

UltraEdit_UESstudio_Help

Table Of Contents

Getting Started	1
Menus/Toolbars	63
Menu Commands	77
File Menu	77
Edit Menu.....	109
Search Menu	120
Insert Menu	148
Project Menu	152
Build Menu	181
Solution Menu.....	190
View Menu.....	193
Format Menu.....	217
Column Menu	228
Macro Menu.....	234
Scripting Menu.....	258
Advanced Menu	258
Configuration	259
Window Menu	366
Help Menu	369
Compiler Configuration	373
CVS Support	393
FAQ.....	395

License Agreement	401
Sales and Support.....	405
Index	407

Getting Started

Using Help



These Help files are organized primarily based on the menu structure in UltraEdit / UESudio. To the left you should see the following major headings:

Getting Started

Menus/Toolbars

Menu Summary

* Compiler Configuration

* CVS Support

FAQ

License Agreement

Sales and Support

The Getting Started section (of which this is the first document) contains useful information to get you up and running quickly and editing text in UltraEdit.

The Menu Summary section may be expanded by double-clicking and consists of information regarding the commands presented in the menus in UltraEdit. Each menu heading (File Menu, Edit Menu, etc.) may be expanded by double-clicking and each topic below the menu headings may be opened by clicking on the topic title in the list. You can also collapse a section heading by double-clicking on it.

The FAQ section includes some of the more common questions asked regarding UltraEdit's functionality and usage.

The License Agreement section includes both the registered and unregistered license agreements for UltraEdit.

The Sales and Support section includes information regarding technical support and purchasing UltraEdit.

* Please note: These features are only supported in UESudio, and not in UltraEdit.

Blocking Text / Column Mode



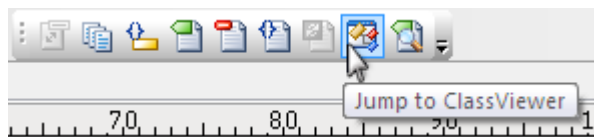
In [Column Mode](#) the operation of the editor is changed for selecting text. When Column Mode is selected highlighted text is based on the character column position of the first character selected to the column of the last character of text selected. Text selected in this mode does not automatically include all text between the start and end position, but includes all text in the columns between the first and last character selected. Column Mode editing is sometimes referred to as block mode editing as the act of selecting text forms a rectangle.

UltraEdit supports two different methods of selecting text in Column Mode. Pressing ALT+C will toggle Column Mode on and text will be selected in a block as described above. Users may also hold down the ALT key while dragging the cursor to make a selection and Column Mode will automatically be toggled on for this selection.

ClassViewer



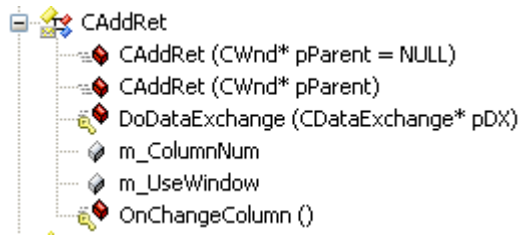
The ClassViewer in UESudio may be invoked by clicking on the Jump to ClassViewer button in the Intellitip toolbar



or by clicking on the Tags tab of the Workspace Manager



This will cause a parsed representation of the active project to be shown.



The items shown in the Workspace Manager under the Tags pane replace Ctags in UESTudio. Double-clicking on one of these items will open the associated project document.

Symbol	Meaning
	Struct
	Class
	Globals Folder
	Global Miscellaneous Folder
	Global Functions Folder
	Global Labels Folder
	Global Macros Folder
	Global Variables Folder
	Enum
	Namespace
	Enum Value
	Variable
	Protected Variable
	Private Variable
	Global Variable
	Function
	Protected Function
	Private Function
	Typedef
	Macro

Code Folding



Code folding, implemented in UltraEdit v11.00 and UESTudio, works in conjunction with the [Hide/Show Lines](#) feature of previous versions.

There are two configuration options under the [Code Folding](#) branch under Configuration in the Advanced menu that determine whether or not code folding and hiding/showing lines is supported in UltraEdit: Enable show/hide lines and code folding and Enable show/hide lines for non syntax highlighted files.

Enable show/hide lines and code folding

If this option is selected under Configuration then the Hide/Show Lines command may be invoked via the [View menu](#) option. Selection of this configuration option also enables code folding and a small gutter will be visible between the line numbers and the edit window where the symbols used for code folding ("+" and "-") will be displayed. By default code folding will operate on any strings (whether characters - "{, }" - or words - "if, endif") defined in the active wordfile as /Open Fold Strings or /Close Fold Strings. Please see [Syntax Highlighting](#) for further information on fold strings.

If a "+" or "-" is displayed in the folding gutter the folding state of the referenced code may be toggled by left-clicking on the folding symbol in the folding gutter. For example, in the following code:

```
276 unsigned int len;
277 {
286 }
```

clicking on the "+" would expand the collapsed code. Clicking on the "-" below:

```
295 {
296     unsigned int i, j;
297
298     for (i = 0, j = 0; j < len; i++, j += 4)
299         output[i] = (((UINT4)input[j]) | (((UINT4)input[j+1]) << 8) |
300                     (((UINT4)input[j+2]) << 16) | (((UINT4)input[j+3]) << 24));
301 }
302
303 /* Note: Replace "for loop" with standard memcpy if possible.
304 */
305 static void MD5_memcpy (output, input, len)
306 POINTER output;
307 POINTER input;
308 unsigned int len;
```


would collapse the expanded code.

There are two code folding options that may be selected from the View menu: Expand All and Collapse All.

Expand All

When this command is invoked from the View menu, all blocks in the active file bounded by Indent/Unindent Strings defined in the active wordfile would be expanded. If text is selected this will affect only the selected region. If no text is selected this command will operate on the whole of the active file.

Please note: When this command is invoked, strings hidden using the [Hide Lines command](#) will be expanded as well as folded sections of code.

Collapse All

When this command is invoked from the View menu, all blocks in the active file bounded by the Indent/Unindent Strings defined in the active wordfile would be collapsed. If text is selected this will affect only the selected region. If no text is selected this command will operate on the whole of the active file.

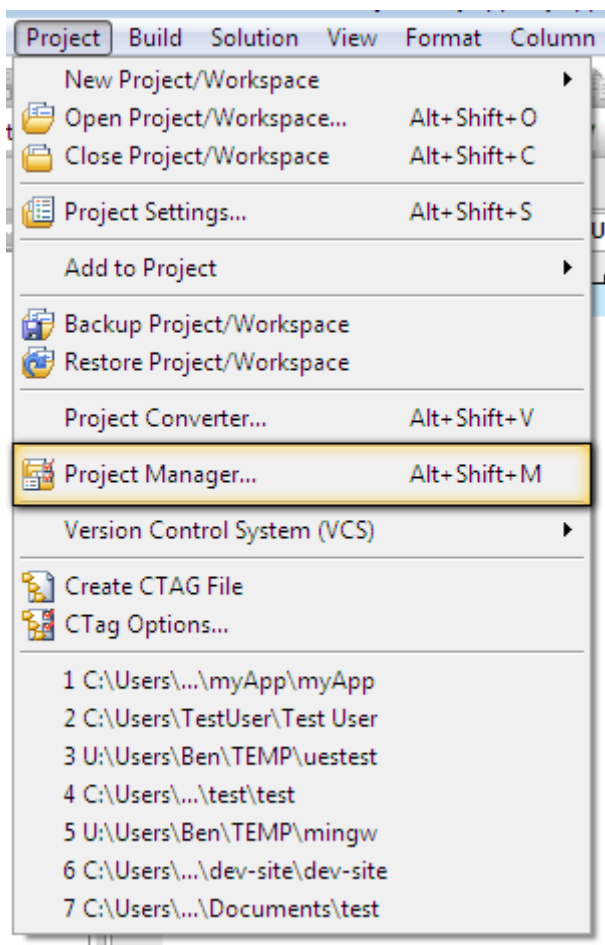
Enable show/hide lines for non syntax highlighted files

If this option is selected under Configuration then the Hide/Show Lines command may be invoked via the [View menu](#) option even on files that have no syntax highlighting (e.g. on standard text files). The advantage here is that lines that have been hidden using the Hide/Show Lines command will be indicated by the "+" in the folding gutter. If this option is not selected under Configuration, text may not be hidden in non-highlighted files and the folding gutter will not be displayed.

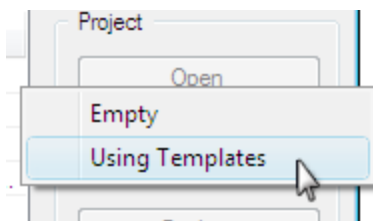
Creating Your First Application



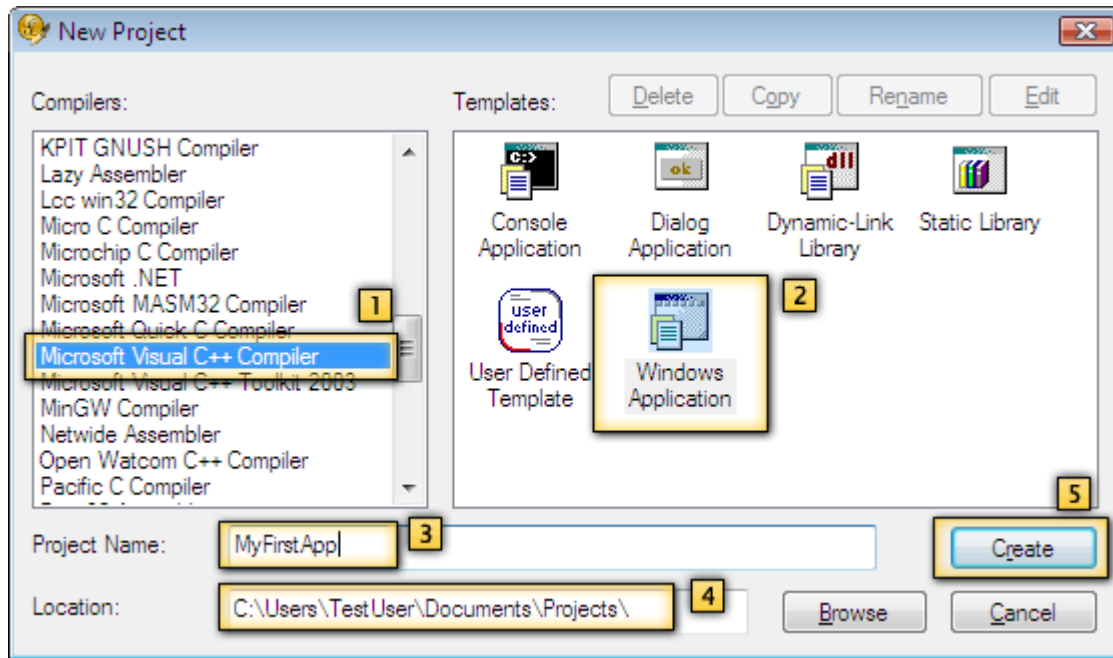
It's easy to Create, Build, and Run an application from within UESTudio! First, click on the Project menu and select [Project Manager](#).



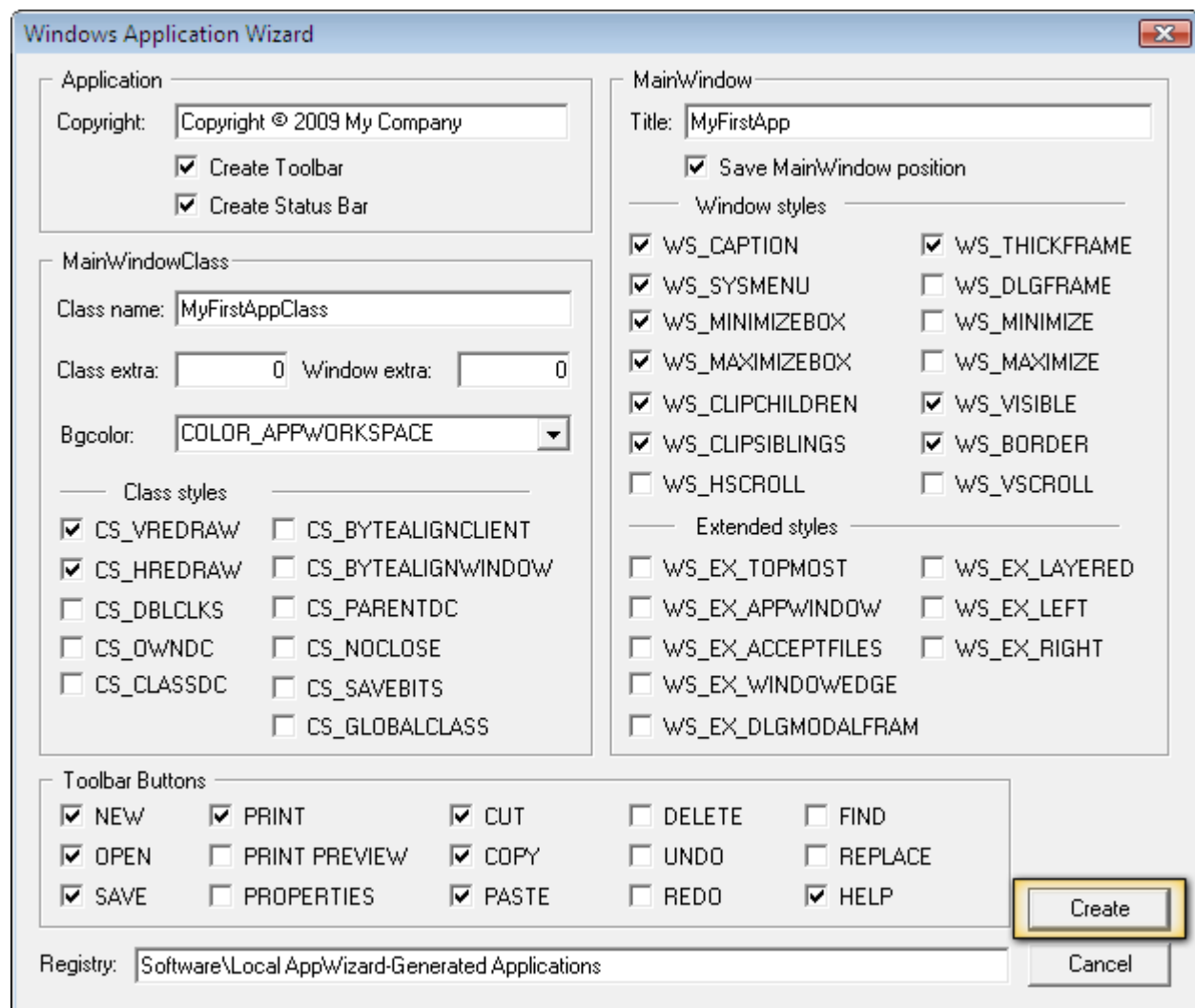
Next, click on the Create button in the Project Manager dialog and select Using Templates from the context menu.



Select a compiler from the Compilers pane [1] in the [New Project dialog](#). (Please note: You must have the compiler you select here installed on your system in order to invoke it from within UESstudio). Then select the type of application you want to create from the Templates pane [2]. Once you've selected the application template, type a name for your project in the Project Name text field [3] and specify the Location [4] or path where you want this project created, and click the Create button [5].

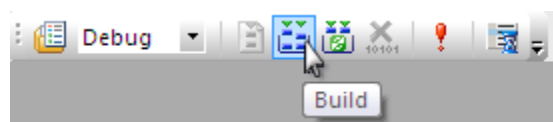


Once you press the Create button, UESudio will launch the Windows Application Wizard dialog where you can specify the general framework of the application you wish to create. Please note: The example below was created with the Microsoft Visual C++ Compiler and the Windows Application template. The options presented in the Application Wizard will be compiler and template specific.

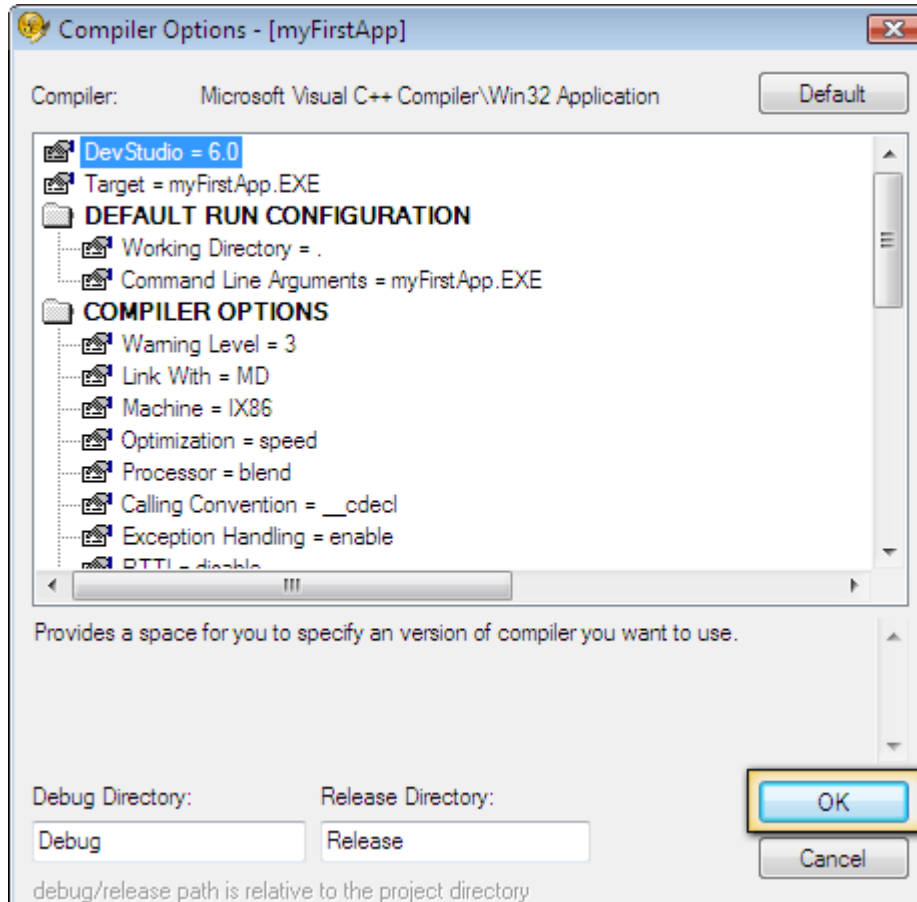


Select the desired options and press the Create button to finish creating your first project and application!

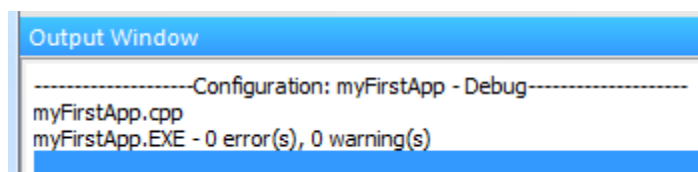
Now it's time to Build your application and you can do this by simply pressing the Build button in the [Build toolbar](#):



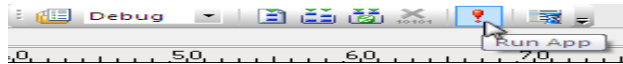
UEStudio will present the Compiler Options dialog and pressing OK will build your first application! Please note: The options shown below are based on the selected Microsoft Visual C++ Compiler. The options presented in the Compiler Options dialog will be compiler-specific.



UEStudio will build and parse your new application and the [Output Window](#) will indicate when the build is complete and list any errors encountered during the build:



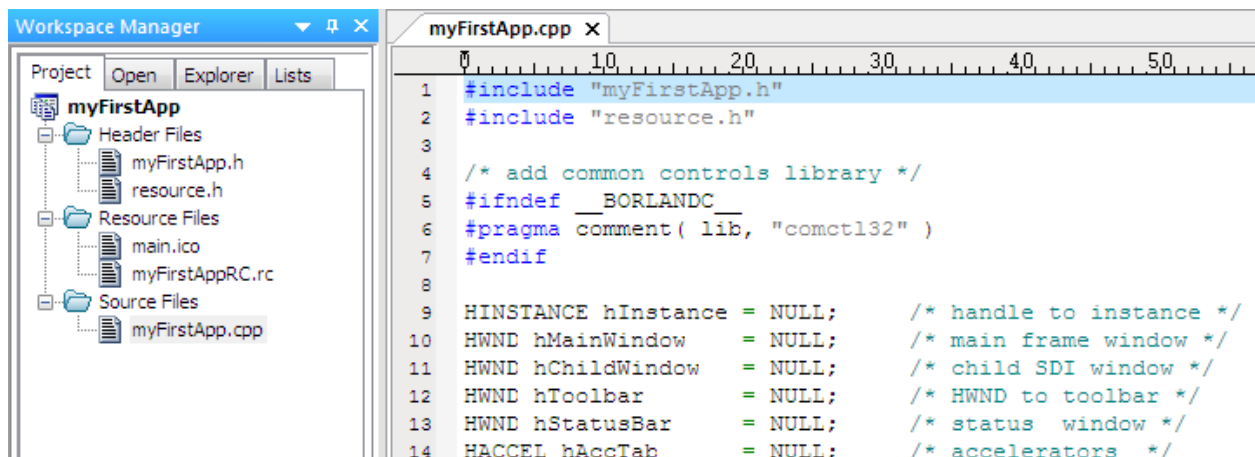
Now that the application is built, all that's left to do is Run the application! You may do this by pressing the Run button in the [Build toolbar](#):



UESudio will launch your newly created and built application!



Please note: Once you've reached this point the menus and toolbar buttons (if included in the application you've created and built) are not actually functional. The framework has been created, but functionality has to be programmed additionally. Double-clicking on the associated source file in the [Workspace Manager](#) will open the source file in UESudio so that further functionality may be added.



Cursor Arrows (Cursor Movement)



The "up", "down", "right" and "left" arrows will move the cursor one character or line as indicated.

Holding down the CTRL key while using the "left" or "right" cursor arrows will move the cursor left and right to the beginning of the previous/next word.

Holding down the CTRL key while using the "up" or "down" keys does not affect the cursor position, but moves the text within the editor window up or down respectively. If the text is moved enough for the cursor to meet the upper or lower edge of the editor window the cursor will then move from line to line as it does when normally using the "up" and "down" arrows.

Holding down the ALT key while using the "left" or "right" cursor arrows will move the cursor to the beginning of the previous/next paragraph.

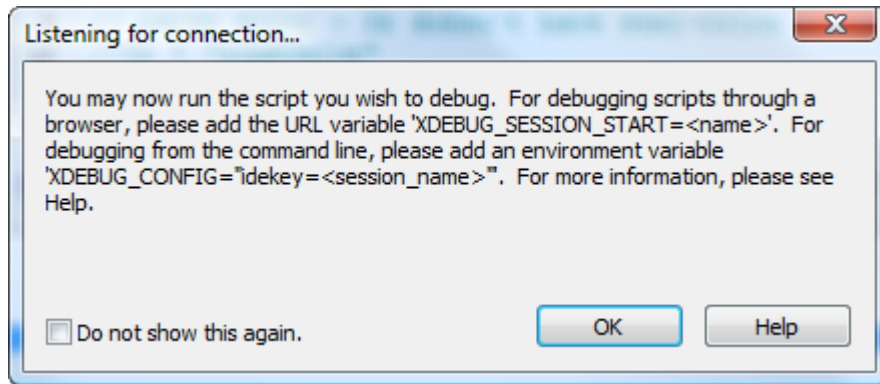
Debugging PHP Scripts



As of version 4.3.0, PHP supports a new SAPI type (Server Application Programming Interface) named CLI which means **Command Line Interface**. If you're running the latest version of [PHP](#), you will be able to execute PHP scripts either in a browser (using the [Apache server](#) and PHP scripting engine) or from the command line (using only the PHP scripting engine). Through the integration of [Xdebug](#), users of UESTudio can debug both browser-based PHP scripts or command line PHP scripts.

Before debugging can occur, the PHP scripting engine and Xdebug must be downloaded and installed. Please [click here](#) for instructions on how to download and install these components and configure them appropriately based on whether you will be debugging browser-based PHP scripts or command line PHP scripts. Once you've installed and configured PHP and Xdebug on your system, debugging your PHP scripts is just a few steps away:

1. Create a default PHP debugging project in UESTudio. This PHP debugging project is only used to store settings specified for the PHP debugger. *PHP scripts do not have to be associated with the active project in any way for debugging to occur.*
 - a. Go to **New Project/Workspace** in the **Project** menu and select "Create New Project..."
 - b. Enter a name for your new project, e.g.: "PHPDebug" and press the **Save** button.
 - c. Click on the **Debug** tab and select "Integrated Xdebug" from the **Select Project Debugger** dropdown and press the **Close** button.
2. In the **Build** menu select **Start Debugger** to initiate the Xdebug debugging session. When Xdebug is invoked a dialog will appear:



At this point you may press "OK" and proceed as indicated in the dialog:

Browser Based PHP Debugging

1. After pressing "OK" in the **Listening for Connection...** dialog, you will see

Xdebug: Listening for connection...

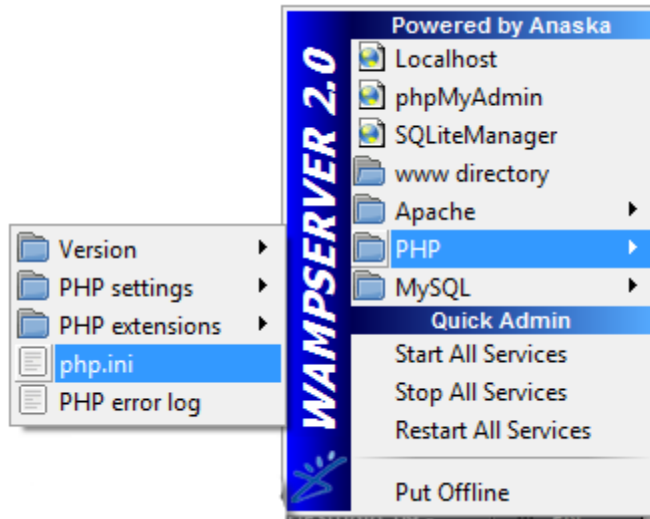
in the status bar. This indicates that Xdebug is ready and waiting to detect a script being run on the appropriate port.

2. Ensure that the installed [WampServer](#) is running. If not running WampServer, ensure that the Apache server and PHP scripting engine are running. The PHP script must be executed to be detected and parsed for debugging.

3. For debugging scripts through a browser, please add the URL variable 'XDEBUG_SESSION_START=<name>', e.g.:

`http://localhost/test.php?XDEBUG_SESSION_START=myPHPTest`

If you prefer not to append `?XDEBUG_SESSION_START=myPHPTest` to all scripts to enable debugging when executed, you can set a flag in your php.ini file to enable debugging of every PHP script when run. If you've installed WampServer, the easiest way to do this is to access the php.ini file through the WampServer menu:



Where Xdebug settings were defined during installation, if the **remote_autostart** option is set to "1" rather than "0":

xdebug.remote_autostart = 1

then every time a PHP file is executed after Build -> Start Debugger has been selected, Xdebug would detect and debug the running script. Please note that the PHP script must be saved to the proper directory for the Apache server installed on your system in order to be run from "http://localhost/". With the default WampServer installation, this would be under:

C:\wamp\www\

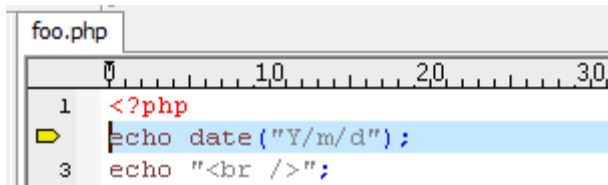
When the URL is appropriately entered into your browser with the appended debug flag, the status bar in UESTudio should change to indicate that Xdebug detected the running script and has begun to debug it. UESTudio will indicate this by displaying:

Xdebug: Initializing session

in the status bar. Xdebug will process the script and update this prompt to:

Xdebug: Step Into

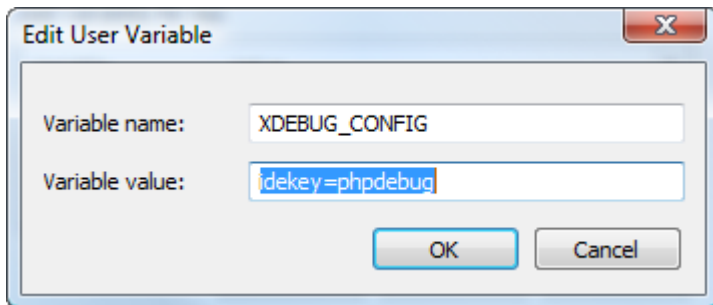
At this point, the debugging results will be displayed in the output window in UESstudio. You will either see an error message displayed, or the output window will indicate "No debug message." which indicates that no errors were found in the debugged script. If this debugged file is not already open in UESstudio, it will automatically be opened at this point in time. If no errors exist in the script, the debugger focus arrow will be displayed on the first line of the script:



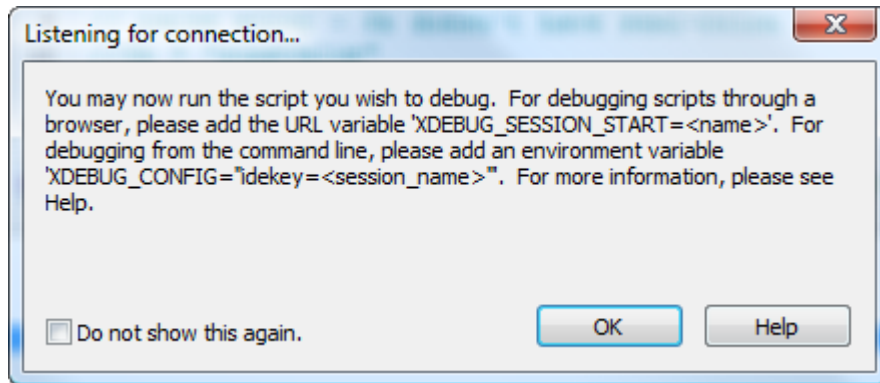
If an error is found, the line associated with the error displayed in the output window will be marked by the debugger focus arrow.

Command Line PHP Debugging

An environment variable must be defined before debugging of command line PHP scripts can occur. The environment variable name should be "XDEBUG_CONFIG" and the value should be "idekey=<session_name>" where <session_name> can be any value, e.g. "idekey=phpdebug".



1. In the **Build** menu select **Start Debugger** to initiate the Xdebug debugging session. When Xdebug is invoked a dialog will appear:



At this point you may press "OK" and proceed as indicated in the dialog:

2. After pressing "OK" in the **Listening for Connection...** dialog, you will see

Xdebug: Listening for connection...

in the status bar. This indicates that Xdebug is ready and waiting to detect a script being run on the appropriate port.

3. Open a command prompt and execute the script to be debugged, e.g.:

```
C:>php c:\php_modules\test.php
```

Unlike browser-based scripts, no debug flag is required for command line scripts because this is automatically handled by the environment variable already defined.

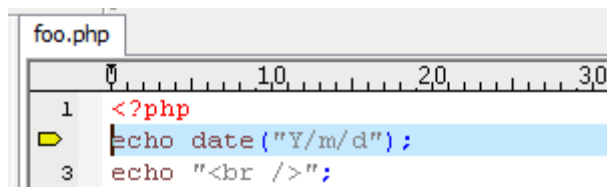
4. When the script has been executed, the status bar in UESTudio should change to indicate that Xdebug detected the running script and has begun to debug it. UESTudio will indicate this by displaying:

Xdebug: Initializing session

in the status bar. Xdebug will process the script and update this prompt to:

Xdebug: Step Into

At this point, the debugging results will be displayed in the output window in UESstudio. You will either see an error message displayed, or the output window will indicate "No debug message." which indicates that no errors were found in the debugged script. If this debugged file is not already open in UESstudio, it will automatically be opened at this point in time. If no errors exist in the script, the debugger focus arrow will be displayed on the first line of the script:



If an error is found, the line associated with the error displayed in the output window will be marked by the debugger focus arrow.

Drag and Drop Editing



UltraEdit supports dragging and drop editing using the mouse.

Drag and drop editing allows the user to select text in the normal manner and then copy or move the selected text by dragging it to a new location with the mouse. No keyboard interaction is required.

Initially the user selects text in the normal manner. Once text has been selected, the user may click the primary mouse button (usually the left button) on the selected text, at which time the mouse pointer changes to a pointer with a small dotted rectangular box at the bottom of the pointer. This pointer indicates you are in the drag and drop editing mode and may drag the selection to a new location.

While keeping the primary mouse button down the selected text may be dragged to a new location. As you move the mouse the cursor will follow the mouse and the cursor indicates the new position of the text if you stop pressing the mouse button (and "drop" the text).

The user may even drag the text from one file to another within UltraEdit. If the mouse is positioned over a different file within UltraEdit, that file will become active and the text may be dropped into that file at the cursor position shown.

At any time, pressing ESCAPE will cancel the drag and drop mode.

Additionally, instead of moving the text being dragged it may be copied to a new location. If at any time while in the drag and drop mode the CTRL key is pressed and held the mouse pointer changes to have a small '+' symbol beside it indicating the text will be copied to the new location indicated by the cursor and also remain in the original position.

Drag and Drop editing is supported in normal, [Hex](#) and [Column](#) editing modes.

Drag and Drop Files



UltraEdit supports dragging and dropping files from the File Manager into an open session of UltraEdit. Files may be selected from the File Manager (multiple files may be selected) and dragged into the UltraEdit window. If the mouse button is released with the mouse over UltraEdit the files selected will automatically be opened for editing with UltraEdit.

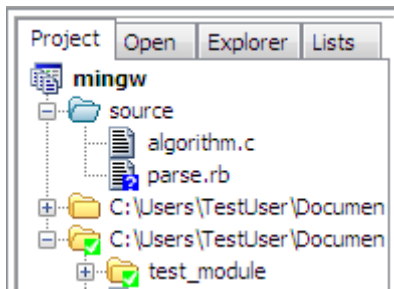
File Tree View Interface



UltraEdit and UESTudio feature an enhanced File Tree View. Visibility of the File Tree View may be toggled by selecting File Tree View from the Views/Lists submenu of the View menu or by pressing Ctrl+U.

Users may toggle between a listing of Project, Open, Explorer and **List** display modes.

Project mode: displays all files and folders in the active project with the project name listed at the top.



If no items are selected in the File Tree View in **Project** mode or if only the project name is selected the following items are shown in the context menu:

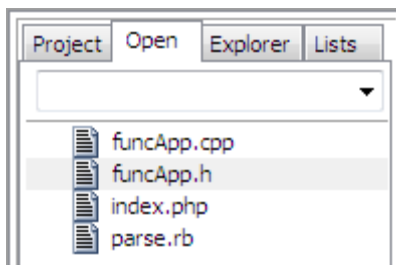
Explore	opens Explorer to active project directory
Refresh	updates the project information (committing the changes) and refreshes the project list
Add Folder	presents New Folder dialog
Add New File	presents New File dialog where users specifies file to be added
Add Active File	adds file currently open in the edit window to the active project
Add All Open Files	adds currently open files to the active project
Add Files to Project	presents Open dialog for specification/selection of file(s) to be added to project. If no subfolders are selected in the included dropdown file(s) will be added to the root of the project.
Open File in Project	presents Open File(s) in Project dialog
* Compiler Options	presents Compiler Options dialog
Settings	presents Project Settings dialog

If a folder or file is selected in the File Tree View under **Project** mode, the following items are shown in the context menu:

Open	opens selected file/folder in active project
Explore	opens Explorer to selected file/folder directory
VCS -> Commit	commits working copy to repository without displaying any prompts or options
VCS -> Update	commits working copy from repository without displaying any prompts or options

VCS -> Update (Special)	allows configuration of multiple VCS-related options using CVS/Subversion Update dialog
VCS -> Diff	shows the difference between the current local working file and the earlier revision stored in the repository.
VCS -> Compare	invokes UltraCompare to show the difference between the current local working file and the earlier revision stored in the repository.
VCS -> Advanced Diff/Compare	presents CVS Diff dialog where user may specify parameters to be used for diff command
VCS -> Remove	removes files and directories from a VCS repository
Remove from Project	removes the selected file/folder from the active project
* Add to Environment	presents the Add/Modify Environment Variable dialog

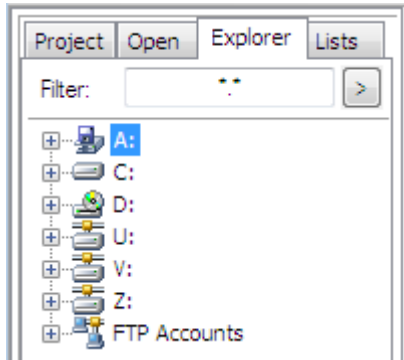
Open mode: displays all currently open files.



The context menu in **Open** mode includes the following items:

Close File	closes selected file(s)
Save File	saves selected file(s)
Print File	prints selected file(s)
Tile Windows Horizontally	horizontally tiles all open files
Tile Windows Vertically	vertically tiles all open files
Make Folder Active File Path	sets active file path to path of currently selected file
Show Names Only	if selected, only file names (and not paths) are displayed
Sort By Extension	sorts listed files by extension

Explorer mode: displays File Tree View for local/network drives and removable media.



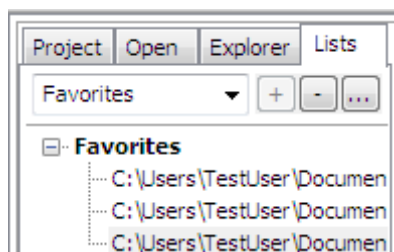
If a file is selected in **Explorer** mode, right-clicking in the File Tree View presents the following context menu:

System Explorer Menu	presents applicable parts of Windows Explorer context menu
Rename File	moves focus to active file name to allow renaming
Open File	opens active file in edit window
Open with Default	opens active file in application registered with Windows for file type
Properties	displays Windows properties dialog for active file
Delete File(s)	deletes selected file(s)
Make folder active file path	sets selected file folder as active file path
Windows Directory	switches focus to Windows directory
System Directory	switches focus to System directory
My Documents	switches focus to My Documents directory for user
Application	switches focus to Application Data directory
Insert into Document	presents submenu where users may select to insert the Full Path, Path, or Name of the selected file into the active document

If a folder is selected in **Explorer** mode, right-clicking in the File Tree View presents the following context menu:

System Explorer Menu	presents applicable parts of Windows Explorer context menu
Refresh	updates listing of all hard drives and removable media
Set Directory as Root	sets the active directory as root on right click and selection
Find in Files in this Directory	launches Find in Files dialog with Directory set to selected directory
Replace in Files in this Directory	launches Replace in Files dialog with Directory set to selected directory
Open Command Prompt Here	opens command prompt set to path for selected directory
Rename File	moves focus to active folder name to allow renaming
Open File	expands directory listing for selected folder
Properties	displays Windows properties dialog for active file
Delete File(s)	deletes selected file(s)
Make folder active file path	sets selected folder as active file path
Windows Directory	switches focus to Windows directory
System Directory	switches focus to System directory
My Documents	switches focus to My Documents directory for user
Application	switches focus to Application Data directory
Insert into Document	presents submenu where users may select to insert the Full Path, Path, or Name of the selected directory into the active document

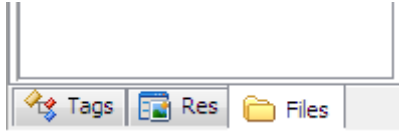
Lists mode: displays Tree View for Favorites and user-created groups.



If focus is in the combo box, you may type the name for a new list heading (e.g., **Personal**) and pressing the "+" button would add this to the list as a new heading. You may then select the heading you would like to add items to and press the browse button "..." to present the context menu with the options **Add Folder** and **Add File**. If an item is selected in the **Lists** pane and the "-" button is pressed, the selected item will be removed from the list.

UEStudio Interface

UESstudio includes three additional tabs at the bottom of the File Tree View that are not present in UltraEdit.



When the * Files tab is selected the three default tabs at the top of the display are active and selectable. In UESstudio the Project, Open, and Explorer modes display files/modules with small colored squares that indicate their status in relation to the CVS repository.

Color	Meaning
Green	file/module is up-to-date and unmodified
Blue	file/module is not currently under CVS control
Yellow	file/module is modified
Red	file/module contains a conflict

When the * Res tab is selected the resources for the active project are listed. If a folder is selected in the File Tree View under Resource mode the following items are shown in the context menu:

Save Resources	saves modified resources in the active project
Insert	inserts a copy of the selected resource folder into the active project below the selected resource folder
Import	presents File Open dialog from which the item to be imported may be selected
Editor Configuration	presents Resource Editors Configuration dialog where users may specify paths for Bitmap Editor, Icon Editor, or Cursor Editor

If a file is selected in Resource mode right clicking in the File Tree View presents the following context menu:

Open	opens the selected Data, Dialog or Menu resource for editing. If a Bitmap, Cursor, Icon or Toolbar resource is selected, the
------	--

	associated editor will be invoked. If an Accelerator or String Table resource is selected the Accelerators dialog or Strings dialog will be presented to allow editing of the selected resource.
Open Binary Data	opens the selected resource in binary/HEX mode
Remove Resource	removes the selected resource from the active project
Save Resources	saves modified resources in the active project
Insert Copy	inserts a copy of the selected resource into the active project below the selected resource
Insert	presents Insert Resource dialog from which users may choose to insert one of the following items: Accelerator, Bitmap, Cursor, Dialog, Icon, Menu, Toolbar, Version, Font, Html
Import	presents File Open dialog from which the item to be imported may be selected
Editor Configuration	presents Resource Editors Configuration dialog where users may specify paths for Bitmap Editor, Icon Editor, or Cursor Editor

When the * Tags tab is selected classes, structs and other symbols defined inside the project workspace are displayed. Below these items globals (Functions, Labels, Macros, and Variables) will be listed as well. If you double-click on a displayed symbol, the appropriate source file will be opened and the line where the symbol is defined/declared will be positioned to. In UESTudio this interface replaces the Ctags mechanism used in UltraEdit.

If a tag is selected right clicking in the File Tree View presents the following context menu:

Go To Definition	opens file where symbol is defined to the line where definition occurs
Find Symbol in Active Document	lists occurrences of symbol in active document in the output window
Find Symbol in Project Files	lists occurrences of symbol in active project in the output window

* Please note: These features are only supported in UESTudio, and not in UltraEdit.

Home and End keys (Cursor Movement)



The "Home" key moves the cursor to the beginning of the line the cursor is on. Holding down the CTRL key while using the "Home" key moves the cursor to the beginning of the document.

The "End" key moves the cursor to the end of the line the cursor is on. Holding down the CTRL key while using the "End" key moves the cursor to the end of the document.

Insert/Overstrike Modes



UltraEdit allows editing in both insert and overstrike modes. The mode of operation is indicated in the status bar at the bottom of the UltraEdit window. The insert/overstrike mode is changed with the INSERT key. The mode is toggled when the INSERT key is pressed.

If insert mode is selected, the second area of the status bar is clear; if overstrike mode is selected, the second area of the status bar shows ' OVR'.

Insert mode causes a key to be inserted into the document at the position of the cursor. Overstrike mode causes the character immediately to the right of the cursor to be overwritten by the new character.

If the editor active window is in [Hex mode](#) the active window is always in overstrike mode. To insert text in Hex mode, use the [Hex Insert/Delete command](#) from the [Edit menu](#).

Inserting Special Characters



Special support has been added to UltraEdit to allow characters not supported on American keyboards to be easily inserted. These are as follows:

Ctrl+Shift+a	Inserts ä character
Ctrl+Shift+A	Inserts Å character (Caps Lock on)

Ctrl+Shift+o	Inserts ö character
Ctrl+Shift+O	Inserts Ö character (Caps Lock on)
Ctrl+Shift+u	Inserts ü character
Ctrl+Shift+U	Inserts Ü character (Caps Lock on)
Ctrl+Shift+s	Inserts ß character
Ctrl+Shift+S	Inserts ß character (Caps Lock on)

It is also possible to insert any character listed in the [ASCII Table](#) in the [View menu](#) by holding down the ALT key while typing in the four digit decimal value of the character on the numeric keypad. For example, " ¿ " may be inserted by holding down the ALT key while typing "0191" and " Æ " may be inserted by holding down the ALT key while typing "0198".

Keyboard Commands



UltraEdit allows the use of keyboard commands for a number of commonly used features. The keyboard commands are listed below:

F1	Help	Show UltraEdit Help
Shift+F1	Context Help	Invoke the context help
Ctrl+N	File New	Create a new document file
Ctrl+O	File Open	Open an existing document file
Ctrl+Q	File Quick Open	Open an existing document file without showing the File Open dialog
Ctrl+F4	File Close	Close an existing document
Ctrl+S	File Save	Save the active document
F12	File Save As	Save the active document as a new file
Ctrl+P	File Print	Print the active document
Ctrl+X	Edit Cut	Cut text from active document into the clipboard
Ctrl+C	Edit Copy	Copy text from active document into the clipboard
Ctrl+V	Edit Paste	Paste text from the clipboard into the active document
Ctrl+0-9	Edit Select Clipboard	Select active clipboard
Ctrl+A	Edit Select All	Select All text in active document
Ctrl+Z	Edit Undo	Undo last action if possible

Ctrl+Y	Edit Redo	Reverse last Undo action if possible
Ctrl+J	Edit Select Word	Select the current word (same as double click)
Ctrl+E	Edit Delete Line	Delete line cursor is on
Ctrl+F11	Edit Delete to Start of Line	Delete from cursor to start of line
Ctrl+F12	Edit Delete to End of Line	Delete from cursor to end of line
Ctrl+W	Toggle Word Wrap	Toggle word wrap on/off
F7	Edit Insert Time/Date	Insert time/date at cursor
Alt+F3 or Ctrl+F	Find	Find a character string
Ctrl+F3	Find Prev	Repeat last find toward beginning of file
F3	Find Next	Repeat last find toward end of file
Ctrl+B	Match Brace	Find matching brace (.,[, { or },],.)
Ctrl+R	Replace	Find and replace a character string with another
Ctrl+G	Goto	Goto the specified line (or HEX address)
Ctrl+U	File Tree View	Toggle File Tree View
Ctrl+F8	Tag List	Toggle Tag List
Ctrl+T	Reformat Paragraph	Reformat the current paragraph or selected text
Ctrl+F5	Format to Lower	Convert selected text to lower case
Alt+F5	Format to Upper	Convert selected text to upper case
F5	Format Capitalize	Capitalize first character of each word in selected text
Shift+F5	Format Invert Case	Invert case of all characters in selected text
Ctrl+K	Edit Spell Check	Invoke the spelling checker
Alt+C	Column Mode	Toggle column/block mode on/off
Ctrl+Backspace	Delete previous word	Delete the word preceding the cursor
Ctrl+Delete	Delete next word	Delete the word following the cursor
Ctrl+I	Insert Literal Character	Insert a literal character at cursor position
INS	Insert/Overstrike Mode	Toggle insert/overstrike mode
Ctrl+H	HEX Edit	Toggle HEX editing mode
Ctrl+D	HEX Insert/Delete	Insert or delete characters in HEX editing mode
Ctrl+F2	Set Bookmark	Toggle bookmark on/off
F2	Goto Bookmark	Goto next bookmark

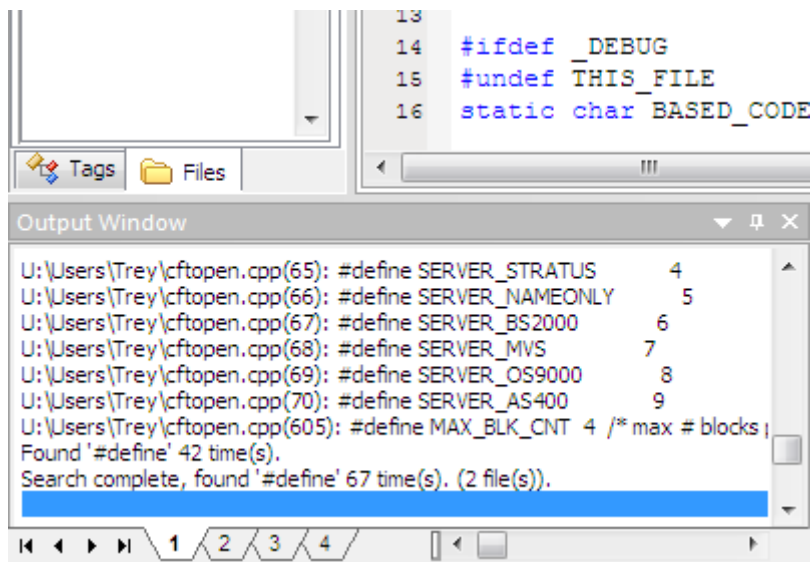
Ctrl+M	Macro Play	Replay a macro
Ctrl+L	Macro Play Multiple Times	Replay a macro the specified number of times
Ctrl+F6	Next Window	Make next document window active
Ctrl+Shift+F6	Previous Window	Make previous document window active
Alt+F4	Exit UltraEdit	Exit UltraEdit
Alt+0-9 or Shift+Alt+0-9	Insert Template	Insert user-defined template
F9	DOS Command	Run DOS window command
Ctrl+F9	Last DOS Command	Repeat last DOS window command
F10	Run Windows Program	Run Windows Program
Ctrl+Shift+A	Inserts ä character	(As ANSI or ASCII based on the font)
Ctrl+Shift+A (Caps Lock on)	Inserts Ä character	(As ANSI or ASCII based on the font)
Ctrl+Shift+O	Inserts ö character	(As ANSI or ASCII based on the font)
Ctrl+Shift+O (Caps Lock on)	Inserts Ö character	(As ANSI or ASCII based on the font)
Ctrl+Shift+U	Inserts ü character	(As ANSI or ASCII based on the font)
Ctrl+Shift+U (Caps Lock on)	Inserts Ü character	(As ANSI or ASCII based on the font)
Ctrl+Shift+S	Inserts ß character	(As ANSI or ASCII based on the font)
Ctrl+Shift+S (Caps Lock on)	Inserts ß character	(As ANSI or ASCII based on the font)
Ctrl+Up	Scroll Up	Scroll up one line keeping cursor in view
Ctrl+Down	Scroll Down	Scroll down one line keeping cursor in view
F4	Change Focus	Toggle focus between active file and Tree View if visible
F8	Display Function List	Display Function List
Alt+Right	Next Paragraph	Position cursor at first non-space character of next paragraph
Alt+Left	Previous Paragraph	Position cursor at first non-space character of current paragraph (if positioned mid-paragraph) or previous paragraph
Alt+Page Up	Top of Window	Position cursor at beginning of first line of display
Alt+Page Down	Bottom of Window	Position cursor at beginning of last

		line of display
Alt+'-' (numeric keypad only)	Scroll to Top	Position active line at top of window
Alt+'+' (numeric keypad only)	Scroll to Bottom	Position active line at bottom of window
Alt+'center' or Alt+'5' (numeric keypad only)	Scroll to Center	Position active line at center of window
Ctrl+'1' (numeric keypad '1' only)	End of previous word	Position cursor at the end of the previous word
Ctrl+'2' (numeric keypad '2' only)	End of next word	Position cursor at the end of the next word
Ctrl+Insert	Telnet copy	Copy selected text in Telnet client to clipboard
Shift+Insert	Telnet paste	Paste clipboard contents into Telnet client

Output Window



UESudio supports a tabbed output window with four different tabs allowing output to be written and stored separately without overwriting the results of previously run functions.



The advantage of this is that multiple functions may be executed and the dynamically generated output of these functions may be stored separately for future reference (during the active editing session). All output will be written to the active window. So clicking on tab "1" and running a [Find in Files](#) would write the results of the Find in Files command to the output window associated with tab "1". Clicking on tab "3" and [building](#) the active project would cause the STDOUT and STDERR output (depending on [compiler settings](#)) to be written to the output window associated with tab "3".

Right clicking in the Output Window displays a context menu with the following options:

Clear Output	erases contents of active window
Copy to Clipboard	copies contents of active window to clipboard
Copy Selected Line to Clipboard	copies selected line to clipboard
Next Message	moves focus to next item in output window listing and opens referenced file to listed line
Previous Message	moves focus to previous item in output window listing and opens referenced file to listed line
Set Default Path for Relative Paths...	presents dialog where user may browse to folder to specify a directory to be used as the default directory for processing relative paths in the output window
Go To Error/Warning	opens referenced file to indicated line/column
Scroll Output	controls whether output window focus scrolls to last line when populated or remains at top of listing
Allow Docking	toggles whether output window may be positioned within edit framework as a docked window
Hide Output Window(s)	toggles visibility of active output windows
Use Spaces instead of Tabs	forces output windows to use spaces instead of tabs when displaying command output
Show Tool tips	toggles whether tool tips are displayed when output lines exceed window width

* Please note: This is only supported in UESTudio, and not in UltraEdit.

Page Up & Page Down keys (Cursor Movement)



The "Page Up" key moves the cursor up one page. Holding down the CTRL key while using the "Page Up" key moves the cursor to the top of the edit window.

The "Page Down" key moves the cursor down one page. Holding down the CTRL key while using the "Page Down" key moves the cursor to the bottom of the edit window.

Registering UltraEdit/UESstudio

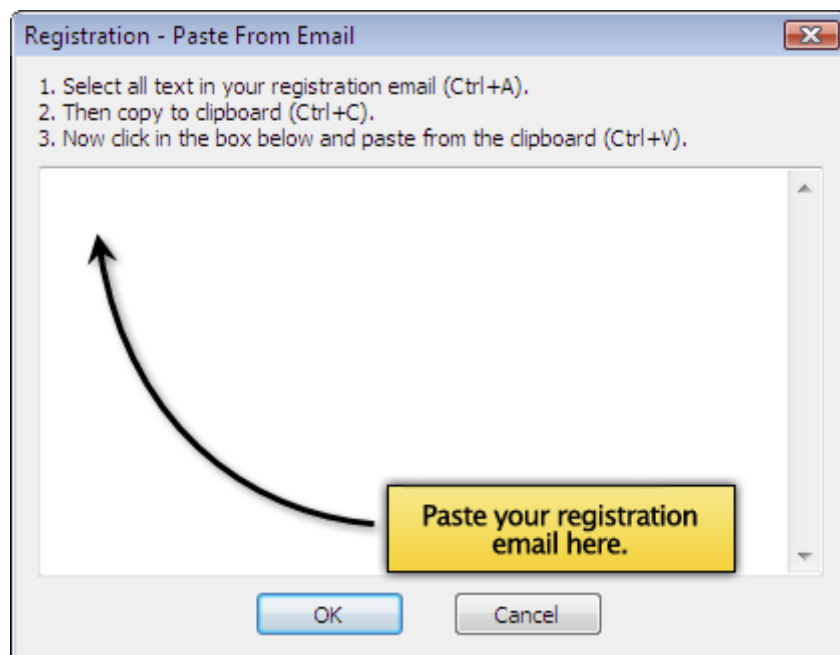


The user name and authorization code provided for UltraEdit/UESstudio are case sensitive and must be entered EXACTLY as received from IDM Computer Solutions, Inc.

Paste From Email

Users may paste the entire registration message to register the product rather than pasting the user name and authorization code separately. When the Register option is selected from the Help menu, the Authorization dialog will be selected and users may press the **Paste From Email** button for the SIMPLEST and MOST RECOMMENDED method for registering the product. This will present the **Registration - Paste From Email** dialog.

As indicated in the dialog (shown below) the entire registration message should be copied to the clipboard and pasted into the **Registration - Paste From Email** dialog. UltraEdit/UESstudio will automatically parse the registration message and register the product accordingly.



Manual Registration

If preferred, the user may still register by copying the user name and authorization code from the message received from IDM using the clipboard (select it and then copy to clipboard), then paste it into the authorization dialog edit box (using Ctrl+V).



Please note that if your user name includes a reference to the number of users, i.e.,

"Acme Printing~ 8 User License"

this is part of the registration name and must be entered along with "Acme Printing".

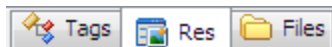
If you do have any problems properly registering UltraEdit/UESstudio, please send an email to support@idmcomp.com indicating what version of UltraEdit/UESstudio you're trying to register and EXACTLY what you're entering for the user name and authorization code. We will reply to you as soon as possible with information to help you register your license.

Resource Editor

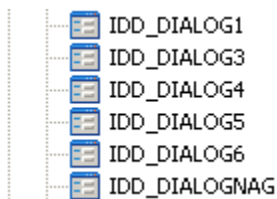


UESstudio features a Resource browser and editor. Add, Remove, Browse resources as desired. Edit dialogs, string tables, menus and more. You can even configure UESstudio to edit bitmaps and icons via user-installed graphic editors.

When a compilable project is loaded in UESstudio, the Resource Editor toolbar is active. If the Resource tab is selected at the bottom of the Workspace Manager



the source for the associated dialog may be opened in UESstudio by double-clicking on the dialog icon or ID in the Workspace Manager.

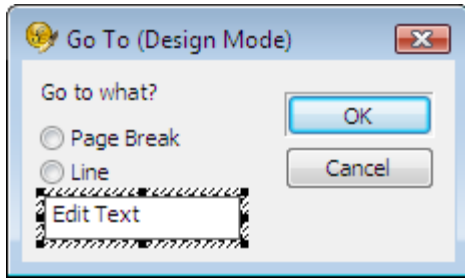


With the source for the dialog to be edited visible in the edit window, two options are available for modifying the dialog.

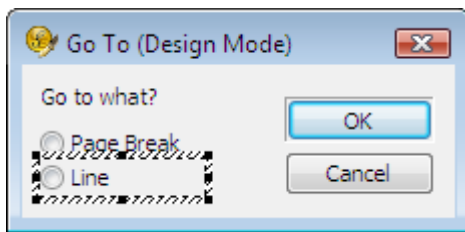
1. The source for the dialog may be modified directly and saved changes will be written back to the project resources.
2. The Test button in the Resource Editor toolbar may be clicked



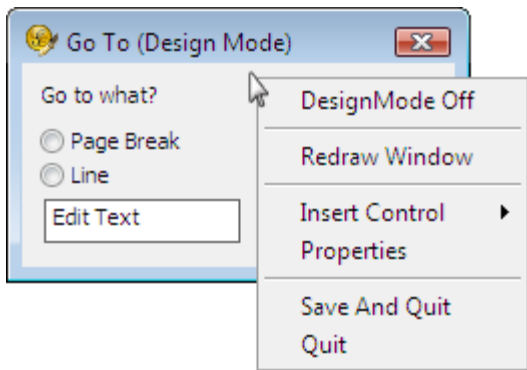
to present the dialog in preview mode.



Individual controls may be clicked on and dragged to new positions within the dialog. New controls may be added and existing controls may be removed:

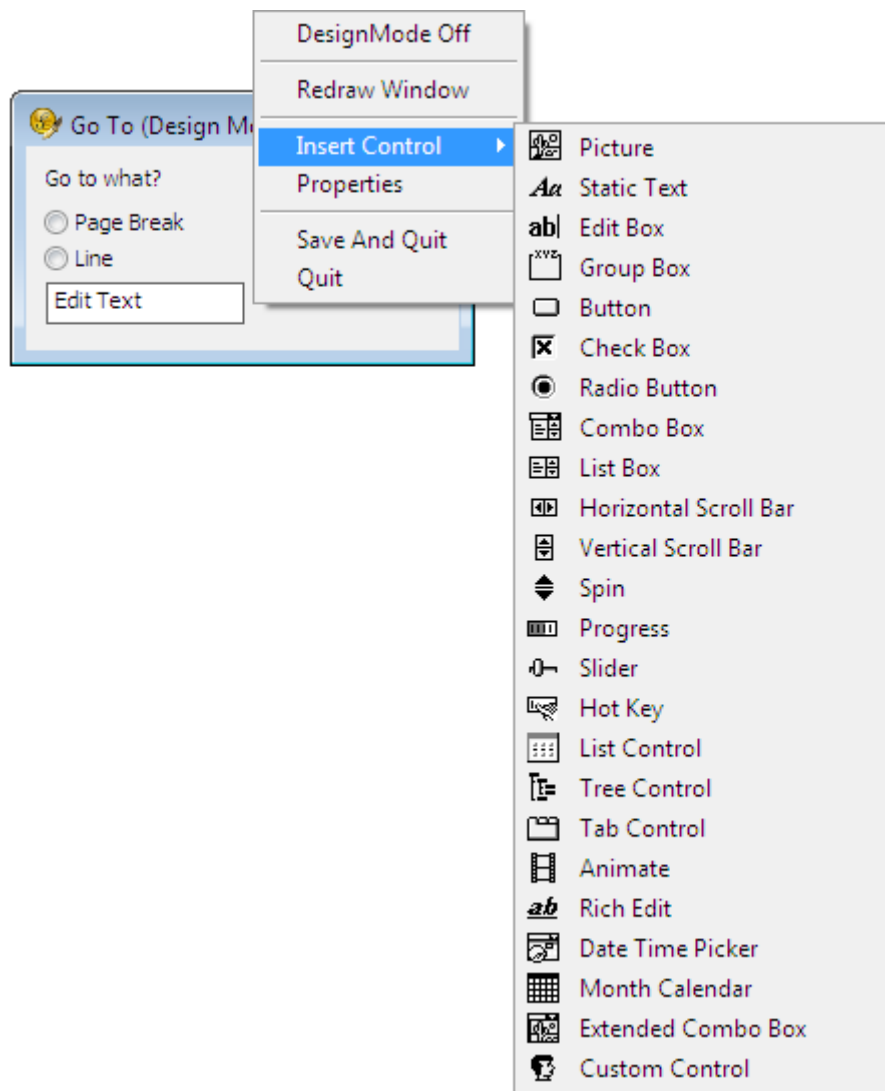


Right clicking on the preview dialog will present the Resource Editor context menu:



This provides a more visual method of designing dialogs and once the desired result is achieved changes to the dialog may be saved by selecting Save and Quit from the context menu.

A broad range of choices is available for adding new controls to the dialog being previewed:



Right Mouse Button Commands



UltraEdit allows the right mouse button to display a pop-up menu. If the right mouse button is clicked while in a document window, the [Edit menu](#) is displayed at the position of the mouse click. If text is selected when the right mouse button is pressed, a new menu item Open File Highlighted is added to the top of the pop-up menu and if selected the file is opened if it exists.

If no text is selected, UltraEdit will examine the text under the cursor and if it finds what it believes may be a file it will add a new menu item "Open " followed by the name of the file it will try and open. If this is a URL the information will be passed to the default browser for opening.

If the right mouse button is clicked while in the UltraEdit main window and outside of a document window, the [File menu](#) is displayed at the position of the mouse click.

Scripting Demo and Tutorial



Scripting in UltraEdit/UEStudio is enabled through embedding of the JavaScript engine. The scripting engine supports the core functionality of JavaScript 1.7. Further information on JavaScript may be found on the associated Mozilla site (<http://developer.mozilla.org/en/docs/JavaScript>).

Scripting Tutorial

For a short **scripting tutorial** please [click here](#).

Demo Scripts

Demo scripts may be found in the UltraEdit installation directory in the **scripts** subdirectory (typically c:\Program Files\IDM Computer Solutions\UltraEdit\scripts). These and any other JavaScript scripts may be opened directly in UltraEdit for editing with integrated syntax highlighting as in the example below:

```

25 // Do operation number of requested iterations.
26 var x = 0;
27 while (x < num) {
28     UltraEdit.document[0].findReplace.find(str);
29     UltraEdit.document[0].selectLine();
30     UltraEdit.document[0].copy();
31     UltraEdit.activeDocument.write(str + "\r\n");
32     x++;
33 }
```

The **hello.js** script demonstrates the simplest form of script supported - a single line script that writes to the active file:

```

21 UltraEdit.activeDocument.write("Hello World!");
```

To invoke this script please go to **Scripting** in the **Advanced** menu and click on the **Scripts...** item in the submenu. This will present the **Scripting** configuration dialog. If you press the **Add** button and navigate to the "scripts" directory below the installation directory you may add "hello.js" to the list of scripts available to be run. You may assign a hotkey to the script by clicking in the Hotkey field and pressing the desired keys. Scripts that are added will

be dynamically updated in the bottom of the **Scripts...** submenu referenced above and may be invoked through hotkeys or by clicking on the script name in this list.

To be fully self-contained this script actually creates a new Edit document to write to and the script may be run without opening any documents.

The **stringvar.js** script shows the ability to capture user input and store it in variables for later use in the script:

```
18 // Get user input.
19 var num = UltraEdit.getValue("How many occurrences do you want to find?",1);
20 var str = UltraEdit.getString("What string do you want to find?",1);
21
22 // Create new file. This will also become the active file.
23 UltraEdit.newFile();
24
25 // Do operation number of requested iterations.
26 var x = 0;
27 while (x < num) {
28     UltraEdit.document[0].findReplace.find(str);
29     UltraEdit.document[0].selectLine();
30     UltraEdit.document[0].copy();
31     UltraEdit.activeDocument.write(str + "\r\n");
32     x++;
33 }
```

This may be invoked as described above for the **hello.js** script. To run this demo, please open any text file and specify the string you wish to search for and the number of occurrences you wish to find. The script will then open a new document and write the specified number of matching lines to the new file.

The **header.js** script demonstrates the ability to find the number of open files using the **document** JavaScript array object which is a property of the **UltraEdit** application object. This is an array of all currently opened documents. **UltraEdit** is the application object that all UltraEdit operations will be based on. Using this information, this script iterates through all open documents writing a predefined header:


```

15 // Get the num of open documents.
16 var num_of_docs = UltraEdit.document.length;
17
18 var dashes = "// -----";
19 dashes += "-----\r\n";
20
21 // Enumerate through all open documents and add the header.
22 var index;
23 for (index = 0; index < num_of_docs; index++) {
24     UltraEdit.document[index].top();
25     UltraEdit.document[index].write(dashes);
26     UltraEdit.document[index].write("// Script Name: \r\n");
27     UltraEdit.document[index].write("// Creation Date: \r\n");
28     UltraEdit.document[index].write("// Last Modified: \r\n");
29     UltraEdit.document[index].write("// Copyright (c)2007\r\n");
30     UltraEdit.document[index].write("// Purpose: \r\n");
31     UltraEdit.document[index].write(dashes);
32 }

```

For information regarding the commands supported for scripting, please see the [Scripting commands](#) Help topic.

Scripting commands



Scripting in UltraEdit/UEStudio is enabled through embedding of the JavaScript engine. This allows users to enjoy the power and flexibility of the full JavaScript language while using the commands specified below to specifically interact with the editor (Application Object commands) or documents open in the editor (Document Object commands). Scripts may be edited in UltraEdit/UEStudio with integrated syntax highlighting for JavaScript in the default wordfile:

```

25 // Do operation number of requested iterations.
26 var x = 0;
27 while (x < num) {
28     UltraEdit.document[0].findReplace.find(str);
29     UltraEdit.document[0].selectLine();
30     UltraEdit.document[0].copy();
31     UltraEdit.activeDocument.write(str + "\r\n");
32     x++;
33 }

```

The scripting engine supports the core functionality of JavaScript 1.7. Further information on JavaScript may be found on the associated Mozilla site (<http://developer.mozilla.org/en/docs/JavaScript>).

For example, this script will generate a sequence of numbers and write them to the active document:

```
function recall(num) {  
  
    UltraEdit.activeDocument.write(num + "\r\n");  
  
}  
  
function num() {  
  
    var i = 0, j = 1, n = 0;  
  
    while (n < 10) {  
  
        recall(i);  
  
        var t = i;  
  
        i = j;  
  
        j += t;  
  
        n++;  
  
    }  
  
}  
  
num();
```

Further demo scripts may be found in the installation directory in the "scripts" subdirectory. For a short scripting tutorial please [click here](#).

Including scripts in scripts

If desired, users may include an external script in a script by reference as shown below:

```
// include externalScript.js
```

or

```
// include C:\full path\to external\script\externalScript.js
```

The **include** command must be preceded by a line comment. If a users wishes to exclude the included script for debugging purposes, the include should be preceded by a doubled line comment, i.e.:

```
// // include externalScript.js
```

Please note that scripts are executed as soon as the inclusion is processed, but inclusions are processed prior to the active script. If an include is inserted into the middle of a script file, it will actually execute **prior** to the script it is included in. Where users desire to build complex scripts in a modular fashion from smaller scripts, the best practice would be to create a master script file that calls the included scripts, i.e.:

```
// include script1.js
```

```
// include script2.js
```

```
// include script3.js
```

Default Variable Values

UltraEdit supports several variable values that are initialized by default every time a script starts in UltraEdit:

Column Mode is always off.

Hex Mode is always off.

Insert Mode is always on.

The regular expression engine is always set to Perl.

These items are set every time a script runs.

Application Object commands

UltraEdit is the application object that all UltraEdit operations will be based on. The following commands act on the editor itself rather than the active document. Unless other parameters are noted, all Application Object commands must be invoked using the following format:

UltraEdit.commandName();

The table below shows the Application Object commands:

Commands	Parameters*	Description
activeDocumentIdx	** This is a READ ONLY property.	Returns index of active document in the document array. Example: var adl = UltraEdit.activeDocumentIdx;
clearClipboard		Clears active clipboard.
clipboardContent	Text in quotes (") if used to set clipboard content.	Returns content of active clipboard. Example: var clip = UltraEdit.clipboardContent; May also be used to SET the content of the active clipboard. Example: UltraEdit.clipboardContent = "Hello World!";
clipboardIdx	** This is a READ ONLY property.	Returns index of active clipboard. Example: var clip = UltraEdit.clipboardIdx;
closeFile	File Path Save Mode 0 - prompt to save 1 - save and close 2 - close without save	Closes active file. Example: UltraEdit.closeFile("C:\\temp\\test.txt",2); *Note: Any backslash used in parameters must be escaped as shown above (i.e. "\\" rather than "\").
columnMode	** This is a READ ONLY property.	Returns BOOLEAN value indicating if column mode is active. Example: var columnActive = UltraEdit.columnMode;
columnModeOff		Turn column mode off. Example: UltraEdit.columnModeOff(); *Note: The current column mode state will be restored once the script has completed processing.
columnModeOn		Turn column mode on. Example: UltraEdit.columnModeOn();
decryptFilePrompt		Present Decrypt File dialog. Example: UltraEdit.decryptFilePrompt();

encryptFilePrompt		Present Encrypt File dialog. Example: UltraEdit.encryptFilePrompt();
frInFiles	<p>.directoryStart string full path of directory where search should begin</p> <p>.encoding int value: Encoding value to be used for search</p> <p>.filesToSearch int value: 0 - Files Listed 1 - Open Files 2 - Favorite Files 3 - Project Files 4 - Solution Files</p> <p>.ignoreHiddenSubs boolean true/false</p> <p>.matchCase boolean true/false</p> <p>.matchWord boolean true/false</p> <p>.regExp boolean true/false</p> <p>.searchSubs boolean true/false</p> <p>.unicodeSearch boolean true/false</p> <p>.useEncoding boolean true/false</p> <p>.searchInFilesTypes string files/types to search in</p> <p>.find("string to find");</p> <p>.replace("string to find", "string to replace");</p> <p>** Only used with replace **</p> <p>.logChanges boolean true/false</p> <p>.preserveCase boolean true/false</p> <p>** Only used without replace **</p> <p>.displayLinesDoNotMatch boolean true/false</p> <p>.reverseSearch boolean true/false</p> <p>.useOutputWindow boolean true/false</p>	<p>Searches through specified files to find the string in quotes "" based on the parameters specified.</p> <p>Find in Files Example:</p> <p>UltraEdit.frInFiles.directoryStart = "c:\\temp\\";</p> <p>UltraEdit.frInFiles.searchInFilesTypes = "*.txt";</p> <p>UltraEdit.frInFiles.useOutputWindow = true;</p> <p>UltraEdit.frInFiles.find("3939");</p> <p>Replace in Files Example:</p> <p>UltraEdit.frInFiles.directoryStart = "c:\\temp\\";</p> <p>UltraEdit.frInFiles.searchInFilesTypes = "*.txt";</p> <p>UltraEdit.frInFiles.useOutputWindow = true;</p> <p>UltraEdit.frInFiles.replace("3939", "7878");</p>
getString	String in quotes (") used in the prompt dialog	Prompts user for string to insert at current location. The "prompt" is the prompt or question that is displayed to the user when the script is run. Examples:

	Optional flags: 0 int 1 int return value	UltraEdit.getString("What is your name?"); or var str = UltraEdit.getString("What is your name?",1); If the int value "1" is used the string entered will not be written to the active file, but will be saved as the variable value and used in the running script.
getValue	String in quotes (") used in prompt dialog Optional flags: 0 int 1 int return value	Prompts user for a value to insert at current location. The "prompt" is the prompt or question that is displayed to the user when the script is run. Examples: UltraEdit.getValue("How old are you?"); or var str = UltraEdit.getValue("How old are you?",1); If the int value "1" is used the string entered will not be written to the active file, but will be saved as the variable int value and used in the running script.
insertMode		Change text editing mode to insert mode for characters typed.
insOvrMode	** This is a READ ONLY property.	Returns BOOLEAN value indicating if insert mode is on. Example: var insertActive = UltraEdit.insOvrMode;
messageBox	Message text in quotes (") Title text in quotes (") (optional)	Presents message dialog with "OK" button. Example: UltraEdit.messageBox("Can't complete process", "Process Abort");

newFile		Open a new blank file.
open	File name in quotes ("")	<p>Open the specified file. The file name must be in quotes. "^c" may be used and UltraEdit will replace this with the contents of the clipboard. Example:</p> <pre>UltraEdit.open("c:\\temp\\test.txt");</pre> <p>or</p> <pre>UltraEdit.open('^c');</pre> <p>or</p> <pre>UltraEdit.open("c:\\temp\\^c");</pre> <p>or</p> <pre>UltraEdit.open("FTP::myserver.com\\home/my path/public_html\\index.html");</pre>
overStrikeMode		Change text editing mode to overstrike mode for characters typed.
perlReOn		Switch regular expressions to Perl compatible style regular expressions.
regexMode	** This is a READ ONLY property.	<p>Returns a number indicating active regular expression type.</p> <p>0 = UltraEdit expressions</p> <p>1 = Unix expressions</p> <p>2 = Perl expressions</p> <p>Example:</p> <pre>var regexType = UltraEdit.regexMode;</pre>
runTool	String in quotes ("") specifying the case sensitive menu name of tool to run.	<p>Run a tool. The tool must be configured from Tool Configuration in the Advanced menu. Example:</p> <pre>UltraEdit.runTool("Script Tool");</pre>
save		Save active file.
saveAll		Save all active files.
saveAs	File name in quotes ("")	<p>Save the active file as the filename specified. The filename must be in quotes. "^s" may be used in the filename. UltraEdit will replace this with the currently selected text in the active Window. Likewise, "^c" may be used and UltraEdit will replace this with the contents of</p>

		<p>the clipboard.</p> <p>UltraEdit.saveAs("c:\\temp\\test.txt");</p> <p>or</p> <p>UltraEdit.saveAs("^s");</p> <p>or</p> <p>UltraEdit.saveAs("^c");</p>
selectClipboard	Clipboard number (0-9)	<p>Select the specified clipboard 0 = Windows clipboard and 1-9 specifies user clipboards. Example:</p> <p>UltraEdit.selectClipboard(2);</p>
ueReOn		Switch regular expressions to UltraEdit style regular expressions.
unixReOn		Switch regular expressions to Unix style regular expressions.

Document Object commands

document is a JavaScript array object which is a property of the UltraEdit application object. This is an array of all currently opened documents. The activeDocument parameter may be used to specify that output should be written to the active file or users may specify a file's index based on file tab order (i.e. document[0], ... document[8]). For example:

```
UltraEdit.activeDocument.write("test");
```

would write the word "test" to the active file, while the following:

```
UltraEdit.document[4].write("test");
```

would allow the user to have multiple files open and write the specified text to the fifth file (based on file tab order) currently open for editing.

Scripts may be commented for testing or documentation using "//".

Once a script has been created, it may be edited. Please note, that "^c" and "^s" may be used with many script commands and will be replaced with the contents of the clipboard (^c) and the text currently selected (^s) when used. This allows users to create a script that may reference a specific string and replace this with one of these two items to allow the string to be dynamically "specified" as the script is run. The following commands act on a document currently open for editing. Unless other parameters are noted, all Document Object commands must be invoked using the following format:

```
UltraEdit.activeDocument.commandName();
```

The table below shows the Document Object commands:

Commands	Parameters*	Description
ansiToOem		Convert file from ANSI to OEM. Example: UltraEdit.activeDocument.ansiToOem();
ASCIIToUnicode		Convert file from ASCII to Unicode.
bottom		Jump to end of file.
codePage	** This is a property of the active/specified document.	Returns value of code page for active document. Example: var cp = UltraEdit.activeDocument.codePage; May be used to set code page to be used for active document. Example: UltraEdit.open("C:\\temp\\korean_file.txt") UltraEdit.activeDocument.codePage = 949;
collapseAll		Collapse all foldable text in active file. Example: UltraEdit.activeDocument.collapseAll();
columnCenterJustify		Center justify selected columns.
columnCut	Numeric value of columns to cut	Cut in column mode the selected columns or the specified number of columns from current cursor position to bottom of the file. Value "0" must be used to cut the selected columns.
columnDelete	Numeric value of columns to delete	Delete in column mode the selected columns or the specified number of columns from current cursor position to bottom of the file. Value "0" must be used to delete the selected columns.
columnInsert	String in quotes ("")	Insert string between quotes into selected columns.
columnInsertNum	Start Number int Increment int LeadingZero boolean true/false	Insert number in selected columns. If there is no selection the insertion will run from the cursor location to the last line of the file. Example:

	HEX boolean true/false	UltraEdit.activeDocument.columnInsertNum(2, 3, false, true);
columnLeftJustify		Left justify selected columns.
columnRightJustify		Right justify selected columns.
copy		Copy selected text to the clipboard. If there is no selection, the line at the current cursor location will be copied if Enable copy/append of current line when no selection is active is configured.
copyAppend		Copy selected text and append it to the clipboard. If there is no selection the line at the current cursor location will be copied if Enable copy/append of current line when no selection is active is configured.
copyFilePath		Copy the active file path/name to the clipboard.
currentChar	** This is a READ ONLY property of the active/specified document.	Returns value of character at cursor. Example: var char = UltraEdit.activeDocument.currentChar;
currentColumnNum	** This is a READ ONLY property of the active/specified document.	Returns value of current column number. The first column is numbered as "1". Example: var col = UltraEdit.activeDocument.currentColumnNum ;
currentLineNum	** This is a READ ONLY property of the active/specified document.	Returns value of current line number. Example: var lineNum = UltraEdit.activeDocument.currentLineNum;
currentPos	** This is a READ ONLY property of the active/specified document.	Returns value of current position in bytes from the beginning of the file. Example: var pos = UltraEdit.activeDocument.currentPos;
cut		Cut the selected text from the file to the clipboard. If there is no selection the line at the current cursor location will be cut.
cutAppend		Cut the selected text from the file and append it to the clipboard. If there is no selection the line at the current cursor location will be cut.
decodeBase64		Convert selected text from Base64. Example: UltraEdit.activeDocument.decodeBase64();
delAllBookmarkLines		Delete currently bookmarked lines in active document. Example: UltraEdit.activeDocument.delAllBookmarkLines();
delAllHiddenLines		Delete lines currently hidden/folded in active

		document. Example: UltraEdit.activeDocument.delAllHiddenLines() ;
deleteText		Delete current character or selected text.
deleteLine		Delete the current line.
deleteToEndOfLine		Delete from the current cursor position to the end of the line.
deleteToStartOfLine		Delete from the current cursor position to the start of the line.
dosToMac		Convert the file (line terminators) to MAC format.
dosToUnix		Convert the file (line terminators) to UNIX format.
dupeLine		Inserts duplicate of active line below cursor.
encodeBase64		Convert selected text to Base64. Example: UltraEdit.activeDocument.encodeBase64();
encoding	** This is a READ ONLY property of the active/specified document.	Returns value of encoding for active document. Example: var enc = UltraEdit.activeDocument.encoding;
endSelect		Stop selecting text (see startSelect for details).
expandAll		Expands all folded text in active file. Example: UltraEdit.activeDocument.expandAll();
fileSize	** This is a READ ONLY property of the active/specified document.	Returns size of referenced file in bytes. Example: var size = UltraEdit.activeDocument.fileSize;
findReplace	<div> <div> .matchCase boolean true/false .matchWord boolean true/false .mode int value: 0 - Current File 1 - In Selection 2 - All Open Files .regExp boolean true/false .searchAscii boolean true/false .searchDown boolean true/false .searchInColumn boolean true/false .fromCol int default: 0 .toCol int default:- </div> </div>	Find the string in quotes "" based on the parameters specified. Example: UltraEdit.activeDocument.findReplace.matchWord = true; UltraEdit.activeDocument.findReplace.find("3 939"); or UltraEdit.document[0].findReplace.matchWor

	<p>1</p> <p>.find("string to find"); .replace("string to find", "string to replace");</p> <p>** Only used with replace **</p> <p>.preserveCase boolean true/false .replaceAll boolean true/false .replaceInAllOpen boolean true/false *supersedes .mode in replace .selectText boolean true/false *replaces only in selected text</p>	<p>d = true;</p> <p>UltraEdit.document[0].findReplace.matchCase = true;</p> <p>UltraEdit.document[0].findReplace.replace("Copper", "Silver");</p> <p>Please note: all properties once set are active for all following finds and replaces until the property is set again to a different value.</p>
fromEBCDIC		Convert text from EBCDIC format.
gotoBookmark	Index of bookmark to jump to or -1 to go to next bookmark	<p>Jump to the next/specified bookmark. The indexes start with 0. If a user enters an index that is greater than the actual number of bookmarks then it is automatically directed to the first bookmark (index 0). Example:</p> <p>UltraEdit.activeDocument.gotoBookmark(0);</p>
gotoBookmarkSelect	Index of bookmark to jump to or -1 to go to next bookmark	<p>Jump to the next/specified bookmark and select text from cursor position to bookmark. The indexes start with 0. If a user enters an index that is greater than the actual number of bookmarks then it is automatically directed to the first bookmark (index 0). Example:</p> <p>UltraEdit.activeDocument.gotoBookmarkSelect(0);</p>
gotoLine	Numeric value of line and column number to jump to	<p>Jump to the specified line and column number. Use line number 0 to jump to the specified column in the current line. Example:</p> <p>UltraEdit.activeDocument.gotoLine(1,5);</p>
gotoLineSelect	Numeric value of line and column number to jump to	<p>Jump to specified line number and column number and select text from cursor position to line/column. Use line number 0 to jump to the specified column in the current line while selecting text. Example:</p> <p>UltraEdit.activeDocument.gotoLineSelect(1,5);</p>
gotoPage	Numeric value of page to jump to	Jump to specified page number.

		Example: <code>UltraEdit.activeDocument.gotoPage(5);</code>
<code>gotoPageSelect</code>	Numeric value of page to jump to	Jump to specified page number and select text from cursor position to page number. Example: <code>UltraEdit.activeDocument.gotoPageSelect(5);</code>
<code>hexDelete</code>	Numeric value specifying number of bytes to delete	Delete the specified number of bytes from the file.
<code>hexInsert</code>	Numeric value specifying number of bytes to insert.	Insert the specified number of bytes into the file. This will insert spaces (HEX 20).
<code>hexMode</code>	** This is a READ ONLY property of the active/specified document.	Returns a BOOLEAN value indicating if Hex mode is active. Example: <code>var hexActive = UltraEdit.activeDocument.hexMode;</code>
<code>hexOff</code>		Turn Hex mode off - switch to text mode.
<code>hexOn</code>		Turn Hex mode on.
<code>hideOrShowLines</code>		Hide the selected lines, or if hidden, show the lines hidden at the line of the cursor.
<code>insertLine</code>		Inserts blank line below current cursor position.
<code>insertPageBreak</code>		Insert a form feed/page break character at the current cursor position in the file.
<code>insertTemplate</code>	Index of template	Insert the specified template into the file. Example: <code>UltraEdit.activeDocument.insertTemplate(0);</code>
<code>invertCase</code>		Invert the case of selected text.
<code>isCharGt</code>	"character"	This checks if the current character at the cursor position is greater than the specified character. Example: <code>if (UltraEdit.document[1].isCharGt('k')){ //do these commands... } else { //do these commands... }</code>
<code>isChar</code>	"string"	This checks if the current character at the cursor position is the character specified. Example:

		<pre> if (UltraEdit.document[1].isChar('k')){ //do these commands... } else { //do these commands... } </pre>
isColNum	number	<p>This checks if the current cursor position is the specific column number. Example:</p> <pre> if (UltraEdit.activeDocument.isColNum(13)){ //do these commands... } else { //do these commands... } </pre>
isColNumGt	number	<p>This checks if the current cursor position is greater than the specific column number. Example:</p> <pre> if (UltraEdit.activeDocument.isColNumGt(13)){ //do these commands... } else { //do these commands... } </pre>
isEof		<p>This checks if the current cursor position is at the end of file. Example:</p> <pre> if (UltraEdit.document[1].isEof()){ //do these commands... } else { //do these commands... } </pre>
isExt	"string"	This checks if the file extension of the

		<p>active file matches the specified string. Example:</p> <pre> if (UltraEdit.document[1].isExt("txt")){ //do these commands... } else { //do these commands... } </pre>
isFound		<p>This checks the results from the last find command in the script and will conditionally execute further commands based on the result. Example:</p> <pre> UltraEdit.activeDocument.findReplace.find("string"); if (UltraEdit.activeDocument.isFound()){ //do these commands... } else { //do these commands... } </pre>
isFTP		<p>This checks if the current file is a file loaded via FTP/SFTP and not a local/network file.</p> <pre> if (UltraEdit.document[1].isFTP()){ //do these commands... } else { //do these commands... } </pre>
isHexModeOn		<p>This checks if the active file is currently set to Hex/binary mode.</p> <pre> if (UltraEdit.activeDocument.isHexModeOn()){ </pre>

		<pre>//do these commands... } else { //do these commands... }</pre>
isName	"string"	<p>This checks if the active file name (not path or extension) matches the specified string. Example:</p> <pre>if (UltraEdit.document[1].isName("foo")){ //do these commands... } else { //do these commands... }</pre>
isNotFound		<p>This checks the results from the last find command in the script and will conditionally execute further commands based on the result. Example:</p> <pre>UltraEdit.activeDocument.findReplace.find("string"); if (UltraEdit.activeDocument.isNotFound()){ //do these commands... } else { //do these commands... }</pre>
isSel		<p>This checks if there is currently any selected text within the active file. Example:</p> <pre>if (UltraEdit.document[1].isSel()){ //do these commands... } else { //do these commands... }</pre>

key	BACKSPACE DEL DOWN ARROW END HOME LEFT ARROW PGDN PGUP RIGHT ARROW UP ARROW CTRL+END CTRL+HOME CTRL+LEFT ARROW CTRL+RIGHT ARROW	Insert a key command into the active file. Generally used for navigation in the file and for backspace or delete. The "CTRL+" modifier may be used as with normal editing to modify the command. For text input the "write" command is used, not the "key" command. Examples: <pre>UltraEdit.activeDocument.key("BACKSPACE");</pre> <pre>UltraEdit.activeDocument.key("CTRL+RIGHT ARROW");</pre>
length	** This is a READ ONLY property of the active/specified document.	Returns number of active documents. Example: <pre>var num_of_docs = UltraEdit.document.length;</pre>
lineTerminator	** This is a READ ONLY property of the active/specified document.	Returns a numeric value indicating line terminator type in active document. Example: <pre>var lt = UltraEdit.activeDocument.lineTerminator;</pre> <p>Supported values:</p> <p>-2 = MAC but content of file contains currently DOS line terminators</p> <p>-1 = UNIX but content of file contains currently DOS line terminators</p> <p>0 = DOS</p> <p>1 = UNIX</p> <p>2 = MAC</p>
matchBrace		Find next matching brace and select the text between them.
moveLineDown		Move current line downward one line in active document. Example: <pre>UltraEdit.activeDocument.moveLineDown();</pre>
moveLineUp		Move current line upward one line in active document. Example: <pre>UltraEdit.activeDocument.moveLineUp();</pre>

oemToAnsi		Convert active file from OEM to ANSI.
paste		Paste the contents of the clipboard into the file.
path	** This is a READ ONLY property of the active/specified document.	Returns full path of specified file. Example: var text = UltraEdit.activeDocument.path;
previousBookmark		Jump to the previous bookmark. Example: UltraEdit.activeDocument.previousBookmark();
previousBookmarkSelect		Jump to previous bookmark and select text from cursor position to bookmark. Example: UltraEdit.document[1].previousBookmarkSelect();
reIndentSelection		Re-indents currently selected text. Example: UltraEdit.activeDocument.reIndentSelection();
returnToWrap		Convert hard returns to word wrap in current selection.
selectAll		Select all text in the file.
selection	** This is a READ ONLY property of the active/specified document.	Returns currently selected text. Example: var text = UltraEdit.activeDocument.selection;
selectLine		Select all text on active line.
selectToBottom		Select all text from the current position to the end of file.
selectToTop		Select all text from the current position to the top of file.
selectWord		Select the current word (same as double clicking a word).
setActive		Sets specified document as active document. Example: UltraEdit.document[1].setActive();
sort	.ascending boolean true/false .col1Start int start column key 1 .col1End int end column key 1 .col2Start int start column key 2	Sort the file, or selected text according to specified parameters. Example: UltraEdit.activeDocument.sort.ascending = true; UltraEdit.activeDocument.sort.ignoreCase = false;

	<div> <div>.col2End</div> <div>int end column key 2</div> </div> <div> <div>.col3Start</div> <div>int start column key 3</div> </div> <div> <div>.col3End</div> <div>int end column key 3</div> </div> <div> <div>.col4Start</div> <div>int start column key 4</div> </div> <div> <div>.col4End</div> <div>int end column key 4</div> </div> <div> <div>.ignoreCase</div> <div>boolean true/false</div> </div> <div> <div>.removeDuplicates</div> <div> int 0 - false 1 - all keys match 2 - any keys match </div> </div> <div> <div>.remKey1</div> <div>boolean true/false</div> </div> <div> <div>.remKey2</div> <div>boolean true/false</div> </div> <div> <div>.remKey3</div> <div>boolean true/false</div> </div> <div> <div>.remKey4</div> <div>boolean true/false</div> </div> <div> <div>.type</div> <div> int 0 - character order 1 - numeric sort 2 - use locale 3 - alt. sort </div> </div>	<div>UltraEdit.activeDocument.sort.removeDuplicates = 1;</div> <div>UltraEdit.activeDocument.sort.remKey1 = true;</div> <div>UltraEdit.activeDocument.sort.remKey2 = true;</div> <div>UltraEdit.activeDocument.sort.type = 0;</div> <div>UltraEdit.activeDocument.sort.col1Start = 1;</div> <div>UltraEdit.activeDocument.sort.col1End = 15;</div> <div>UltraEdit.activeDocument.sort.col2Start = 35;</div> <div>UltraEdit.activeDocument.sort.col2End = 50;</div> <div>UltraEdit.activeDocument.sort.sort();</div>
<div>sortAsc</div> <div>sortDes</div>	<div> Sort Type 0 - Sort based on character order. 1 - Sort based on numeric value, not character order. 2 - Specifies that the sort should be locale specific. 3 - Specifies that the sort should use the alternate sort method. With the alternate sort method only a single sort key may be used. Ignore Case boolean true/false Remove Duplicates boolean true/false Sort Keys int Up to four pairs of start/end keys may be specified. </div>	<div>Sort the file, or selected text in ascending or descending order.</div> <div>Example:</div> <div>UltraEdit.activeDocument.sortAsc(0, true, true, 1, -1);</div> <div>Example:</div> <div>UltraEdit.activeDocument.sortDes(1, true, false, 4, 8);</div>

spacesToTabs		Convert (leading) spaces within the file to tabs. This is based on the Tab Stop Value defined under Word Wrap/Tab Settings in Configuration. If the Tab Stop Value is set to three, a group of three spaces together would be required to convert to a tab. Fewer than three spaces together would not be converted.
spacesToTabsAll		Convert all spaces within the file to tabs. This is based on the Tab Stop Value defined under Word Wrap/Tab Settings in Configuration. If the Tab Stop Value is set to three, a group of three spaces together would be required to convert to a tab. Fewer than three spaces together would not be converted.
startSelect		Start selection. This turns the selection mode on. Any cursor movement or positioning will be with selection on and the text is selected. endSelect will stop the selection mode. The selected text will remain selected until another command causes it not to be selected as with normal editing.
tabsToSpaces		Convert all tabs within the file to spaces.
timeDate		Insert the time and date into the file at cursor location.
toCaps		Capitalize each word in the selected text.
toEBCDIC		Convert text to EBCDIC format.
toggleBookmark		Set or remove a bookmark at the current line.
toLower		Convert the selected text to lower case.
top		Jump to top of file.
toUpper		Convert the selected text to upper case.
trimTrailingSpaces		Trim trailing spaces from each line of current file.
unicodeToASCII		Convert file from Unicode to ASCII.
unixMacToDos		Convert the active file (line terminators) from Mac/Unix to DOS format.
wrapToReturn	Column number at which wrap is performed. Zero for the column number indicates the wrap is to occur at the window boundary.	Convert selection from word wrap to hard returns. Example: UltraEdit.activeDocument.wrapToReturn (60);
write	Text to write in quotes ("")	Write specified text at cursor location. Example:

		<pre>UltraEdit.activeDocument.write("This is a test.");</pre> <p>or</p> <pre>UltraEdit.activeDocument.write("^c");</pre> <p>This would use the contents of the clipboard for the write command.</p>
xmlConvertToCRLF		Convert single-line XML file to indented XML format.

Output Window Object commands

outputWindow is a JavaScript array object which is a property of the UltraEdit application object. Unless other parameters are noted, all Output Window Object commands must be invoked using the following format:

```
UltraEdit.outputWindow.commandName();
```

The table below shows the Output Window Object commands:

Commands	Parameters*	Description
clear		Clears contents of output window. Example: <code>UltraEdit.outputWindow.clear();</code>
copy		Copies contents of output window to active clipboard. Example: <code>UltraEdit.outputWindow.copy();</code>
showOutput	boolean true/false	Determines visibility of user specific data that is written to the output window. Currently that only includes outputWindow.write(). Example: <code>UltraEdit.outputWindow.showOutput=false;</code>
showStatus	boolean true/false	Determines visibility of all status information in output window (script name, success/failure of

		script, and error information). Example: UltraEdit.outputWindow.showStatus=true;
showWindow	boolean true/false	Toggles visibility of output window. Example: UltraEdit.outputWindow.showWindow(true);
visible	** This is a READ ONLY property.	Returns a BOOLEAN value indicating if output window is visible. Example: UltraEdit.outputWindow.visible;
write	Text to write in quotes ("")	Write specified text to output window. This will only support one line at a time and may not include line terminators. Example: UltraEdit.outputWindow.write("This is a test.");

Tab Indentation



UltraEdit provides multi-line indentation and out-denting with the tab key. If multiple lines are selected, the Tab Key will indent these lines with the equivalent of one tab stop. If the Shift Key is pressed at the same time as the Tab Key the lines will be out-dented by one tab stop.

Text Selection/Highlighting



Holding down the "Shift" key while [cursoring with the arrows](#) highlights the text the cursor passes showing that the highlighted text is selected for edit. Holding down the CTRL and the "Shift" key while using the left or right cursor arrows highlights one complete word at a time for editing, moving to the beginning of the previous/next word.

Holding down the CTRL and the "Shift" key while using the up or down cursor arrows highlights all text between the starting and ending position of the cursor. Holding down the CTRL and the "Shift" key while using the "Home" key will highlight all text from the beginning cursor position to the beginning of the document being edited.

Holding down the CTRL and the "Shift" key while using the "End" key will highlight all text from the beginning cursor position to the end of the document being edited. When the "Shift" Key is pressing at the same time as performing the following functions the current selection (if present), or a new selection will be extended to select text up to the cursor position following the command:

Find Next/Previous

Goto

Goto Bookmark

The "Shift" key should be pressed at the time the action for command takes place, not on the command that shows the dialog. For example, if Ctrl+G displays the Goto Dialog and OK is clicked to perform the [Goto](#), the "Shift" Key is pressed when the OK is clicked, NOT when the Ctrl+G is pressed to display the dialog.

Unicode and UTF-8 Support



UltraEdit provides support for Unicode (16-Bit wide character) files, and for UTF-8 files and allows direct editing of Unicode files and UTF-8 files as well as conversion routines between ASCII/ANSI and Unicode or UTF-8. UltraEdit attempts to detect the file type when the file is loaded. It will look for the FF FE marker for Unicode files. For UTF-8 it will look for one of the following three occurrences:

- 1) File marker (BOM) EF, BB, BF.
- 2) String occurrences "charset=utf-8" or "encoding=utf-8"
- 3) It will look for the occurrence of valid UTF-8 multi-byte characters in the first 64KB of a file.

If the file is found to be Unicode, it will be treated as such and the status bar will indicate this with "U-" being displayed in front of the file type in the status bar.

If the file is found to be UTF-8, it will be treated as such and converted internally to Unicode (16-Bit) for editing. The status bar will indicate this with "U8-" being displayed in front of the file type in the status bar. When the file is saved, it will be converted back from Unicode to UTF-8 and saved in this format.

Handling internally as Unicode allows the Unicode character messages from the OS to be used for character input. Additionally, the WM_UNICHAR message used in some versions of Windows and with some applications is supported by UltraEdit allowing Unicode characters to be entered on versions of the OS that do not support Unicode natively (Windows 95/98/Me).

The following conversions to and from Unicode/UTF-8 are available:

ASCII to Unicode	converts from ASCII to Unicode
UTF-8 to Unicode	converts from UTF-8 to Unicode (16-Bit)
Unicode to ASCII	converts from Unicode to ASCII
UTF-8 to ASCII	converts from UTF-8 to ASCII
ASCII to UTF-8 (Unicode Editing)	converts from ASCII to UTF-8 with the file internally in UNICODE format for editing
Unicode/UTF-8 to UTF-8 (Unicode)	converts the file from either Unicode or UTF-8 (non Unicode internally) to UTF-8 with the file internally in UNICODE format for editing
Unicode/ASCII/UTF-8 to UTF-8 (ASCII)	converts from Unicode, ASCII or UTF-8 (Unicode format internally) to UTF-8 but leaves the file in non-unicode (ASCII display) format for editing.

Unicode file support within UltraEdit is provided on Windows 95, 98, Me, NT and 2000.

As not all operating systems natively support Unicode certain functions convert from Multi-Byte to Unicode or from Unicode to Multi-Byte. If the active code page (see Windows documentation/help) does not support the characters then the conversion will not convert correctly.

The functions that convert between Multi-Byte and Unicode include Find/Replace and Column Insert.

UE3 Functionality



UE3 refers to UltraEdit customized for installation on a U3 smart drive. UE3 supports the functionality of UltraEdit except as follows:

- File associations are not supported.
- Shell integration (context menu access from Windows Explorer) is not supported as the U3 conventions forbid access to the parts of the registry which must be modified to support shell integration.

In nearly all other respects UE3 allows the use of UltraEdit as if it was installed on a standard desktop hard drive allowing users access to the most needed features of UltraEdit - macros, scripting, syntax highlighting, column mode, code folding, UNICODE support, FTP, Telnet and more - all on your U3 smart drive.

The primary difference is that when UE3 is shut down and the U3 smart drive is ejected, any references to files/settings that don't reside on the U3 smart drive are removed from the UE3 INI file (ue3.ini) when this is written back to the U3 smart drive. For example:

- Files added to the Favorite files list during an editing session that belong to the host system will be removed from the application when the U3 drive is ejected.
- Open files (except in the case of files opened via FTP) will not be remembered between sessions.

Note: U3, the U3 smart Logo, and all U3 formative trademarks are trademarks or registered trademarks of U3 LLC worldwide.

Menus/Toolbars

Menu Configuration



UltraEdit allows any command available from the menus to be available on the toolbar. It also allows multiple user menus to be configured. If you right click on a menu, a context menu will appear with the following options:

Basic

Advanced

Manage Menu Configurations

Customize Menu

The Advanced option should be selected initially and this will provide all menus and toolbars. If Basic is selected, a subset of the default menus and toolbars will be displayed for use. If desired, users may press ALT+F8 to present a context menu similar to that described above. This is a single switch Profile menu which will allow users to change menus and toolbars at the same time (i.e. switch to "Basic" or "Advanced" for both menus and toolbars in a single operation).

Menu Configurations

If Manage Menu Configurations is selected from the context menu the Menu Configurations dialog is presented. If a name is entered in the New Name field then the currently active menus would be saved under this configuration name. If a preconfigured menu name is selected (such as Basic) and the Reset button is pressed, this menu configuration would be restored to its default state. If the Reset All button is pressed, all menu configurations shipped with UltraEdit would be restored to their default state and user-configured menu configurations would be deleted.

The user may then right click on a menu and select Customize and modify the menus for this configuration as described below. If Customize Menu is selected the Menu Configuration dialog is presented and users may customize the currently active menus as described below.

Menu Configuration Dialog

The Menu Configuration dialog consists of two panes labeled Menus and Commands along with a text entry field for New Menu/Submenu. If Use check marks on menus is selected, check marks will be shown for selected menu items rather than the depressed menu icons. This option helps resolve some issues seen with screen readers for disabled users. Also included in the dialog are up/down and right/left arrows and an Edit button. These items and their functionality are all described below:

Menus/Commands Configuration Panes

The currently active menus are displayed in the Menus pane with the items currently assigned to them in an Explorer interface. Each menu or submenu heading is indicated by a line terminated in a square to the left of the menu/submenu name. Double clicking on the heading indicator will cause the included items to be expanded/collapsed appropriately.

If a menu heading or item is selected and the Edit button is pressed the name for the menu/item may be modified. Pressing Return on the keyboard will save the menu item change and pressing the OK button will commit the change so that this is reflected in the menus.

When a menu heading is selected the entire menu may be moved to a new location using the up/down and right/left arrows. When a menu item is selected that item may be moved to a new location within the current menu by pressing the up/down buttons. A menu item may be moved to the extreme top or bottom of its current menu. If a menu item is part of a submenu, it cannot be moved outside the submenu. If a menu item is part of a main menu, moving it up or down will cause it to "skip" any submenus it encounters.

The Commands pane displays all the functions currently available for use in the menus. To move an item from the Commands pane to the Menus pane, click on the desired Commands item and press the left arrow in the center of the dialog to add the selected item. Selected menu items will be added directly below the currently selected item in the Menus pane. If there is no selection in the Menus pane, the item will be added at the top of the first menu listed. Once items are added they may be repositioned within a menu as desired using the up/down arrows.

New Menus

To create a new menu/submenu, type the menu/submenu name (i.e. "Test1") in the New Menu field and press the left arrow button to the right of the New Menu field. This will add a new, empty menu to the bottom of the Menus list. Menus may be removed by selecting them and pressing the right arrow button in the center of the dialog.

The data for the configurable menus is stored in the % APPDATA% directory by default under IDM Comp\UltraEdit in the uedit32.mnu file. (For UESudio this would be under the UESudio directory rather than the UltraEdit directory).

When a menu is modified, its name is stored in the menu list followed by the date, e.g. "Advanced Saved [03.09.06]". Backups or saved versions of menus may be selected by simply right-clicking on a menu and selecting the desired copy.

Toolbar Configuration



UltraEdit/UEStudio allows any command available from the menus to be available on the toolbar. It also allows multiple user toolbars to be configured. If you right click on the toolbar a context menu will appear with the following options:

Basic

Advanced

Manage Toolbar Configurations

Customize Toolbar

Large Buttons

The Advanced option should be selected initially and this will provide all the standard toolbars. If Basic is selected, a subset of the default toolbars will be displayed for use. If desired, users may press ALT+F8 to present a context menu similar to that described above. This is a single switch Profile menu which will allow users to change menus and toolbars at the same time (i.e. switch to "Basic" or "Advanced" for both menus and toolbars in a single operation).

If Manage Toolbar Configurations is selected the Toolbar Configurations dialog is presented. If a name is entered in the New Name field then the currently active toolbar(s) would be saved under this configuration name. The user may then right click on the toolbar and select Customize and modify the toolbar(s) for this configuration as described below.

If Customize Toolbar is selected the Toolbars Configuration dialog is presented and users may customize the currently active toolbars as described below.

The Large Buttons item may be used to toggle the size of buttons in the active toolbars.

Toolbar Configuration Dialog

The Toolbar Configuration dialog consists of two panes labeled Toolbars and Commands along with a checkbox for Large Buttons and text entry fields for New Toolbar and Ext. Bmp. Also included in the dialog are up/down and right/left arrows. These items and their functionality are all described below:

Toolbars/Commands Configuration Panes

The toolbars which have been specified are displayed in the Toolbars pane with the items currently assigned to them in an Explorer interface. Clicking on the "-" to the left of a toolbar name will collapse that toolbar and clicking on the "+" will expand the toolbar to display the items included in it. If an item is selected in a toolbar, pressing the up/down arrow will move the item accordingly in the toolbar list. It is not possible to use the up/down arrows to move an item from one toolbar to another adjacent toolbar.

The Commands pane displays all the functions currently available for use in the toolbars. To move an item from the Commands pane to the Toolbars pane, click on the desired Commands item and press the left arrow in the center of the dialog to add the selected item. Selected toolbar items will be added directly below the currently selected item in the Toolbars pane. If there is no selection in the Toolbars pane, the item will be added at the top of the first toolbar listed. Once items are added they may be repositioned within a toolbar as desired using the up/down arrows.

New Toolbars

To create a new toolbar, type the toolbar name (i.e. "Test1") in the New Toolbar field and press the left arrow button to the right of the New Toolbar field. This will add a new, empty toolbar to the bottom of the Toolbars list. Toolbars may be removed by selecting them and pressing the right arrow button in the center of the dialog.

Ext. Bmp

If desired, new bitmaps may be associated with toolbar items (such as User Tools) so that this is displayed in the toolbar. To do so, select the toolbar item in the Commands pane and add it to the appropriate toolbar as described above. Select the item in the Toolbars pane and specify the path to the desired bitmap in the Ext. Bmp field. If you know exactly where this is located you may directly specify this and press Return to assign this bitmap to the selected toolbar item. If you're not sure where this is located, you may press the "..." button to browse to the desired bitmap and then press Return to assign this bitmap to the selected toolbar item.

Large Buttons

Selecting the Large Buttons checkbox will set ALL toolbars to use large buttons.

The data for the configurable toolbars is stored in the % APPDATA% directory by default under IDM Comp\UltraEdit in the uedit32.tbr file. (For UESTudio this would be under the UESTudio directory rather than the UltraEdit directory).


















When a toolbar is modified, its name is stored in the menu list followed by the date, e.g. "Advanced Saved [03.09.06]". Backups or saved versions of toolbars may be selected by simply right-clicking on a toolbar and selecting the desired copy.

HTML Toolbar



The HTML toolbar provides a quick method for inserting commonly needed code in HTML. The following functions are included:











	View Tag List	toggles visibility of HTML Tag List
	Bold	inserts tags around selection to render text bold
	Italic	inserts tags around selection to italicize selected text
	Underline	inserts tags around selection to underline selected text
	Increase Indents	inserts BLOCKQUOTE tags around selected text
	Bullets	inserts tags to create a bulleted list
	Numbering	inserts tags to create a numbered list
	Foreground Color	displays Color dialog so user may set the text foreground color. If no color is selected the appropriate tags will be added to the text and the color value will have to be typed by the user
	Background Color	displays Color dialog so user may set the text background color. If no color is selected the appropriate tags will be added to the text and the color value will have to be typed by the user
	Align Left	inserts tags around selection to left align text
	Align Center	inserts tags around selection to center align text

	Align Right	inserts tags around selection to right align text
	Justify	inserts tags around selection to justify text
	HTML Div	inserts DIV tag to create text label
	Button	inserts a button input tag and sets the cursor appropriately to type the button text
	Text Field	inserts single line textbox
	Text Area	inserts multi-line textbox (textarea)
	Password Field	inserts single line password textbox
	Hidden	inserts hidden textbox
	Checkbox	inserts a checkbox
	Radio Button	inserts a radio button
	Image	inserts IMAGE tag and sets cursor so the user may type the path to the image
	Form Image	presents open dialog for insertion of form image
	Anchor	inserts tags around selected text to create an anchor
	Close Tag	finds the nearest unclosed tag and closes it only supported by UESstudio, not by UltraEdit
	Convert Special Characters in Selected Text to HTML Entities	converts special characters in selected text to HTML entities (i.e. "ß" becomes "ß")
	Encode/Decode	converts selection to %xx where xx is equivalent to the hexadecimal number representing the character
	HTML Protector	protects HTML code (for example embedded JavaScript) against browsing

Tools Toolbar



The Tools toolbar provides a quick method for invoking tools for use in or along with UltraEdit. The following functions are included:

	Color Selector	presents the Color Selector dialog which may be used to insert color values in Decimal, HEX, HTML, or RGB format
	Style Builder	presents the Style Builder dialog which may be used to create styles in conformance with the CSS2 standard
	HTML Tidy	presents the HTML Tidy dialog which through which the user may specify the location for a configuration file (if options other than default are desired) and also configure custom options for HTML Tidy operations
	GUID Generator	presents a context menu for inserting a Globally Unique Identifier into the active document in one of four available formats
	Number Converter	converts a selected non-zero number to one of four formats: binary, octal, decimal, or hexadecimal
	Artistic Style Formatter	reformats source code (many configurable options). Please note: The Artistic Style EXE is installed in the GNU subdirectory and must be selected prior to its first use.
	* .NET Code Protector	presents the .NET Framework Code Protector dialog. This protects .NET executables against simple disassembly of source code. Works directly with the executable and doesn't need .NET framework installed.
	* Project Converter	converts a Visual Studio project to UESTudio format
	XMLint	invokes XMLint parser. For more information on commands and use please see http://xmlsoft.org/xmlint.html
	JavaScript Lint	invokes JavaScript Lint to check for mistakes in JavaScript source code. For more information see http://www.javascriptlint.com/









* Please note: These features are only supported in UESTudio, and not in UltraEdit.

Build Toolbar



If a project has been configured to be compilable (Project Settings dialog) the build toolbar may be used to execute compile/build functions on the files associated with the active project.

The Build toolbar includes the following items:






	* Project Settings	presents Project Settings dialog
	* Build Mode	sets Build mode to "Debug" or "Release"
	* Compile	saves the active document if needed and then compiles it. If a compiler is not selected or the compiler options are not set for the project the Select Compiler dialog is displayed, then the Project Settings dialog is displayed and then the Compile process is started.
	* Build	saves the active document if needed and builds the project target(s). If the compiler is not set for the project or project settings are not saved it will work in the same way as the Compile command.
	* Stop Build	stops running build process
	* Run App	runs the application created with the selected compiler or invokes the build process first if the application does not yet exist
	* Rebuild All	removes all files created by the compiler and then builds the project
	* Commands	presents a popup menu with a variable number of items depending on the compiler configuration that is used and which commands are set for this compiler's configuration .





* Please note: These features are only supported in UESstudio, and not in UltraEdit.

IntelliTip Toolbar



The IntelliTip toolbar provides a quick method for invoking items for use along with IntelliTips in UESstudio. The following functions are included:

	.source -> .header	displays context menu allowing user to toggle between source and associated header file
	File List	displays file list
	Display/ redisplay function tip	displays function tip for active function
	Symbols List	displays symbols from active file
	Keywords List	displays keywords from active file


	Display functions defined in this file	displays functions defined in active file
	Reparsing active document	force IntelliTips to reparsing active file
	Jump to ClassViewer	toggles File Tree View to display Tags pane
	Find Symbol definition in project files	displays IntelliTips list indicating where selected symbol is defined. Double-clicking item in list opens referenced file to definition point.

* Please note: These features are only supported in UESTudio, and not in UltraEdit.

Resource Editor Toolbar



The Resource Editor toolbar allows users to edit resources for the active project. The following functions are included:


	Test	presents active dialog in Design Mode
---	------	---------------------------------------







* Please note: These features are only supported in UESTudio, and not in UltraEdit.

Script Toolbar



The Script toolbar presents a submenu from which options related to Ruby and PHP coding may be selected. The following functions are included:

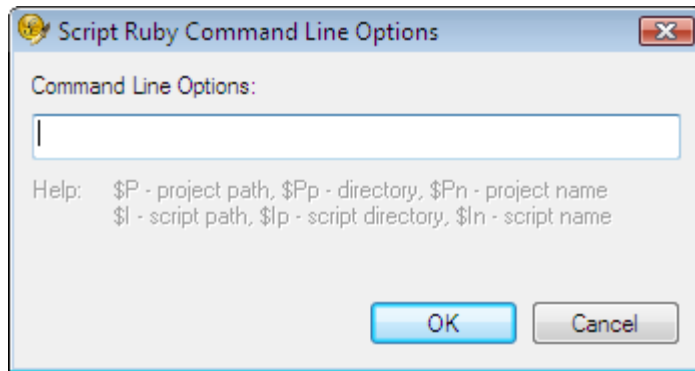
	Run Ruby Commands	<p>presents submenu from which the commands below may be selected</p> <p>Note: If the ruby.exe has not been installed, selecting any of the options below will result in an error dialog indicating that the ruby.exe could not be found. The latest version of the Ruby EXE for Windows may be downloaded</p>
---	-------------------	---

		here: http:// www.ruby-lang.org/
	Check Syntax	checks syntax of active Ruby script
	Run Script	runs active Ruby script
	Command Line Options	presents dialog (shown below) for specification of command line options
	PHP Commands	<p>presents submenu from which the commands below may be selected</p> <p>Note: The first time one of the commands below is selected a dialog is shown where the user must browse to locate the php.exe on the local system. If you have not installed PHP on your system please go to http://www.php.net/ downloads.php and download the latest version and follow the installation instructions.</p> <p>If PHP has not been installed on your system the functions below will not operate and will only prompt for the location of the php.exe.</p>
	Run Script	runs active PHP script
	Syntax Check	checks syntax of active PHP script
	Syntax Check All	checks syntax of all scripts in all open PHP files
	Parse Document	parses active file and presents HTML equivalent of scripted events in documentname-parsed.php file
	Stripped Comments	strips comments and whitespace from active file and reformats it and presents result in documentname-parsed.php file
	Highlighted Source	converts script to HTML text bracketed by <code> and </code> allowing script to be displayed in an HTML file highlighted as it appears in UltraEdit
Select PHP executable...	Select PHP Executable	presents a dialog for user to browse to location of php.exe.

Ruby Command Line Options

If **Command Line Options** is selected from the Script toolbar submenu, then the following dialog will be presented. Users familiar with Ruby may use this interface to launch Ruby with specified command line

options and run this on the active script. For further information regarding the Ruby programming language please see: [http:// www.ruby-lang.org/](http://www.ruby-lang.org/).











* Please note: These features are only supported in UESTudio, and not in UltraEdit.

Debugger Toolbar



The Debugger toolbar provides an integrated method for invoking commands for use with the WinDbg debugger in UESTudio. The following functions are supported:

	Start Debugger	runs debugger - if debugger is not set yet, a dialog will be displayed where the debugger for the active project may be selected.
	Go	automatically starts the debugger and runs the application
	Restart	ends a debugging session and then restarts
	Stop Debugging	ends a debugging session
	Break	pauses target execution and returns control to debugger
	Step Into	executes code one statement at a time, following execution into function calls
	Step Out	executes remaining lines of a function in which the current execution point lies
	Step Over	executes the next line of code but does not follow execution through any function calls
	Run to Cursor	resumes execution of code from the current statement to the selected statement






	Set Breakpoint	adds breakpoint at current cursor location
	Remove Breakpoint	removes breakpoint at current cursor location
	Call Stack Window	displays the Call Stack window to display a list of all active procedures or stack frames for the current thread of execution
	Variables Window	displays the Variables window to view local variables and their values for each procedure in the current stack frame
	Watch Window	displays the Watch window to view the values of selected variables or watch expressions
	Registers Window	displays the Registers window to view registers
	Memory Window	displays the Memory window to view memory contents of an address range
	Disassembly Window	displays the Disassembly window to view code bytes for the process being debugged and the assembly instructions associated with them






* Please note: These features are only supported in UESstudio, and not in UltraEdit.

Web Search Toolbar



The Web Search toolbar provides a quick method for invoking tools for use in or along with UltraEdit. The following functions are included:

	Google Search	submits search query to google.com using currently selected text
	Yahoo Search	submits search query to yahoo.com using currently selected text
	Wikipedia Search	submits search query to wikipedia.org using currently selected text
	Dictionary.com Search	submits search query to dictionary.com using currently selected text
	Thesaurus Search	submits search query to thesaurus.com using currently selected text

		text
	Whois Search	submits search query to whois.net using currently selected text
	NetworkSolutions Whois Search	submits search query to networksolutions.com using currently selected text
	PHP.net Search	submits search query to php.net using currently selected text
	cplusplus.com Search	submits search query to cplusplus.com using currently selected text
	MSDN Online Search	submits search query to msdn.microsoft.com using currently selected text

Please note:

* GOOGLE and the Google Logo is a trademark of Google Inc.

* Yahoo! and the Yahoo! Logo is property of Yahoo! Inc.

* Wikipedia and the Wikipedia Logo is a registered trademark of the Wikimedia Foundation, Inc.

* DICTIONARY.COM, THESAURUS.COM, the DICTIONARY.COM logo, and the THESAURUS.COM logo are trademarks and, where indicated, registered trademarks of Lexico Publishing Group, LLC. or its subsidiaries.

* WHOIS.NET is a service mark WHOIS.NET.

* NETWORKSOLUTIONS.com and the Network Solutions Marks are owned by Network Solutions.

* The PHP logo are Copyright© The PHP Group.

* cplusplus.com, and the cplusplus.com logo are copy right of cplusplus.com.

* search.msdn.microsoft.com and the Windows logo are owned by Microsoft Corporation.

For information regarding customizing the toolbar, either by adding or deleting toolbar buttons, please [click here](#).

Menu Commands

File Menu

File menu



The File menu includes the following commands:

New	creates a new document
Open	opens an existing document
Quick Open	opens an existing document allowing entry of name without browse dialog
Close	closes an opened document
Close All Files	closes all opened documents
Close All Files Except This	closes all opened documents except the active one
FTP/Telnet - FTP/SFTP Browser	presents FTP/SFTP Browser
FTP/Telnet - Open from FTP	opens an existing document from an FTP site
FTP/Telnet - Save As to FTP	saves an opened document to a specified file name on an FTP site
FTP/Telnet - FTP Account Manager	shows or hides the FTP/SFTP Account Manager
FTP/Telnet - SSH/Telnet Console	shows or hides the SSH/Telnet console
FTP/Telnet - SSH/Telnet Account Manager	shows or hides the SSH/Telnet Account Manager
Revert to Saved	reverts to saved file
Save	saves an opened document using the same file name
Save As	saves an opened document to a specified file name
Save All	saves all modified documents using their current file names
Save Selection As	saves selection to a new file
Make Copy/Backup	makes a copy/backup of active file
Encryption - Encrypt File	encrypts active file using user-specified passphrase
Encryption - Decrypt File	decrypts active file using user-specified passphrase
Rename File	renames active file
Compare Files	compares two ASCII files
Sort	sorts active file based on current sort options
Sort - Advanced Sort/Options	presents advanced sorting and sort options dialog

Conversions - UNIX/MAC to DOS	converts the file format from UNIX or MAC to DOS
Conversions - DOS to MAC	converts the file format from DOS to MAC
Conversions - DOS to UNIX	converts the file format from DOS to UNIX
Conversions - EBCDIC to ASCII	converts the file format from EBCDIC to ASCII
Conversions - ASCII to EBCDIC	converts the file format from ASCII to EBCDIC
Conversions - OEM to ANSI	converts the file format from OEM to ANSI
Conversions - ANSI to OEM	converts the file format from ANSI to OEM
Conversions - ASCII to Unicode	converts the file format from ASCII to Unicode
Conversions - UTF-8 to Unicode	converts the file format from UTF-8 to Unicode
Conversions - Unicode to ASCII	converts the file format from Unicode to ASCII
Conversions - UTF-8 to ASCII	converts the file format from UTF-8 to ASCII
Conversions - ASCII to UTF-8	converts the file format from ASCII to UTF-8
Conversions - Unicode/UTF-8 to UTF-8	converts the file format from Unicode/UTF-8 to UTF-8
Conversions - Unicode/ASCII/UTF-8 to UTF-8	converts the file format from Unicode/ASCII/UTF-8 to UTF-8
Conversions - UNICODE to UNICODE Big Endian	converts the file format from Unicode to Unicode (Big Endian)
Conversions - UNICODE Big Endian to UNICODE	converts the file format from Unicode (Big Endian) to Unicode
Delete Active File	deletes the disk file associated with the active window
Send Mail	sends file via mail to recipients
Print	prints a document
Print All Files	prints all opened documents
Print Preview	displays the document on the screen as it would appear printed
Print Setup/Configuration - Page Setup	sets up page headers and footers for printing
Print Setup/Configuration - Print Setup	selects a printer and printer connection
Print Setup/Configuration - Set Printer Font	selects the printer font to use
Print Setup/Configuration - Mirror Display Font	uses display font for the printer
Favorite Files	opens favorite files, or manages the list
Recent Files	opens recent files
Recent Projects/Workspace	opens recent projects or workspaces
Exit	exits UltraEdit

New command (File menu)



Use this command to create a new document in UltraEdit. You can open an existing document with the [Open command](#).

Shortcut: CTRL+N

Open command (File menu)



Use this command to open an existing document (file) in a new window. You can open multiple documents at once. Use the Window menu to switch among the multiple open documents. See [Window 1, 2, ... command](#).

This command presents the File Open dialog. This allows multiple files to be selected and opened at one time.

Files may also be opened by dragging them from the File Manager into UltraEdit's window and dropping them (releasing the mouse button).

You can create new documents with the [New command](#).

Shortcut: CTRL+O

Quick Open command (File menu)



Use this command to quickly open a file without the delay of showing a file browse dialog. The directory from which the file is to be opened is shown in the dialog. Relative paths as well as the filename may be entered.

With this command it is also possible to open with wildcards such as "*.txt".

The checkbox Open all matching files from subdirectories instructs UltraEdit to recursively search all subdirectories of the specified path (or default path) and open all files that match the specified name.

The file selected will be opened or an error displayed.

Shortcut: CTRL+Q

Close command (File menu)



Use this command to close all windows containing the active document. UltraEdit suggests that you save changes to your document before you close it. If you close a document without saving you lose all changes made since the last time you saved it. Before closing an untitled document UltraEdit displays the Save As dialog box and suggests that you name and save the document.

Shortcut: CTRL+F4

Close All Files command (File menu)



Use this command to close all files currently opened for editing in UltraEdit. UltraEdit suggests that you save changes to your document before you close it. If you close a document without saving you lose all changes made since the last time you saved it. Before closing an untitled document UltraEdit displays the Save As dialog box and suggests that you name and save the document.

Close All Files Except This command (File menu)



Use this command to close all files currently opened for editing in UltraEdit except the active file. UltraEdit suggests that you save changes to your document before you close it. If you close a document without saving you lose all changes made since the last time you saved it. Before closing an untitled document UltraEdit displays the Save As dialog box and suggests that you name and save the document.

FTP Browse (File menu/FTP submenu)



Use this command to open the FTP Browser. The FTP Browser consists mainly of three panes. The leftmost pane displays the directory tree for the local machine. The middle pane displays the content of the selected folder on the local machine. The right pane displays the content of the active folder on the remote system (FTP server).

Multiple files may be selected and moved between the local and remote systems using the buttons on the vertical divider between the middle and right panes. The "A" radio button on this divider is used for an ASCII transfer (converting line terminators for the receiving system) and the "B" radio button is used for a binary transfer, which transfers the selected files byte-for-byte with no changes. An address bar including a button to move up one directory, a refresh button, and a text field indicating the path to the active folder are located above the panes for the local and remote systems.

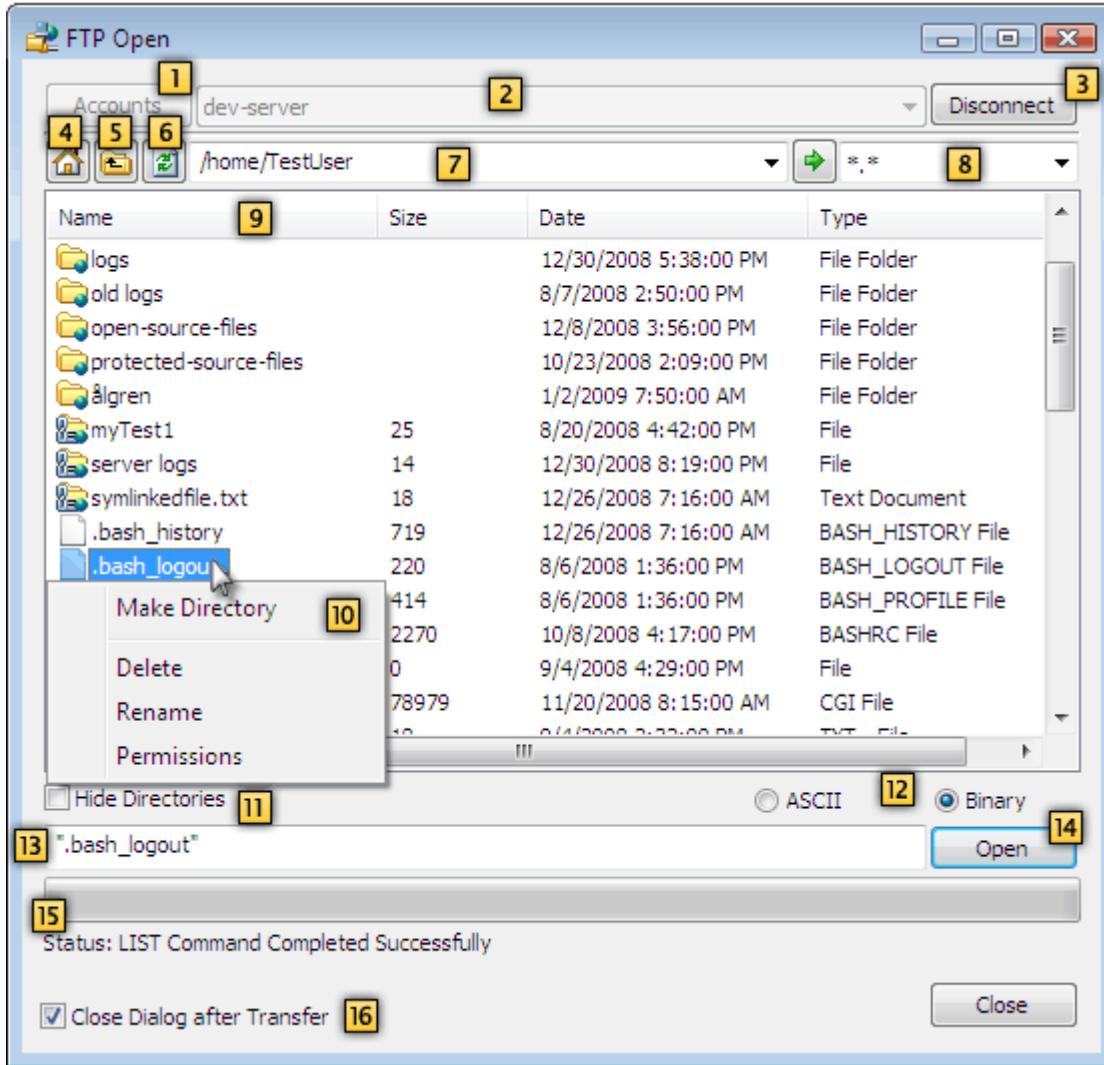
The following buttons are included:

Accounts	presents the FTP/SFTP Account Manager dialog
Connect	connects to selected FTP/SFTP account
Disconnect	disconnects from active FTP/SFTP account
Close	closes FTP/SFTP Browser
Change	presents Change FTP Directory dialog to allow direct specification of path to change to on remote system
Abort	terminates active transfer process

Open from FTP (File menu/FTP submenu)



Use this command to open a file via FTP. This command presents a custom dialog that allows selection of the site/account to browse.



The **Accounts** [1] button may be pressed to open the FTP Account Manager where new accounts may be specified. A new account must be configured for each site to be connected to as each site may have unique settings.

A list of configured accounts is shown in the center dropdown [2] and any of the configured accounts may be selected. Once the desired account is selected, the **Connect** [3] button must be pressed to initiate the connection to the server. When a connection is established, this will be replaced with a **Disconnect** [3] button which may be used to terminate the connection.

Pressing the **Return to Initial Directory** [4] button will navigate to the directory initially loaded for the active session.

Pressing the **Parent Directory [5]** button will navigate to the parent folder of the active directory. Pressing the **Refresh [6]** button refreshes (rereads) the directory information from the host server.

Double-clicking on any of the directories shown in the list will attempt to change to the specific directory. Additionally the Change Directory [7] combobox may be used to type in a directory to change to. This will automatically store a list of all paths typed into it.

The directory list displayed may be filtered to show a subset of files by specifying a filter in the Filter [8] text field. The Filter text field supports Perl regular expressions as well as standard wildcards. This filter only affects the listing downloaded from the server and displayed in the integrated FTP client. This will automatically store a list of all filters typed into it. This list may also be sorted by clicking on any of the column headings [9] for the list (Name, Size, Date, Type).

New directories may be created on the host server by right-clicking in the list and selecting **Make Directory** from the context menu [10]. Additionally the Hide Directories [11] checkbox will cause directories not to be shown.

To open a file, either double-click on the file or select it and click on the Open [14] button. If the full path to the file to be opened is known, this may be typed in the **Open [13]** text field and navigation to the file is not required. The transfer progress will be shown as the file is transferred.

While an operation is in progress a warning will be shown if an attempt to start another operation is made, or canceling the dialog is requested. While files are being transferred, or a site is being connected to, the dialog will show the current status [15]. The transfer will be executed using the specified transfer type [12] (ASCII or Binary). Binary transfers the file byte-for-byte with no modification. ASCII causes the file to be appropriately converted for line terminations between different systems. Depending on the FTP server/client, an **ASCII** transfer may corrupt the file as this is a 7-bit transfer. For this reason, **Binary** is the preferred transfer type.

By default after the file is retrieved and open the dialog will be closed if the option **Close Dialog after Transfer [16]** is set. This option may be cleared to have the dialog remain open.

If there are problems connecting to the server, it may be useful to select [Show FTP log in output window](#) under the **FTP** section in the Configuration dialog. This will display a log of the session showing all commands sent to and data received from the FTP server in the output window. This can provide helpful diagnostic information and the contents of this window may be copied using the clipboard.

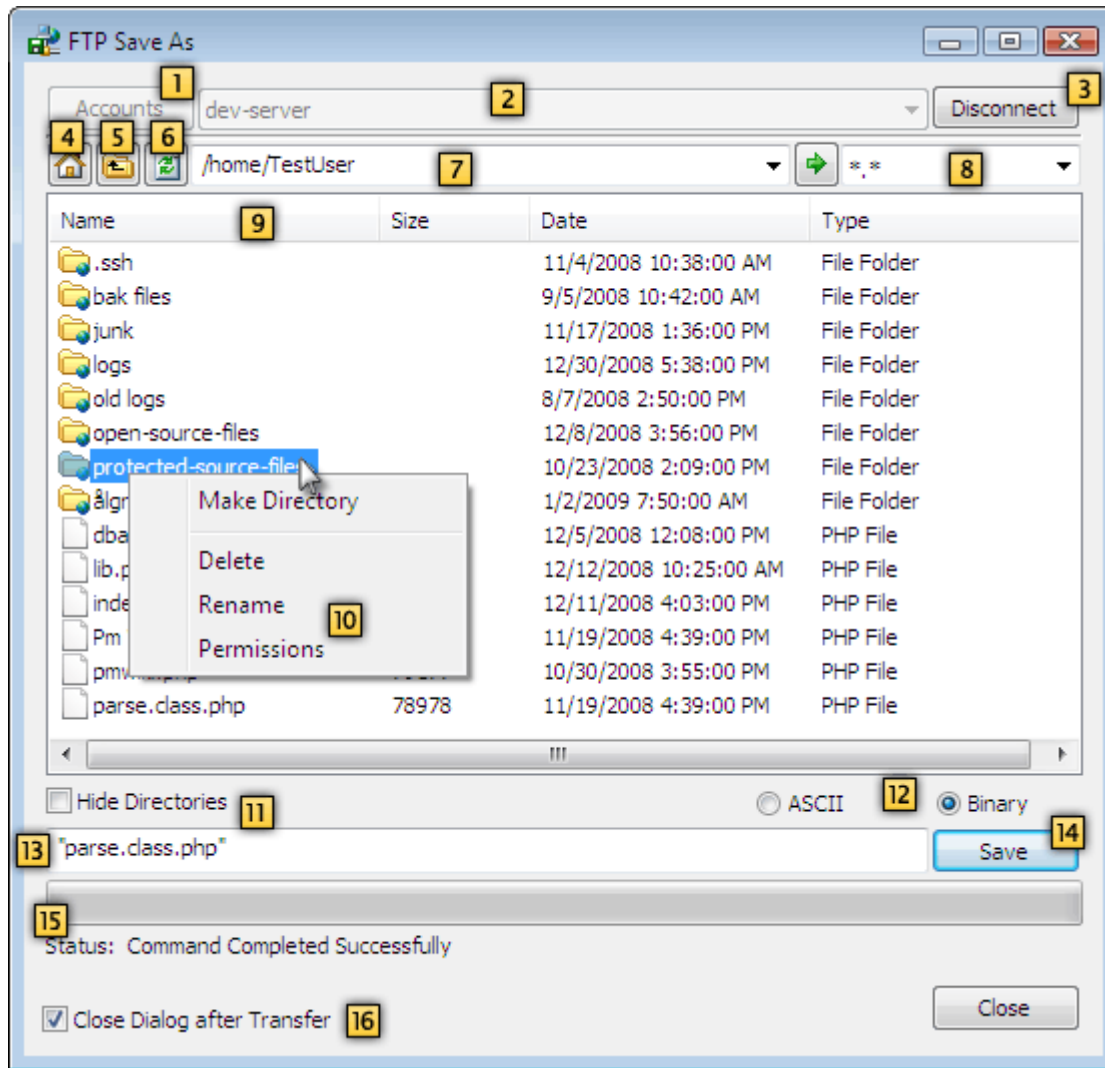
The context menu in this dialog includes the following options:

Make Directory	creates a new directory on the host server. No checking of the name or syntax is performed. Any errors are reported back from the host server.
Delete	deletes a remote file or directory on the host server
Rename	renames a remote file or directory on the host server
Permissions	allows the file permissions to be set on a remote file. This will only work on servers that recognize the FTP "Site" command chmod. Typically this would only be UNIX/Linux servers.

Save As to FTP (File menu/FTP submenu)



Use this command to save an open document/file via FTP. This command presents a custom dialog that allows selection of the site/account to browse.



The **Accounts** [1] button may be pressed to open the FTP Account Manager where new accounts may be specified. A new account must be configured for each site to be connected to as each site may have unique settings.

A list of configured accounts is shown in the center dropdown [2] and any of the configured accounts may be selected. Once the desired account is selected, the **Connect** [3] button must be pressed to initiate the connection to the server. When a connection is established, this will be replaced with a **Disconnect** [3] button which may be used to terminate the connection.

Pressing the **Return to Initial Directory** [4] button will navigate to the directory initially loaded for the active session.

Pressing the **Parent Directory [5]** button will navigate to the parent folder of the active directory. Pressing the **Refresh [6]** button refreshes (rereads) the directory information from the host server.

Double-clicking on any of the directories shown in the list will attempt to change to the specific directory. Additionally the Change Directory [7] combobox may be used to type in a directory to change to. This will automatically store a list of all paths typed into it.

The directory list displayed may be filtered to show a subset of files by specifying a filter in the Filter [8] text field. The Filter text field supports Perl regular expressions as well as standard wildcards. This will automatically store a list of all filters typed into it. This list may also be sorted by clicking on any of the column headings [9] for the list (Name, Size, Date, Type).

New directories may be created on the host server by right-clicking in the list and selecting **Make Directory** from the context menu [10]. Additionally the Hide Directories [11] checkbox will cause directories not to be shown.

By default the **Save [13]** text field will be populated with the name of the file currently active in the editor. To save a file, navigate to the desired directory on the host server and click on the **Save [14]** button. If the intent is to replace a file already on the server, it should be selected in the list. Its name will populate the **Save [13]** text field and clicking on the **Save [14]** button will initiate the transfer. The transfer progress will be shown as the file is transferred.

While an operation is in progress a warning will be shown if an attempt to start another operation is made, or canceling the dialog is requested. While files are being transferred, or a site is being connected to, the dialog will show the current status [15]. The transfer will be executed using the specified transfer type [12] (ASCII or Binary). Binary transfers the file byte-for-byte with no modification. ASCII causes the file to be appropriately converted for line terminations between different systems.

By default after the file is transferred the dialog will be closed if the option **Close Dialog after Transfer [16]** is set. This option may be cleared to have the dialog remain open.

If there are problems connecting to the server, it may be useful to select [Show FTP log in output window](#) under the **FTP** section in the Configuration dialog. This will display a log of the session showing all commands sent to and data received from the FTP server in the output window. This can provide helpful diagnostic information and the contents of this window may be copied using the clipboard.

The context menu in this dialog includes the following options:

Make Directory	creates a new directory on the host server. No checking of the name or syntax is performed. Any errors are reported back from the host server.
Delete	deletes a remote file or directory on the host server
Rename	renames a remote file or directory on the host server
Permissions	allows the file permissions to be set on a remote file. This will only work on servers that recognize the FTP "Site" command chmod. Typically this would only be UNIX/Linux servers.

Revert to Saved command (File menu)



Use this command to close the active document and reload it losing any changes you have made. This command is not available if the file is an unnamed file. A prompt will be displayed confirming the loss of changes.

Save command (File menu)



Use this command to save the active document to its current name and directory. When you save a document for the first time, UltraEdit displays the Save As dialog so you can name your document. If you want to change the name and directory of an existing document before you save it, choose the [Save As command](#).

Shortcuts: CTRL+S

Save As command (File menu)



Use this command to save and name the active document. UltraEdit displays the Save As dialog so you can name your document.

To set a default directory for use by the Save As command use the [Default Save Directory](#) setting.

To save a document with its existing name and directory, use the [Save command](#).

Shortcut: F12

Save All command (File menu)



Use this command to save all modified documents to their current name and directory. When you save a document for the first time, UltraEdit displays the Save As dialog so you can name your document. If you want to change the name and directory of an existing document before you save it, choose the [Save As command](#).

Save Selection As command (File menu)



Use this command to save the current selection to a new file. UltraEdit displays the Save As dialog so you can name your document.

Make Copy/Backup command (File menu)



This allows the user to make a backup copy of the active file (with any current changes) without affecting the active file.

The active file maintains the current file name and does not change name unlike the [Save As command](#).

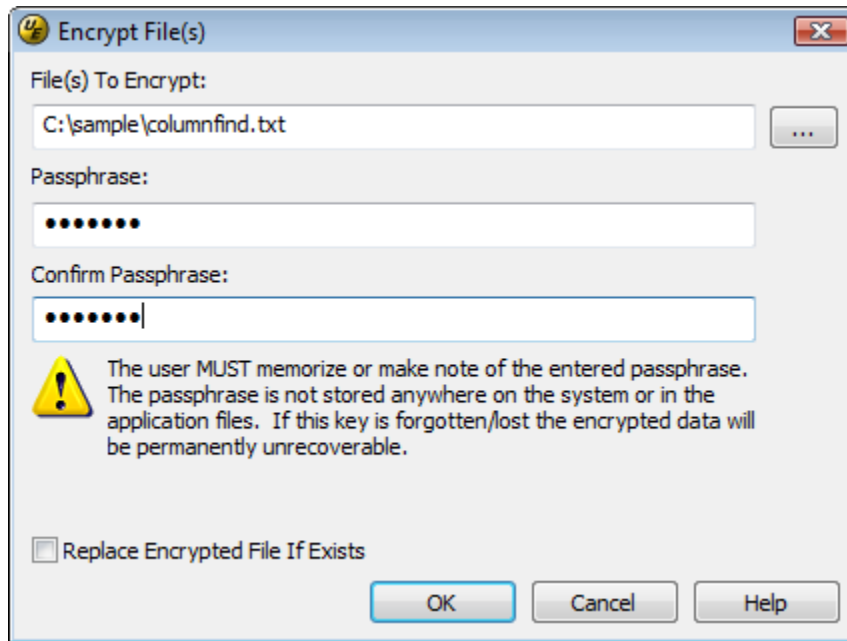
Encrypt File command (File menu/Encryption)



This command may be used to encrypt the specified file (or specified files) using the Advanced Encryption Standard (AES) algorithm.

Encrypt File

When this option is selected, a dialog is displayed where the user is prompted to specify the file(s) to be encrypted and enter a passphrase to be used as the encryption key. The passphrase is case-sensitive and may be between 8 and 32 characters in length and may include any characters which may be typed.



By default the path for the active file will populate the **File(s) to Encrypt** field. Users may specify the full path to a different file (or files) to be encrypted in this field. The easiest way to do this would be to press the browse button beside the **File(s) to Encrypt** field which will present an Open dialog where users may navigate to the file(s) they wish to be encrypted. If desired, multiple files may be encrypted by selecting them in the Open dialog (using the CTRL key and mouse or SHIFT key and mouse to select multiple items) and then pressing the Open button to pass the specified files back to the **Encrypt File(s)** dialog.

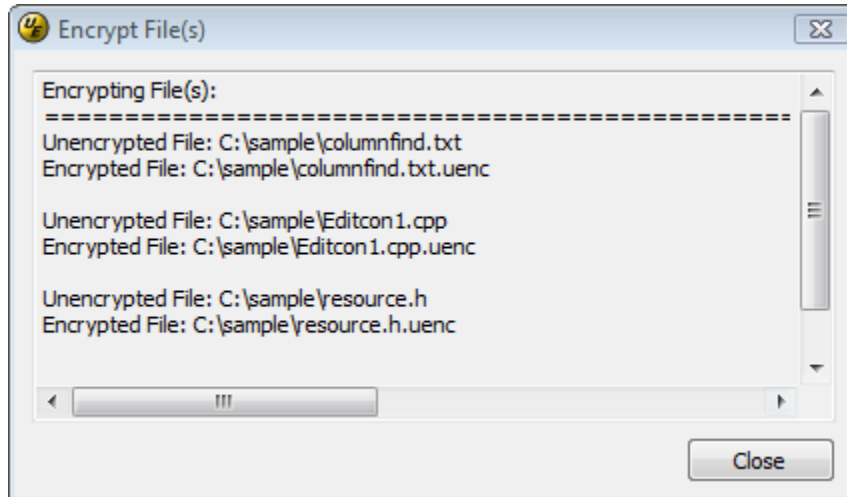
If the files to be encrypted are specified manually, the full path must be specified and multiple files should be separated by a semi-colon and a space. (Example: c:\sample\columnfind.txt; c:\sample\resource.h; c:\sample\main.cpp;")

The user must then enter and confirm the passphrase to be used for encrypting the specified files.

PLEASE NOTE: The user **MUST** memorize or make note of the encryption passphrase specified. The specified passphrase is not stored anywhere on the system or in the application files. If this passphrase is forgotten/lost the encrypted data will be permanently unrecoverable.

If **Replace Encrypted File If Exists** is selected, then any encrypted files that already exist in the specified path with names and extensions matching the files selected for encryption would automatically be overwritten.

Once the specified file or multiple files are encrypted, an encryption status dialog will be displayed listing the files that are encrypted if the output window is not currently visible. If the output window is visible, these items would be displayed in the output window.



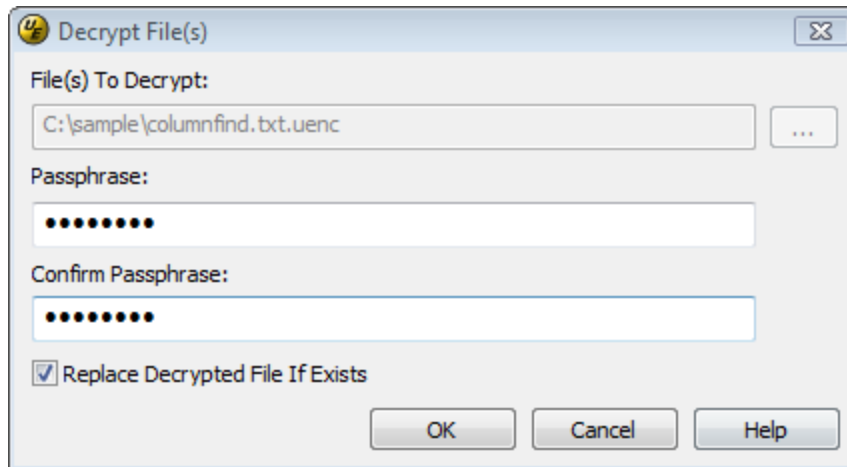
Decrypt File command (File menu/Encryption)



This command may be used to decrypt the content of the active file (or specified files) using the Advanced Encryption Standard (AES) algorithm.

Decrypt Files

When this option is selected, a dialog is displayed where the user is prompted to specify the file(s) to be decrypted and enter a passphrase to be used as the decryption key. The passphrase is case-sensitive and may be between 8 and 32 characters in length and may include any characters which may be typed.



Users may specify the full path to a different file (or files) to be decrypted in the **File(s) to Decrypt** field. The easiest way to do this would be to press the browse button beside the **File(s) to Decrypt** field which will present an Open dialog where users may navigate to the file(s) they wish to be decrypted. If desired, multiple files may be decrypted by selecting them in the Open dialog (using the CTRL key and mouse or SHIFT key and mouse to select multiple items) and then pressing the Open button to pass the specified files back to the **Decrypt File(s)** dialog.

If the files to be decrypted are specified manually, the full path must be specified and multiple files should be separated by a semi-colon and a space. Encrypted files will be saved with a ".uenc" extension.

The user must then enter and confirm the passphrase to be used for decrypting the specified files. The passphrase entered and confirmed for decryption must be an exact match for the passphrase entered during encryption.

If **Replace Decrypted File If Exists** is selected during decryption, any plain text files that already exist in the specified path (where the encrypted files exist) with names and extensions matching the files selected for decryption would automatically be overwritten.

Rename File command (File menu)



This command allows the user to rename the active file. It presents a dialog allowing the user to specify the new name. Only the name should be entered, the path will be the same as the existing path.

If the file cannot be renamed, or the new name is invalid, an error message will be shown.

Compare Files command (File menu)

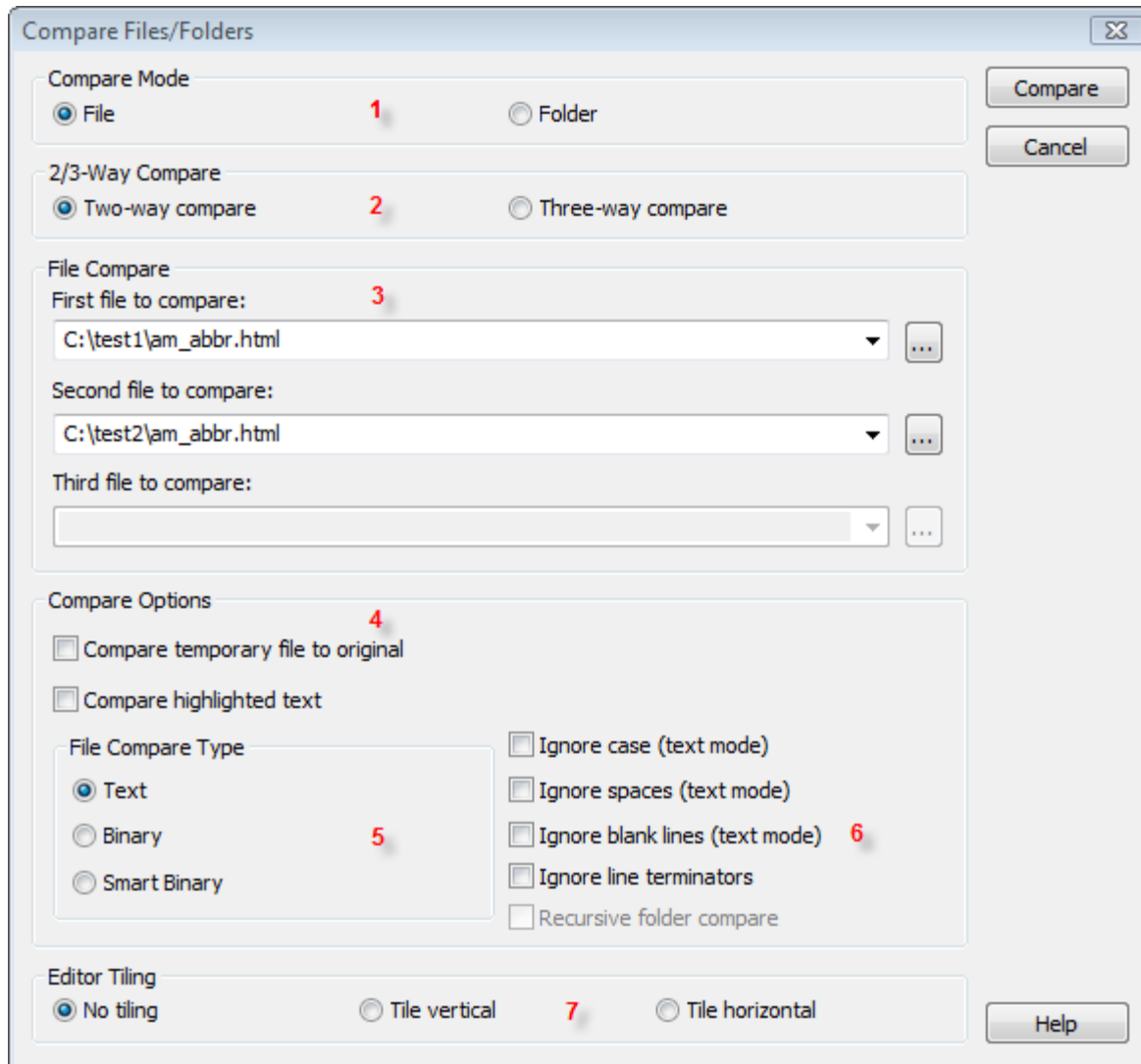


UltraEdit and UESstudio both come with UltraCompare Lite at no additional cost.

Please [click here](#) to learn more about the differences between UltraCompare Lite and UltraCompare Pro.

Please [register UltraCompare Professional now](#) to get full file, folder, and merge capabilities.

The Compare Files menu item presents a dialog allowing the user to specify files/folders to be compared. A separate window is shown displaying the compared files with different lines highlighted and marked to indicate the differences in the files/folders.



The file compare dialog allows the user to select the following options:

Compare Mode [1]

This option specifies if the items being compared are Files or Folders.

2/3 Way Compare [2]

This option is used to specify whether two or three files/folders will be compared.

File/Folder Compare [3]

The dropdowns/browse buttons in this section may be used to specify the files/folders to be compared. The items displayed and the associated title will be updated based on the selection in the 2/3 Way Compare section above.

File Compare Options [4]

Compare temporary file to original

If this option is selected only the first file specified will be read and a copy of the modified state of the file will be compared against the disk copy.

Compare highlighted text

If this option is selected only the sections highlighted in the files specified for comparison will be compared.

File Compare Type [5]

These options control whether the file comparison should be a Text compare, Binary compare, or Smart Binary compare. A Text comparison is a line-by-line comparison of the characters displayed in the text file. A Binary comparison is a byte-by-byte binary comparison of the content of the compared files. A Smart Binary comparison will show where there are matching blocks of code even if they are offset by inserted/deleted data and don't match in a byte-by-byte comparison

File Compare Options [6]

These options allow the user to define "ignore" options for text compare or a recursive compare for folder compare.

Editor Tiling [7]

These options control how or if the compared files will be tiled in UltraEdit when they're presented for comparison in UltraCompare.

Sort File command (File menu)



This command sorts each line of the active file according to the current sort options. Any size file may be sorted. See the [Advanced Sort/Options](#) for details.

If a portion of the file is selected, only the selected portion is sorted. The rest of the file will not be modified.

Advanced Sort/Options (File menu)



This command allows you to set the sort options and perform a sort of the active file. The sort is performed on the complete file unless a portion is selected. If a portion is selected, just the selected portion will be sorted. However ALL the lines selected will be sorted, not just selected columns if in column mode. The UNDO command is not supported with sort operations.

When this option is selected, a dialog is presented with the following options:

Sort Order

Ascending This sets the sort for ascending order ('a' to 'z', 1-9) and is the default.

Descending This sets the sort for descending order ('z' to 'a', 9-1).

Remove Duplicates

If this is checked, then the associated radio button options **Where All Selected Keys Match** and **Where Any Selected Keys Match** become active and may be selected.

If **Where All Selected Keys Match** is selected, duplicate strings/characters must exist in ALL active sort keys for a line to be considered a duplicate and be removed. If **Where Any Selected Keys Match** is selected, a line is considered a duplicate if duplicate strings/characters exist in ANY of the active sort keys. Active keys are designated using the checkboxes under the **RD** heading. This allows the user greater flexibility in determining which defined keys are used for the **Remove Duplicates** sort operation.

Ignore Case

If the Ignore Case option is set, lines are considered identical if the characters are the same with the only difference being the case of them. This option should be checked if you want the search to be insensitive

to case, i.e. "CAT" is considered the same as "cat". Do not check this option if the sort should consider "cAT" different from "cat".

Numeric Sort

If this is checked, the sort will perform a numeric sort on the selected columns. Typically a sort is non-numeric and the values 1,11,111 would be sorted before 2,22,222 as 1 comes before 2. If a numeric sort is selected, the values would be sorted in numeric order (1,2,11,12,111,222). The numeric sort is not possible with a locale specific sort or when using the alternate sort method. The numeric sort option should not be selected when sorting only alphabetical characters.

Sort Columns

Up to 4 sort keys may be selected to sort on. Each sort key contains a range of columns that will be sorted. The normal value for the range is from column 1 to 9216.

The start and end columns for unused keys should be set to 0. If multiple keys are used, the sort order is based on the first key, followed by the second key, etc.

To sort by complete line set the start column to 1 and the end column to -1. If any key has a value of -1 for the end column the sort will be from the start column to the end of line. If any key is set to sort the complete line the rest of the sort keys will be ignored.

If the start column is 0 for any sort key (and not sorting by line) the start column will be the column of the cursor.

Sorting performance is better when less keys are used and when less columns are sorted on.

The sort uses memory mapped files and virtual memory. In some cases it may be necessary to use an alternate sort that allows the user to specify where the temporary files will be used. If this is the case the Alternate Sort not using Virtual Memory should be checked and only a single sort key may be used (and non-numeric). This is exactly the same as earlier versions that only supported single keys.

Use Locale (Slower)

This specifies that the sort should be locale specific. For some languages (e.g. German) this is required to change the default sort order. The locale to be used can be specified under the [Advanced Menu](#) and [Set Code Page/Locale](#). When using this setting the sort may be slower than when not doing so. A locale specific sort is not possible if Alternate Sort not using Virtual Memory is selected.

These options are saved between UltraEdit sessions and are used by the [Sort File command](#).

UNIX/MAC to DOS command (File menu)



Use this command to correct the end of line characters for display in UltraEdit and other text editors.

Some documents generated from UNIX-based systems (and some other systems) do not terminate each line with a carriage return and line feed. UltraEdit requires all lines to be terminated with a carriage return and line feed character.

This command searches the file for all carriage return and line feed characters. If a carriage return character is found without an accompanying line feed character, a line feed character is added. If a line feed character is found without a preceding carriage return character a carriage return character is inserted. If a line feed is found preceding a carriage return the characters are reversed.

To convert a file to UNIX format, use the [DOS to UNIX command](#).

To convert a file to MAC format, use the [DOS to MAC command](#).

DOS to MAC command (File menu/Conversions submenu)



Use this command to convert the file to a Macintosh format. This will replace the carriage return and line feed combinations to a carriage return only.

To convert a file from UNIX or MAC format, use the [UNIX/MAC to DOS command](#).

DOS to UNIX command (File menu/Conversions submenu)



Use this command to convert the file to a UNIX format. This will replace the carriage return and line feed combinations to a line feed only.

To convert a file from UNIX or MAC format, use the [UNIX/MAC to DOS command](#).

EBCDIC to ASCII command (File menu/Conversions submenu)



This function converts the complete file from EBCDIC format to ASCII. This is useful in dealing with files that are in the EBCDIC file format (usually on Mainframe computer systems).

ASCII to EBCDIC command (File menu/Conversions submenu)



This function converts the complete file from ASCII to EBCDIC format. This is useful in dealing with files that need to be exported in the EBCDIC file format (usually on Mainframe computer systems).

OEM to ANSI command (File menu/Conversions submenu)



Use this command to convert a file that uses the OEM character set (usually DOS generated text files) to the ANSI (Windows) character set.

To convert a file from ANSI to OEM, use the [ANSI to OEM command](#).

ANSI to OEM command (File menu/Conversions submenu)



Use this command to convert a file that uses the ANSI (Windows) character set to the OEM (DOS) character set.

To convert a file from OEM to ANSI, use the [OEM to ANSI command](#).

ASCII to Unicode command (File menu/Conversions submenu)



This function will convert the complete file from ASCII to Unicode (16-Bit wide characters).

The conversion uses the active code page. For information on code pages see the Windows documentation/help.

The status bar will show a "U-" in front of the file type (DOS/UNIX/MAC) to indicate the file is Unicode.

A file may be converted to ASCII with the [Unicode to ASCII command](#).

UTF-8 to Unicode command (File menu/Conversions submenu)



This function will convert the complete file from UTF-8 to Unicode. When the file is saved it will remain as Unicode unless specifically converted to another format by the user. The status bar will show a "U-" in front of the file type (DOS/UNIX/MAC) to indicate the file is Unicode.

Unicode to ASCII command (File menu/Conversions submenu)



This function will convert the complete file from Unicode to ASCII.

The conversion uses the active code page. For information on code pages see the Windows documentation/help.

For Unicode files the status bar will show a "U-" in front of the file type (DOS/UNIX/MAC) to indicate the file is Unicode. Following the conversion this indicator would not be present.

A file may be converted to Unicode with the [ASCII to Unicode command](#).

UTF-8 to ASCII command (File menu)



This function will convert the complete file from UTF-8 to ASCII. When the file is saved it will remain as ASCII unless specifically converted to another format by the user. It will use the current code page for conversion. It's possible that not all characters may be converted correctly. In this case a default character (?) will be used.

ASCII to UTF-8 command (File menu/Conversions submenu)



This function will convert the complete file from ASCII to UTF-8. When the file is saved it will remain as UTF-8 unless specifically converted to another format by the user. The status bar will show "U8-" in front of the file type (DOS/UNIX/MAC) to indicate the file is UTF-8.

Internally the file will be in UNICODE format and converted back when the file is saved to UTF-8 encoding. In HEX mode the file will be shown with its current internal representation.

Unicode/UTF-8 to UTF-8 command (File menu/Conversions submenu)



This function will convert the complete file from Unicode or UTF-8 (ASCII representation) to UTF-8 (with the file internally as Unicode). When the file is saved it will remain as UTF-8 unless specifically converted to another format by the user. The status bar will show "U8-" in front of the file type indicator.

Internally the file will be in UNICODE format and converted back when the file is saved to UTF-8 encoding. In HEX mode the file will be shown with its current internal representation.

Unicode/ASCII/UTF-8 to UTF-8 command (File



menu/Conversions submenu)

This function will convert the complete file from Unicode, ASCII or UTF-8 (Unicode representation internally) to UTF-8 encoding. The file will however not be internally stored as Unicode and as such the multi-byte UTF-8 characters will display as upper ASCII characters and not the UTF-8 character they represent. When the file is saved it will remain as UTF-8 unless specifically converted to another format by the user. The status bar will NOT indicate this file format as it has no different structure from an ASCII file.

UNICODE to UNICODE Big Endian command (File menu/Conversions submenu)



This function will convert the complete file from Unicode encoding to Unicode (Big Endian) encoding. Endian refers to the order in which bytes are stored. On Windows platforms, which are mostly little endian, UTF-16LE is just called "Unicode" and UTF-16BE is just called "Unicode (Big Endian)". This is much less confusing for the majority of people who do not work cross-platform.

UNICODE Big Endian to UNICODE command (File menu/Conversions submenu)



This function will convert the complete file from Unicode (Big Endian) to Unicode encoding. Endian refers to the order in which bytes are stored. On Windows platforms, which are mostly little endian, UTF-16LE is just called "Unicode" and UTF-16BE is just called "Unicode (Big Endian)". This is much less confusing for the majority of people who do not work cross-platform.

Delete Active File command (File menu/Special Functions submenu)



Use this command to close the active document and delete its associated file from the disk. This command prompts the user to ensure the file should really be deleted. If the file is deleted from the disk, it is no longer available. USE THIS COMMAND WITH CARE.

Send Mail command (File menu/Special Functions submenu)



Use this command to send the active file as a MAIL attachment. The file and all changes are sent. This feature is only supported on systems that have MS-MAIL installed or another mail program that installs itself with MAPI support.

Print command (File menu)



Use this command to print a document. This command presents a Print dialog, where you may specify the range of pages to be printed, the number of copies, the destination printer, and other printer setup options.

Shortcut: CTRL+P

Print All Files command (File menu)



This command prints all the open documents with the current page setup and printer settings. To change the settings use the [Page Setup](#) or [Print Setup](#) commands from the [File menu](#).

Print Preview command (File menu)



Use this command to display the active document as it would appear when printed. When you choose this command, the main window will be replaced with a print preview window in which one or two pages will be displayed in their printed format. The print preview toolbar offers you options to view either one or two pages at a time; move back and forth through the document; zoom in and out of pages; and initiate a print job.

Page Setup command (File menu/Print Setup/Configuration submenu)



The Page Setup menu item presents a dialog allowing the user to set up page headers, footers and margins for printing. The page header, if configured, is printed at the top of every page. The page footer, if configured, is printed at the bottom of every page. If a page footer or header is configured, a single line is printed across the page between the header or footer and the text to be printed.

The header or footer may be disabled by selecting the appropriate check box. If the header or footer is enabled, the separator line that is normally printed between the header/footer and the page text may optionally be disabled.

The page header and footer may include text defined by the user, and additionally special characters to allow the file name and page numbers to be printed in the header or footer.

The special characters are:

& f This is replaced in the header or footer with the full file name including path.

& n This is replaced in the header or footer with the file name only (no path).

& p This is replaced in the header or footer with the page number.

& t This is replaced in the header or footer with the total number of pages.

& l This aligns the text following the "& l" to the left side of the header/footer area.

& c This aligns the text following the "& c" to the center of the header/footer area.

& r This aligns the text following the "& r" to the right side of the header/footer area.

(Note - the alignment characters are not required, and any or all of them may appear in a single header or footer. However, if more than one is present, they must be in the order above -- "& l" must be before "& c" and "& c" must be before "& r").

Additionally, the header or footer may include the file date and time or system date and time. Two radio buttons allow the selection of either the file date, or the system date to be included in the header or footer. The radio buttons select the source of the date and time, additional special characters used in the header or footer specify if the date and time are to be printed and what format is used.

The characters used to specify the time and date format are shown below, the absence of any of these characters result in the time and date not being printed.

The formatting codes date and time are listed below:

Format	Description
%a	Abbreviated weekday name
%A	Full weekday name
%b	Abbreviated month name
%B	Full month name
%c	Date and time representation appropriate for the locale
%d	Day of the month as a decimal number (01-31)
%H	Hour in 24-hour format (00-23)
%I	Hour in 12-hour format (01-12)
%j	Day of the year as a decimal number (001-366)
%m	Month as a decimal number (01-12)
%M	Minute as a decimal number (00-59)
%p	Current locale's AM/PM indicator for a 12-hour clock
%S	Second as a decimal number (00-59)
%U	Week of the year as a decimal number, with Sunday as the first day of the week (00-51)
%w	Weekday as a decimal number (0-6; Sunday is 0)
%W	Week of the year as a decimal number, with Monday as the first day of the week (00-51)
%x	Date representation for current locale
%X	Time representation for current locale
%y	Year without the century as a decimal number (00-99)
%Y	Year with the century as a decimal number

% Z	Time zone name or abbreviation; no characters if time zone is unknown
-----	---

%%	Percent sign
----	--------------

Examples:

Date is July, 4th 1994. Time is 9:00pm

```
"%X , %x" prints "21:00 , 07/04/94"
```

Date is July, 4th 1994. Time is 9:00pm

```
"% c" prints                                07/04/94 21:00
```

Date is Jan, 10th 1991. Time is 5:00am

```
"%A, %B %d, %Y %I:%M%p" prints    Tuesday, January 10, 1991 05:00:am
```

Page Margins

The page margins may be set up in either inches or millimeters (mm). Each margin, top, bottom, left and right may be set independently. The margins are applied before the headers and footers are generated. If the printer is set up to have a minimum margin that is greater than the margin selected in UltraEdit the printer margin will be in effect.

Page Break Character

The page break character may be set to any ASCII character. The page break character defaults to a form-feed character ,12 (0C HEX). This character may be changed if required but should not be set to any printable character as a page break will be inserted on every occurrence of this character when printing occurs. It may be desirable to change this character if the font being used does not allow the page break character to be seen.

Wrap Text When Printing

This option when set will automatically cause the printed output to wrap the text to additional lines of any line that exceeds the width of the printable area of the page.

Print Line Numbers

This option when set causes the line numbers to be printed for each line on the left hand side of each page. If a line is wrapped, the line number is printed with the first line only. This setting is independent from the Display Line Number setting.

Print 2 Pages on 1 Sheet

This option allows two logical pages to be printed on a single sheet of paper. The logical page data is scaled to print two pages side-by-side. This scaling may mean that a little more or less data may fit side-by-side than when the pages are printed on a single sheet.

This option is particularly useful for saving paper on printouts.

Print Syntax Highlighting

This option instructs UltraEdit to print the file with the color used for syntax highlighting if a color printer is used. This will also show shades of gray on a Postscript monochrome printer.

Print Setup command (File menu/Print Setup/Configuration submenu)



Use this command to select a printer and a printer connection. This command presents a Print Setup dialog, where you specify the printer and its connection.

Set Printer Font command (File menu/Print Setup/Configuration submenu)



The Set Printer Font menu item presents the Choose Font dialog that allows selection of different fonts and font sizes for use by the printer. The selection made will apply only to the printer fonts. Selection of

the printer fonts unchecks the [Mirror Display Font](#) menu item on the Print Setup/Configuration submenu of the [File menu](#). Only fonts applicable to the printer will be available.

Font selections may be reset to default values by selecting the [Reset Fonts](#) menu item.

Mirror Display Font command (File menu/Print Setup/Configuration submenu)



The Mirror Display Font menu item instructs UltraEdit to use the font that is selected for the screen on the printer. This overrides any font selection already made for the printer with the [Set Printer Font](#) menu item.

Favorite Files command (File menu)



The Favorite Files feature allows a user to store the filenames of frequently opened files or favorite files that the user wishes to quickly reference and open in the future. When the Favorite Files menu item is selected, a dialog is displayed that allows the user to open one or more of the favorite files and modify the list of favorite files.

The dialog shows a list of files that are currently in the favorite file list. The following controls are used to manage the list:

Open

The open button opens the selected files from the list and closes the dialog. Multiple files may be selected from the list and opened at once. Any changes to the list are saved. DOUBLE CLICK of any files in the file list will perform the open command. If one of the selections is a directory, a file open dialog is shown initialized to the specified directory.

Close

The close button closes the dialog. Any changes to the list are saved.

Add File

The add file button presents a file open dialog allowing the user to browse for a file to add to the list. If a valid file is selected the file is added to the list.

Add Active File

The add active file button adds the filename of the active file in the editor to the file list.

+All Open Files

The +all open files button adds the filenames of all files in the editor to the file list.

Add Folder

The add folder button allows a user to browse for a folder and add this to the list. This allows the user to quickly open files from a favorite folder.

Remove File

The remove file button removes any selected files from the file list.

Recent Files command (File menu)



Use this command to select the document you wish to edit from up to 16 of the most recently edited documents. The number of documents shown here is configurable in the [Miscellaneous](#) topic under the Toolbars/Menus branch under Configuration in the [Advanced menu](#). Select the document you wish to edit by highlighting and clicking on it in the submenu that appears when you click on Recent Files.

Note: The files shown in this submenu may be appended to the [File menu](#) by selecting the appropriate option in the [Miscellaneous](#) topic under the Toolbars/Menus branch under Configuration in the [Advanced menu](#). This requires restarting UltraEdit. Please see the Configuration topic for more details.

Recent Projects/Workspace command (File menu)



Use this command to select the project you wish to edit from the most recently edited projects. Select the project you wish to edit by highlighting and clicking on it in the sub-menu that appears when you click on Recent Projects. The number of projects viewable here matches the number of recent files set in the [Miscellaneous](#) topic under the Toolbars/Menus branch in Configuration under the Advanced menu.

Note: The Project menu may be moved to this position as a submenu with the Recent Projects appended to the end of the Project menu by selecting the appropriate option in the [Miscellaneous](#) topic under the Toolbars/Menus branch in Configuration under the Advanced menu. This requires restarting UltraEdit.

Edit Menu

Edit menu



The Edit menu includes the following commands:

Undo	reverses previous editing operation
Redo	reverses the previous Undo command
Cut	deletes data from the document and moves it to the clipboard
Copy	copies data from the document to the clipboard
Paste	pastes data from the clipboard into the document
Paste & Copy	pastes the clipboard text to the selection (replacing it) and copies the selected text to the clipboard
Paste Special	presents a submenu where users may select an option to paste either HTML Source or Raw RTF rather than the selected text only
Cut & Append	deletes data from the document and moves it to the clipboard, appending it to any existing data in the clipboard
Copy & Append	copies data from the document to the clipboard, appending it to any existing data in the clipboard
Copy File Path/Name	copies the full path and name of the active file to the clipboard
Clipboards	allows the user to select the active clipboard
Select All	selects all text in the active document
Select Line	selects the line with the cursor
Select Word	selects the word with the cursor
Select Range	selects a user-specified range
Persistent Selection	allows selection of text without use of SHIFT key
Delete	deletes the selected text or character at cursor
Delete Line	deletes the line with the cursor
Delete to Start of Line	deletes from the cursor position to the start of line

Delete to End of Line	deletes from the cursor position to the end of line
Delete all bookmarked lines	deletes all currently bookmarked lines in the active document
Delete all hidden lines	deletes all lines in the active document currently hidden by code folding operations
Move Line Up	moves active line up one line
Move Line Down	moves active line down one line
Duplicate Line	duplicates active line below cursor position
Hex Edit	toggles the state of the editing mode for the current window between HEX and ASCII
Hex Edit/EBCDIC	toggles the state of the editing mode for the current window between HEX/EBCDIC and ASCII
Hex Insert/Delete	presents a dialog allowing insertion or deletion of bytes in a file while in HEX editing mode
Hex Copy Selected View	copies the HEX view of data to the clipboard
Spell Check	invokes the spelling checker and checks the complete file or selected text
Read Only	toggles the READ ONLY state of the active document
Toggle Word Wrap	toggles the Word Wrap state of the active document
Encode Base64	converts selected text to Base64
Decode Base64	converts selected text from Base64
Comment Add	adds line comments to selected lines
Comment Remove	removes line comments from selected lines
Comment Selection	encloses selected text in block comments. If block comments already exist in selection and nested comments are not supported, line comments will be inserted.
Uncomment Selection	removes comments inserted into file using Comment Selection

Undo command (Edit menu)



Use this command to reverse the last editing action, if possible. UltraEdit supports multiple levels of undo (the specific number varies based on the items being undone). Repeated Undo commands will attempt to step backwards through the previous commands and undo one at a time. The [Redo command](#) may be selected to redo a command that has been undone. If commands are undone and then a new command is performed, the Redo will no longer allow previously undone actions to be redone.

Shortcuts: CTRL+Z or ALT+Backspace

Redo command (Edit menu)



The Redo command allows commands that have previously been undone with the [Undo](#) to be redone. If commands are undone and then a new command is performed, the Redo will no longer allow previously undone actions to be redone.

Shortcuts: CTRL+Y

Cut command (Edit menu)



Use this command to remove the currently selected data from the document and put it on the clipboard. This command is unavailable if there is no data currently selected.

If Cut & Append to clipboard is selected the data is appended to the contents of the clipboard otherwise cutting data to the clipboard replaces the contents previously stored there.

Shortcut: CTRL+X

Copy command (Edit menu)



Use this command to copy selected data onto the clipboard. This command is unavailable if there is no data currently selected.

If Copy & Append to clipboard is selected the data is appended to the contents of the clipboard otherwise copying data to the clipboard replaces the contents previously stored there.

Shortcuts: CTRL+C

Paste command (Edit menu)



Use this command to insert a copy of the clipboard contents at the insertion point. This command is unavailable if the clipboard is empty.

Shortcuts: CTRL+V

Clipboards (Edit Menu)



UltraEdit provides up to nine user clipboards for use with the [Cut](#), [Copy](#) and [Paste](#) commands as well as the standard Windows Clipboard. The active clipboard may be changed from the [Edit Menu](#), or the Right Click context menu. Additionally, Ctrl+1-9 will switch to the user clipboards and Ctrl+0 will switch to the Windows clipboard.

When changing the active clipboard the contents of the clipboards are not modified. With the exception of the standard Windows Clipboard the contents of the clipboards are destroyed when UltraEdit is closed.

The active clipboard is shown in the Status Bar following the Line and Column indicator separated with a comma.

Shortcut: CTRL+0-9

Select All command (Edit menu)



The Select All command highlights the complete file. This allows an easy mechanism to select the complete file and [Cut](#) it to the clipboard.

Use the [Select Line command](#) to select a single line.

Shortcuts: CTRL+A

Select Line command (Edit menu)



The select line command highlights the line in which the cursor is positioned. The end of line characters are also highlighted. This allows an easy mechanism to select a line and [Cut](#) it to the clipboard.

Use the [Select All command](#) to select the complete file.

A line may also be selected by triple-clicking the mouse. This will cause the complete line to be selected including the hard return at the end of line.

Select Range command (Edit menu)



This presents a dialog allowing the user to specify a range within the file that should be selected/highlighted.

The user is able to specify the line and column for the start and end of the selection. When the dialog is first displayed the entry fields default to the current selection.

If not in [Column Mode](#), UltraEdit will switch automatically to Column Mode if the start and end columns are specified with different values.

Persistent Selection command (Edit menu)



This option allows selection to remain active and supports navigating in other parts of the file without affecting the active selection. Once this mode is active, the user defines the selected text (through keyboard commands or cursor movement) and then the menu option should be selected again or the keyboard shortcut should be invoked to terminate the definition of the selection.

It is not necessary to hold the **SHIFT** key while defining this selection. The anchor point for the active selection may be changed by holding down the **CTRL** key and left-clicking with the mouse at the desired file position.

Once the Persistent Selection has been defined, if the user moves the cursor to another position and types, the selection will be dismissed and text inserted at the current cursor position. If the selection is defined and the cursor is not moved, any key entered would replace the selected text (or in the case of **BACKSPACE** or **DELETE** would remove the selected text).

If desired, the user may extend the defined selection by selecting the menu option and clicking outside the current selection and then select the menu option again to terminate definition of the selection.

Shortcuts: CTRL+P, CTRL+S

Delete command (Edit menu/Delete submenu)



The delete command deletes the selected text or if no text is selected it deletes the character at the cursor.

Delete Line command (Edit menu/Delete submenu)



The delete line command deletes the line in which the cursor is positioned. The end of line characters are also deleted.

Use the [Select Line command](#) to select the line.

Shortcut: CTRL+E

Delete to Start of Line command (Edit menu/Delete submenu)



The delete to start of line command deletes all contents of the current line from the position of the cursor to the beginning of the line containing the cursor.

Shortcut: CTRL+F11

Delete to End of Line command (Edit menu/Delete submenu)



The delete to end of line command deletes all contents of the current line from the position of the cursor to the end of the line containing the cursor.

Shortcut: CTRL+F12

Duplicate Line (Edit menu)



The duplicate line command inserts a copy of the active line below the current cursor position. The cursor may be positioned anywhere on the active line when this feature is invoked. This command may be invoked from the [Edit menu](#) or the user may define a keyboard shortcut under [Key Mapping](#) in Configuration.

Shortcut: Alt+Shift+D

Hex Edit command (Edit menu)



The Hex Edit command selects the editing mode for the active window. The mode is toggled between Hexadecimal (HEX) and ASCII. The ASCII mode of editing is typically used for any ASCII (text) based files. In the ASCII mode UltraEdit allows normal operation of all functions and allows any characters to be typed.

The Hex mode of editing is typically used for non-ASCII files, or binary files. These files typically contain non-printable characters and are not text files.

The following restrictions apply when HEX mode is selected:

Insert mode (rather than overstrike mode) is not supported.

When the HEX mode of editing is selected for a given window, UltraEdit splits the screen area into three areas as shown:

File Offset:	Hexadecimal Representation	; ASCII Representation
000000h:	30 31 32 33 34 35 36 37 38 39 30 31 32 33 34 35	;123456789012345

The number of HEX characters shown in a line may be specified under the [Hex Edit Display](#) topic under the Editor Display branch of Configuration in the Advanced menu.

The file offset area shows the current byte offset of the character at the beginning of the line to form the beginning of the file.

The hexadecimal representation shows the hexadecimal value of each file byte as a two character field separated from the next byte by a space (' ').

The ASCII representation provides the view of printable characters. Non-printable characters are shown with a period character ('.').

If Character Properties is selected from the [Search menu](#), a dialog will be displayed showing the decimal value and hexadecimal value of the active byte as well as how this will be displayed (this may vary based on font and script). Along with this the offset of the active byte is displayed in both decimal and hexadecimal formats.

Editing in either the ASCII area or the HEX area automatically updates the other area after each valid character is typed.

All modifications are performed in OVERSTRIKE mode. To insert or delete characters in Hex mode use the [Hex Insert/Delete command](#) from the [Edit menu](#).

[Cut](#), [copy](#) and [paste](#) are supported in Hex mode.

Shortcut: CTRL+H

Hex Edit/EBCDIC command (Edit menu)



The Hex/EBCDIC editing mode works almost identically to the Hex Editing mode with the exception it will convert the EBCDIC data for viewing purposes only and show this as ASCII. The content of the file will not be modified, however the right hand side that shows the ASCII representation in Hex Mode will show the data converted to ASCII from EBCDIC.

For details of editing in Hex mode, see [Hex Edit command](#).

While in this mode, editing in the ASCII area will allow data to be entered as ASCII and automatically converted to EBCDIC in the file. Editing in the Hex area will allow data to be entered directly as Hex and the ASCII representation (converted from EBCDIC) will be shown in the ASCII area.

Find/Replace in this mode will allow ASCII strings to be used and correctly found/replaced according to the settings in the Find/Replace dialog box. Regular Expressions however cannot be used in this mode, the results would be indeterminate.

Hex Insert/Delete command (Edit menu)



The Hex Insert/Delete command allows the insertion or removal of characters at the current cursor position while in [Hex mode](#). This command presents a dialog that allows selection to insert or delete characters and requires the number of characters that are to be inserted or deleted to be specified.

If insert is selected, UltraEdit will insert the specified number of characters in the file and initialize them to a hexadecimal value of 20 (ASCII space). The characters may then be modified as required.

If delete is selected, the specified number of characters will be removed from the file beginning at the cursor position.

Shortcut: CTRL+D

Hex Copy Selected View command (Edit menu)



This function is available when editing in [Hex mode](#). The normal [Copy command](#) copies the actual data in the file, not the Hex representation shown on the display. This command formats the actual data in the clipboard in the same way it is shown on the screen in HEX mode. This allows the representation to be copied into a text or similar document.

As the data being copied may only be a partial line this command computes the address for the start of the data and this will be how it is formatted in the clipboard.

The following shows a row of data in HEX Mode (the title is for clarification only):

File Offset:	Hexadecimal Representation	; ASCII Representation
000000h:	30 31 32 33 34 35 36 37 38 39 30 31 32 33 34 35	;123456789012345

If the 4567 were selected and copied to the clipboard with this function the format in the clipboard (ready to paste into another document) would be:

000004h: 34 35 36 37 ; 4567

Spell Check command (Edit menu)



UltraEdit incorporates the Aspell spelling checker. The spelling checker is invoked from the [Edit Menu](#) and will check only selected text (if there is a selection) or the complete file if no text is selected.

If a word is not found in the selected dictionary the Check Spelling dialog is displayed with various options that allow the word to be changed, ignored, or added to a user dictionary.

The spelling checker supports different languages. These languages are selected from the [Spell Checker](#) topic under Configuration.

Shortcut: CTRL+K

Read Only command (Edit menu)



The Read Only command toggles the state of the read only status of the file. If the read only status is set the file cannot be modified. If the file has already been modified the modifications are maintained and the file may be saved.

The default behavior of the read only status may be configured in the [Miscellaneous](#) topic in the File Handling branch under Configuration in the Advanced menu. This determines if the file is read only when it is opened.

Comment Add/Remove commands (Edit menu)



These commands allow line comments (as defined in [Syntax Highlighting](#)) to be added to or removed from the selected lines. The file being edited must be enabled with Syntax Highlighting and line comments must be defined for the particular language.

If the line comments are appropriately defined, these functions will add the line comment characters to the beginning of each line or remove the line comment characters from the beginning of each line. When removing comment characters, the characters will only be removed if they are the first characters in the line.

Search Menu

Search menu



The Search menu includes the following commands:

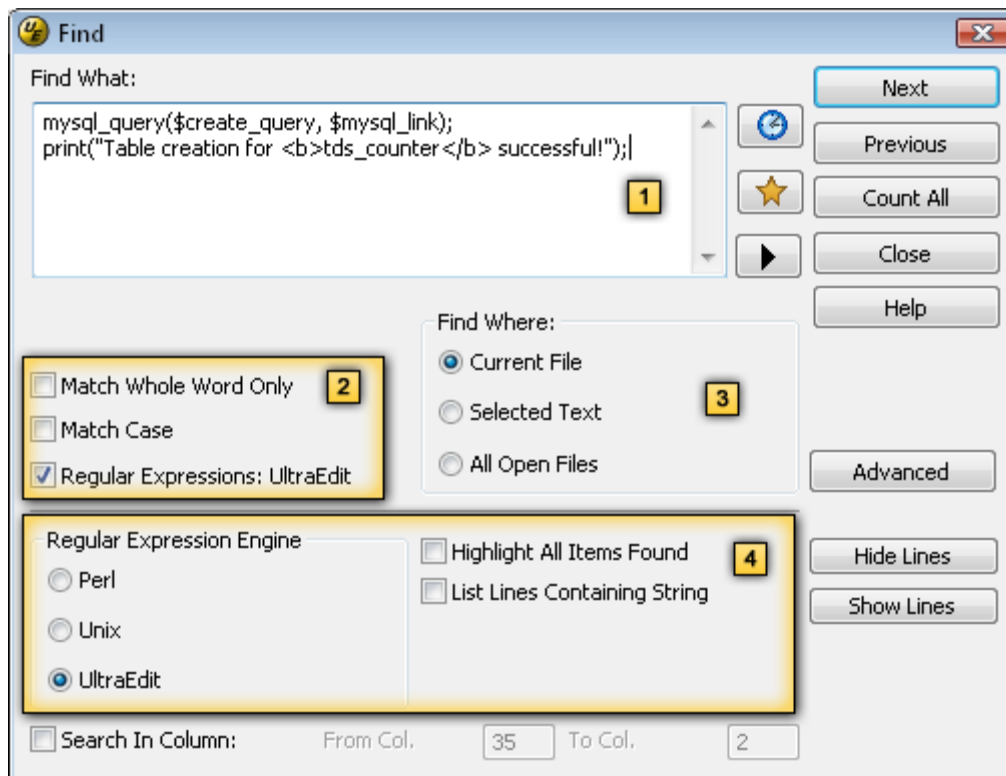
Find	searches active file for a text string
Find Next	searches active file for next occurrence of text string
Find Prev	searches active file for previous occurrence of text string
Replace	searches active file for a text string and replaces it with another text string
Incremental Search	searches active file finding the closest match as characters are typed
Find in Files	searches files on disk for a text string
Replace in Files	searches files on disk for a text string and replaces with another text string
Match Brace	finds and highlights text between matching braces
Goto Line/Page/Bookmark	positions the cursor at the specified line/column number, page break (or file position in HEX mode), or bookmark
Toggle Bookmark	toggles a bookmark at the current line
Previous Bookmark	moves the cursor to the previous bookmark
Next Bookmark	moves the cursor to the next bookmark
Clear All Bookmarks	clears all bookmarks in the active file
Edit Bookmarks	presents Edit Bookmark dialog where bookmarks may be renamed or deleted
Word Count	counts the number of words in the active file
Function List	finds and shows list of functions in the active file/project
Find Symbol	finds symbol in tag file
Character Properties	displays properties of active byte
Back (last position)	goes to last position editing/repositioning occurred
Forward (next)	goes to next position editing/repositioning occurred (after using

position)	Back command)
---------------------------	---------------

Find command (Search menu)



This command displays a dialog box for searching the document for the specified text.



The string to search for is specified in the **Find What [1]** field. A single or multi-line string up to 20,000 bytes may be used. The user will determine which direction the search will be executed by pressing the **Next** or **Previous** button to initialize the search. In HEX mode, the search is performed in the direction specified with the **Direction** radio buttons. If the string is not found, the search wraps around to the other end of the file and continues the search. If the string is not matched a message indicating that the string is not found is displayed.


In addition to searching for printable characters, tab characters and newline (Carriage Return/Line Feed) characters may be searched for. To search for a Tab character, use "^t" characters in the dialog. To search for the newline characters, use "^p" characters for CR/ LF, "^n" for LF only and "^r" for CR only in the dialog. To search for a "^" character, use "^" in the dialog.


Additionally, the characters "^s" will be replaced with any text that is highlighted when a find is executed and the characters "^c" will be replaced with the contents of the clipboard (up to 30,000 characters). This is useful particularly for macros. When recording a macro, the option exists to include a find of a specified character string, or using "^s" the macro records the "^s" and when replayed will determine the text highlighted at the time the macro is playing back, likewise when using "^c", these characters will be replaced with the contents of the clipboard at the time the macro is playing back.

Special character summary:

Symbol	Function
^^	searches for a "^" character
^s	searches for selected (highlighted) text when a macro is running
^c	searches for the contents of the clipboard when a macro is running
^b	matches a page break
^p	matches a newline (CR/LF) (paragraph) (DOS Files)
^r	matches a newline (CR Only) (paragraph) (MAC Files)
^n	matches a newline (LF Only) (paragraph) (UNIX Files)
^t	matches a tab character


Note - ^ refers to the character '^' NOT Control Key + value.

If the  button is pressed the find history is displayed.

If the  button is pressed the **Favorites** submenu with the following options is presented:

Add to Favorites	adds current item in combobox to Favorites list
Edit Favorites	presents Edit Find/Replace Favorites dialog where users can name favorite search strings, organize the list and remove items from the list
Favorites	presents submenu displaying the Favorites list. Strings may be selected from the list to populate the combobox



If the  button is pressed a list of regular expression options appropriate to the selected **Regular Expression Engine** is displayed. This button is subdued unless the **Regular Expressions** checkbox is selected.

To select all text between the cursor position and the search target, hold down the **SHIFT** key at the time the search is started with the Find Next button.

Search Direction

The default search direction is down (toward the end of the document). The option to search up or down the document is given. If a search has already occurred in the document, the default search direction is that of the previous search.

Count All

Count All presents a dialog indicating the number of times the current search string occurs in the active file.

Find Where [3]

The user may specify if the search should be executed in the **Current File**, Selected Text or **All Open Files** in the Find Where section and the replace will be limited to the specified area. If **All Open Files** is selected, pressing **Find Next** will automatically move focus to the first occurrence in the next open file. This functionality replaces the **Find Next through all opened files** option from older versions.

The following search options **[2]** exist:

Match Whole Word Only

By default, the search will not be restricted to matching whole words only. This option allows the search to be restricted to whole words only. This is useful for searching for 'a', or 'A' without finding all words including 'a'. If a search has already occurred in a document, the default selection will match the previous search.

Match Case

By default, the search will not be case sensitive. The option to make the search case sensitive is given. If a search has already occurred in the document, the default selection for case sensitivity is that of the previous search.

Regular Expressions

If this option is selected the search will be executed using the regular expression type selected under the **Advanced** section.

Advanced Options [4]

Pressing the **Advanced** button extends the Find in Files dialog to display additional search options.

Regular Expressions

Perl Compatible Regular Expressions

If this option is selected the editor will use Perl style regular expressions based on the Boost C++ Libraries.

UltraEdit Style Regular Expressions

If this option is selected the editor will use UltraEdit style regular expressions.

UNIX Style Regular Expressions

If this option is selected the editor will use UNIX style regular expressions.

For information regarding creation of regular expressions see Regular Expressions or Perl Regular Expressions help.

Highlight All Items Found

This option causes all items matching the search string to be highlighted (shown as selected text) when the **Find Next** button is pressed.

List Lines Containing String

This option causes UltraEdit to search for all occurrences of the string in the active document and to list them in a dialog box. By double-clicking on one of the result lines (or selecting it and clicking on Goto) UltraEdit will position to that line in the file. The **Lines Containing Find String** dialog includes a **Shown Lines** checkbox option. If this is selected, the line numbers associated with the matched strings will be displayed in the **Lines Containing Find String** dialog.

Search In Column

If this option is selected the search will be restricted to the specified start and end columns. The number specified for the end column must be greater than the number specified for the start column (except when searching an entire line). To search an entire line, "0" and "-1" should be used. It is important to note that column numbering begins with "0" so **From Col.** would have to be set to "0" to search from the beginning of a line.

If the editor active window is in [Hex mode](#), an additional option exists:

ASCII

By default, in Hex mode, the search will be for hexadecimal characters. Optionally the find may be configured to search for ASCII characters.

When a find operation is complete the text is highlighted.

Initially after the find is performed, a slim dialog box is displayed enabling easy selection of find and find next. This may be disabled by changing the INI settings.

Hide Lines

If this button is pressed all lines matching the specified search string will be hidden in the active file using code folding.

Show Lines

If this button is pressed only lines matching the specified search string will be displayed in the active file. All non-matching lines will be hidden using code folding.

Shortcut: ALT+F3 or CTRL+F

Find Next command (Search menu)



This command repeats the previous [Find command](#) in the forward direction toward the end of the file. The next occurrence of the find text will be found (if present) in the forward direction.

To select all text between the cursor position and the search target, hold down the SHIFT key at the time the search is started with the Find Next menu item or shortcut key.

Shortcut: F3

Find Prev command (Search menu)



This command repeats the previous [Find command](#) in the backward direction toward the beginning of the file. The next occurrence of the find text will be found (if present) in backward direction.

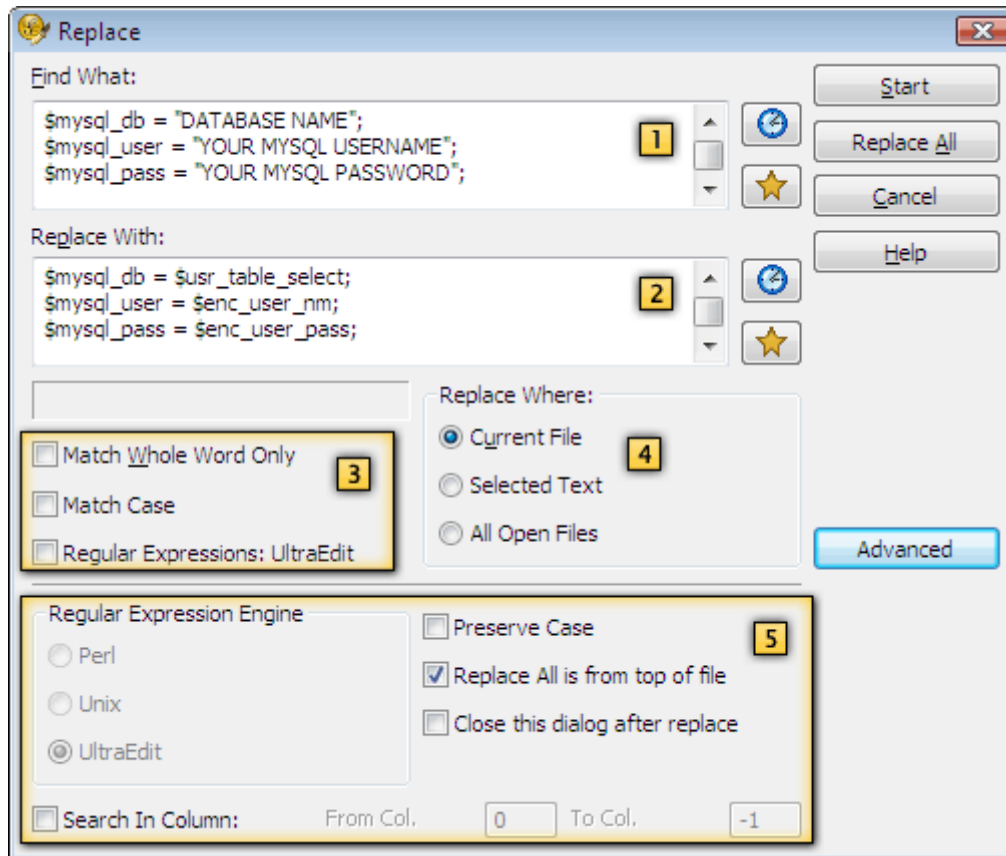
To select all text between the cursor position and the search target, hold down the SHIFT key at the time the search is started with the Find Prev menu item or shortcut key.

Shortcut: CTRL+F3

Replace command (Search menu)



This command displays a dialog box for searching the document for the specified text and replacing it with a different text.




The string to search for is specified in the **Find What [1]** field and the replace string to be used is defined in the **Replace With [2]** field. Single or multi-line strings up to 20,000 bytes may be used in both fields.

The search is performed in the direction specified. If the string is not found the search wraps around to the other end of the file and continues the search. If the string is not found a message is displayed.

In addition to searching and replacing printable characters, tab characters and newline (Carriage Return/Line Feed) characters may be searched for or replaced. To specify a Tab character, use `"^t"` characters in the dialog. To specify the newline characters, use `"^p"` characters for CR/LF, `"^n"` for LF only and `"^r"` for CR only in the dialog. To search for a `"^"` character, use `"^^"` in the dialog. To specify a `"^"` character, use `"^^"` in the dialog.

Additionally, the characters `"^s"` will be replaced with any text that is highlighted when a find is executed and the characters `"^c"` will be replaced with the contents of the clipboard (up to 30,000 characters). This is useful particularly for macros. When recording a macro, the option exists to include a find of a specified character string, or using `"^s"` the macro records the `"^s"` and when replayed will determine the text highlighted at the time the macro is playing back, likewise when using `"^c"`, these characters will be replaced with the contents of the clipboard at the time the macro is playing back.

If the  button is pressed the **Find What / Replace With** history is displayed.

If the  button is pressed a submenu with the following options is presented:

Add to Favorites	adds current item in combobox to Favorites list
Edit Favorites	presents Edit Find/Replace Favorites dialog where users can name favorite search strings, organize the list and remove items from the list
Favorites	presents submenu displaying the Favorites list. Strings may be selected from the list to populate the combobox

Special character summary:

Symbol	Function
^^	searches for a "^" character
^s	searches for selected (highlighted) text when a macro is running
^c	searches for the contents of the clipboard when a macro is running
^b	matches a page break
^p	matches a newline (CR/LF) (paragraph) (DOS Files)
^r	matches a newline (CR Only) (paragraph) (MAC Files)
^n	matches a newline (LF Only) (paragraph) (UNIX Files)
^t	matches a tab character

Note - ^ refers to the character '^' NOT Control Key + value.

Find Next

Find next finds the next occurrence of the specified text.

Replace

Replace finds the next occurrence of the specified text and replaces it with the new text. The user may also specify Selected Text under the Replace Where **[4]** selection and the replace will be limited to the selected area. If the following items are NOT selected in Configuration under the Advanced menu:

Find dialog defaults to word under cursor (Search -> [Miscellaneous](#))

Always set 'Find/Replace Where' to 'Current File' in Find/Replace dialog (Search -> [Auto Reset Settings](#))

the **Replace Where** option will automatically default to **Selected Text** when a selection exists.

If **All Open Files** is selected, pressing **Replace** will automatically move focus to the first occurrence in the next open file. This functionality replaces the **Replace Next through all opened files** option from older versions.

Replace All

Replace all finds every occurrence of the specified text and replaces each occurrence with the new text.

The replace all provides the additional options:

Replace in current file - this replaces all occurrences within the current file only. It determines if the replace should be from the top of the file, or the cursor position based on the setting in this dialog box.

Replace all within the selected text - this replaces only the occurrences of the string that fall within the area of selected text at the time the replace is started.

Replace all in all the open files - this replaces all occurrences of the string in all open files.

Last

The last command performs a single replace and closes the dialog without moving on to the next occurrence of the string.

The following replace options **[3]** exist:

Match Whole Word Only

By default, the search will not be restricted to matching whole words only. This option allows the search to be restricted to whole words only. This is useful for searching for 'a', or 'A' without finding all words including 'a'. If a search has already occurred in a document, the default selection will match the previous search.

Match Case

By default, the search will not be case sensitive. The option to make the search case sensitive is given. If a search has already occurred in the document, the default selection for case sensitivity is that of the previous search.

Regular Expressions

If this option is selected the search will be executed using the regular expression type selected under the Advanced section.

Advanced Options [5]

Pressing the **Advanced** button extends the Find in Files dialog to display additional search options.

Regular Expressions

Perl Compatible Regular Expressions

If this option is selected the editor will use Perl style regular expressions based on the Boost C++ Libraries.

UltraEdit Style Regular Expressions

If this option is selected the editor will use UltraEdit style regular expressions.

UNIX Style Regular Expressions

If this option is selected the editor will use UNIX style regular expressions.

For information regarding creation of regular expressions see Regular Expressions or Perl Regular Expressions help.

Preserve Case

The option to preserve cases causes UltraEdit to match the case of the replace string with the case of the found string. The rules for this are as follows:

- 1) If the word found is all lowercase the replacement word will be all lower case
- 2) If the word found is all UPPERCASE the replacement word will be all UPPERCASE
- 3) If the word found is MixedCase the replacement word will MATCH the casing of the replacement word.

Example: (Find "The", replace with "Then")

"the" will be replaced with "then"

"THE" will be replaced with "THEN"

"The" will be replaced with "Then"

The mixed case example requires the replacement word specified to have the mixed case in the replacement string that is required for the result so that a word that may have multiple mixed case characters may be correctly replaced, not just capitalization as in:

Replace "NextCharacter" with "PreviousCharacter" would replace:

"nextcharacter" with "previouscharacter"

"NEXTCHARACTER" with "PREVIOUSCHARACTER"

"NextCharacter" with "PreviousCharacter"

Replace All is From Top of File

If this option is selected the specified search will be executed from the top of the active file rather than the current cursor position.

Close this dialog after replace

If this option is selected the Replace dialog will automatically be closed after a single replace is executed.

Search In Column

If this option is selected the search will be restricted to the specified start and end columns. The number specified for the end column must be greater than the number specified for the start column (except when searching an entire line). To search an entire line, "0" and "-1" should be used. It is important to note that column numbering begins with "0" so **From Col.** would have to be set to "0" to search from the beginning of a line.

If the editor active window is in [Hex mode](#), an additional option exists:

ASCII

By default, in [Hex mode](#), the search will be for hexadecimal characters. Optionally the find may be configured to search for ASCII characters.

When a find operation is complete the text is highlighted. In Hex mode additional characters may be highlighted if the specified search string crosses a line. In this case, all characters displayed from the first character of the search string on the first line to the last character of the search string on the last line are highlighted. If a replace is requested, only the search string will be replaced.

Initially after the replace is performed, a slim dialog box is displayed enabling easy selection of find next and replace operations. This may be disabled by changing the INI settings.

[Undo](#) is supported for Replace All. However if the replace is too large, Undo will not be supported and the Undo buffer will be flushed.

Shortcut: CTRL+R

Incremental Search (Search menu)



This command is similar to the [Find command](#). It performs a find, however instead of waiting for the complete search string to be entered it will search for the closest match as the string is typed, repositioning to the found string as it is typed. This makes it a lot quicker in many cases to find the correct occurrence of a string.

If an error is made while typing the string, backspace may be used to modify what has already been entered. If backspace is used, the search will begin again from the original position finding the first occurrence of the modified string.

At any time during the search, pressing Escape or any of the cursor keys will cause the search to stop. If focus is changed from the edit window this will also stop the search.

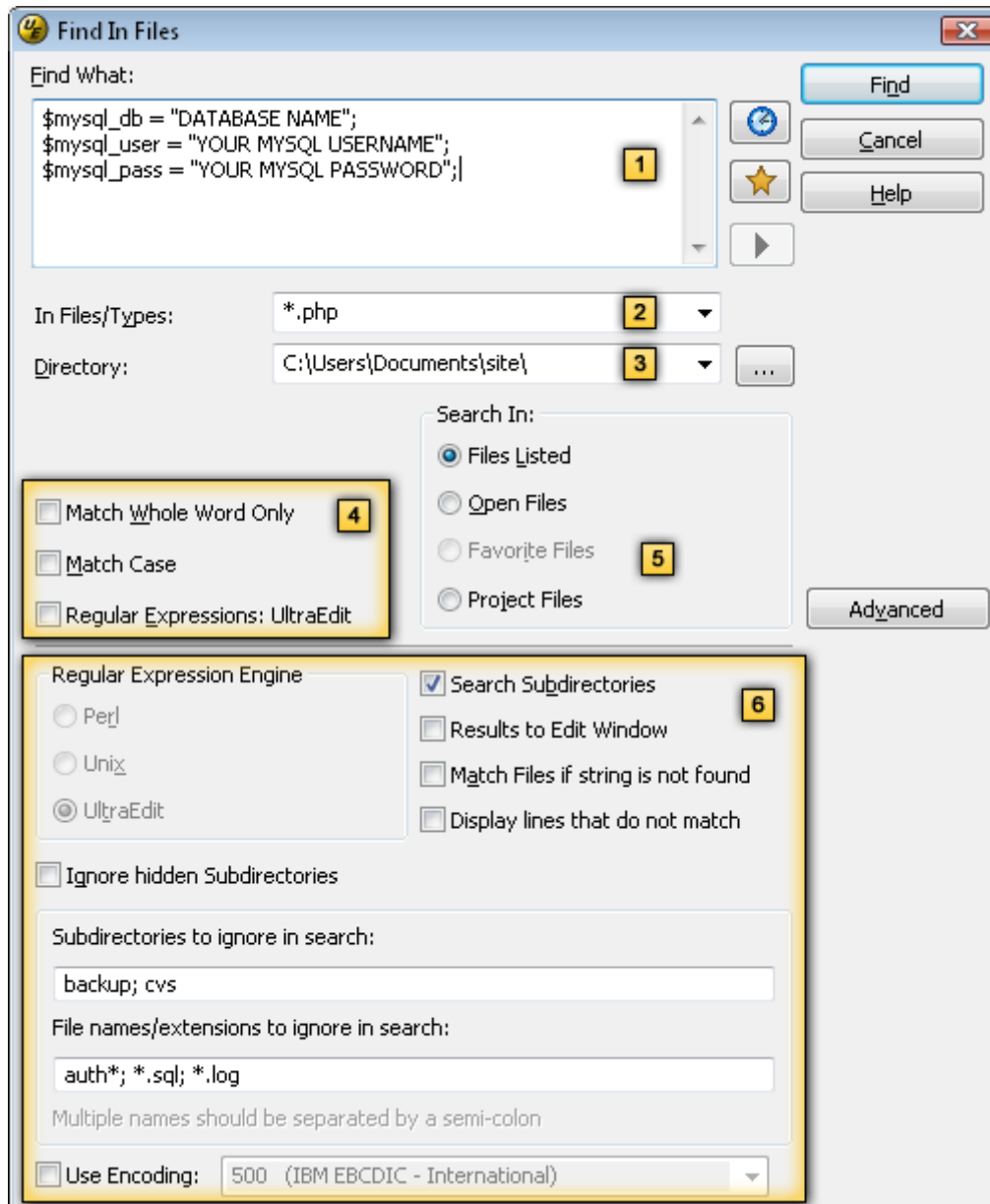
While the search is active, the search string is shown in the status bar.

At the end of an incremental search, the Back command may be used to position to the original position prior to the start of the search.

Find in Files command (Search menu)



This command displays a dialog box for searching for a text string in files on a disk.



The string to search for is specified in the **Find What [1]** field. A single or multi-line string up to 20,000 bytes may be used. The files to search are specified in the In Files/Types **[2]** and Directory **[3]** fields. The directory field specifies the directory to search and the In Files/Types field specifies the filename or types. The filename may contain wildcards and may also contain multiple file types separated by semicolons, i.e.:


.cpp;.h


The Directory [3] field may contain multiple search paths separated by semicolons, i.e.:

C:\mydir1\; c:\mydir2\


Please note: If the **Find [1]** string is left blank the editor will return a list of all filenames that match the specified criteria. This can be useful in generating a directory listing.

When the find is executed, the existing output is cleared if the output is being sent to the output window. If being sent to an edit window, a new file is created if one does not already exist with the Find Results title. If it does exist, the results are appended to the end of the existing file.

If the  button is pressed the find history is displayed.

If the  button is pressed a submenu with the following options are presented:

Add to Favorites	adds current item in combobox to Favorites list
Edit Favorites	presents Edit Find/Replace Favorites dialog where users can name favorite search strings, organize the list and remove items from the list
Favorites	presents submenu displaying the Favorites list. Strings may be selected from the list to populate the combobox

If the  button is pressed a list of regular expression options appropriate to the selected **Regular Expression Engine** is displayed. This button is subdued unless the **Regular Expressions** checkbox is selected.

A search may be limited to file groups as follows:

Search In [5]:

Files Listed - This will search the files listed in In Files/Types and Directory fields (default selection).

Open Files - This will search all open files. The In Files/Types and Directory fields will be ignored.

Favorite Files - This will search all files listed under Favorite Files in the File menu. The In Files/Types and Directory fields will be ignored.

Project Files - This will search all files within the open project. The In Files/Types and Directory fields will be ignored.

Solution Files - This will search all files within the open solution. The In Files/Types and Directory fields will be ignored.

***Solution Files option only supported in UESstudio.**

The following search options **[4]** exist:

Match Whole Word Only

By default, the search will not be restricted to matching whole words only. This option allows the search to be restricted to whole words only. This is useful for searching for 'a', or 'A' without finding all words including 'a'. If a search has already occurred in a document, the default selection will match the previous search.

Match Case

By default, the search will not be case sensitive. The option to make the search case sensitive is given.

Regular Expressions

If this option is selected the search will be executed using the regular expression type selected under the Advanced section.

Advanced Options [6]

Pressing the **Advanced** button extends the Find in Files dialog to display additional search options.

Regular Expressions

Perl Compatible Regular Expressions

If this option is selected the editor will use Perl style regular expressions based on the Boost C++ Libraries.

UltraEdit Style Regular Expressions

If this option is selected the editor will use UltraEdit style regular expressions.

UNIX Style Regular Expressions

If this option is selected the editor will use UNIX style regular expressions.

For information regarding creation of regular expressions see Regular Expressions or Perl Regular Expressions help.

Search Subdirectories

By default, only files specified in the specified directory are searched. This instructs the search to search all subdirectories of the specified directory. If the **Ignore hidden Sub Directories** option is selected, then sub directories which are hidden (as defined in Microsoft Explorer) will not be included in the search.

If desired, the user may exclude subdirectories by specifying them in the **Subdirectories to ignore in search** field. Multiple names should be separated by a semi-colon. Users may also specify file names or extensions which should be ignored in the search in the **File names/extensions to ignore in search** field. For example: "*.cpp;index.*;" would instruct the search not to match files with a "*.cpp" extension or files that are named "index" (regardless of extension).

Results to Edit Window

This option instructs UltraEdit to create an edit window (new file) for the results of the find. This allows editing of the results, saving or printing the results. The default action is to capture the output in a list box window (Output Window) that allows the user to double click on the result and have the file opened automatically.

Match Files if string is not found

This option instructs UltraEdit to list only files that DO NOT include the specified search string.

Display lines that do not match

This option instructs UltraEdit to list each individual line that does not match the search parameters. Whereas **Match Files if strings is not found** only lists files that do not include the specified search string, this option returns data on a line-by-line basis.

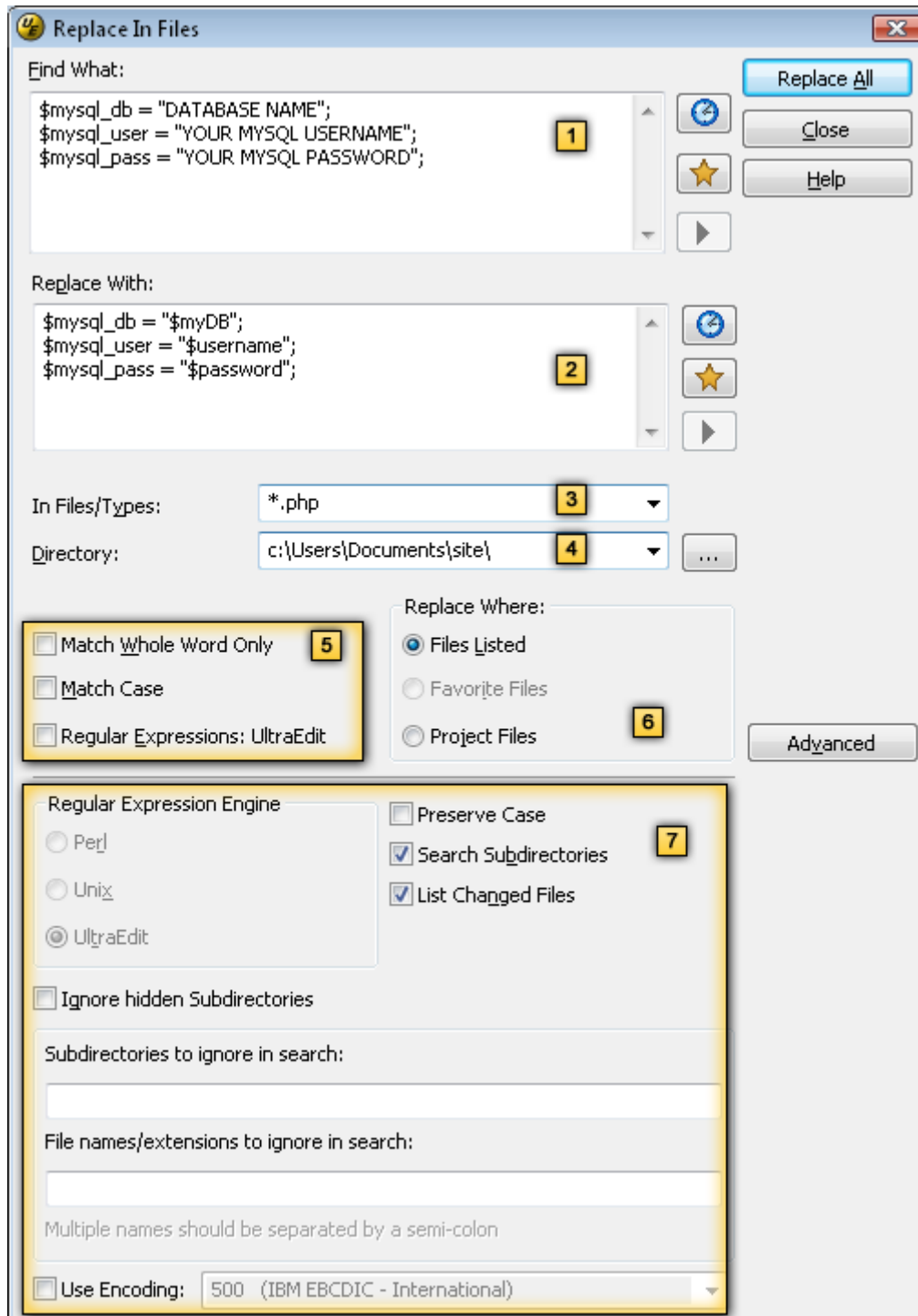
Use Encoding

By default this option is disabled. When the checkbox is selected, the dropdown may be used to specify the encoding to be used when searching for the specified string.

Replace in Files command (Search menu)



This command displays a dialog box for searching for a text string in files on a disk and replacing it with another text string.



The string to search for is specified in the **Find What [1]** field and the replace string to used is defined in the **Replace With [2]** field. Single or multi-line strings up to 20,000 bytes may be used in both fields.


The files to search are specified in the In Files/Types and Directory fields. The **Directory [4]** field specifies the directory to search and the In Files/Types **[3]** field specifies the filename or types. The


filename may contain wildcards (an asterisk) and may also contain multiple file types separated by semicolons, i.e.:

.cpp;.h


The Directory [4] field may contain multiple search paths separated by semicolons, i.e.:

C:\mydir1\; c:\mydir2\

If the  button is pressed the find history is displayed.

If the  button is pressed a submenu with the following options are presented:

Add to Favorites	adds current item in combobox to Favorites list
Edit Favorites	presents Edit Find/Replace Favorites dialog where users can name favorite search strings, organize the list and remove items from the list
Favorites	presents submenu displaying the Favorites list. Strings may be selected from the list to populate the combobox

If the  button is pressed a list of regular expression options appropriate to the selected **Regular Expression Engine** is displayed. This button is subdued unless the **Regular Expressions** checkbox is selected.

Replace Where [6]

Users may specify if changes should be made to Files Listed, **Open Files**, **Favorite Files**, Project Files, or Solution Files. If Files Listed is selected then the **File Types** and specified **Directory** will be used to determine the files to be modified by the Replace operation. If Project Files or **Solution Files** is selected then the Replace would affect all files included in the active project/solution.

When the find/replace is executed, each file where the find string is found is modified and each occurrence of the string is replaced with the replace string. If the file is [read only](#), the file is not modified

and a message box is posted. At the end of the operation, a dialog box is displayed indicating how many times the replace occurred and how many files were modified.

The following search options **[5]** exist:

Match Whole Word Only

By default, the search will not be restricted to matching whole words only. This option allows the search to be restricted to whole words only. This is useful for searching for 'a', or 'A' without finding all words including 'a'. If a search has already occurred in a document, the default selection will match the previous search.

Match Case

By default, the search will not be case sensitive. The option to make the search case sensitive is given.

Regular Expressions

If this option is selected the search will be executed using the regular expression type selected under the Advanced section.

Advanced Options [7]

Pressing the **Advanced** button extends the Find in Files dialog to display additional search options.

Regular Expressions

Perl Compatible Regular Expressions

If this option is selected the editor will use Perl style regular expressions based on the Boost C++ Libraries.

UltraEdit Style Regular Expressions

If this option is selected the editor will use UltraEdit style regular expressions.

UNIX Style Regular Expressions

If this option is selected the editor will use UNIX style regular expressions.

For information regarding creation of regular expressions see Regular Expressions or Perl Regular Expressions help.

Preserve Case

The option to preserve cases causes UltraEdit to match the case of the replace string with the case of the found string. The rules for this are as follows:

- 1) If the word found is all lowercase the replacement word will be all lower case
- 2) If the word found is all UPPERCASE the replacement word will be all UPPERCASE
- 3) If the word found is MixedCase the replacement word will MATCH the casing of the replacement word.

Example: (Find "The", replace with "Then")

"the" will be replaced with "then"

"THE" will be replaced with "THEN"

"The" will be replaced with "Then"

The mixed case example requires the replacement word specified to have the mixed case in the replacement string that is required for the result so that a word that may have multiple mixed case characters may be correctly replaced, not just capitalization.

Replace " NextCharacter" with " PreviousCharacter" would replace:

" nextcharacter" with " previouscharacter"

" NEXTCHARACTER" with " PREVIOUSCHARACTER"

" NextCharacter" with " PreviousCharacter"

Search Subdirectories

By default, only files specified in the specified directory are searched. This instructs the search to search all subdirectories of the specified directory. If the **Ignore hidden Sub Directories** option is selected, then sub directories which are hidden (as defined in Microsoft Explorer) will not be included in the search.

If desired, the user may exclude subdirectories by specifying them in the **Subdirectories to ignore in search** field. Multiple names should be separated by a semi-colon. Users may also specify file names or extensions which should be ignored in the search in the **File names/extensions to ignore in search** field. For example: "*.cpp;index.*;" would instruct the search not to match files with a "*.cpp" extension or files that are named "index" (regardless of extension).

List Changed Files

This instructs the editor to list all the files that were modified in the Output Window. If set, each file change will be listed in the Output Window followed by the number of occurrences found and changed.

Ignore hidden Sub Directories

If this option is selected hidden subdirectories will not be affected by the Replace in Files operation.

Sub Directory names to ignore in search

If this option is selected hidden subdirectories will not be affected by the Replace in Files operation.

Use Encoding

By default this option is disabled. When the checkbox is selected, the dropdown may be used to specify the encoding to be used when searching for the specified string.

Match Brace command (Search menu)



This command searches forward in the file from the cursor position to find an open or close brace. If an open brace is found, the search continues in a forward direction until a matching close brace is found at the same nest level. If the first brace found is a close brace, the search is performed in the reverse direction until a matching open brace is found at the same nest level.

When the matching brace is found, all the text between and including the two braces is highlighted.

By default an open brace is '(', '[', or '{' and a close brace is ')', ']' and '}'. Users may specify different characters/strings for open/close braces on a per language basis using the Open Brace Strings and Close Brace Strings commands in the wordfile as described under [Syntax Highlighting](#).

Additionally, UltraEdit will automatically highlight a matching bracket/brace if the match is in view when this cursor is positioned to the right or left of a bracket/brace. No screen positioning will be performed, the bracket/brace will only be highlighted if it is in view.

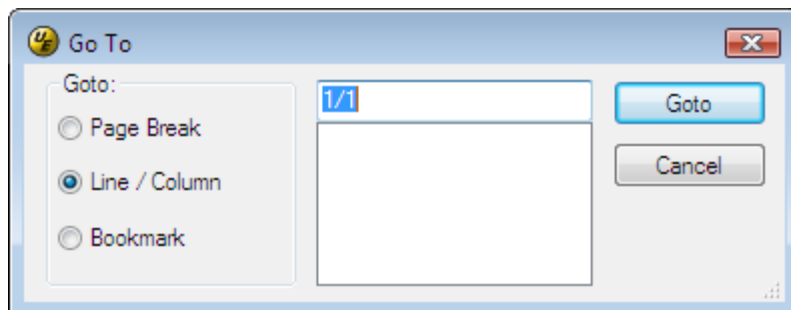
If the file is designated as an HTML file (see Syntax Highlighting) matching '<' and '>' characters will be highlighted also.

Shortcut: CTRL+B

Goto Line/Page/Bookmark command (Search menu)



The Goto Line/Page/Bookmark menu item presents a dialog that allows the user to specify the line/column number, page break, or bookmark at which the cursor should be positioned.



When **Page Break** is selected, the cursor is positioned at the line with the specified page break character.

When **Line/Column** is selected, the cursor is positioned at the line/column specified and the line is scrolled into view. If the line number is greater than the number of lines in the file, the cursor is positioned at the end of the file. If the column number is greater than the number of columns in the file, the cursor is positioned at the end of the specified line.

When **Bookmark** is selected, the cursor is positioned at the line with the specified bookmark.

When the Shift Key is pressed at the same time as performing the Goto function, the current selection (if present) or a new selection will be extended to select text up to the cursor position following the command.

In [Hex mode](#) the goto dialog allows the user to select the offset into the file at which the cursor should be positioned. If the offset is greater than the file size, the cursor is positioned at the end of the file.

Shortcut: CTRL+G

Toggle Bookmark command (Search menu)



UltraEdit allows you to set a "bookmark" to mark a position in a file. Once a bookmark has been set at a line, you may use menu or keyboard commands to move to that line.

To set a bookmark:

1. Move the cursor to the line where you want to set a bookmark.
2. From the Search menu, choose Toggle Bookmark.

Shortcut: CTRL+F2

To remove all the bookmarks, use the Clear All Bookmarks menu item.

To move to the next bookmark after the insertion point, choose Next Bookmark from the [Search menu](#).

Shortcut: F2

To move to the previous bookmark after the insertion point, choose Previous Bookmark from the [Search menu](#).

Shortcut: ALT+F2

When the Shift Key is pressed at the same time as performing the Goto Bookmark function the current selection (if present) or a new selection will be extended to select text up to the cursor position following the command.

Word Count command (Search menu)



The word count feature analyzes the document and selection and determines the number of Words, Lines and Characters (with and without spaces) and displays the result. If a portion of the file is selected, the results are shown for that portion of the file and the complete file.

The number of lines determined are real lines terminated with the line terminator. If a line is wrapped in the display, this is not counted as multiple lines.

Function List command (Search menu)



This command shows the list of functions in the active document. The results are displayed in dockable window either in alphabetical order, or in order they occur in the document. By double-clicking on one of the result lines UltraEdit will position to that line in the file and the focus will change to the file. If Enter is pressed while focus is in the Function List UltraEdit will position to that line and focus will remain in the function list. CTRL+Enter will also position to the line in the file and the focus will change to the file.

Optionally the Function List may be set to search all documents in the active project. This is set by right-clicking in the Function List and selecting the appropriate menu item.

Additionally, by right-clicking in the Function List the user may select to sort (or not) the Function List in alphabetical order.

UltraEdit searches for a particular pattern to determine if a line is considered a function. The pattern that is searched for is a regular expression string(s) that may be modified by the user.

The regular expression string(s) is defined in the active wordfile and each language may have its own definition.

By default the regular expression string uses UltraEdit style regular expressions. If desired, Perl compatible Regular Expressions may be used for the function strings by adding the following:

```
/Regexp Type = Perl
```

See the [Syntax Highlighting](#) section for details on changing the string used for recognizing a function.

While navigating through the file, UltraEdit will attempt to highlight the current function that the cursor is in. In reality it will find the previous function in the file (to the cursor position) that it recognized as a function and highlight this. If the cursor is between functions, or after the last function, the previous function recognized will be highlighted. If the option for all files in the project to be shown in the function list is set, the position of the cursor will not be highlighted.

Find Symbol (Search menu)



This command searches the tag file for the active symbol and if only one match is found the file will be opened in UltraEdit and the file positioned to the symbol. If multiple matches occur, a dialog box will be shown giving the user the option to position to the appropriate tag.

Character Properties (Search menu)



This command causes a dialog to be displayed showing the decimal value and hexadecimal value of the active byte as well as how this will be displayed (this may vary based on font and script). Along with this the offset of the active byte is displayed in both decimal and hexadecimal formats.

Back/Forward command (Search menu)



UltraEdit maintains a rotating list of positions that the user last edited, changed position from, or jumped to another file from with the [Find Symbol command](#). This allows the user to quickly jump back to a previous position or file after changing position within a file, or to another file.

The Back command repositions the cursor to the previous position/file in the list. When a user jumps back, the Forward command is enabled enabling the user to navigate forward again to the position the user just came from.

When a change is made, or a reposition occurs that updates the list of remembered positions, the new position is added to the list at the current list position. This enables the user to continue to navigate forwards/backwards as desired.

When editing within a file, only the last edit position is remembered since the last reposition, not every position for each character added or change made in sequence is remembered.

Insert Menu

Insert menu



The Insert menu includes the following commands:

Date/Time	inserts the date/time in the active file
Line	inserts blank line below cursor position
Page Break	inserts a page break at the cursor position
File	inserts a file into the active file
String At Every Increment	inserts a string at the specified increment in the active file
Literal Character	inserts a literal character or control code (such as Control C) into the active file

Template	inserts a user-defined template into the active file
Individual Templates	presents submenu for selecting named templates to be inserted into the active file
Color	inserts a color value into the active file

Date/Time command (Insert menu)



The date/time command inserts the date and time at the cursor position. The date and time are formatted according to the control panel settings. A single space is inserted between the date and time.

To insert only the date or the time a [Template](#) definition feature may be used.

Shortcut: F7

Line command (Insert menu)



The Line command inserts a new blank line below the current cursor position. The cursor may be positioned anywhere on the active line when this feature is invoked. This command may be invoked from the [Insert menu](#) or the user may define a keyboard shortcut under [Key Mapping](#) in Configuration.

Shortcut: Ctrl+Enter

Page Break command (Insert menu)



The Page Break command inserts a page break character at the current cursor position. The page break character can be configured in the [Page Setup dialog](#).

When the document is printed, the page break character is replaced with a page break. Printing continues on the next page following a page break character.

File command (Insert menu)



Use this command to insert an existing document (file) into the active document.

This command presents a dialog allowing selection of the file to insert. The **Format** dropdown is used to specify the encoding of the file which is being inserted into the active file. This will enable UltraEdit to convert the file to the format of the target file if necessary.

The complete contents of the selected file are inserted in the active document at the cursor position.

String at Every Increment command (Insert menu)



When handling large database type files it is sometimes necessary to split the file into "records" or known line lengths as the file has been output without any delimiters/separators between records.

This feature allows a user-specified string to be inserted at a constant user-specified increment in the file from some start point in the file to some end point in the file.

Start Point:

The start point for this operation may be the beginning of the file or the position of the cursor. This is helpful if the file has a header in it that should not be treated as part of the data. The default behavior is that the beginning of the file is the start point.

End Point:

The end point may be specified as one of the following:

The end of the file (default behavior)

A known offset into the file

After the specified number of records have converted

When a particular string is found, in this case the insertion stops at or before the beginning of the string if found, if not found the option to continue or not is given.

Example:

A database file has records of known length of 2000 characters. Each record has been output consecutively with no separation between the records.

For editing purposes you want to put each record onto a single line.

With this feature you can specify the string to insert is "^p" (insert a newline - paragraph) and that the increment at which to insert the string is every 2000 characters.

UltraEdit will add the new line at the end of each record until the end of file is encountered.

Literal Character command (Insert menu)



This command allows a literal character (such as Control C) to be inserted into the file. Pressing Ctrl+I tells UltraEdit that the next character is to be inserted directly into the file without any translation.

This is useful for inserting control codes that may be needed by printers or other devices.

Example:

Ctrl+I followed by Ctrl+C inserts a Control C character into the file

Ctrl+I followed by ESC inserts an escape character into the file

Shortcut: Ctrl+I

Template command (Insert menu)



Use this command to insert a template. It presents a dialog allowing you to select the template you wish to insert from the list of templates.

Optionally, the hot keys Alt+0 through Alt+9 and Shift+Alt+0 through Shift+Alt+9 may be used instead of this menu command.

Color command (Insert menu)



This function allows a user to select a color value and insert it into a document. The dialog allows the user to view or choose a color and then have the value of the color inserted into the document.

The color value is shown in one of the three specified formats:

- 1) Decimal - This is the decimal value for the color value in RGB (Red,Green,Blue).
- 2) Hexadecimal - This is the hexadecimal value for the color value in RGB (Red,Green,Blue).
- 3) HTML - This is the format required for display in an HTML document. The # is added to the value.

When the dialog is first displayed, the color value will default to the selected color (if there is a value selected) in the active file. UltraEdit will try and determine the format of this also based on the selection.

When the user clicks on OK, the Color Value shown will be inserted into the document, replacing the current selection if a selection is made.

Project Menu

Project menu



The Project menu allows you to manage your projects or workspaces. This provides an easy way to manage and group associated files.

This functionality combines the grouping of files for quick access via the [Files dialog](#) and the saving/restoring of the open files state.

New Project/Workspace	creates and opens a new project file
Open Project/Workspace	opens an existing project file
Close Project/Workspace	closes a project
Project Settings	adds, removes, or opens files within the project
Add To Project	presents submenu to insert files/folders/groups into active project
* Insert	presents submenu to insert C++, Java, or C# Class into active project
Back Up Project/Workspace	copies all active files in the current project/workspace to a compressed backup file (. bup) stored in the same directory as the . prj file for the active project/workspace
Restore Project/Workspace	presents Restore Project Files dialog to reload a project from a copy that was previously backed up
* Project Converter	launches Visual Studio to UESTudio Project Converter dialog
* Project Manager	launches Project Manager dialog from which users may open, create, backup, restore, or batch build projects
* Checkout	checks out module from repository
* Update	commits working copy from repository without displaying any prompts or options
* Commit	commits working copy to repository without displaying any prompts or options.
* Update(Special)	allows configuration of multiple CVS-related options using CVS Update dialog
* Tag	creates tag (branch) in CVS tree
* Import	imports new module into repository
* Export	checks out module excluding CVS information
* Accounts/Settings	launches CVS Manager dialog
Create CTAG File	creates tag file
CTAG Options	sets up options for Ctags

* Please note: These features are only supported in UESTudio, and not in UltraEdit.

New Project/Workspace command (Project menu)



The new project command allows you to create a new project/project file.

A project allows you to manage your projects or workspaces. This provides an easy way to manage and group associated files.

In many instances a group of files may be related and while editing a particular file there may be reason to edit or view other associated files. These files can be grouped as a project/workspace.

Grouping these files into a project allows you to easily access and/or open all the associated files when needed without having to search for them.

The project support provides combined project and workspace functionality. In many cases it may be convenient to use the automatic workspace support as described below:

Workspace

The workspace functionality saves the state of all open files in the project file when the project is closed including window position, cursor position and which files are open.

When a project is opened all the files that were previously opened when the project was closed are opened automatically. The files do not have to be specifically added to the project via the [Files dialog](#).

Project

The project functionality allows files to be associated together even if they are not currently open in the editor. Files may be added to the project or removed from the project via the [Files dialog](#). The files within the project can easily be opened via the [Files dialog](#).

Open Project/Workspace command (Project menu)



The Open Project command allows you to open an existing project that must previously have been created with the [New Project/Workspace command](#). When a new project is opened, the current project (if opened) will be closed. The option will be given to close files that are currently open.

When the project is opened, any files that were previously open when the project was last closed will be opened automatically.

Close Project/Workspace command (Project menu)



The Close Project command closes the currently opened project. The state of all files will be saved in the project file allowing them to be opened automatically when the project is next [opened](#). The option will be given to close files that are currently open.

Project Settings command (Project menu)



The Project Settings command presents the Project Settings dialog. This dialog is used to configure the settings for the active project. Files/folders may also be added to or removed from the active project using the context menu options in the [File Tree View](#).

Where possible, the path to files in a project will be relative to the location of the project file. In some instances, if the relative path cannot be resolved, the path would be the FULL path.

In some cases it is advantageous to group multiple sets of files together within a single project. UltraEdit provides for grouping of these files in named groups. These groups may be added or deleted as needed and files may be added to each group.

The dialog shows a list of files that are part of the project. The following controls are used to manage the list:

Open

The Open button opens the selected files from the list and closes the dialog. Multiple files may be selected from the list and opened at once. Any changes to the list are saved. DOUBLE CLICK of any files in the file list will perform the open command.

Close

The Close button closes the dialog. Any changes to the list are saved.

Update

This updates the project information (committing the changes) and refreshes the project list.

New Group

The New Group button presents a dialog box to the user allowing the user to specify the name of the group about to be added. Nested groups are permitted.

Up to 2000 files may be added to a project.

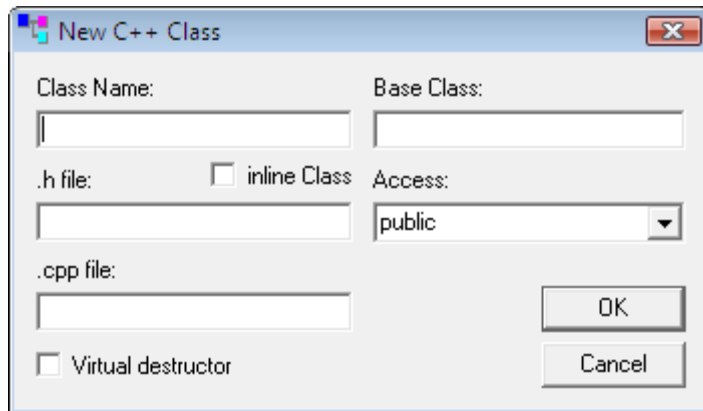
Insert command (Project menu)



This command presents a submenu where the user may select to insert a new C++ Class or a Java/ C# Class. Please note: This feature is only supported in UESstudio, and not in UltraEdit.

New C++ Class

If the "C++ Class" option is selected the New C++ Class dialog is presented. This dialog is used to create a new class and automatically generates the associated .h and .cpp files.



Class Name

The Class Name field is used to set the name of the new class. This name will be used to create the associated .h and .cpp files.

Base Class

This field is used to set the base class for the new class.

Access

This dropdown is used to set the access modifiers of the new class. Access modifiers are keywords that specify the access other classes have to the class member functions. This may be set to public, private, or protected.

Public	<p>When preceding a list of class members, the public keyword specifies that those members are accessible from any function. This applies to all members declared up to the next access specifier or the end of the class.</p> <p>When preceding the name of a base class, the public keyword specifies that the public and protected members of the base class are public and protected members, respectively, of the derived class.</p>
Private	<p>When preceding a list of class members, the private keyword specifies that those members are accessible only from member functions and friends of the class. This applies to all members declared up to the next access specifier or the end of the class.</p>

	When preceding the name of a base class, the private keyword specifies that the public and protected members of the base class are private members of the derived class.
Protected	<p>The protected keyword specifies access to class members in the member-list up to the next access specifier (public or private) or the end of the class definition. Class members declared as protected can be used only by the following:</p> <ul style="list-style-type: none"> - Member functions of the class that originally declared these members. - Friends of the class that originally declared these members. - Classes derived with public or protected access from the class that originally declared these members. - Direct privately derived classes that also have private access to protected members. <p>When preceding the name of a base class, the protected keyword specifies that the public and protected members of the base class are protected members of its derived classes.</p>

.h file

This field is used to set the name of the header file for the new object's class. By default, this name is based on the name you provide in Class Name.

.cpp file

This field is used to set the name of the implementation file for the new object's class. This is where the definition of the class would be stored.

Inline Class

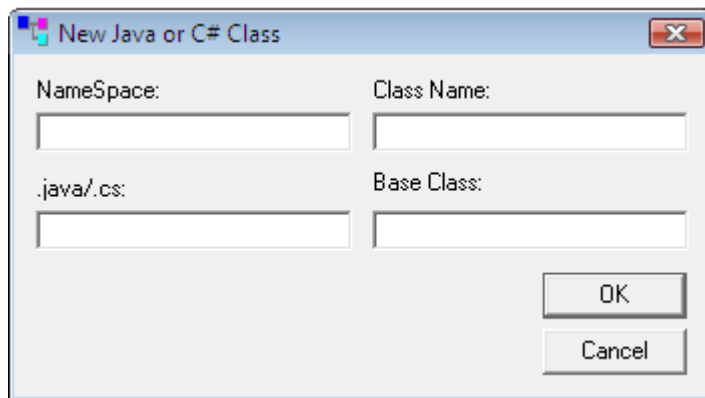
If this option is selected both the class constructor and the class definition will be generated as inline functions in the header file.

Virtual Destructor

This option specifies whether the class destructor is virtual. Using virtual destructors helps ensure that the correct destructor is called when instances of derived classes are deleted.

New Java or C# Class

If the "New Java or C# Class" option is selected the New Java or C# Class dialog is presented.



Namespace

This field is used to set the namespace keyword which is used to declare a scope. This namespace scope lets you organize code and gives you a way to create globally-unique types. If a namespace is declared the . java/.cs field will automatically be populated with a C# file name based on the specified class name.

Class Name

The class name specified in this field will be used to create the file in which the class is defined/prototyped.

. java/.cs

A new . java/.cs file is automatically generated for each new class as specified. If a namespace is declared this field will automatically be populated with a C# file name based on the specified class name. If a namespace is not declared this field will automatically be populated with a Java file name based on the specified class name.

Base Class

This field is used to specify the Base Class which specifies the class from which the new class derives.

Add To Project command (Project menu)



This command presents a submenu from which the user may select items to add to the active project. Please note: In UltraEdit this submenu only includes the New Folder and Files options described below.

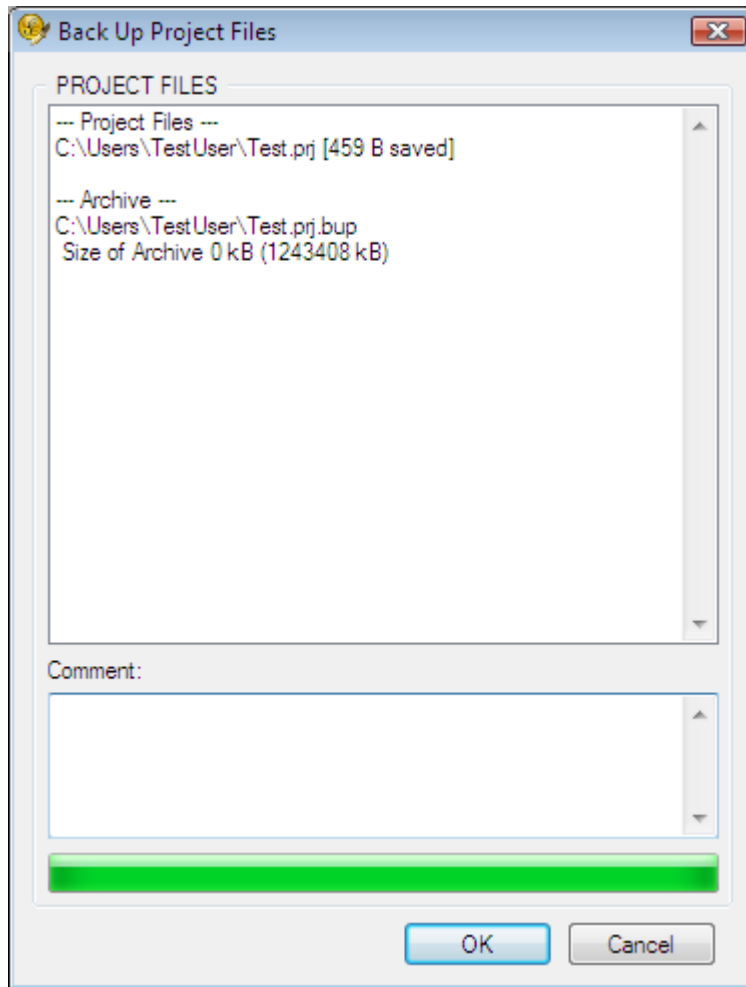
The Add To Project submenu contains the following items:

New Folder	presents New Folder dialog to allow the user to select a folder from currently mapped drives to add to a project or (if the Group item is selected) specify a new Group for placekeeping purposes in the active project
Files	presents File Open dialog to allow the user to select a file from currently mapped drives to add to the active project
Dynamic Link Library	presents dialog to allow the user to select a DLL to from currently mapped drives to add to the active project

Backup Project/Workspace command (Project menu)



This command copies all files in the current project/workspace to a compressed backup file (with a ". bup" extension) stored in the same directory as the . prj file for the active project/workspace and presents the Backup Project Files dialog to display information regarding the created project backup file.

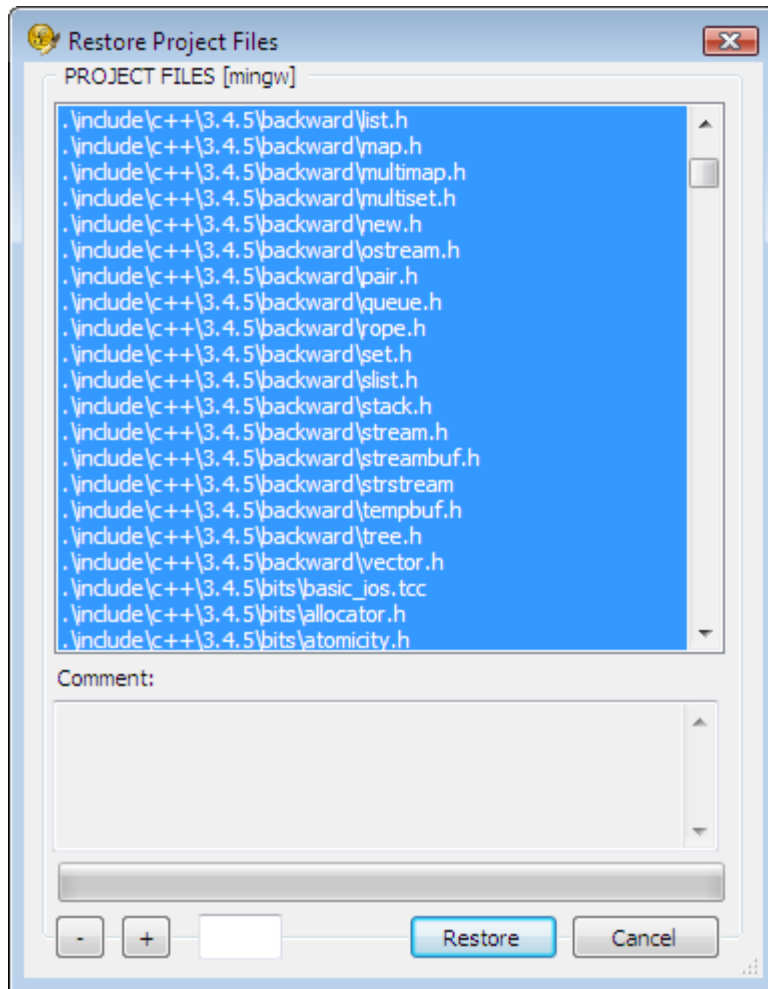


Having a backup copy of a project/workspace allows users to develop with the security of knowing that if something doesn't work as planned a working project/workspace can be restored using the [Restore Project/Workspace command](#).

Restore Project/Workspace command (Project menu)



This command presents the Restore Project Files dialog to reload a project from a copy that was previously backed up using the [Backup Project/Workspace command](#).



The Restore Project Files dialog lists all the files stored in project/workspace from the previous backup.

It is not necessary to restore all items from a compressed backup file if this is not desired. Pressing the "-" button will cause all items in the current listing to be deselected. Pressing the "+" button will cause all items listed in the dialog to be selected. Individual files may be selected/ deselected by clicking on them with the mouse.

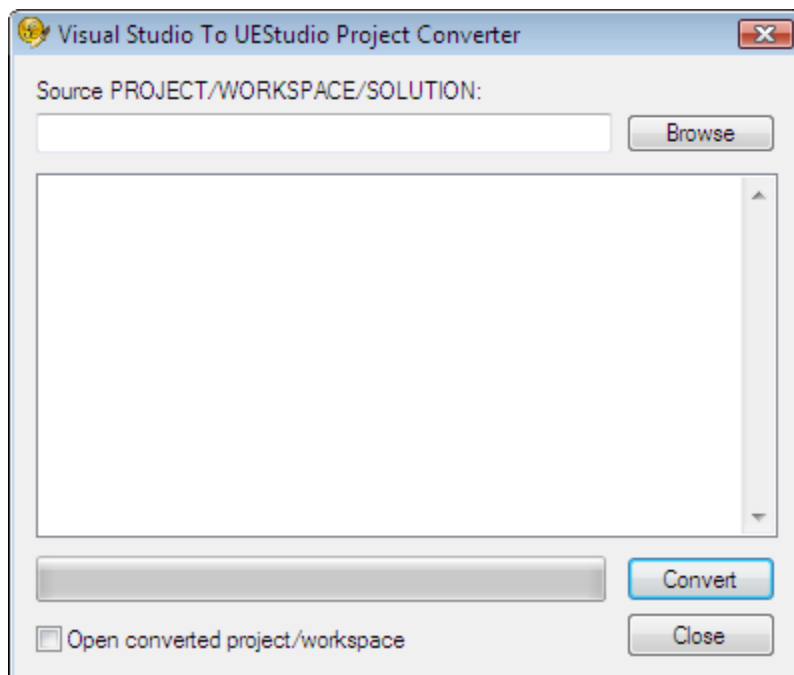
The text field to the right of the "+" button is a filter text field where file extensions may be entered. If an extension is specified (e.g. cgi) and the "+" button is pressed, all files with that extension will be selected. If an extension is specified (e.g. cgi) and the "-" button is pressed, all files with that extension will be deselected.

Once the desired items in the project are selected pressing the Restore button will reload them from the compressed backup file.

Project Converter command (Project menu)



This command may be used to convert an existing Visual Studio project to UEStudio project format so that the application may be built/modified within UEStudio.



Pressing the "..." button will allow the user to browse to select the Visual Studio Project, Workspace, or Solution that should be converted for UEStudio handling.

If the Open converted project/workspace option is selected the selected item will be opened in UEStudio after conversion.

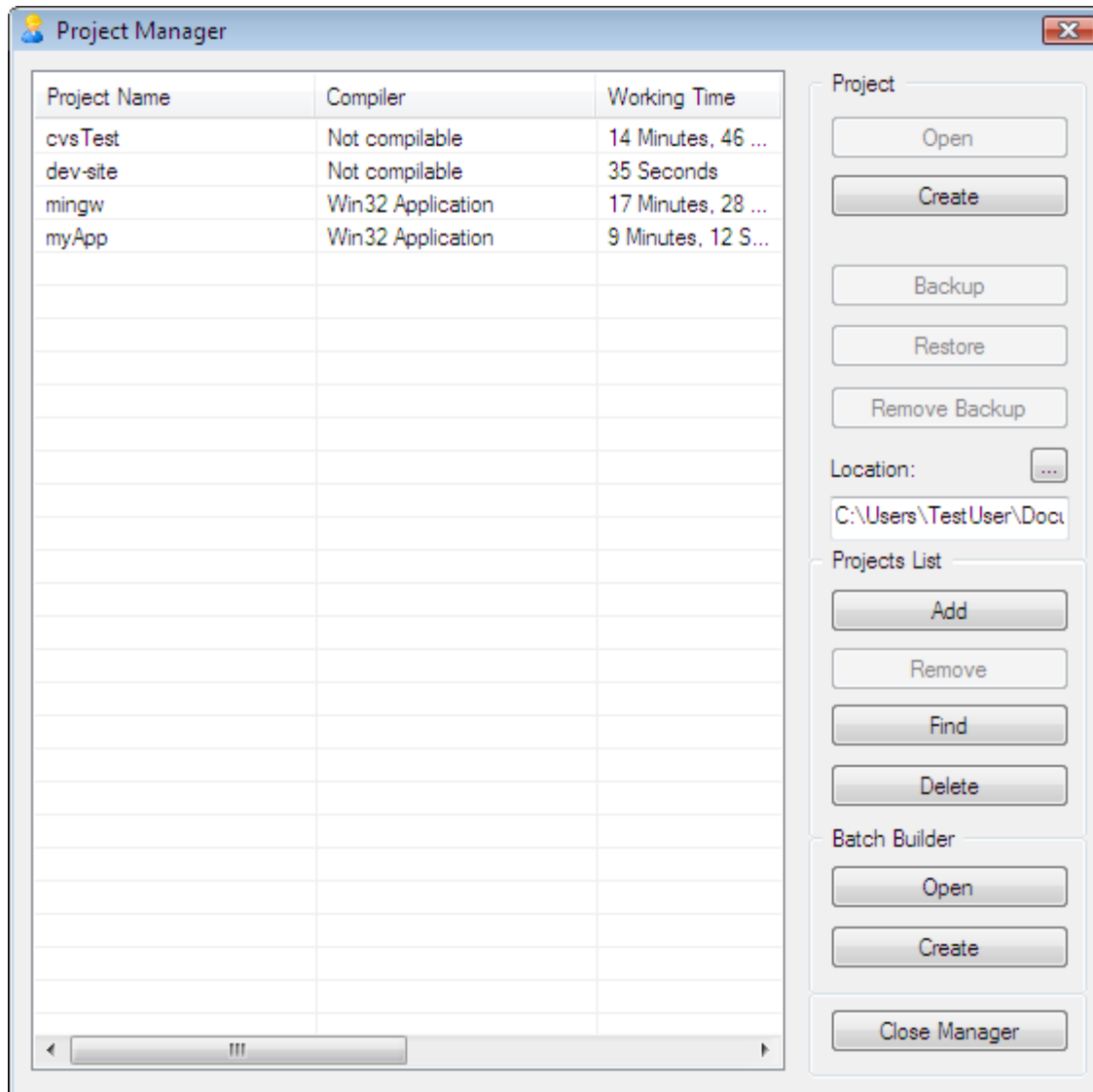
Pressing the Convert button will begin the conversion process and pressing the Close button dismisses the Project Converter dialog.

* Please note: This feature is only supported in UESstudio, and not in UltraEdit.

Project Manager command (Project menu)



Selecting this option in the Project menu launches the Project Manager dialog.



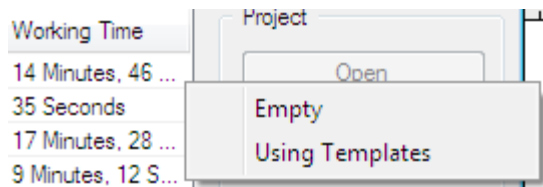
Project List

The main part of the Project Manager dialog is the Projects list [1]. This list includes the Project Name, Compiler, Working Time, and Project Path. The Project Manager is intended to be used to help manage all projects currently in progress. The functions related to Project management are divided into three categories: Project [2] - related to opening, creating, backing up, and restoring project files, Projects List [3] - related to adding projects to or removing them from the Projects List, and Batch Builder [4] - related to opening/creating lists for batch building.

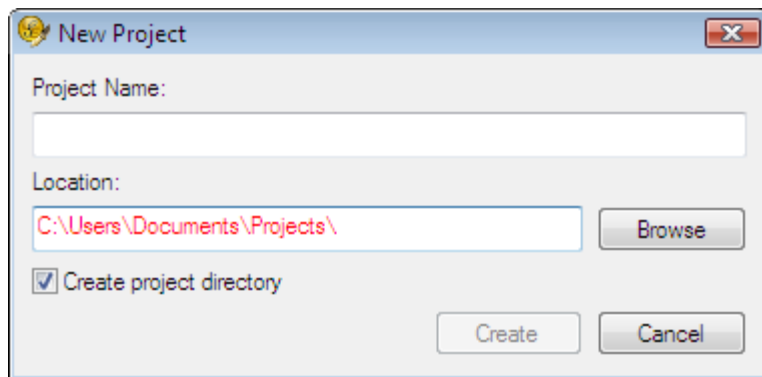
Project Functions

Projects may be opened from the Project Manager by double-clicking on a project name or by selecting a project name and pressing the Open button. This will open the selected project in UESTudio.

If the Create button is pressed, a context menu is presented from which the user may select Empty or Using Templates.

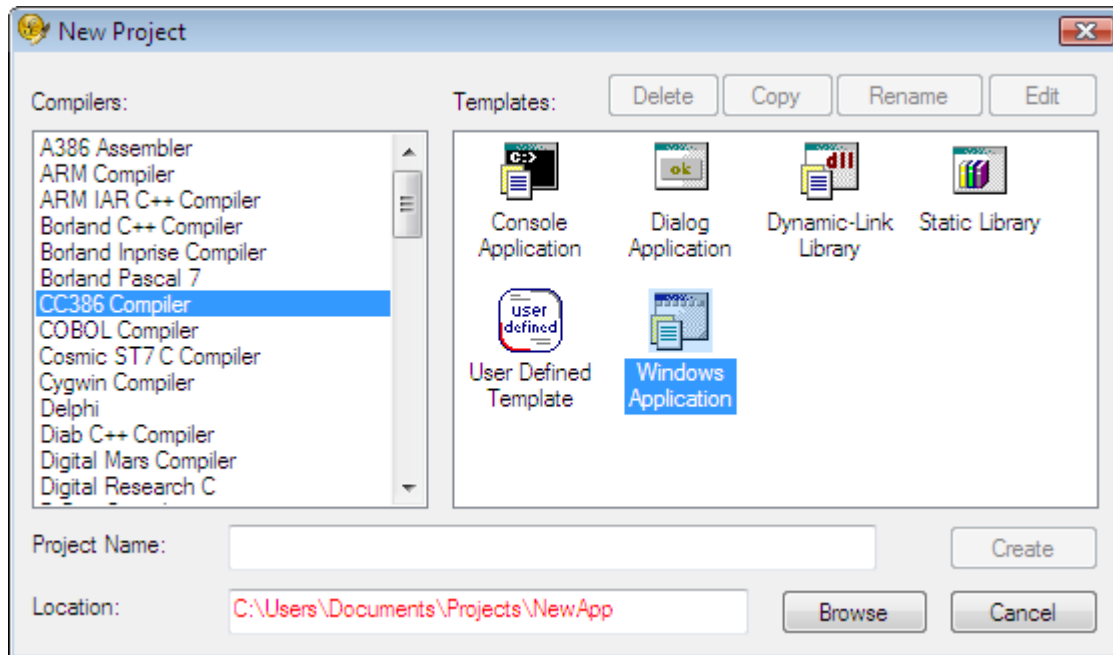


If Empty is selected a dialog for creating a new, empty dialog is presented.



Here the Project Name [1] and Location [2] may be specified. If Create project directory [3] is NOT selected then the specified project will be created in the specified directory. If Create project directory IS selected then a directory with the name of the project will be created on the specified path. For example, if the project name "App1" were specified the path to the project file would be " C:\ Development\Working Group1\App1\App1.prj".

If the Using Templates option is selected, a dialog allowing the specification of the new project and the template to be used to create it is presented.

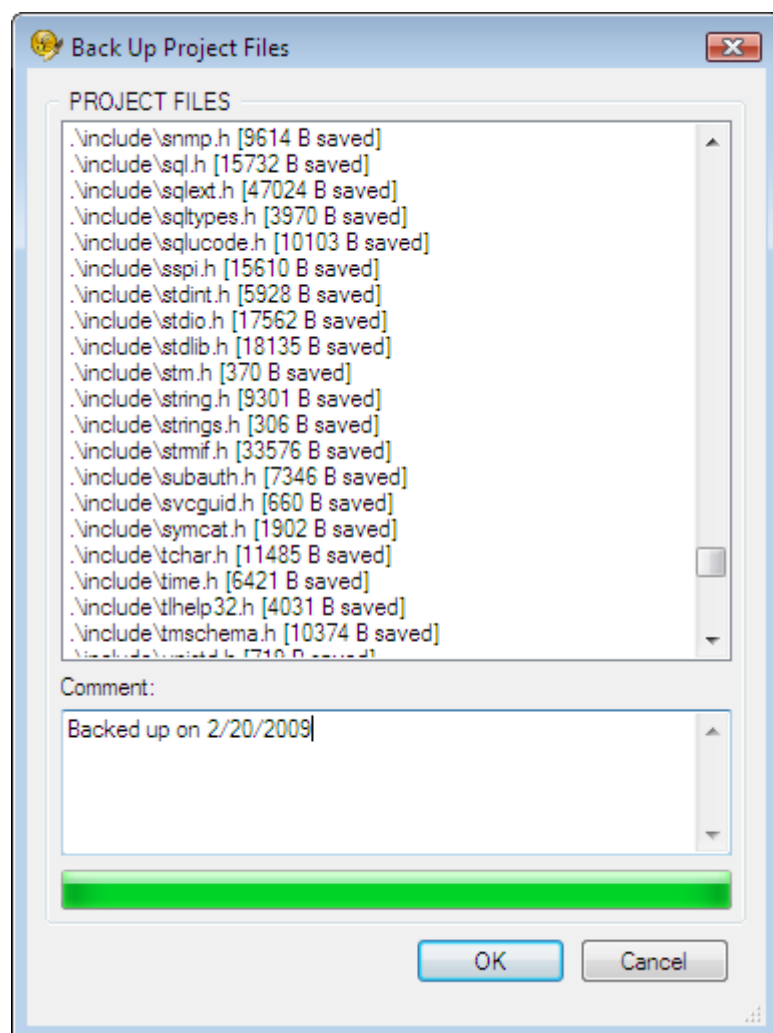


Project Templates

Depending on the type of compiler selected you may use one of the offered templates. Select the desired Compiler [1], next select the type of Template you want to use to create a project with [2] and type the name of this project in the Project Name field [3]. Don't forget to set the Location [4] of the project by typing it or press the Browse button and select it. It is recommended you set a location when you run project templates the first time and then use the same one always. This location is the "parent" directory of your projects, because UESTudio creates its own directory for a project with the same name as the project. Most of the templates automatically create a project and open in UESTudio but some templates will need some options set before a project will be created. One template is included for all compilers - User Defined Template.

Backup/Restore Functions

If the Backup button is pressed the files in the active project will immediately be backed up. If the Location text field is empty, backups are created with the same name as the active project. If a path is specified, the Project Manager will generate names with a date stamp and if Restore is pressed, the user may select the version of the backup desired to restore the selected project. Remove Backup may be pressed to select backups that should be removed. Backups for the active project will be listed in a context menu from which the user may select the backup to remove.



Projects List Functions

If projects have already been created, the Add button may be used to add a single project to the Projects List. The Find button may be used to add all projects found in the specified directory. To remove projects from the Projects List, the Remove button may be used. If a single project is selected when the Remove button is pressed, the Project Manager will open the project directory in Windows Explorer to allow the user to more easily remove project files from the disk if desired. If more than one project is selected, Project Manager will remove these projects only from the Projects List. Pressing the Delete button will delete the entire active Projects List.

Batch Builder Functions

Pressing the Create button in the Batch Builder section with a project selected in the Projects List will quickly create a Batch Builder list. Pressing the Open button will open the Batch Builder dialog.

Checkout command (Project menu/CVS submenu)



This command launches the CVS Checkout dialog. Users may create/modify a CVS account and checkout options through the CVS Checkout dialog.

The screenshot shows the 'CVS Checkout' dialog box. It has a title bar with a close button. The dialog contains several sections: 'Account:' with a dropdown menu showing 'site-dev*' and a 'Modify' button; 'Module:' with a dropdown menu showing 'assets' and an 'Update' button; 'Local Root Path:' with a text field showing 'C:\Users\Documents\Projects' and a 'Browse' button; 'Add Module' with three radio buttons: 'New Project', 'Current Project' (which is selected), and 'None'; 'Checkout Options' with two checked checkboxes: 'Use module name for checkout folder' and 'Checkout default HEAD branch'; and two empty text fields for 'Folder:' and 'Branch/Tag:'. At the bottom are 'Checkout' and 'Cancel' buttons.

The desired account may be selected from the Account dropdown. If a CVS account doesn't already exist, pressing the Modify button will launch the [CVS Manager dialog](#) where users may add/remove an account.

The Module dropdown lists the top level directories in the CVS repository. Pressing the Update button will update the list of modules in the CVS repository if this feature is supported by the CVS server. Modules will be created as subdirectories of the Local Root Path.

The Local Root Path field may be used to specify the path to the local directory where documents checked out through CVS should be written. The "..." button may be used to browse to the desired directory if this is preferred.

Checkout Options

If Use module name for checkout folder is selected, then the module name will be the name of the folder under which the files being checked out will be written in the local copy directory defined in the Local Root Path field. If this is not selected the Folder field will be active and the name to be used should be defined here.

If Checkout default HEAD branch is selected the most current revision of the specified module will be used as the beginning point for opening the desired files. If this is not selected the Branch/Tag field will be active and the branch desired for opening should be specified here.

Add Files To

These settings relate strictly to UESstudio projects and are not directly related to CVS. If New Project is selected then files checked out through this dialog will be added to a new Project. When the Checkout button is pressed the [Project Settings](#) dialog will be launched and the new project parameters may be defined. If Current Project is selected the specified files would be added to the active project.

Pressing the Checkout button will retrieve the files under the specified branch. When the specified files are retrieved, each directory should contain a CVS subdirectory in which the CVS information related to the files in the active directory will be stored. This will include a list of files, versions and tags in the active directory.

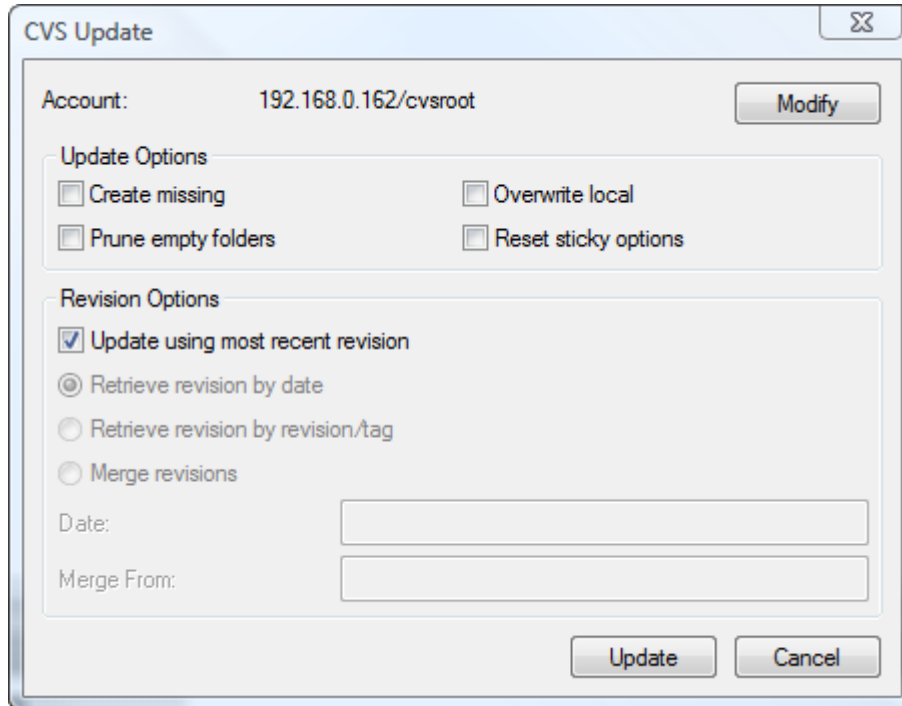
* Please note: This feature is only supported in UESstudio, and not in UltraEdit.

Update command (Project menu/CVS submenu)



There are two Update commands in the CVS submenu of the Project menu. The first one listed will only be active if a CVS Project is loaded. (A project is considered a CVS Project if the CVS Support option is selected and a valid CVS module is specified in the Project Settings dialog.) If a CVS Project is loaded and this option is selected the Update will be executed without any prompts or options as this will use the options specified for the CVS Project.

If the second Update option is selected in the CVS submenu the CVS Update dialog will be presented. This may be used to specify the parameters used to update the specified module or file in CVS.



The desired account may be selected from the Account dropdown. If a CVS account doesn't already exist, pressing the Modify button will launch the [CVS Manager dialog](#) where users may add/remove an account.

The Module/File field may be used to specify the module or file that will be updated from CVS. You may browse to select an existing module or file by pressing the "..." button.

The Update using most recent revision option is checked by default and specifies that the most recent revision will be retrieved from CVS. If this isn't selected then the Revision Options section below becomes active.

Revision Options

Retrieve revision by date retrieves the most recent revision no later than the date specified in the Merge To field.

Retrieve revision by revision/tag retrieves the revision/tag/branch specified in the Merge To field. This option will remain active for the specified file/module until overridden or reset using Reset sticky options.

Merge revisions enables merging a single revision into the current working directory or merging two separate revisions into the current working directory. If Merge from default ancestor is checked, the revision specified in the Merge To field is merged with the current working directory. If it is unchecked the revision specified in the Merge From field is merged with the revision specified in the Merge To field.

Update Options

Overwrite local overwrites any locally modified files with the most current CVS revision and creates a backup of the modified file prefixed with .#.

Reset sticky options removes any tag, revision, or date information from the CVS information maintained in the working copy.

Prune empty folders removes directories in the working copy that are empty in the CVS repository.

Create missing retrieves any files or directories that are missing from the working copy.

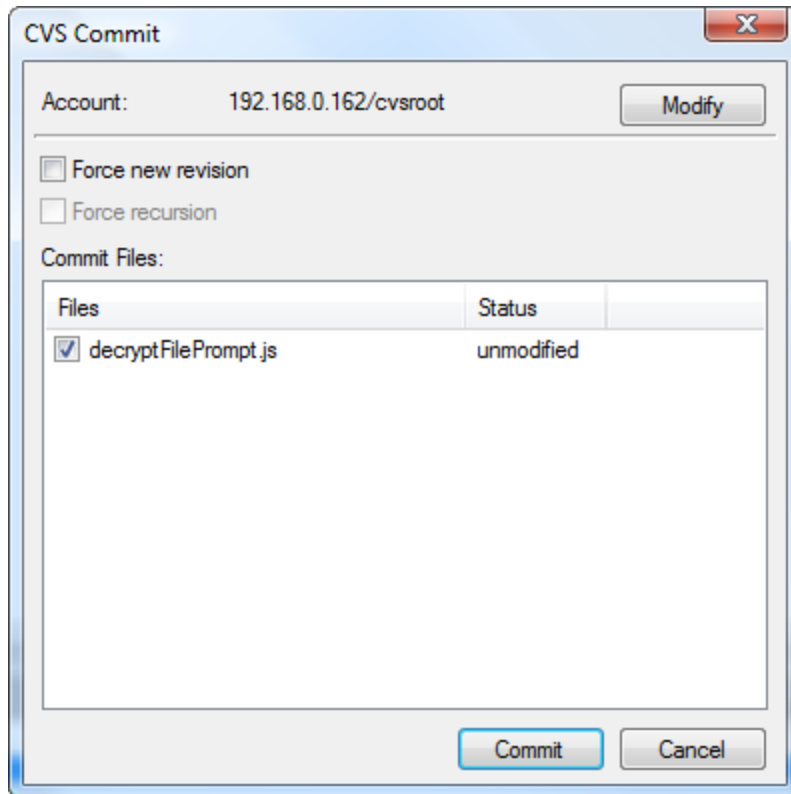
* Please note: This feature is only supported in UESstudio, and not in UltraEdit.

Commit command (Project menu/CVS submenu)



There are two Commit commands in the CVS submenu of the Project menu. The first one listed will only be active if a CVS Project is loaded. (A project is considered a CVS Project if the CVS Support option is selected and a valid CVS module is specified in the Project Settings dialog.) If a CVS Project is loaded and this option is selected the Commit will be executed without any prompts or options as this will use the options specified for the CVS Project.

If the second Commit option is selected in the CVS submenu the CVS Commit dialog will be presented. This may be used to specify the parameters used to commit a module or file to CVS.



The desired account may be selected from the Account dropdown. If a CVS account doesn't already exist, pressing the Modify button will launch the [CVS Manager dialog](#) where users may add/remove an account.

The Module/File field may be used to specify the module or file that will be committed to CVS. You may browse to select an existing module or file by pressing the "..." button.

By default the Commit to HEAD option is selected which will commit the specified module/file to the most current branch of the CVS repository.

The Force new revision option increments the revision of the specified module or file even if no changes are detected.

The Force recursion option only becomes active if Force new revision is enabled and checking this will recursively increment the revision level of all modules and files below the specified module.

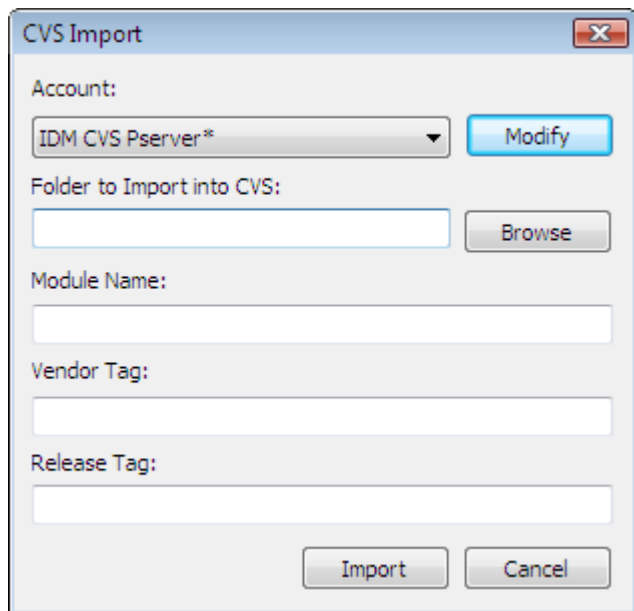
For Subversion accounts, the only option available is the **Non-recursive** option. If this option is selected, the Commit command will not recurse into directories (i.e. if a directory is selected to be committed, none of its subdirectories would be committed).

* Please note: This feature is only supported in UESstudio, and not in UltraEdit.

Import command (Project menu/CVS submenu)



The Import command in the CVS submenu presents the CVS Import dialog. This may be used to specify the parameters to be used to import items to CVS.



The desired account may be selected from the Account dropdown. If a CVS account doesn't already exist, pressing the Modify button will launch the [CVS Manager dialog](#) where users may add/remove an account.

All files and directories below the path specified in the Folder to Import into CVS field will be added to the CVS repository under the name specified in the Module Name field.

The Vendor Tag is the initial branch tag used to identify the module upon import.

The Release Tag allows for a unique identification of the files from this import into the CVS repository and for any subsequent imports.

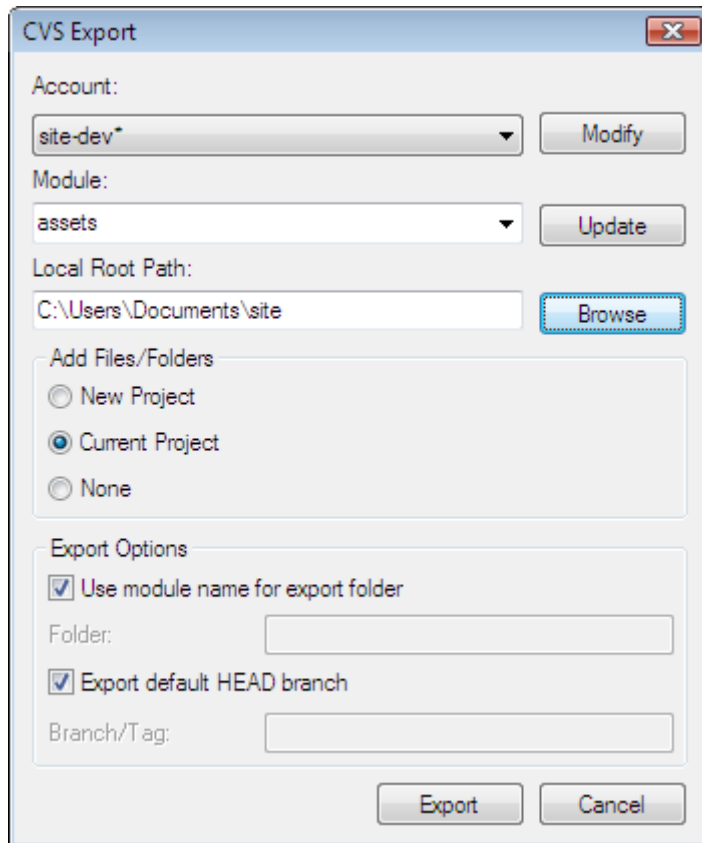
The only difference between the CVS Import dialog and the Subversion Import dialog is that the Subversion dialog does not include the Vendor Tag and Release Tag items.

* Please note: This feature is only supported in UESTudio, and not in UltraEdit.

Export command (Project menu/CVS submenu)



This command launches the CVS Export dialog. Users may create/modify a CVS account and export options through the CVS Export dialog.



The desired account may be selected from the Account dropdown. If a CVS account doesn't already exist, pressing the Modify button will launch the [CVS Manager dialog](#) where users may add/remove an account.

The Module dropdown lists the top level directories in the CVS repository. Pressing the Update button will update the list of modules in the CVS repository.

The Local Root Path field may be used to specify the path to the local directory where documents checked out through CVS should be written. The "..." button may be used to browse to the desired directory if this is preferred.

Export Options

If Use module name for export folder is selected, then the module name will be the name of the folder under which the files being checked out will be written in the local copy directory defined in the Local Root Path field. If this is not selected the Folder field will be active and the name to be used should be defined here.

If Export default HEAD branch is selected, the top level folder of the specified module will be used as the beginning point for opening the desired files. If this is not selected the Branch field will be active and the branch desired for opening should be specified here.

Add Files To

These settings relate strictly to UESTudio projects and are not directly related to CVS. If New Project is selected then files checked out through this dialog will be opened under a new Project. When the Export button is pressed the [Project Settings](#) dialog will be launched and the new project parameters may be defined. If Current Project is selected the specified files would be opened under the active project.

Pressing the Export button will open the files under the specified branch as specified.

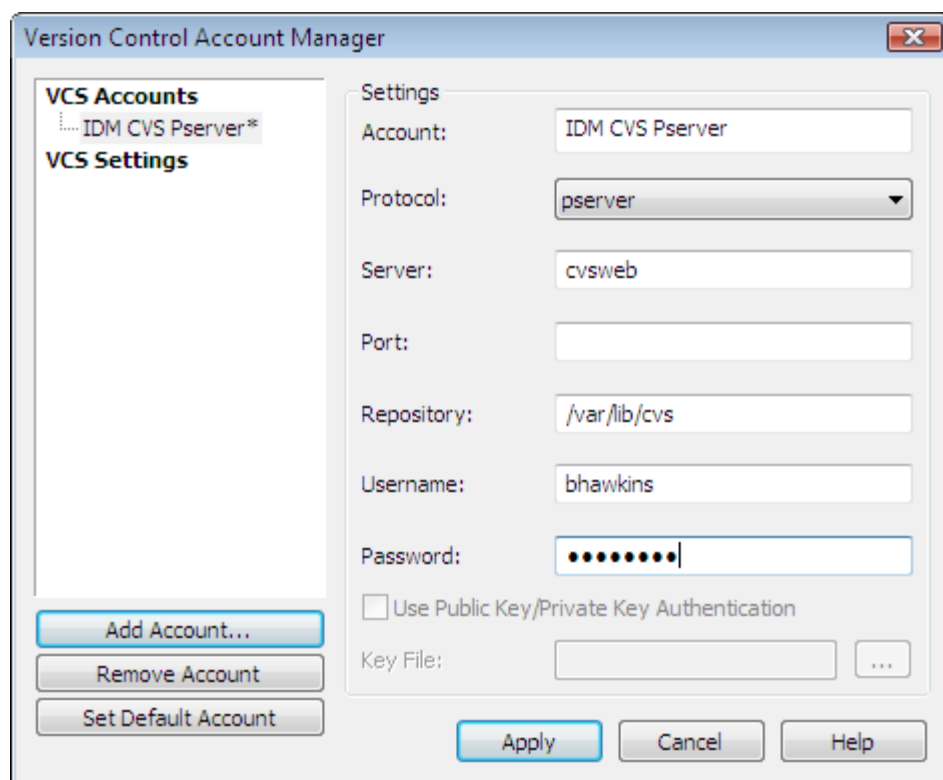
Please note: The only difference between [Checkout](#) and Export is that an export opens the specified files without any CVS information. If files are exported the CVS subdirectory with the file names, versions, and tags will not be created in the exported directories. Exported files cannot be updated, committed, or otherwise returned to the CVS repository.

* Please note: This feature is only supported in UESTudio, and not in UltraEdit.

Accounts/Settings command (Project menu/CVS submenu)



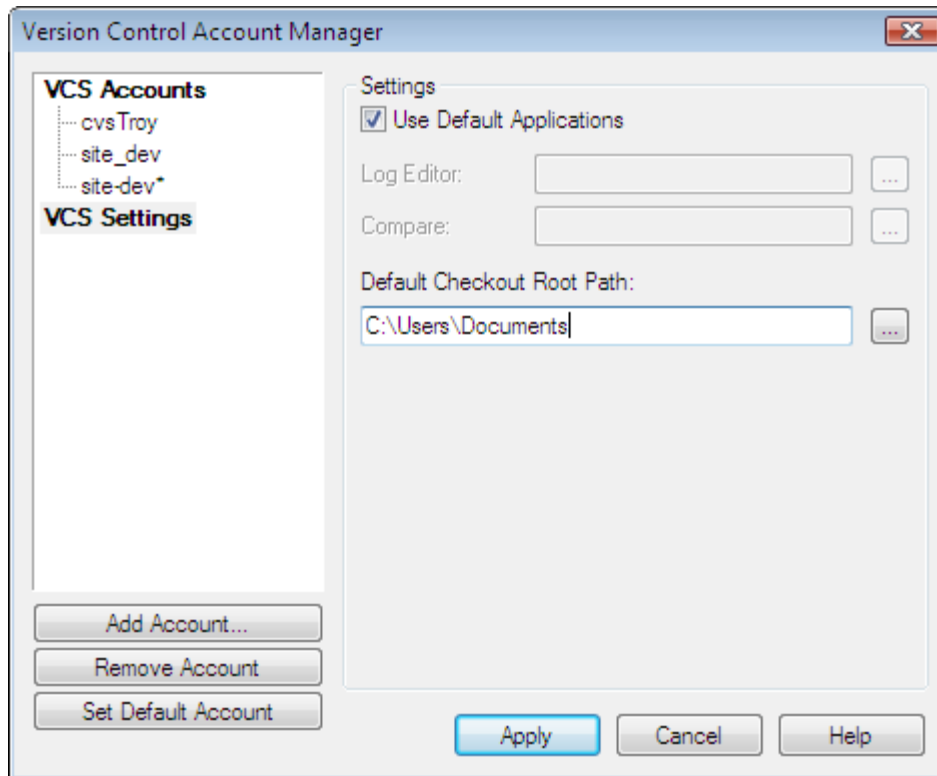
The Version Control Account Manager dialog may be used to create accounts for checking out files from the CVS/Subversion Server and to specify the settings to be used with these accounts.



If the Add Account button is pressed a new account will be created with settings as follows:

Setting Items	Usage
Account	name to be used for account
Protocol	may be either pserver or ext. pserver is the standard CVS communications mechanism using username/password login and is a non-secure, plain text method with no encryption. ext launches an external SSH (secure shell) application where username and authorization code are specified. This is encrypted and will require the password to be entered with each invocation.
Server	IP address or fully qualified domain name
Port	normally blank
Repository	path to the CVS repository on the CVS server
Username	username for login to account, required for both pserver and ext login
Password	password - required for each usage of ext account (stored for pserver accounts)

If an account is selected in the CVS Accounts list and the Set Default Account button is pressed, the specified account will be set as the default CVS account.



If VCS Settings is selected the VCS Settings pane is displayed. By default the Use Default Applications option will be selected. If this is not selected users must specify/browse to the desired executable for the Log Editor and Compare application.

The Default Checkout Root Path is the default path for CVS Checkout or Export. This can be overridden in the individual CVS command dialogs.

* Please note: This feature is only supported in UESTudio, and not in **UltraEdit**.

Ctags support



UltraEdit supports Ctags, either as part of a project, or independently.

What is Ctags?

Ctags generates a tag file (or index) file of C/C++ objects and functions found in C/C++ header and source files that allows these items to be referenced quickly within UltraEdit.

Ctags EXE

UltraEdit relies on an external tag file generator. There are many versions of Ctags and we recommend "Exuberant Ctags" available from [http:// ctags.sourceforge.net/](http://ctags.sourceforge.net/).

UltraEdit can make use of an existing tagfile, or if an EXE for creating Ctags is available it can also create the tag file. Using ctags.exe from Exuberant Ctags and placing this exe in the directory containing UltraEdit will by default allow UltraEdit to create the tag file.

Ctags Configuration within UltraEdit

The Project menu allows the user to change the Ctags option and to create the tag file. The Ctags option dialog allows for the following three items to be defined:

Default Tagfile – This specifies the tagfile to be used if there is not a specific project tagfile active (defined in the project settings). This must be present if there is not a project specific tagfile for UltraEdit to lookup symbols.

CTAG EXE Path/Name – This specifies the full path and name of the Ctag exe that will be run to create the tag file from within UltraEdit. If this is not present UltraEdit can still reference a tagfile for symbols but it can not create one. The default setting for this is "ctags.exe" in the directory containing UltraEdit.

CTAG Parameters – This specifies the parameters to be passed to the Ctag exe for it to function correctly. The default settings are set up for "Exuberant Ctags" as described above.

% fi on the parameter line is replaced when the ctag exe is run with the filename/path of a file that contains a list of files that the ctag.exe will use to create its tags from. If a project is loaded in UltraEdit, UltraEdit creates a temporary file with a list of the project files and passes this in place of % fi.

% fo on the parameter line is replaced with the full filename/path of the tag file to be created. This would be the tagfile specified in the project, or the Default Tagfile defined above.

Creating the Tag File

The tag file can be created directly from the [Project Menu](#) with the Create CTAG file item, or when a project is opened the tag file can be created automatically if the project settings specify this.

Finding a Symbol/Tag

The whole purpose of Ctags is to be able to quickly find a function or object that is defined elsewhere in the file or another file. Assuming the tag file exists, selecting Find Symbol from the [Search menu](#) will search the tag file for the symbol and if only one match is found the file will be opened in UltraEdit and the file positioned to the symbol. If multiple matches occur, a dialog box will be shown giving the user the option to position to the appropriate tag.

Build Menu

Build menu



Please note: This menu and the options described below are only available in UESTudio and the following information doesn't apply to UltraEdit.

The Build menu contains commands which may be used to debug, compile, or run the files in the active project or workspace. Before most options in the Build menu will be available for use, you must set the Project as compilable under the [Project Files/Settings dialog](#) in the Project menu and select a compiler to be used for your project.

The Build menu includes the following commands:

Compile	runs the selected compiler on the active file
Build	builds all files in active workspace based on the active target and dependencies
Rebuild All	rebuilds all files for the active target without reference to dependencies

Batch Build	builds list of files (EXEs, DLLs) based on target and dependencies
Clean	removes intermediate files and target executables
Start Debugger	starts selected debugger on active file
Debug - Debug Windows	presents submenu for Watch, Call Stack, Memory, Disassembly, Registers, and Variables windows
Debug - Go	automatically starts the debugger and runs the application
Debug - Restart	ends a debugging session and then restarts
Debug - Stop Debugging	ends a debugging session
Debug - Break	pauses target execution and returns control to debugger
Debug - Step Into	executes code one statement at a time, following execution into function calls
Debug - Step Over	executes the next line of code but does not follow execution through any function calls
Debug - Step Out	executes remaining lines of a function in which the current execution point lies
Debug - Run to Cursor	resumes execution of code from the current statement to the selected statement
Debug - Set Breakpoint	adds breakpoint at current cursor location
Debug - Remove Breakpoint	removes breakpoint at current cursor location
Run Application	runs selected target executable
Run Commands	presents submenu to run or debug selected target or to show makefile
Select Compiler	presents Select Compiler dialog
Set Compiler Options	presents Compiler Options dialog
Advanced Build Options	presents Advanced Build Options dialog
Set Compiler Paths	presents Compiler Paths dialog
Regenerate Makefile	recreates control file for compile process based on current configuration options
Display Tag List	opens Tag List file for active project

* Please note: These features are only supported in UESudio, and not in **UltraEdit**.

Compile command (Build menu)



This command saves the active document if needed and then compiles it. If a compiler is not selected or the compiler options are not set for the project the [Select Compiler dialog](#) is displayed, then the Project Settings dialog is displayed and then the Compile process is started.

Build command (Build menu)



This command saves the active document if needed and builds the project target(s). If a compiler is not selected or the compiler options are not set for the project the [Select Compiler dialog](#) is displayed, then the Project Settings dialog is displayed and then the Build process is started.

Rebuild All command (Build menu)



This command removes all files created by the compiler and then builds the project.

Batch Build command (Build menu)



This command builds a list of executables (EXEs, DLLs) based on target(s) and dependencies.

Clean command (Build menu)

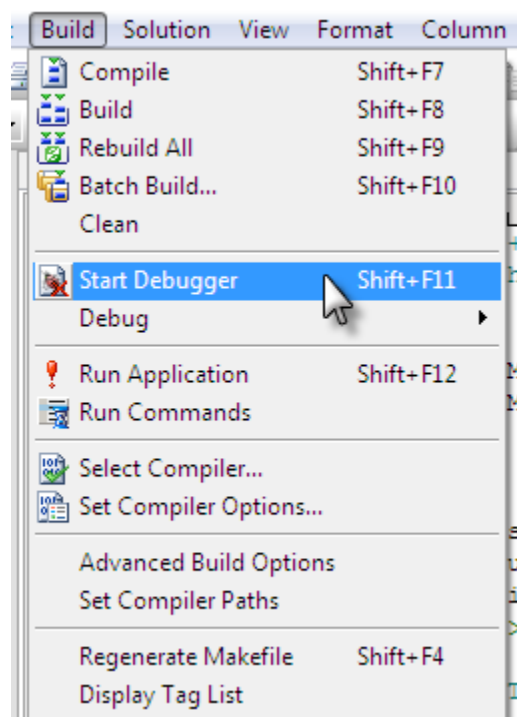


This command removes intermediate files and target executables for the active project.

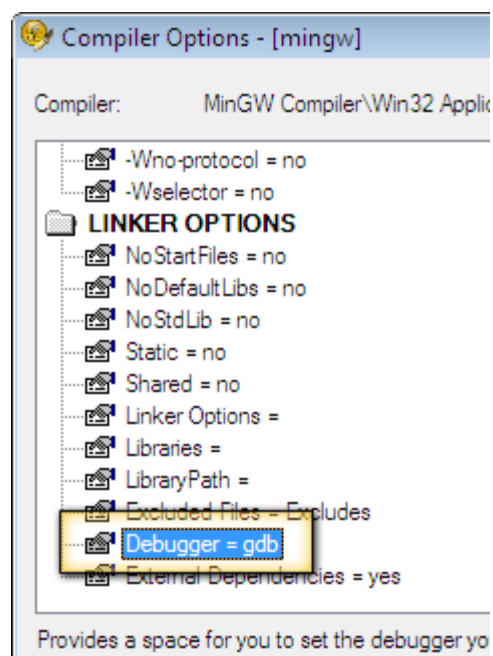
Start Debugger command (Build menu)



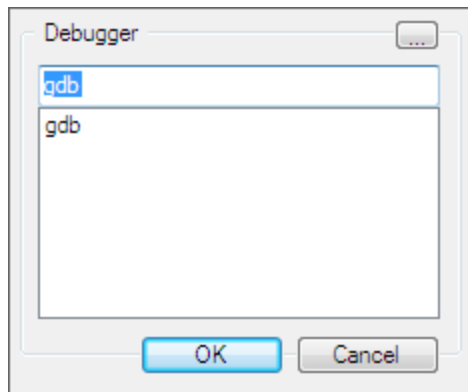
This command starts selected debugger on the active file. If debugger is not set, a dialog will be displayed where the debugger for the active project may be selected.



In most cases, where the use of a debugger is appropriate, this will be specified as most compilers come packaged with an associated debugger and the Debugger key in the Compiler Options (under Linker Options) will show this (for example for Visual Studio, the debugger is set to "msdev").



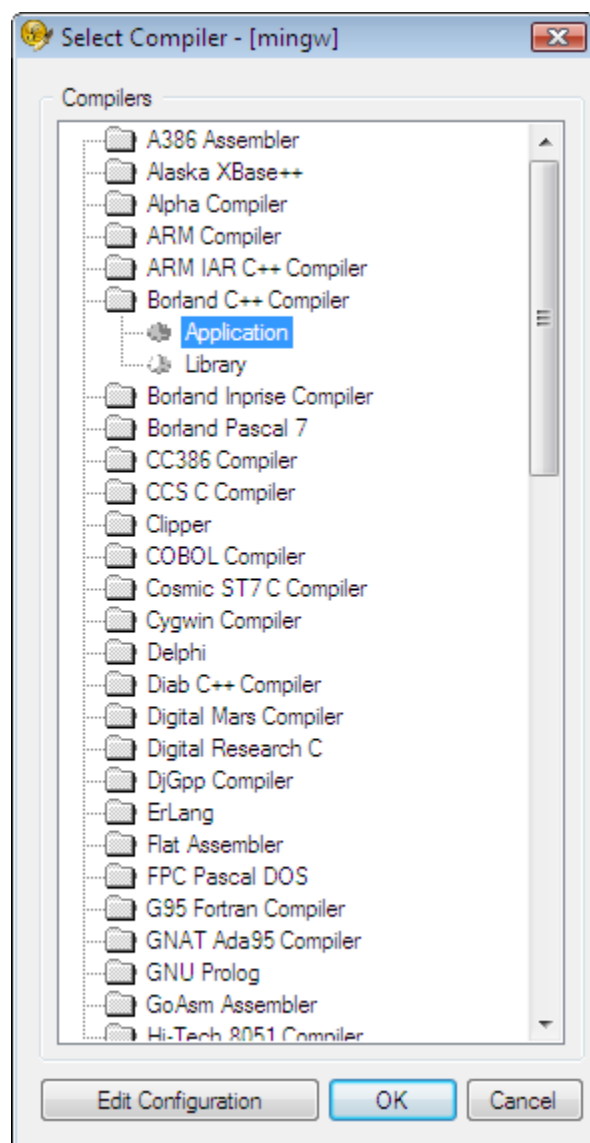
Double clicking on the Debugger key presents a dialog where the user may specify a different debugger to be used if desired.



The user may click on the Debug button in the Tools toolbar to invoke the debugger.



UEStudio will check to see if a debugger is set. If this isn't set in the compiler options, UESTudio will display a browse dialog so a debugger may be selected. If a debugger is set, UESTudio will prompt if this debugger should be set as the default for future projects. If "Yes" is selected, the selected debugger will be saved in the compiler configuration file so that when a new project is selected this will be remembered and is not limited to the active project.



If the Edit Configuration button is pressed in the Select Compiler dialog the compiler configuration file will be displayed here and advanced users who are very familiar with their compiler/debugger functionality may edit this file directly if desired.

```
Microsoft Visual C++ Compiler\Win32 Application

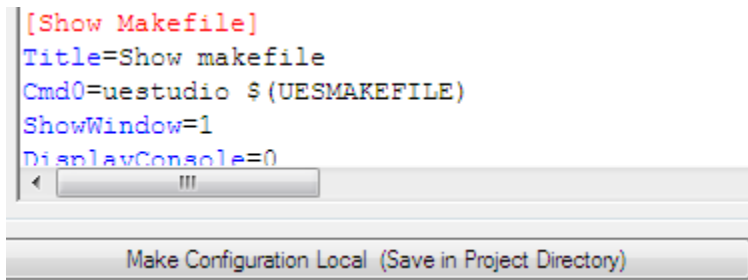
[Start Debugger]

Cmd0 = $(Debugger) $T
Depends = $T
ShowWindow = 1
Title = Debug $T
```

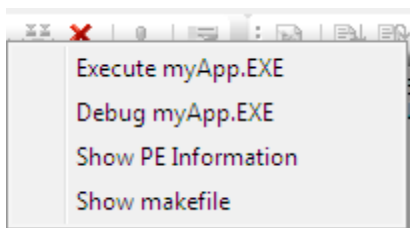
For example, users may set dependencies, and decide if they want to run the debugger as a console debugger (gdb) or windowed debugger (WinDbg).

In the configuration file, users may select the Debug key and modify the [Start Debugger] section. For example, if the default is set to a windowed debugger and you wish to use a console debugger you would have to change/add the key "DisplayConsole=1" in that section.

If the Make Configuration Local button is pressed in the Select Compiler dialog the compiler configuration file will be saved into the project directory and changes made after pressing this will be saved locally with the project.



Once a debugger is selected, the user may invoke the debugger from the Commands submenu in the Tools toolbar:



Run Application command (Build menu)



This command runs the application created with the selected compiler or invokes the build process first if the application does not yet exist.

Run Commands command (Build menu)



This command presents a popup menu with a variable number of items depending on the compiler configuration that is used and which commands are set for this [compiler's configuration](#).

Select Compiler command (Build menu)



This command presents the [Select Compiler dialog](#) which may be used to select the compiler to be used with the active project.

Set Compiler Options command (Build menu)

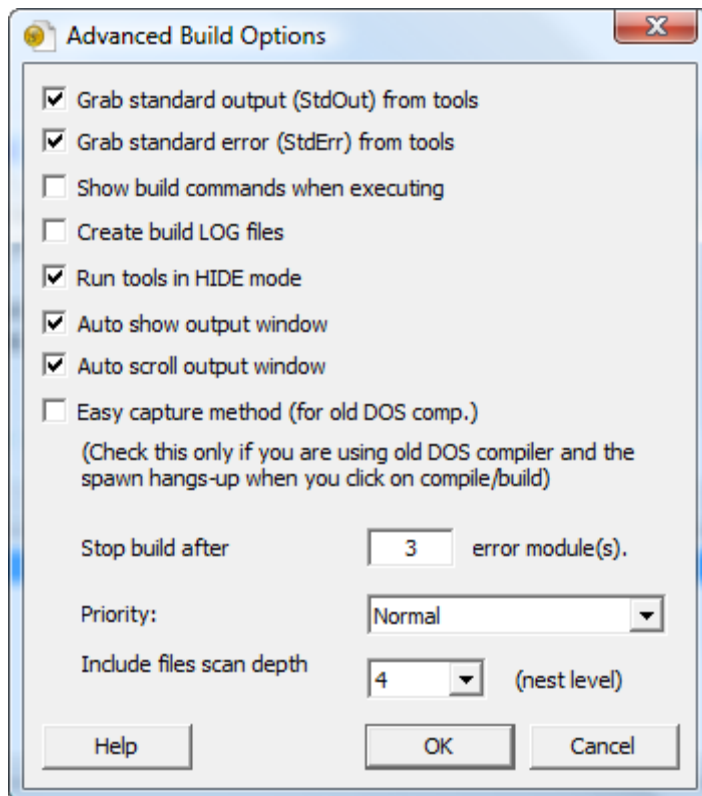


This command presents the Compiler Options dialog which may be used to configure options to be used with the compiler selected to be used with the active project.

Advanced Build Options command (Build menu)



This command presents the Advanced Build Options dialog. These options allow the user to have greater control over the Build process.



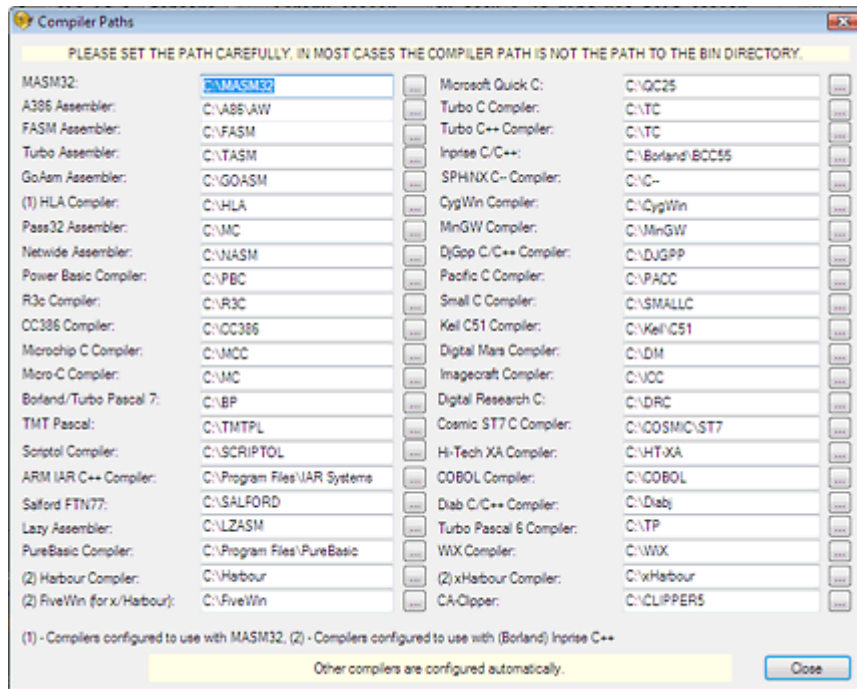
The Advanced Build Options dialog has the following options:

Option	Usage
Grab standard output from tools	Most compilers write only information here, but it is recommended to leave this checked. If unchecked, none of the messages assigned to STDOUT from compilers and build tools will be displayed.
Grab standard error from tools	Same as above, but for STDERR. Compilers write WARNINGS and ERRORS here.
Show build commands when executing	By default this option is not checked. If you want to show which commands are used when you click BUILD, REBUILD, etc., this item should be selected.
Create build LOG files	If this option is selected, HTML files with commands will be created when COMPILE, BUILD, REBUILD, and other commands are executed.
Run tools in HIDE mode	By default all compilers and other build tools are executed in HIDE mode and their output is grabbed. If this option is not selected this would not be the case.
Auto show output window	If you wish to see the output from STDOUT or STDERR when build commands are executed, this option should be checked.
Auto scroll output window	By default when UEStudio writes to the output window, it always sets the active line position to the line that is written, but in some configurations this may slowdown the build process.
Easy capture method (for old DOS compilers)	This option enables an alternate method to capture output from the compiler. Please check this ONLY if your compiler or tool "hangs up" when UEStudio runs it.
Stop build after # error module(s)	When the compiler writes error/warning messages to the output window, UEStudio monitors these messages. If the number of errors exceeds the specified threshold, the build process will be stopped after completion of the current compilation where the threshold is exceeded.
Priority	When building a large project, but the build process slowing down other processes is not acceptable, the Priority should be set to IDLE. Higher priorities may be selected as desired.
Include files scan depth (nest level)	When source files are compiled, UEStudio scans for included files to check dependencies. This value specifies how many levels the search should recurse when searching for these. Please note that a higher level will require a longer time for the scan.

Set Compiler Paths command (Build menu)



This command presents the Set Compiler Paths dialog. This may be used to specify the path for many different compilers.



The path to the desired compiler may be directly entered in the text field to the right of the compiler name or users may press the "... " button to browse to the desired compiler executable.

Solution Menu

Solution menu



Please note: This menu and the options described below are only available in UESudio and the following information doesn't apply to UltraEdit.

What is a Solution?

UESudio supports two different kinds of containers (solutions and projects) to aid in management of development resources. Solutions and projects contain items that represent the items needed to create your application.

Solutions are designed to manage the way sets of related projects are configured, built, and deployed. Depending upon your requirements, a solution in UESudio can include just one project, or several projects built jointly by a development team. A complex application might even require multiple solutions.

A solution container can contain multiple projects and a project container typically contains multiple items. Using these containers, you can take advantage of the integrated development environment.

Solution Menu

The Solution menu includes the following commands:

* New Solution	presents dialog for creation of a new solution (.prw)
* Open Solution	presents dialog for selection of an existing solution to be opened
* Close Solution	closes the currently active solution
* Add Project	presents a dialog to allow users to browse to a project and add it to the active solution
* Remove Project	removes the current project from the active solution

* Please note: These features are only supported in UESTudio, and not in UltraEdit.

New Solution command (Solution menu)



This command opens a Save As dialog where the user may define a new Solution file (. prw) in which multiple project files (. prj) may be associated.

* Please note: This feature is only supported in UESTudio, and not in UltraEdit.

Open Solution command (Solution menu)



This command presents an Open dialog where the user may select a Solution file (. prw) to be opened.

* Please note: This feature is only supported in UESTudio, and not in UltraEdit.

Close Solution command (Solution menu)



This command closes the active solution.

* Please note: This feature is only supported in UESstudio, and not in UltraEdit.

Add Project command (Solution menu)



This command presents an Open dialog where the user may select a project file (. prj) to be opened in the active solution.

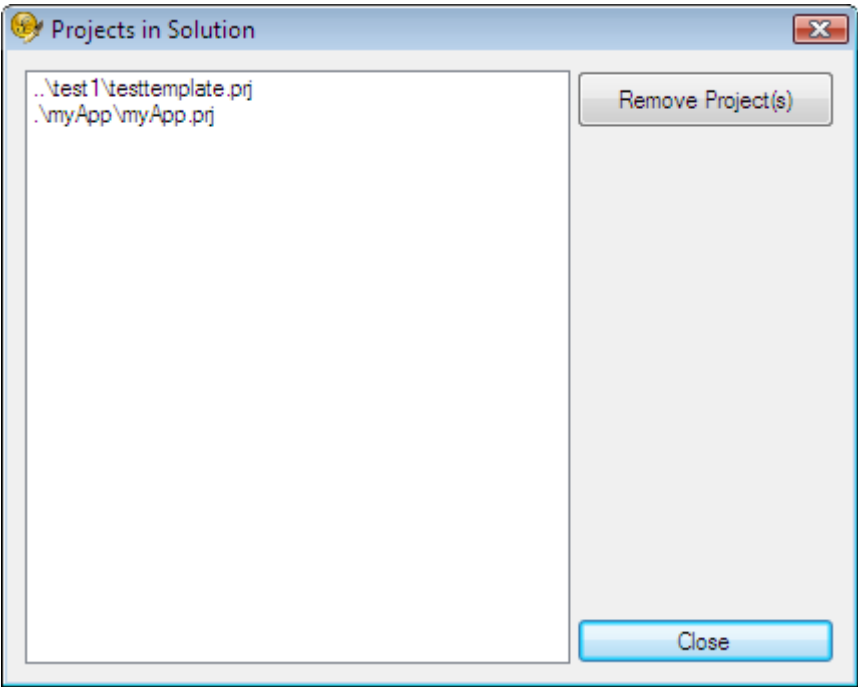
If a project is opened in a solution, this project is set in the solution (INI based) file as active, so that the next time the solution is opened, this project is opened also.

* Please note: This feature is only supported in UESstudio, and not in **UltraEdit**.

Remove Project command (Solution menu)



This command presents the Projects in Solution dialog.



Selected projects (. prj) are removed from the active Solution (. prw) when the Remove Projects button is pressed in this dialog.

* Please note: This feature is only supported in UESTudio, and not in UltraEdit.

View Menu

View menu



The View menu includes the following commands:

Environments	presents Environment Selector dialog where user may specify active environment (menus/toolbars/dockable windows)
Views/Lists - Toolbars	shows or hides the toolbar
Views/Lists - Status Bar	shows or hides the status bar
Views/Lists - Open File Tabs	shows or hides the open file tabs
Views/Lists - File Tree View	shows or hides the file tree view
Views/Lists - ASCII Table	shows or hides the dockable ASCII table

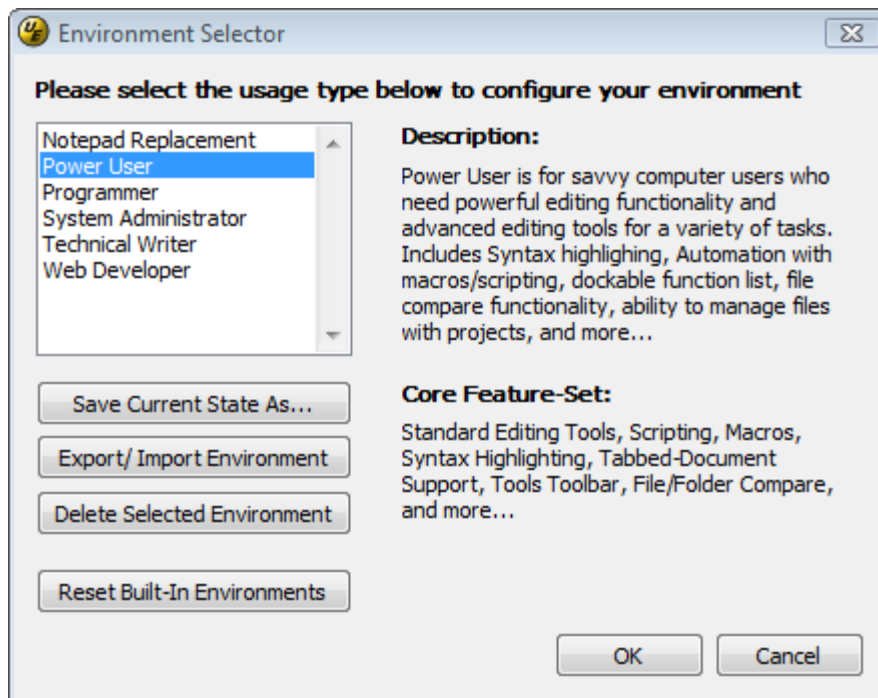
Views/Lists - Function List	shows or hides the function list
Views/Lists - Macro List	shows or hides listing of all currently loaded macros
Views/Lists - Tag List	shows or hides the tag list
Views/Lists - Template List	shows or hides the template list
Views/Lists - Clipboard History	shows or hides the clipboard history list
Views/Lists - Script List	shows or hides listing of all currently specified scripts
Views/Lists - XML Manager	shows or hides the XML Manager used for navigating nodes in XML files
Views/Lists - Bookmark Viewer	shows or hides the Bookmark Viewer
Views/Lists - Output Window	shows or hides the output window
Views/Lists - SSH/Telnet Console	shows or hides the SSH/Telnet console. Settings for SSH/Telnet sessions may be specified using the SSH/Telnet Account Manager
View As (Highlighting File Type)	displays file with selected syntax highlighting
Set Code Page	presents Code Page Selection dialog to allow changing code page for active document
Set Colors	sets the screen colors for edit window items
Set Font	selects the screen font to use
Set HEX/Column Mode Font	selects the screen font to use for HEX and Column mode
Reset Fonts (default)	resets the screen and printer fonts to default settings
Increase Font Size	increases font size (zoom in)
Decrease Font Size	decreases font size (zoom out)
Show Spaces/Tabs	displays spaces and tabs as visible characters
Show Line Endings	displays new line characters as visible characters
Show Page Breaks as Lines	displays page breaks as lines
Show Column Markers	toggles visibility of active column markers
Set Column Markers	presents dialog where column marker settings are defined
No Scroll Left Side of Marker 1	toggles fixed pane/columns left of first column marker
Line Change Indicator	toggles visibility of markers indicating modified lines in active document
Display Ruler	toggles the column ruler indicator at the top of each window
Display Line Numbers	toggles the line number display at the left of each window
Spacing - Single	displays documents with single spacing
Spacing - 1 1/2 Spacing	displays documents with 1 1/2 spacing
Spacing - Double Spacing	displays document with double spacing
Hide/Show Lines - Hide/Show Selection	hides or shows the selected lines
Hide/Show Lines - Collapse All	collapses any expanded function or structure in active file
Hide/Show Lines - Expand	expands any collapsed section in active file

All	
File Change Polling	checks active file for changes at interval (seconds) defined in configuration
Auto Scroll Polled Files	scrolls to last line of polled file when changes are detected (similar to UNIX tail command)
Always on Top	toggles setting to display UltraEdit on top of all other applications
Full Screen Mode	toggles full screen edit mode
Toggle Browser View	toggles view of active file from HTML content to browser view (interprets using IE integration)
Refresh Browser View	refreshes browser view of active file

Environments (View menu)



This command presents the Environment Selector dialog where the user specifies the active editing environment.



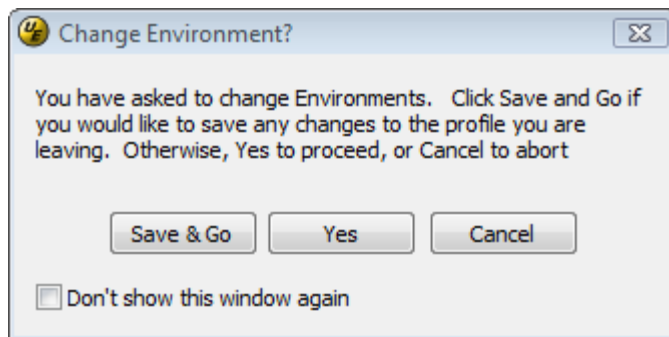
The **Environment Selector** dialog allows users to choose between predefined editing environments. An environment includes specific menus, toolbars and dockable windows as well as customized templates

for selected environments. This provides a great time-saver as users may switch between different display options for all of these items at once rather than configuring each of these individually.

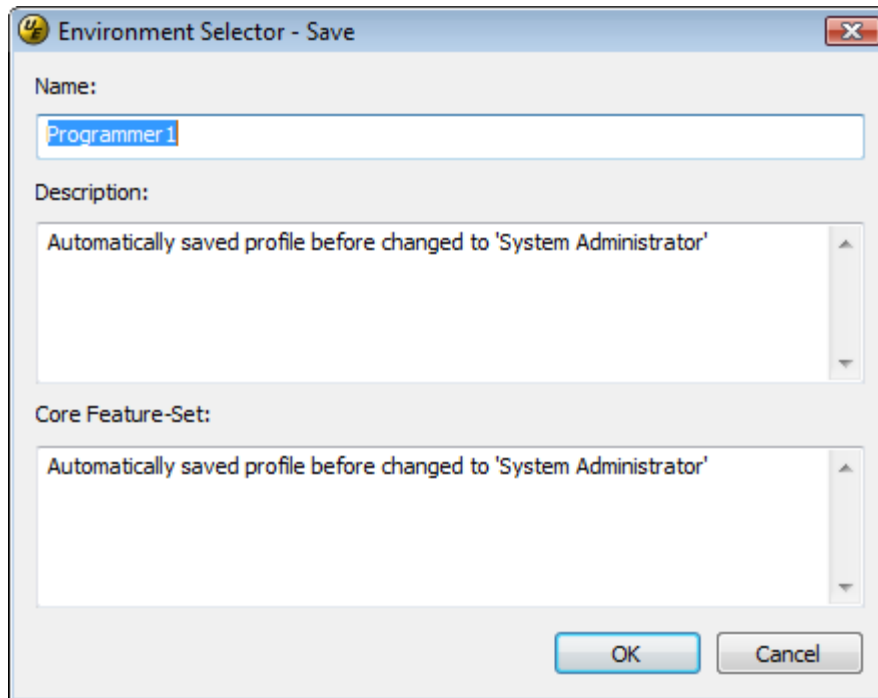
When the Environment Selector is first displayed, the active environment will be selected in the left pane with a description of the environment and its core feature-set displayed to the right.

Switching Environments

The active environment may be changed by selecting a new environment from the list and pressing the **OK** button. When this is done the user will be prompted to confirm he wishes to change environments.



If **Yes** is selected, the active environment will be changed to the one selected in the **Environment Selector** dialog. If **Cancel** is pressed, no change will occur. If **Save & Go** is selected, any changes the user has made to the current environment will be saved and the user will be prompted to name the newly created environment and optionally populate the **Definition** and **Core Feature Set** for that environment.



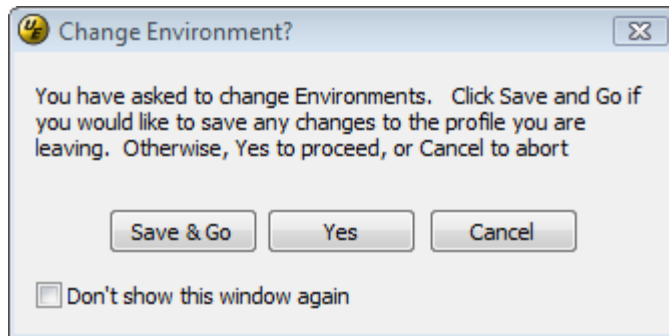
Users may also open a new instance of UltraEdit using a different environment by holding the CTRL key while clicking on the **Select Environment** button in the MAIN toolbar and selecting an environment from the list.

Customizing Environments

Default environments may be customized if desired. When an environment is active, a user can customize environment items individually. Menus and toolbars may be customized by right-clicking on any menu heading (File, Edit, etc.) or toolbar and selecting **Customize Menu** or **Customize Toolbar** respectively. Dockable windows may be opened and positioned as desired to meet the user's needs.

Once the active environment is customized, the **Environment Selector** dialog should be invoked by selecting it from the View menu or by pressing ALT+F8. The user may then edit the name of the active environment in the text box at the top of the selection list and press the **Save Current State As** button to save the customized version of the active environment.

Alternatively, users may simply modify the elements of an environment and select an environment from the Environment Selector to switch to. The "**Change Environment?**" dialog will be displayed and the user may select the **Save & Go** option.



This will populate the **Environment Selector - Save** dialog with the description and feature set of the active environment and allow the user to save the modified environment under another name, or overwrite the existing environment, if desired.

Any defined environment may be deleted by selecting it in the selection list and pressing the **Delete Selected Environment** button.

If the **Reset Built-In Environments** button is pressed, all built-in environments will be restored to their default state. **Please note:** Any customer environments that have been created in addition to the default environments will be preserved in their current state.

Exporting/Importing Environments

If a user wishes to export an environment they have created to share with a friend or co-worker, this may be done by pressing the **Export/Import Environment** button. This will present a popup menu where the user may choose one of two options: **Export Selected Environment** or **Import Environment**.

If the user selects the **Export Selected Environment** button, the **Browse for Folder** dialog is presented and the user can select the folder to export the environment file to or create a new folder to export to.

The selected environment will be created as a file with the name of the selected environment and the extension ".uee". For example, the **Power User** environment would be exported as "Power User.uee".

If the **Import Environment** button is pressed, the **Open** dialog will be displayed so the user may navigate to the *.uee file that should be imported for the desired environment. The settings in the imported file will be automatically parsed and the environment will then be available for selection in the Environment Selector dialog.

Toolbars command (View menu/ Views/Lists submenu)



Use this command to display and hide the Toolbars, which include buttons for some of the most common commands in UltraEdit, such as File Open. A check mark appears next to the menu item when the Toolbar is displayed.

See [Toolbar Configuration](#) for help on modifying the toolbar.

Status Bar command (View menu/ Views/Lists submenu)



Use this command to display and hide the Status Bar, which describes the action to be executed by the selected menu item or depressed toolbar button, and keyboard latch state. A check mark appears next to the menu item when the Status Bar is displayed.

See Status Bar for help on using the status bar.

Open File Tabs command (View menu/ Views/Lists submenu)



This menu item enables or disables the tab control that allows the user to select between open documents. The tab control allows easy selection between open documents while maintaining all other window switching and display functionality.

If you right-click on the open file tabs a pop-up menu is displayed with several file options.

The tab control may be dragged from its default docked position at the top of the main window and docked at the bottom of the window, or left standalone. If it is standalone, it may be resized vertically and horizontally as desired.

Additionally, the individual tabs may be dragged to a new location. To drag a tab, left click on the tab you wish to move and while holding the mouse button down, move it over the tab you wish to insert before

and release the mouse button. When dragging tabs, the tab will always be inserted before the tab that you "drop" the tab on. The tab will only be moved if you release the mouse button while positioned over an existing tab. If the **New file tabs look** option is selected under **Application Layout** in **Configuration**, the individual tabs may not be dragged to a new position.

When the tab control is in its default docked position, scroll arrows appear if more documents are open than can be displayed.

The tab control may be resized. If resized while docked so that more than one row of tabs are shown the scroll arrows will no longer be available.

Right-clicking on an open file tab will allow the file tabs to be sorted. These will be sorted alphabetically when the command is performed. Due to the operation of the underlying control used, if multiple lines are displayed the underlying control will reorder the ROWS of tabs but internally keep the order the same. This means that the sort will work but the rows may not be in order and will change order if a file from a different row is selected.

Double clicking on the File Tab for an open document will close the document.

If the cursor is positioned above a File Tab a tooltip will be displayed showing the full path of the related file.

Locking the File Tabs

In UltraEdit a right click in empty space by the open file tabs will cause a tip **Undockable** to appear. If this item is checked, the open file tabs may be dragged from their default docked position to a floating dialog. Double-clicking on the title bar of this dialog will return the file tabs to their default docked position. If **Undockable** is NOT checked, the file tabs are "locked" and may not be removed from their default docked position.

File Tree View command (View menu/ Views/Lists submenu)



This menu item enables or disables the File Tree View.

The File Tree View shows a list of files or directories for the selected drive in a tree type format. The Drive/File List shows the drives available and additionally allows a selection for displaying all the open files or files in the current project.

Double clicking on any of the files in the Tree View will open or activate the file. Multiple files may be selected and opened at once. (To select multiple files, use the left mouse button and the shift key to select between two files, or the control key and the left mouse button to select multiple non contiguous files).

The Filter entry box allows the user to specify a file filter that will be applied to the files shown when viewing one of the drives. It does not apply when showing project or open files.

Several functions are available by right clicking on the appropriate file. These are dependant on the list being shown.

Open Files

If the list of files shown are the currently open files the following functions are available when the right click menu is shown:

Close	closes an opened document
Save	saves an opened document using the same file name
Print	prints a document
Tile Horizontal	arranges windows in non-overlapped horizontal tiles
Tile Vertical	arranges windows in non-overlapped vertical tiles
Show Names Only	displays file names only (no path)
Allow Docking	permits docking of File Tree View rather than "floating" view
Hide File List	dismisses File Tree View
Make Folder Active Path	sets File Tree View to folder of active file

All of the above functions work on ALL the selected files.

Project Files/Drive

If the list of files shown are the currently open files the following functions are available when the right click menu is shown:

Open Selected File - This will open the selected file(s) in the editor.

Properties - This will show the file properties for the FIRST item selected.

Common

At any time right clicking on the Tree View will provide options to hide the view, or to disable the docking. If it is not docked it will show as a window unattached to other windows.

ASCII Table command (View menu/ Views/Lists submenu)



This feature displays a pop-up dialog which contains an ASCII Table. This may also be invoked as a dockable window from the Views/Lists submenu.

This table contains the following columns:

Character

This column shows the characters as they would be displayed using the selected font. The font may be changed using the Select Font button.

Decimal

This is the decimal value of the character in the range of 0 to 255.

Hexadecimal

This is the hexadecimal value of the character in the range of 0 to FF.

Name

This is the recognized name for the character. This is for the first 32 characters only.

Ctrl Character #

This is the recognized control character that the character represents. This is for the first 32 characters only.

To view the characters with a different font use the Select Font button. When the ASCII Table starts up it defaults to the font currently selected within the editor.

If you wish to insert any of the characters in to the active file select the character and click the Insert Char button or double-click the character in the table.

Function List command (View menu/ Views/Lists submenu)



Use this command to toggle Function List visibility. This function is different from the [Function List command](#) in the [Search menu](#) as that option populates the Function List based on the Function Strings defined in the wordfile (see [Syntax Highlighting](#) for more information).

The context menu in the Function List includes the following options:

Sort List	sorts functions alphabetically. If this is not selected, functions are listed in logical order.
List for all Project Files	lists functions for all files in active project. If this is not selected, the functions listed are limited to the active file.
Copy to Clipboard	copies contents of function list to clipboard
Refresh Function List	refreshes function list (equivalent to F8)
Flat List	toggles display of functions between flat and hierarchical modes

Macro List command (View menu/ Views/Lists submenu)



This command toggles the visibility of the **Macro List**. The Macro List displays all currently loaded macros. Macros may be loaded from the [Macro menu](#). Double-clicking on any listed macro will activate that macro similar to the **Play Again** command in the Macro menu. Right-clicking in the Macro List presents a context menu with the option **Modify Macro**. If this option is selected, the [Edit/Create Macro dialog](#) will be displayed.

Tag List command (View menu/ Views/Lists submenu)



This command displays a dialog showing a list of predefined "tags". A tag is simply a short string of characters and the name "tag" is used as this feature is commonly used for "HTML Tags".

The purpose of the tag list is to provide a method to easily insert into a document commonly used tags/strings without having to type them.

The Modify command in the right-click context menu allows the list of tags to be modified or a different taglist file to be selected. The tags consist of a start and end tag.

To insert a tag, select the tag you wish to insert and double click on the tag or press enter. When the tag is inserted into the active document, the start tag (if defined) is inserted at the beginning of any text selected and the end tag (if defined) is inserted at the end of the current selection. If no selection is made the tags are inserted at the cursor location.

By double-clicking on one of the tags UltraEdit will insert the tag and the focus will change to the file. If Enter is pressed while focus is in the Tag List UltraEdit will insert the tag and focus will remain in the Tag List. CTRL+Enter will also insert the tag and the focus will change to the file.

Multiple groups of tags may be defined and the group may be selected from the drop-down list.

Adding New Tags

The default file for the tags is the taglist.uet file. This is stored in the directory where the uedit32.ini / uestudio.ini file is located which is by default %APPDATA%\IDMComp\UltraEdit or %APPDATA%\IDMComp\UEStudio. Additional tag lists have been provided by users and may be found at <http://www.ultraedit.com/downloads/extras.html#taglists>. If a user navigates to this page and selects the Tag List they would like to add (for example ASP Tags) they can save this to their local system.

The full tag list should be pasted into the taglist.uet file just above the "Group List" section at the bottom. The EXACT title must be added to the [Group List]. For example, the ASP tag list title is:

[Tag Group - ASP - Active Server Pages]

The original [Group List] should be modified from:

```
[Group List]
0=COLD FUSION - Tags
1=HTML - Special Characters
2=HTML - Tags
3=UE/UES Macro Commands
4=UE/UES Script Commands
5=
```

to:

```
[Group List]
0=COLD FUSION - Tags
1=HTML - Special Characters
2=HTML - Tags
3=UE/UES Macro Commands
4=UE/UES Script Commands
5=ASP - Active Server Pages
6=
```

when adding ASP tags. The empty lines at the end of the Group List and the Tag List are important and should not be forgotten.

Template List command (View menu/ Views/Lists submenu)



This command toggles the visibility of the **Template List**. The Template List displays all templates that have been defined under **Display/Modify Templates** in the [Advanced menu](#). Double-clicking on any listed template will insert that template into the active document at the current cursor position. Right-clicking a template name and selecting **Insert Template** from the context menu will insert the selected template. Right-clicking a template name and selecting **Modify Template** presents the **Define Template** dialog where a user may define the text inserted for the specified template.

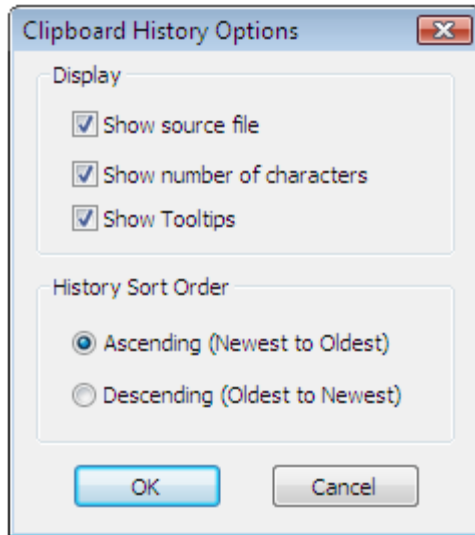
The user may choose whether to use global templates or templates specific to the active environment. Right-clicking in the template list and selecting **Global Templates** will cause the list to be populated with the templates defined for global use. If using global templates, an option will be available in the context menu to choose templates specific to the active environment. For example, in the **Programmer** environment, a **Programmer Templates** option would be available.

Clipboard History command (View menu/ Views/Lists submenu)

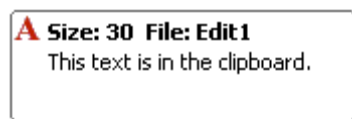


This command toggles the visibility of the **Clipboard History**. The Clipboard History displays a listing for any text copied to clipboard while the history list is active. Double-clicking on any listed item inserts it into the active document.

Right-clicking in the Clipboard History presents a context menu where the user may choose to **Clear Clipboard History** if desired. Selecting **Options** in the context menu presents the **Clipboard History Options** dialog.



The **Clipboard History Options** dialog is used to specify the content to be displayed in the Clipboard History for all clipboard items:



Right-clicking on a history entry and selecting **Delete History Entry** from the context menu will remove the selected item.

Script List command (View menu/ Views/Lists submenu)



This command toggles the visibility of the **Script List**. The Script List displays all currently specified scripts. Right-clicking in the Script List presents the context menu which includes the following options:

Run	runs selected script
Add Script	presents File Open dialog where user may specify script to add to list
Edit Script	opens selected script in editor window
Remove Script	removes selected script from list

Bookmark Viewer command (View menu/ Views/Lists submenu)



This command toggles the visibility of the **Bookmark Viewer**. The Bookmark Viewer displays all bookmarks currently specified for the active document. If no bookmarks are defined the list will display "No Bookmarks!". Right-clicking in the Bookmark Viewer presents the context menu which includes the following options:

Columns	presents submenu allowing user to select items visible in list
Columns -> Column Order	presents dialog allowing user to change the order of items visible in list
List Bookmarks for all Files	displays list of bookmarks in all open files. If this option is selected, the name of the file the bookmark exists in will also be displayed.
Delete Bookmarks	deletes currently selected bookmark
Edit Bookmark	opens Edit Bookmark dialog where name of bookmark may be edited

If desired, the bookmark name may be modified using the Edit Bookmark command. If focus is moved to a bookmarked line, that line will automatically be highlighted in the Bookmark Viewer.

Output Window command (View menu)



This menu item enables or disables the Output Window. The Output Window is used to capture the output from user tools (see Tools Configuration) or from the [Find in Files](#) output.

When lines are added to the Output Window from a tool (not [Find in Files](#)) the list is scrolled down automatically enabling the user to see the end of the results without having to scroll.

Double clicking on a line containing the filename and line number will cause UltraEdit to attempt to open the file specified at the line number. If the filename is fully qualified the filename will be located and the number after this will be used for the line number. If the filename is not fully qualified UltraEdit will attempt to determine the filename from the first word in the line that contains a period. UltraEdit will then open the specified file in the directory of the active file. If the file does not exist in the directory of the active file, UltraEdit will try and open the file from the [Project Directory](#) if it is specified.

By double clicking on one of the result lines UltraEdit will position to that line in the file and the focus will change to the file. If Enter is pressed while focus is in the Output Window UltraEdit will position to that line and focus will remain in the Output Window. CTRL+Enter will also position to the line in the file and the focus will change to the file.

View As (Highlighting File Type) command (View menu)



This allows the user to override the [Syntax Highlighting](#) selection made by UltraEdit based on the file extension and view any file as one of the available languages.

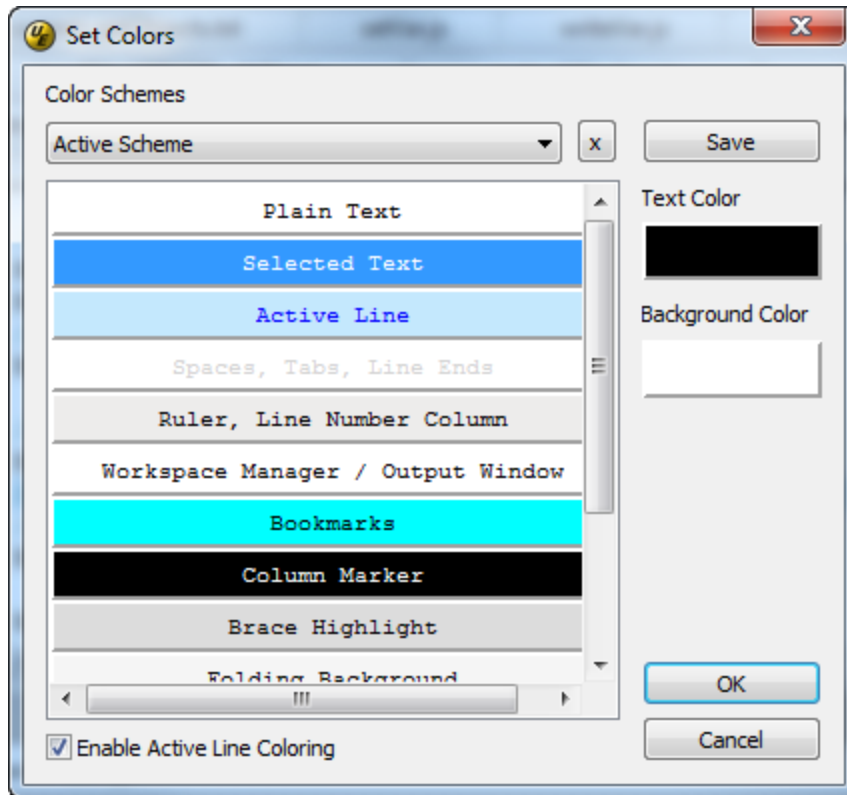
The menu selection allows selection for any of the languages available within Syntax Highlighting and a selection of no highlighting.

If the user uses this feature to override the syntax highlighting the override will stay in effect until the file is closed or saved under a new name.

Set Colors command (View menu)



This menu item presents the **Set Colors** dialog where the user may configure the color of various items in the editor:



While most of the color options are self-explanatory it may be helpful to know that **Brace Highlight** is used to set the color used for automatic brace matching and **Folding Background** is used to set the color used for the folding gutter background. The **Read Only Background** color is substituted for the background color if a file is opened in Read Only mode.

The **Line Change Indicator** options (**Saved** / **Unsaved**) are used to specify colors shown at the beginning of a line to indicate changes made in the current editing session. A different color is used to indicate changes that have been saved and those that have not yet been saved. If the undo buffer is cleared during the editing session, the line change indicator colors would be cleared.

When a labeled color block (i.e. **Plain Text**, **Selected Text**, etc.) is selected, users may specify colors to be used for the selected item using the **Color**, **Text Color**, and **Background Color** options displayed to the right. Clicking on the color block displayed under a label presents a **Color** dialog where the user may select a new color to be used for that option. For the background colors and bookmark color selections, only solid colors are valid.

The background and bookmark colors are used for all files. The text color is used for all files that are not recognized by [Syntax Highlighting](#).

Set Font command (View menu)



The Set Font menu item presents the Choose Font dialog that allows selection of different fonts and font sizes. The selection made will apply to the screen fonts and will also be applied to the printer font if the [Mirror Display Font](#) menu item is checked on the View Menu. If this item is not checked, the printer fonts have been selected independently from the screen fonts.

Font selections may be reset to default values by selecting the [Reset Fonts](#) menu item.

Note: If the printer does not support the font Windows automatically adjusts to the closest font.

Set Hex/Column Mode Font (View menu)



The Set Hex/Column Mode Font menu item presents the Choose Font dialog that allows selection of different fonts and font sizes. The selection made will apply when editing in [Hex mode](#) or in Column mode. UltraEdit will automatically switch to this font when Hex or Column modes are selected.

For normal font selection use [Set Font](#) from the [View Menu](#).

Reset Fonts command (View menu)



The Reset Font menu item instructs UltraEdit to reset user selections of the screen and printer fonts to the default settings. Any user selections of the screen fonts made with the [Set Font](#) menu item or printer fonts made with the [Set Printer Font](#) menu item will be lost.

Increase Font Size command (View menu)



These items allow a user to zoom in and out of a document by increasing and decreasing the font size by about 1 point size with each change. These settings are retained for all documents once they are changed.

Decrease Font Size command (View menu)



These items allow a user to zoom in and out of a document by increasing and decreasing the font size by about 1 point size with each change. These settings are retained for all documents once they are changed.

Show Spaces/Tabs command (View menu)



This menu items allows the displaying of spaces, tabs and new line characters as visible characters on the screen. It does not change the document but allows you to clearly distinguish between the different characters.

Spaces are shown as: ' .'

Tabs are shown as: ' »'

New lines are shown as: ' ¶'

If there is not enough room to show the ' »' with the tab, the tab will appear as normal, and non visible.

Show Page Breaks as Lines command (View menu)



The setting instructs UltraEdit to show the page break character (as defined in [Page Setup](#)) as a line drawn across the screen. This will not modify the line numbering of the file. The line numbers are based on the line terminators only.

If this setting is checked, UltraEdit will draw a straight line horizontally across the screen immediately below the current line. Any text following the page break character will start on the line following the page break line.

If this setting is not checked, the page break character will be printed as with any other character and text will be displayed immediately prior to, and following the character according to what is present.

Set Column Markers command (View menu)



This menu item presents a dialog allowing the user to set a **Column Group** as active and specify the associated **Column Markers** which will be displayed when that group is set active. The **Set Column Markers** dialog is made up of two list controls - **Column Groups** and **Column Markers** - that may be used to define groups of column markers which may be stored and set active for different types of documents.

Column Groups

In the **Column Groups** list, a group name is selected by clicking on it and may be edited if it is clicked on again. This allows column groups to be assigned unique names. This list has four associated buttons:

Set Active	sets selected group active and displays associated column markers in the edit window
Add Group	inserts a new Column Group item into the list
Modify Group	toggles the selected column group name to edit mode so it may be renamed
Delete Group	removes the selected column group from the list

Column Markers

The **Column Markers** list consists of two separate columns, **Enabled** and **Column Number**. When a **Column Group** is selected in the first list, the column markers associated with that group may be defined within the lower list.

When a column marker is defined a checkbox appears in the **Enabled** column and the user specifies a column number at which the marker should be displayed in the **Column Number** column. The associated marker is only displayed if checked. In the **Column Markers** list, a column number is selected by clicking on it and may be edited if it is clicked on again. This list has three associated buttons:

Add Marker	inserts a new Column Marker item and its associated checkbox into the list
Modify Marker	toggles the selected column marker number to edit mode so it may be modified
Delete Marker	removes the selected column marker from the list

No Scroll Left Side of Marker 1 command (View menu)



UltraEdit provides for [Column Markers](#) that display a vertical line at a specified column.

This menu item works in conjunction with the column markers. When this item is checked (enabled) the columns to the left of the 1st column marker are fixed, and do not scroll horizontally while the columns to the right of the column marker will scroll as normal when the horizontal scroll bar is used or when the cursor is positioned to the right hand side of the window forcing a horizontal scroll.

If this feature is enabled, columns to the right of the marker will scroll horizontally while columns to the left will not. While the horizontal scroll is in effect, any mouse selection or clicking of the primary mouse button within the fixed portion of the display will cause the horizontal scroll position to be reset allowing viewing and editing of the continuous columns.

If the 1st column marker is disabled or set to 0 this function will have no effect.

Display Ruler command (View menu)



This menu item toggles the column ruler that displays the column numbers at the top of each window.

The ruler indicates the column number in even increments. For fixed pitch fonts this will show the correct column indication. For variable pitch fonts, this will show the column numbers based on the average character width for the active font.

The ruler colors may be set with the [Set Color dialog](#).

Display Line Numbers command (View menu)



This menu item toggles the line number display on the left hand side of the window.

The line numbers are shown in a fixed area on the left hand side of the Window for each file if this item is checked. The font used is automatically selected based on the current font and is slightly smaller than the selected font for display.

The color of the line numbers may be set with the Set Color dialog.

To print a file with line numbers, make the appropriate selection under the [Page Setup dialog \(File Menu\)](#).

Spacing commands (View menu)



These items provide for Single, 1 ½ and Double spacing. This allows a document to be viewed and printed with additional spacing between each line of text.

If Single spacing is selected, each line is displayed and printed with the normal character height and line spacing.

If 1 ½ spacing is selected, each line is displayed and printed with the 1 ½ the normal character height for the line. The actual characters are the normal height but spacing allocated for the line is increased giving additional room for notes to be hand written between lines.

If Double spacing is selected, each line is displayed and printed with the 2 times the normal character height for the line. The actual characters are the normal height but spacing allocated for the line is increased giving additional room for notes to be hand written between lines.

Hide/Show Selection command (View menu / Hide/Show Lines submenu)



This function allows a user to select some number of lines and to hide these such that they will not be visible on the screen, or if lines are hidden they will be shown.

Hiding Lines

To hide a set of lines, select the lines you wish to be hidden and invoke the feature from the Menu Item or by using the shortcut (CTRL+ '-' numeric keypad). When the lines are hidden, a "+" will be displayed in the folding gutter to the left of the edit window. When editing a file that does not have syntax highlighting, the following configuration options must be selected under Advanced -> Configuration -> Editor Display -> Code Folding:

Enable show/hide lines and code folding

Enable show/hide lines in non syntax highlighted files

Multiple sections may be hidden at one time. Changes made to the file will update the hidden regions in most cases such that the hidden lines are maintained. In a few instances (file sort for example) this may not be possible and the hidden lines may be offset requiring them to be hidden again.

Clicking on the "+" in the folding gutter beside hidden lines will cause the section no longer to be hidden. This may be re-hidden by using this Hide Lines feature without a selection – in this case, the last set of lines that was hidden and then shown will be hidden again.

This command can be used from within [Macros](#).

Un-Hiding Lines

To display a set of lines that is hidden, click on the "+" in the folding gutter beside the hidden lines, or position to it via some other means and select Hide/Show Selection under Hide/Show Lines in the View menu.

Saving the Hidden Lines

If the user has the configuration option set to Save Bookmarks the hidden line information will also be saved and reloaded when the file is next loaded.

Collapse All command (View menu / Hide/Show Lines submenu)



This command collapses all blocks in the active file bounded by the Open/Close Fold Strings defined in the active wordfile. If text is selected this will affect only the selected region. If no text is selected this command will operate on the whole of the active file.

For more information see [Code Folding](#).

Expand All command (View menu / Hide/Show Lines submenu)



This command expands all blocks in the active file bounded by Open/Close Fold Strings defined in the active wordfile. If text is selected this will affect only the selected region. If no text is selected this command will operate on the whole of the active file.

Please note: When this command is invoked, strings hidden using the [Hide Lines command](#) will be expanded as well as folded sections of code.

For more information see [Code Folding](#).

File Change Polling command (View menu)



This command toggles File Change Polling for the active file. This can be very useful for log files or any other files that might be dynamically updated. When this is selected the active file will be checked periodically (based on the interval specified under the [File Change Detection](#) topic under the File Handling branch under Configuration in the Advanced menu) and updated with any changes that have been made to the file.

Please note: This command will be grayed out in the View menu unless the associated Configuration item - Poll for file changes at force interval in seconds - has been selected and a polling interval has been set.

Auto Scroll Polled Files command (View menu)



This command toggles the auto scroll state of all files for which polling has been activated using the [File Change Polling command](#). If this is selected all polled files will automatically be scrolled to display the last line each time the files are polled. In the case of dynamically updated files, such as log files, this ensures that the latest additions to the file are always visible when the polled file is active. This functionality is similar to the UNIX tail command.

Always On Top command (View menu)



This command toggles the setting to display UltraEdit on top of all other applications.

Full Screen Mode command (View menu)



This function allows the active file to be edited using the whole screen. The toolbar, status bar and other windows that would normally be displayed as part of the UltraEdit application will be hidden while in this mode.

A small window with an icon on it will be shown to allow the user to switch back to normal editing. Additionally, the ESCAPE key may be used to cancel the full screen mode.

When the full screen mode is cancelled, the main UltraEdit window and editing window will be returned to its previous state.

Format Menu

Format menu



The Format menu offers the following commands:

Reformat Paragraph	reformats paragraph to wrap a column
Paragraph Formatting	aligns paragraph (Left/Right/Center/Fill)
Convert CR/LFs to Wrap	converts CR/LFs to Word Wrap for selected text
Convert Wrap to CR/LFs	converts Word Wrap to hard CR/LFs for complete file
Tabs to Spaces	converts tabs to spaces
Spaces to Tabs (All)	converts all spaces to tabs
Spaces to Tabs (Leading)	converts leading spaces to tabs
Trim Trailing Spaces	trims the trailing spaces from each line
OEM to ANSI	converts OEM Character Set to ANSI Character Set
ANSI to OEM	converts ANSI Character Set to OEM Character Set
To Upper Case	converts the selected text to upper case
To Lower Case	converts the selected text to lower case
Capitalize	capitalizes the first character of each word in selected text

Invert Case	inverts the case (lower/upper) of each character in selected text
Add Indents	presents a dialog to allow adding indentation from selected rows of text
Remove Indents	presents a dialog to allow removing indentation from selected rows of text
HTML Validation - Run HTML Tidy	checks and tidies HTML files
HTML Validation - CSE HTML Validator (and options)	runs the CSE HTML Validator and sets up its options
JavaScript Lint Tool	invokes JavaScript Lint to check for mistakes in JavaScript source code. For more information see http://www.javascriptlint.com/
XML Convert to CR/LFs	formats single line XML by inserting hard returns as appropriate
XMLlint Tool	invokes XMLlint parser. For more information on commands and use please see http://xmlsoft.org/xmllint.html
ReIndent Selection	re-indents selected text in active file

Reformat Paragraph command (Format menu)



Use this command to format the current paragraph (or selected text) to the specified column number boundary. The column number and settings are specified in the [Paragraph Setup/Formatting dialog](#). A paragraph is determined as having two hard returns in succession (i.e. a blank line).

All hard returns are removed for the paragraph, and the paragraph is reformatted to the appropriate column boundary. Please note that the reformat process will remove all bookmarks from bookmarked lines within the area selected for reformatting. An Undo executed after the reformat process completes will not restore removed bookmarks.

The whole file may be reformatted using the [Select All command](#) to first select the complete file, then reformat it.

If the [Auto Indent](#) feature is enabled the paragraphs will be automatically indented based on the first line of the paragraph.

The settings used in the [Paragraph Formatting dialog](#) will be used for this command.

Paragraph Setup/Formatting dialog (Format menu/Paragraph Formatting submenu)



The paragraph formatting allows the user align the paragraph in the following manners:

Left Align	Text will be left aligned against the left margin and will not extend past the right margin.
Center Align	Text will be centered between the left and right margins.
Right Align	Text will be right aligned against the right margin and will not extend past the left margin.
Fill	Text will be aligned between the right and left margins with spaces added between words to ensure alignment to both margins. The last line of the paragraph will be aligned only to the left margin.

Each of the separate alignment commands may be selected individually from the menu, or invoked from the Paragraph Setup/Reformatting dialog.

The Paragraph Setup/Reformatting dialog specifies the parameters used for reformatting. These parameters are used when one of the paragraph formatting commands are used.

The parameters are:

Hanging Indent	This applies to the left alignment only. If set, the alignment will allow a second margin to be specified that is used for the second and successive lines of a paragraph. This causes the first line of a paragraph to be indented with one setting and the rest of the paragraph to have a different setting. If this is set, the hanging indent value may be entered in the dialog.
Margins - Use paragraph left margin	This instructs UltraEdit to use the existing left margin of the paragraph that is being reformatted rather than specifying a new margin value.
Margins - Specify margin	This setting instructs UltraEdit to use the left margin value entered in the dialog as the indentation for the left margin when reformatting the paragraph.
Margins - Right margin	This setting specifies the right margin to be used when reformatting the paragraph. This is the last column number (starting at 1) at which text may be placed.

The margin settings above will be used for all reformatting commands including the [Reformat Paragraph command](#).

All hard returns are removed for the paragraph, and the paragraph is reformatted based on the settings.

The complete file may be reformatted using the [Select All command](#) to first select the complete file, then reformat it.

Convert CR/LFs to Wrap command (Format menu)



Use this command to remove the CR/LF (carriage return/line feed) characters from the file allowing the text to be Word Wrapped by UltraEdit. This feature removes all CR/LF characters for text that is selected. If no text is selected this function has no effect.

To remove all CR/LF characters from the file select the complete file. To do this, position the cursor at the beginning of the file and press CTRL+SHIFT+END at the same time.

To add CR/LF (carriage return/line feed) characters to the file use the Convert Wrap to CR/LFs command.

Convert Wrap to CR/LFs command (Format menu)



Use this command to [Word Wrap](#) the text based on either the width of the display window or a specified column number and write CR/LF (carriage return/line feed) characters to the file. If a space is found at the position of the wrap a single space is removed. This allows the document to be saved as viewed in the word wrap state with CR/LF characters inserted at the end of each word wrapped line.

When this item is selected, a dialog box is displayed allowing selection of the wrap parameters. By default, the text will be wrapped based on the current view. That is to say, the text will be wrapped at the window boundary as shown on the screen. This dialog allows selection of a column number at which the text should be wrapped. In this case the text will be wrapped at the specified column or at the preceding word boundary on the line.

This feature works on the complete file if no text is selected. If a selection is made, the feature only converts the selected area.

To remove CR/LF (carriage return/line feed) characters from the file use the [Convert CR/LFs to Wrap command](#).

Tabs to Spaces command (Format menu)



Use this command to convert all of the TAB characters within a FILE to spaces. This uses the [Tab Settings](#) to determine how the tab character/tab stop is to be interpreted. The conversion is performed such that the formatting will remain the same.

If a portion of the file is selected/highlighted only the selected portion is converted, otherwise the complete file is converted.

To convert spaces to tabs use the [Spaces to Tabs \(Leading\)](#) or [Spaces to Tabs \(All\) command](#).

Spaces to Tabs (All) command (Format menu)



Use this command to convert all of the spaces in a file to tabs. This uses the [Tab Settings](#) to determine how the tab character/tab stop is to be interpreted.

The conversion is performed such that the formatting will remain the same. This will mean that where there are insufficient spaces to replace with a TAB character and retain the same formatting the spaces will be left in place.

If a portion of the file is selected/highlighted only the selected portion is converted, otherwise the complete file is converted.

If you wish to convert only the space at the beginning of a line use the [Spaces to Tabs \(Leading\) command](#).

To convert tabs to spaces use the [Tabs to Spaces command](#).

Spaces to Tabs (Leading) command (Format menu)



Use this command to convert all of the (leading) spaces in a file to tabs. This uses the [Tab Settings](#) to determine how the tab character/tab stop is to be interpreted.

The leading spaces refers to the space at the beginning of a line. Any spaces after the first non-space character in a line will not be converted. To convert all spaces in a line use the [Spaces to Tabs \(All\) command](#).

The conversion is performed such that the formatting will remain the same. This will mean that where there are insufficient spaces to replace with a TAB character and retain the same formatting the spaces will be left in place.

If a portion of the file is selected/highlighted only the selected portion is converted, otherwise the complete file is converted.

To convert tabs to spaces use the [Tabs to Spaces command](#).

Trim Trailing Spaces command (Format menu)



Use this command to remove the trailing white space (tabs, spaces) from the ends of each line. This function removes the trailing white spaces from the end of every line in the file and positions back to the original cursor position when it is complete.

The [Save](#) topic under the File Handling branch in Configuration under the Advanced menu provides a setting allowing spaces to be trimmed automatically when a file is saved.

To Upper Case command (Format menu)



This command changes all selected text to upper case. There is no limit on the amount of text that may be selected. If [Column Mode](#) is selected, the selected columns only are converted.

Shortcut: ALT+F5

To Lower Case command (Format menu)



This command changes all selected text to lower case. There is no limit on the amount of text that may be selected. If [Column Mode](#) is selected, the selected columns only are converted.

Shortcut: CTRL+F5

Capitalize command (Format menu)



This command changes the first letter of all words in the selected text to upper case and the remaining letters to lower case. There is no limit on the amount of text that may be selected. If [Column Mode](#) is selected, the selected columns only are converted.

Shortcut: F5

Invert Case command (Format menu)



This command inverts the case of all selected text. There is no limit on the amount of text that may be selected. If [Column Mode](#) is selected, the selected columns only are converted.

Shortcut: SHIFT+F5

Add Indents command (Format menu)



The Add Indents menu item presents a dialog that allows the user to specify the number of spaces to add at the beginning of each of the selected (highlighted) lines. UltraEdit starts adding the specified number of spaces to each line beginning with the first highlighted line and ending with the last line that has at least one character selected (highlighted).

The Indents will be added to all lines stated above, irrespective to and indentations that may already exist.

To remove indentations, use the [Remove Indents](#) command on the [Format Menu](#).

Remove Indents command (Format menu)



The Remove Indents menu item presents a dialog that allows the user to specify the number of spaces or tabs to remove from the beginning of each of the selected (highlighted) lines. If the checkbox specifying removal of all indents is checked, UltraEdit will remove ALL indentation at the beginning of the selected lines. UltraEdit starts removing the indents beginning with the first highlighted line and ending with the last line that has at least one character selected (highlighted).

If no indentation is found at the beginning of a selected line, no changes are made to that line.

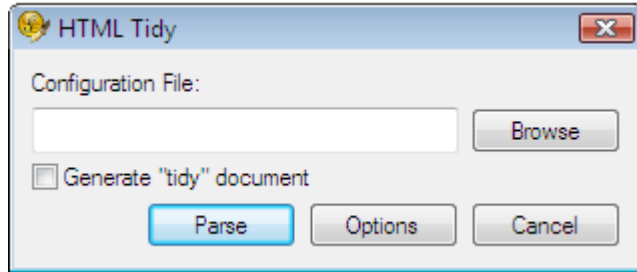
To add indentations, use the [Add Indents command](#) on the [Format Menu](#).

HTML Tidy command (Format menu/HTML Validation submenu)



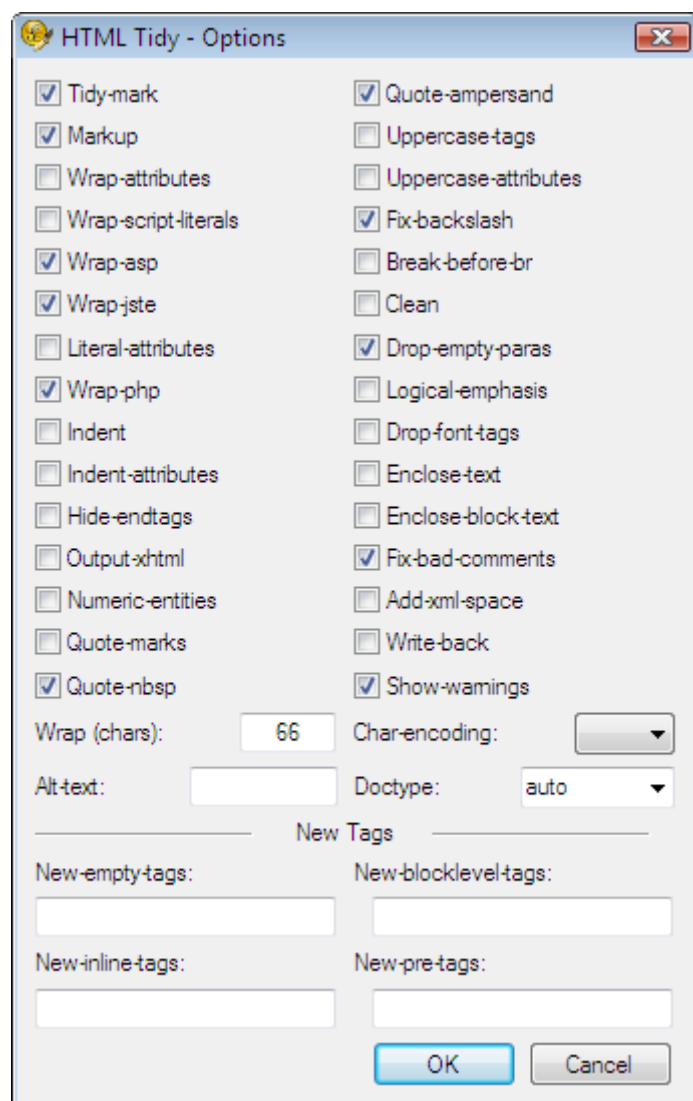
HTML Tidy is an excellent utility that formats, checks and tidies up HTML code. This was developed by Dave Raggett and has been integrated into UltraEdit. More information on HTML Tidy may be found at: <http://tidy.sourceforge.net/>.

HTML Tidy may also be invoked by pressing the HTML Tidy button in the [Tools toolbar](#) and this will present the HTML Tidy dialog:



To define a new config file, type the path to the file in the Configuration text field or press the "... " button to browse to the file you wish to use. Please note that this file must actually exist - specifying a config file that doesn't exist will cause an error when the parse is executed.

Pressing the Return button on the keyboard will save the specified path. Pressing the Options button will present the HTML Tidy options dialog. For a summary of the configuration options available for HTML Tidy please see <http://tidy.sourceforge.net/docs/quickref.html> .



When HTML Tidy is run from the Format menu or by pressing the Parse button in the HTML Tidy dialog, the active document will be saved and HTML Tidy will parse the file. The output window will be used to display any errors or warnings that are found. Double clicking on line with the error or warning in the output window will position the cursor to the appropriate line in the file.

If Generate "tidy" document is selected in the HTML Tidy dialog, HTML Tidy will generate a corrected version of the active file and UltraEdit will display this in a new file/window titled "*** HTML Tidy Output ***". A new "*** HTML Tidy Output ***" file will be created each time HTML Tidy is used. If this is selected and **Write-back** is selected as well, the output would be written back to the original file window, but the changes aren't saved until the modified file is saved. If neither **Write-back** nor Generate "tidy" document are selected no output will be created from the HTML Tidy process.

CSE HTML Validator command (Format menu/HTML Validation submenu)



CSE HTML Validator is an excellent utility that validates HTML, XML and other files. Through these menu items UltraEdit provides integration of the Validator such that it may be run directly from UltraEdit and its output will be captured and shown on the Output Window.

To make use of this feature, the Validator must be installed on your system. All support and information regarding the use of the Validator must be requested directly from the developers of CSE HTML Validator.

For further details, see:

[http:// www.htmlvalidator.com/](http://www.htmlvalidator.com/)

UltraEdit provides menu items to set up the configuration, options, and Job Types for the Validator but does not determine how these are used by the Validator.

XML Convert to CR/LFs command (Format menu)



This command may be used to reformat single line XML. When this option is invoked the editor scans the document and inserts hard returns as appropriate to reformat the document to a more readable format.

ReIndent Selection command (Format menu)



This command allows the user to reformat the selected text within the active file. This is typically used by programmers to reformat existing text according to the indentation settings in configuration and the automatic indent settings for [Syntax Highlighting](#) (see the indentation section under Syntax Highlighting for details).

When the reindenting is performed, UltraEdit will reindent the file line by line, starting with the indentation of the first selected line.

If the indentation of the first selected line is less than the indent setting, or falls between multiples of the indent setting then the indentation will be rounded down to match the value or multiple of the indent setting. (If the indent value is 4, and the first selected line is indented to 2 then the line will be assumed not to be indented as it will be rounded down to 0. If the line was indented to 6 then the line will be adjusted to be indented to 4 as it falls between the indent values of 4 and 8).

UltraEdit does try and determine if the statement causing a change in the indentation is commented out, or part of a string and ignores it in this case. However, if the file contains preprocessor conditional statements (# ifdef, #ifndef, etc.) UltraEdit does not attempt to determine which text is valid and which is not. This may cause lines to be adjusted incorrectly.

To avoid problems with conditional statements it is suggested that all conditional sections include the same number of open and close conditional statements. If an "if (test) {" statement is in the conditional text, make sure the closing brace ("}") is also in the conditional text and there should not be a problem.

In some cases it may be desirable not to indent lines such as compiler directives or comments. UltraEdit provides for this capability by allowing strings to be ignored to be defined in the active wordfile. This is covered under the Syntax Highlighting section.

Column Menu

Column menu



The Column menu offers the following commands:

Column Mode	toggles column/block mode editing
Insert/Fill Columns	specifies the text to insert into each column
Delete Columns	specifies how many/which columns to delete
Cut Columns	specifies how many/which columns to cut to the clipboard
Insert Number	specifies number range to insert into each row
Sum Columns/Selection	sums the total of all the numbers that appear in the selected area (columns or lines)
Convert to Fixed Column	converts character delimited file to fixed columns
Convert to Character Delimited	converts fixed columns to character delimited format
Left Justify	left justifies column

Center Justify	center justifies column
Right Justify	right justifies column

Column Mode command (Column menu)



This command changes the operation of the editor for selecting text. When column mode is selected highlighted text is based on the character column position of the first character selected to the column of the last character of text selected. Text selected in this mode does not automatically include all text between the start and end position, but includes all text in the columns between the first and last character selected.

Column mode editing is sometimes referred to as block mode editing as the act of selecting text forms a rectangle.

[Cut](#), [Copy](#) and [Paste](#) work with the selected text.

Column mode editing requires that [Word Wrap](#) be turned off and a fixed pitch font be selected.

When column mode is enabled, additional column features are enabled; [Insert/Fill Columns](#), [Delete Columns](#), [Cut Columns](#) and [Insert Number](#).

Shortcut: ALT+C

Insert/Fill Columns command (Column menu)



This command presents a dialog that allows a text string to be entered. This text will be inserted into every line that is selected, or if no selection is made it will be inserted into every row starting at the current cursor position and proceeding to the last row of the file. The text string will be inserted into every row at the column position indicated by the cursor when this command is invoked. If text does not exist at this column in any row, spaces ' ' will be added to pad the row allowing the text to be inserted correctly.

If at the invocation of this command, text is highlighted, the columns indicated by the highlighted text will be replaced with the specified text.

Delete Columns command (Column menu)



This command presents a dialog that allows the number of columns to be deleted to be specified. The columns will be deleted from every line that is selected, or if no selection is made they will be deleted from the file from every row starting at the current cursor position and proceeding to the last row of the file. If a value of zero '0' is entered, columns will be deleted based on the highlighted text when this command is invoked. This command does not save the deleted text in the clipboard.

Cut Columns command (Column menu)



This command presents a dialog that allows the number of columns to be deleted to be specified. The columns will be deleted from every line that is selected, or if no selection is made they will be deleted from the file from every row starting at the current cursor position and proceeding to the last row of the file. If a value of zero '0' is entered, columns will be deleted based on the highlighted text when this command is invoked. This command **does** save the deleted text in the clipboard.

Insert Number command (Column menu)



This command allows a sequential number to be inserted into every line selected, or if no lines are selected, into every line from the current line to the last line of the file at the column position indicated by the cursor. This command presents a dialog that allows the initial value to be entered and the increment for each line. This provides an easy method to add line numbers into a text file.

If any text is highlighted at the invocation of this command, the highlighted text will be replaced by the inserted line numbers. The spacing required for the line numbers is calculated automatically based on the number of lines that will have a number inserted, the initial value and the increment. All numbers may be positive or negative non decimal point numbers (32-bit signed = negative number in hexadecimal notation has 8 digits).

If the option to add **Leading Zeros** is selected the appropriate leading zeros will be added to each number to pad the field to the same width (in characters) as the maximum number to be inserted. If this option is not selected, trailing spaces will be used.

The user may select **Decimal** or **Hexadecimal** to specify the format of the inserted numbers. If "62" is entered in the **First Number** field, and Hexadecimal is selected as the output format, then 3E (which is the number 62 in HEX) will be the first number of the sequential numbers inserted.

Sum Columns/Selection command (Column menu)



This command sums the total of all the numbers that appear in the selected area (columns or lines) and presents the total in a dialog box.

As some number formats have a comma, or other character that is part of the number or a separator (i.e. one thousand may be represented as 1,000) this may be configured. The dialog allows the user to specify any character that should be ignored in a number and by default is initialized to a comma (,).

The SUM button recalculates the total. It allows the separator to be modified and the total recalculated.

The TOTAL is shown in an editable entry box allowing the total to be formatted and copied to the clipboard and pasted into the document if required.

Convert to Fixed Column command (Column menu)



This function allows the user to convert a file from character delimited fields to fixed columns. This is particularly useful where column manipulation is required, but difficult with the different field widths for each row. This complements the [Convert to Character Delimited](#) feature.

This function presents the user with a dialog box with the following controls:

Scan

This causes the editor to scan the file (or first line) and determine the maximum field widths for each field. This will determine the maximum width required for each field and place the results (comma separated) in the Field Widths entry box.

Scan First Line Only

This instructs the editor to search just the first line for the field widths.

Ignore Separator in Quotes (')

If this option is selected, separator characters which are enclosed in single quotes will not be taken into account when scanning to determine column spacing for alignment.

Ignore Separator in Double Quotes (")

If this option is selected, separator characters which are enclosed in double quotes will not be taken into account when scanning to determine column spacing for alignment.

Keep Delimiter with Fixed Columns

This indicates that when converting to fixed width columns the delimiter character for the fields should not be removed. If this is not checked, the conversion will remove the delimiter between fields.

Separator Character

This specifies the delimiter character between fields. If this is a TAB character "^t" should be entered. In all other cases, a single character should be entered. The editor uses this character to determine the field boundaries.

Field Widths

This is a list of comma separated field widths used for the conversion. If there are more fields than entries, the last entry will be used as the field width for the remaining fields. This allows the user to set a single entry if all fields are required to be the same width.

Please note: This field must not be left empty when the conversion to fixed columns is executed. It may be populated automatically by pressing the **Scan** button after specifying the conversion options, or the user may manually specify field widths if desired.

Convert

This performs the conversion from character delimited text to fixed column according to the settings above. The complete file will be converted.

Convert to Character Delimited command (Column menu)



This function allows the user to convert a file from fixed field widths to character delimited fields. This complements the [Convert to Fixed Column](#) feature.

This function presents the user with a dialog box with the following controls:

Separator Character

This specifies the delimiter character between fields. If this is a TAB character "^t" should be entered. In all other cases, a single character should be entered. The editor uses this character to separate the fields when converting from fixed widths to character delimited.

Field Widths

This is a list of comma separated field widths used for the conversion. If there are more fields than entries, the last entry will be used as the field width for the remaining fields. This allows the user to set a single entry if all fields are required to be the same width.

Convert

This performs the conversion from fixed column to character delimited text according to the settings above. The complete file will be converted.

Left Justify command (Column menu)



This command left justifies the text inside the selected columns. Any tabs within the selected range will be converted to the appropriate number of spaces to give the same alignment before the justification takes place. This is necessary as the tab expansion and alignment of text would potentially change after justify the text.

This function only works when in [Column Mode](#).

Center Justify command (Column menu)



This command center justifies the text inside the selected columns. Any tabs within the selected range will be converted to the appropriate number of spaces to give the same alignment before the justification takes place. This is necessary as the tab expansion and alignment of text would potentially change after justify the text.

This function only works when in [Column Mode](#).

Right Justify command (Column menu)



This command right justifies the text inside the selected columns. Any tabs within the selected range will be converted to the appropriate number of spaces to give the same alignment before the justification takes place. This is necessary as the tab expansion and alignment of text would potentially change after justify the text.

This function only works when in [Column Mode](#).

Macro Menu

Macro menu



The Macro menu includes commands that let you record, playback, load and/or save them.

What is a macro?

A macro is a series of keystrokes or actions that have been 'recorded' in the order performed. A macro allows you to 'record' a series of keystrokes or actions and then 'playback', or repeat, the recorded keystrokes and/or actions. A macro is typically used to perform a similar set of actions multiple times on each occurrence of a given text string.

UltraEdit supports multiple macros and one macro may invoke another macro.

Macros may be automatically loaded and executed when the editor starts up. See Command Line Parameters for more information.

The Macro menu offers the following commands:

Load	loads a macro file
Load and Append to Existing	loads and appends a macro file to any macros already resident
Save All	saves all resident macros to a file
Quick Record	records a macro without showing dialog for name, etc.
Set Auto Load	specifies macro to load at startup
Set Macro for File Load/Save	specifies macro to run when a file is loaded or saved
Edit Macro	edits a macro
Delete All	deletes all resident macros
Delete Macro/Modify Properties	deletes a macro or modifies its properties
Record	starts recording a macro
Stop Recording	stops recording a macro
Play Again	runs last macro again
Play Any/Multiple Times	runs any macro or runs a macro multiple times
Show Post Macro Record Options	prompts user with options after macros are recorded
Warn on Exit if Macro Not Saved	warns if macros are not saved when editing

Load command (Macro menu)



The Macro Load command presents a dialog box for loading a previously saved set of macros from a disk file. The dialog box allows selection of a file from any file device known to the system. The default filename presented is ULTRAEDIT.MAC, although any filename is acceptable.

When a file is selected for loading a check is made to ensure the file is an UltraEdit macro file.

Load and Append to Existing command (Macro menu)



This function allows a macro file to be loaded appending the file to the macros already resident, unlike the Macro [Load command](#) that deletes all resident macros before loading the file.

A dialog box is presented for loading a previously saved set of macros from a disk file. The dialog box allows selection of a file from any file device known to the system. The default filename presented is ULTRAEDIT.MAC, although any filename is acceptable.

When a file is selected for loading a check is made to ensure the file is an UltraEdit macro file.

Save All command (Macro menu)



The Macro Save All command presents a dialog for saving to a disk file the macros that have been recorded using the Macro [Record command](#). The dialog box allows selection of a file from any file device known to the system. The default filename presented is UltraEdit.MAC, although any filename is acceptable.

This command is available only if a macro has been recorded or loaded.

Quick Record command (Macro menu)



The Macro Quick Record command starts 'recording' keystrokes and actions for later playback. When this is active the menu changes to "Stop Quick Recording". Clicking on this, or using the shortcut for Macro Quick Record will stop recording the macro.

The command works in an identical manner to the Macro [Record command](#) with the exception that it does not prompt for a name, shortcut key, etc.

When this is played back, no cancel dialog will be shown. In the macro list, this will be shown with a blank name and when recorded will overwrite any existing quick macro, or macro recorded with a blank name.

The purpose of this command is to provide the same ability as recording a macro but with a single keystroke to record and stop recording.

The macro is played back by selecting the Macro [Play Again](#) or Macro [Play Any/Multiple Times command](#) as with normal macros.

Set Auto Load command (Macro menu)



The Macro Set Auto Load command allows you to specify a macro file that should be loaded automatically when UltraEdit is started.

This also allows a specific macro to be executed a specified number of times, if required. A macro file may be loaded automatically without executing a macro.

This method of loading a macro at startup overrides the Command Line parameters for loading a macro. If no macro is specified to be loaded here, the command line parameter setting, if specified, will be used.

Set Macro for File Load/Save command (Macro menu)



This command allows the user to specify a macro to be executed each time a file is loaded or saved.

The dialog box shown provides the following options:

Macro Filename

This specifies the full path and name of the macro file containing the macro(s) to be run when a file is loaded or saved. If macros are to be run when a file is loaded and saved, they must be in the same file.

Macro Name to Execute on Load

This specifies the name of the macro to run when a file is loaded. The macro if specified is run AFTER a file is loaded. It is run for each file loaded.

of Times

This specifies the number of times for each file to execute the macro specified for file load.

Macro Name to Execute on Save

This specifies the name of the macro to run when a file is saved. The macro if specified is run BEFORE the file is saved. It is run for each file saved.

of Times

This specifies the number of times for each file to execute the macro specified for file save.

Note – If the user only wants to run these macros for certain types of files the user can test in the macro for the file name or extension and take appropriate action in the macro.

Edit Macro command (Macro menu)



The Edit Macro command allows a new macro to be created and edited or an existing macro that has previously been generated with this command or with the Macro [Record command](#) to be edited.

In many cases, it is easier for the user to generate the macro by recording it with the Macro [Record command](#) and then to edit it if necessary. The learning curve is less with this approach.

Commands

A macro is a series of commands that may later be played back automatically to modify the file. The left hand side of the dialog shows the current macro being edited and an editable window containing the commands that make up the macro.

The commands in the macro must be one of the predefined commands. These predefined commands are shown on the right hand side of the dialog. If the mouse is double clicked on the command it is automatically added to the macro, or clicking on the arrow to the left of the commands will add it to the macro.

Parameters

Many of the commands have numeric parameters. The numeric parameters are quite obvious, such as "GotoLine # #" or "GotoPage #". However, some of the commands such as Find and Replace, Sort etc. have additional parameters that are not numeric. These additional parameters are listed below the command list for easy selection.

Creating a New Macro

To create a new macro from this dialog click the **New Macro** button. This displays the Macro Definition dialog allowing you to define the macro name and hot key if required.

When this new macro is created, four macro commands are automatically added to the macro:

- Insert Mode or OverStrike Mode
- ColumnModeOn or ColumnModeOff
- HexOn or HexOff
- UnixReOff or UnixReOn or PerlReOn

These four commands are added based on the current state of the editor. The purpose is to establish a known state each time the macro is played. If a macro is recorded in one mode and played back in another the results may be unpredictable. Establishing the mode at the beginning of the macro helps to ensure the expected results are achieved. These commands may be modified as required.

Once a macro has been created, it may be edited. Please note, that "^c" and "^s" may be used with many macro commands and will be replaced with the contents of the clipboard (^c) and the text currently selected (^s) when used. This allows users to record a macro that may reference a specific string and replace this with one of these two items to allow the string to be dynamically "specified" as the macro is run.

Modifying The Macro

To modify the macro, commands may be typed directly into the macro edit box or selected from the command list. Commands are not case sensitive but must be spelled exactly as shown in the command list.

The table below shows the commands that may be added to any parameters:

Commands	Parameters*	Description
""		<p>Insert text between the quotes ("").</p> <p>The closing quote must be the last non-space character on the line.</p> <p>The characters to be inserted may cross multiple lines. To insert a quote ("), place it as needed in the text being inserted and ensure the last character is also a quote. To insert just one quote character (") the line should look like this:</p> <pre>""</pre> <p>To insert a new line, add a line between the open quote and close quote. To insert two lines with 1 and 2 at the beginning of the lines the command should be as follows:</p> <pre>"1 2"</pre>
AnsiToOem		Convert file from ANSI to OEM
ASCIIToUnicode		Convert file from ASCII to Unicode
Bottom		Jump to end of file
ClearClipboard		Clear the contents of the active clipboard within UltraEdit
Clipboard	Clipboard number (0-9)	Select the specified clipboard 0 = Windows clipboard and 1-9 specifies user clipboards
CloseFile	Blank (no parameter) Save forces the file to be saved before closing NoSave closes the file without prompting if the file should first be saved	Close the active file
ColumnCenterJustify		Center justify selected columns
ColumnCut	Numeric value of columns to cut	Cut in column mode the selected columns or the specified number of columns from current cursor position to bottom of the file. Value "0" must be used to cut the selected columns.
ColumnDelete	Numeric value of columns to delete	Delete in column mode the selected columns or the specified number of columns from current cursor position to bottom of the file. Value "0" must be

		used to delete the selected columns.
ColumnInsert	String in quotes ("")	Insert string between quotes into selected columns
ColumnInsertNum	start number increment LeadingZero (optional) HEX (optional)	Insert number in selected columns. For example to insert a number starting with number 20 with an increment of -1 (= decrement by 1) with leading zeros in hexadecimal notation at the current column on every selected line or from current line to last line following command must be used: ColumnInsertNum 20 -1 LeadingZero Hex
ColumnLeftJustify		Left justify selected columns
ColumnModeOff		Turn the column mode off
ColumnModeOn		Turn the column mode on
ColumnRightJustify		Right justify selected columns
Copy		Copy selected text to the clipboard
CopyAppend		Copy selected text and append it to the clipboard
CopyFilePath		Copy the active file path/name to the clipboard
Cut		Cut the selected text from the file to the clipboard
CutAppend		Cut the selected text from the file and append it to the clipboard
DecodeBase64		Converts the selected text from Base64
DecryptFilePrompt		Presents the Decrypt File dialog prompting user for the full path of the file to decrypt and the passphrase to be used as the decryption key
DelAllBookmarkLines		Delete currently bookmarked lines in active document
Delete		Delete current character or selected text
DeleteLine		Delete the current line
DeleteToEndofLine		Delete from the current cursor position to the end of the line
DeleteToStartofLine		Delete from the current cursor position to the start of the line
DosToMac		Convert the file (line terminators) to MAC format
DosToUnix		Convert the file (line terminators) to UNIX format

DupeLine		Inserts duplicate of active line below cursor
Else		Else function for an If ... command IfSel ... Else do these commands if the IfSel failed. EndIf
EncodeBase64		Converts the selected text to Base64
EncryptFilePrompt		Presents the Encrypt File dialog prompting user for the full path of the file to encrypt and the passphrase to be used for encryption
EndIf		Ends the block of instructions controlled by the If... statements
EndLoop		End of loop - go back to beginning of the loop - see the Loop command
EndSelect		Stop selecting text (see StartSelect for details)
ExitLoop		Exit loop and go to first command after the loop (after the EndLoop command) - see the Loop command
ExitMacro		Exit macro without processing any additional commands or returning an error
Find	<p>"" (Search string in quotes)</p> <p>AllFiles indicates that search will include all open files (optional)</p> <p>Ascii indicates an ASCII search in hex mode (optional, hex mode only)</p> <p>MatchCase indicates the search is case sensitive (optional)</p> <p>MatchWord indicates the search string will only match instances that are a complete word (optional)</p> <p>PreserveCase if used the case of the replace string matches the case of the found string, only useful with a following Replace command (optional)</p> <p>RegExp indicates a regular expression search (optional)</p> <p>Up indicates search in reverse direction (optional)</p> <p>SearchInColumn indicates search should be restricted to specified columns (optional). The next two parameters must be the numbers for the starting and the ending column. The column numbering starts with "0"</p>	<p>Find the string in quotes "" based on the parameters specified.</p> <p>Some examples:</p> <p>Find a single alphabetic character in any case:</p> <p>Find RegExp "[A-Z]"</p> <p>Find a single alphabetic character in upper case only:</p> <p>Find MatchCase RegExp "[A-Z]"</p> <p>Find a single character which is NOT 'a' or 'A' or CRLF (UltraEdit syntax):</p> <p>Find RegExp "[~a^p]"</p> <p>Find a single character which is NOT 'a'</p>

	<p>(start of line). The ending column can have the value "-1" which means end of line.</p> <p>Select indicates the text should be selected from the current selection/position to the end of the search target</p> <p>SelectText indicates that search will be restricted to currently selected text (optional)</p>	<p>or LF (Unix/Perl syntax):</p> <p>Find MatchCase RegExp "[^a\n]"</p>
FindInFiles	<p>"" (Initial Directory string in quotes)</p> <p>"" (In Files/Types string in quotes)</p> <p>"" (Search string in quotes)</p> <p>DisplayLinesDoNotMatch limits output to lines not matching the search string (optional)</p> <p>FavFiles if used the search is restricted to files specified under Favorite Files in the File menu (optional)</p> <p>IgnoreHiddenSubDirs if used the search will skip over subdirectories defined as "hidden" in the Windows file system (optional)</p> <p>MatchCase indicates the search is case sensitive (optional)</p> <p>MatchIfNotFound if used the search will return files which do NOT include the specified search string (optional)</p> <p>MatchWord indicates the search string will only match instances that are a complete word (optional)</p> <p>OpenFiles indicates the search should be limited to files currently opened for editing (optional)</p> <p>OutputWin if used returns output of search to output window; if not used output is returned to a new edit window (optional)</p> <p>ProjFiles if used the search is restricted to files specified as part of the active project (optional)</p> <p>Recursive if used the search will continue through all subdirectories of the specified search directory (optional)</p> <p>RegExp indicates a regular expression search (optional)</p> <p>SolFiles if used the search is restricted to files specified as part of the active solution (optional, only UESTudio)</p> <p>Unicode if used the specified search string will be interpreted as multi-byte Unicode string and a Unicode find is</p>	<p>Searches through specified files to find the string in quotes "" based on the parameters specified. The format must be as follows:</p> <p>FindInFiles BoldParameters "Initial Directory" "Files/Types" "Search String"</p> <p>For example:</p> <p>FindInFiles OutputWin "c:\temp\test\" "*.html" "home"</p> <p>The Recursive parameter is ignored when the search is in all open files, project files or solution files.</p> <p>The initial directory and the file/type arguments should be empty strings "" when using FavFiles, OpenFiles, ProjFiles or SolFiles.</p>

	executed (optional) UseEncoding if used the specified number will be used to determine the encoding used for the search string (optional)	
FromEBCDIC		Convert text from EBCDIC format
GetString	String in quotes (") used in the prompt dialog	Prompts the user for a string that is inserted in the file at the current location. The "prompt" is the prompt or question that is displayed to the user when the macro is run GetString "What country do you live in?"
GetValue	String in quotes (") used in prompt dialog	Prompts the user for a value that is inserted in the file at the current location. The "prompt" is the prompt or question that is displayed to the user when the macro is run. GetValue "How old are you?"
GotoBookMark	Index of bookmark to jump to or -1 to go to next bookmark	Jump to the next/specified bookmark. The indexes start with 0. If a user enters an index that is greater than the actual number of bookmarks then it is automatically directed to the first bookmark (index 0).
GotoBookMarkSelect	Index of bookmark to jump to or -1 to go to next bookmark	Jump to the next/specified bookmark and select text from cursor position to bookmark. The indexes start with 0. If a user enters an index that is greater than the actual number of bookmarks then it is automatically directed to the first bookmark (index 0).
GotoLine	Numeric value of line and column number to jump to	Jump to the specified line and column number. Use line number 0 to jump to the specified column in the current line.
GotoLineSelect	Numeric value of line and column number to jump to	Jump to specified line number and column number and select text from cursor position to line/column. Use line number 0 to jump to the specified column in the current line while selecting text.
GotoPage	Numeric value of page to jump	Jump to specified page number
GotoPageSelect	Numeric value of page to jump	Jump to specified page number and select text from cursor position to page number
HexDelete	Numeric value specifying number of bytes to delete	Delete the specified number of bytes from the file

HexInsert	Numeric value specifying number of bytes to insert	Insert the specified number of bytes (spaces) into the file
HexOff		Turn Hex mode off - switch to text mode
HexOn		Turn Hex mode on
HideShowCollapseAll		Collapse all foldable lines in active document
HideShowExpandAll		Expand all folded/collapsed lines in active document
HideShowSelection		Hide the selected lines, or if hidden, show the lines hidden at the line of the cursor
IfCharGt	"character" or number	<p>This checks if the current character at the cursor position is greater than the specified character or number specified. The check against a number is not to check the value of a string but the value of the underlying character.</p> <p>IfCharGt " " ... do these commands if the current character is greater than a space. ... EndIf</p> <p>IfCharGt 32 ... do these commands if the current characters decimal value is greater than 32 - a space. ... EndIf</p>
IfCharIs	"string" or number	<p>This checks if the current character at the cursor position is in the specified string, or matches the decimal number.</p> <p>IfCharIs "xyz" ... do these commands if the current character is x, y, or z. ... EndIf</p> <p>IfCharIs 34 ... do these commands if the current character is decimal value 34 - a double quote. ... EndIf</p>
IfColNum	number	This checks if the current cursor position is the specific column

		<p>number.</p> <p>IfColNum 10 ... do these commands if the current cursor position is column 10. Column 1 is the first column. ... EndIf</p>
IfColNumGt	number	<p>This checks if the current cursor position is greater than the specific column number.</p> <p>IfColNumGt 10 ... do these commands if the current cursor position is greater than column 10. Column 1 is the first column. ... EndIf</p>
IfEof		<p>This checks if the current cursor position is at the end of file.</p> <p>IfEof ... do these commands if at end of file. ... EndIf</p>
IfExtIs	"string"	<p>This checks if the file extension of the active file matches the specified string. If the extension matches, the condition is satisfied and the macro will execute the steps following the IfExtIs statement.</p> <p>IfExtIs "txt" ... do these commands if the current file extension is txt. ... EndIf</p>
IfFound		<p>This checks the results from the last find command in the macro and will execute the next set of instructions up to an "EndIf" if the last search was successful:</p> <p>Find "something" IfFound ... do these commands if "something" was found. ... EndIf</p> <p>NOTE - To avoid a "Not Found" message when running the macro,</p>

		make sure the "Continue if a Find with Replace Not Found" check box is set for the macro properties.
IfFTP		<p>This checks if the current file is a file loaded via FTP/SFTP and not a local/network file.</p> <p>IfFTP ... do these commands if the file is an FTP loaded file. ... EndIf</p>
IfNamels	"string"	<p>This checks if the active filename (not path or extension) matches the specified string. If the name matches, the condition is satisfied and the macro will execute the steps following the IfNamels statement.</p> <p>IfNamels "test" ... do these commands if the current filename is test (irrespective of the file extension). ... EndIf</p>
IfNotFound		<p>This checks the results from the last find command in the macro and will execute the next set of instructions up to an "EndIf" if the last search was NOT successful:</p> <p>Find "something" IfNotFound ... do these commands if "something" was NOT found. ... EndIf</p> <p>NOTE - To avoid a "Not Found" message when running the macro, make sure the "Continue if a Find with Replace Not Found" check box is set for the macro properties.</p>
IfSel		<p>This checks if there is currently any selected text within the active file.</p> <p>IfSel ... do these commands if "something" is selected. ... EndIf</p>
InsertLine		Inserts blank line below current cursor position

InsertMode		Change text editing mode to insert mode for characters typed
InsertPageBreak		Insert a form feed/page break character at the current cursor position in the file
InvertCase		Invert the case of selected text
Key	BACKSPACE DEL DOWN ARROW END HOME LEFT ARROW PGDN PGUP RIGHT ARROW UP ARROW Ctrl+END Ctrl+HOME Ctrl+LEFT ARROW Ctrl+RIGHT ARROW	Insert a key command into the macro. Generally used for navigation in the file and for backspace or delete. The "Ctrl+" modifier may be used as with normal editing to modify the command. For text input the "" string command is used, not the Key command.
Loop	#	Loop # times where # is a number. A value of 0 indicates Loop forever. For example: Loop 10 ... do these commands 10 times ... EndLoop If the Loop command exists, the application checks for ExitLoop or ExitMacro until an EndLoop command is reached. If neither of these exist, the macro loop will be limited to one iteration only.
MatchBrace		Find matching brace and select the text between them
MoveLineDown		Moves active line down one line from current position
MoveLineUp		Moves active line up one line from current position
NewFile		Open a new blank file
NextDocument		Switch to next file tab document
NextWindow		Switch to next window
OemToAnsi		Convert file from OEM to ANSI
Open	Filename in quotes ("")	Open the specified file. The filename must be in quotes. "^s" may be used in the filename. UltraEdit will replace this with the currently selected text in the active

		window. Likewise, "^c" may be used and UltraEdit will replace this with the contents of the clipboard.
OverStrikeMode		Change text editing mode to overstrike mode for characters typed
Paste		Paste the contents of the clipboard into the file
PerlReOn		Switch regular expressions to Perl compatible style regular expressions
PlayMacro	Numeric value specifying number of times to play it Macro name (") in quotes	Play another macro. Specify the number of times to play the macro first, then the macro name. The macro name must be in quotes and is case sensitive and the macro must already exist.
PreviousBookmark		Jump to the previous bookmark
PreviousBookmarkSelect		Jump to the previous bookmark and select text from cursor position to bookmark
PreviousDocument		Switch to previous file tab document
PreviousWindow		Switch to previous window
ReIndentSelection		Re-indents currently selected text in active file
Replace	All Replace all occurrences. AllFiles Replace in all open files (requires All). SelectText Replace in selected text (Requires All). Replace String in quotes (") - String to replace the found string	The Replace command replaces the previously found string with the string specified. This command MUST follow a Find command. As with the Find string, the replace string must be in quotes. The closing quote must be the last non-space character on the line. The replace string may cross multiple lines however, if the last non-space character in any line is a quote it will terminate the string. All special characters permitted by the Replace Command are permitted here.
ReplInFiles	"" (Initial Directory string in quotes) "" (In Files/Types string in quotes) "" (Search string in quotes) "" (Replace string in quotes on the same line as end of search string) FavFiles if used the search is restricted to files specified under	Searches through specified files to find the string in quotes "" based on the parameters specified and replace this as specified. The format must be as follows: ReplInFiles BoldParameters "Initial

	<p>Favorite Files in the File menu (optional)</p> <p>IgnoreHiddenSubDirs if used the search will skip over subdirectories defined as "hidden" in the Windows file system (optional)</p> <p>Log creates a list of changed files in the output window (optional)</p> <p>MatchCase indicates the search is case sensitive (optional)</p> <p>MatchWord indicates the search string will only match instances that are a complete word (optional)</p> <p>PreserveCase if used the case of the replace string matches the case of the found string (optional)</p> <p>ProjFiles if used the search is restricted to files specified as part of the active project (optional)</p> <p>Recursive if used the search will continue through all subdirectories of the specified search directory (optional)</p> <p>RegExp indicates a regular expression search (optional)</p> <p>SolFiles if used the search is restricted to files specified as part of the active solution (optional, only UESudio)</p> <p>Unicode if used the specified search and replace string will be interpreted as multi-byte Unicode strings and a Unicode replace is executed (optional)</p> <p>UseEncoding if used the specified number will be used to determine the encoding used for the search string (optional)</p>	<p>Directory" "Files/Types" "Search String" "Replace String"</p> <p>For example:</p> <p>ReplInFiles "c:\temp\test\" "*" .html" "line x line y" "line y line x"</p> <p>The Recursive parameter is ignored when the search is in all open files, project files or solution files.</p> <p>The initial directory and the file/type arguments should be empty strings "" when using FavFiles, OpenFiles, ProjFiles or SolFiles.</p>
ReturnToWrap		Convert hard returns to word wrap
RunTool	String in quotes ("") specifying the case sensitive menu name of tool to run	Run a tool. The tool must be configured from Tool Configuration in the Advanced menu .
Save		Save active file
SaveAll		Save all open files
SaveAs	Filename in quotes ("")	Save the active file as the filename specified. The filename must be in quotes. "^s" may be used in the filename. UltraEdit will replace this with the currently selected text in the active window. Likewise, "^c" may be used and UltraEdit will replace this with the contents of the clipboard.

SelectAll		Select all text in the file
SelectLine		Select all text on active line
SelectToBottom		Select all text from the current position to the end of file
SelectToTop		Select all text from the current position to the top of file
SelectWord		Select the current word (same as double clicking a word)
SortAsc	<p>IgnoreCase Ignore case of characters in the sort.</p> <p>RemoveDup Remove duplicate lines.</p> <p>RemDupByAllKeys Remove duplicate lines where all specified keys match.</p> <p>RemDupByAnyKeys Remove duplicate lines where any specified keys match.</p> <p>RemKey1 Sort columns specified for key 1 used when removing duplicates.</p> <p>RemKey2 Sort columns specified for key 2 used when removing duplicates.</p> <p>RemKey3 Sort columns specified for key 3 used when removing duplicates.</p> <p>RemKey4 Sort columns specified for key 4 used when removing duplicates.</p> <p>Numeric Sort based on numeric value, not character order. Cannot be used with Alternate or Locale sort.</p> <p>Alternate Specifies that the sort should use the alternate sort method. With the alternate sort method only a single sort key may be used.</p> <p>Locale Specifies that the sort should be locale specific. Cannot be used together with the Alternate parameter.</p> <p>Pairs of sort keys (numeric values), each value is separated by a space.</p>	<p>Sort the file, or selected text in ascending order.</p> <p>For example:</p> <p>SortAsc IgnoreCase RemoveDup Numeric 1 -1 0 0 0 0 0</p>
SortDes	<p>IgnoreCase Ignore case of characters in the sort.</p> <p>RemoveDup Remove duplicate lines.</p> <p>RemDupByAllKeys Remove duplicate lines where all specified keys match.</p>	<p>Sort the file, or selected text in descending order.</p> <p>For example:</p> <p>SortDes RemoveDup Locale 0 -1 0 0 0 0 0</p>

	<p>RemDupByAnyKeys Remove duplicate lines where any specified keys match.</p> <p>RemKey1 Sort columns specified for key 1 used when removing duplicates.</p> <p>RemKey2 Sort columns specified for key 2 used when removing duplicates.</p> <p>RemKey3 Sort columns specified for key 3 used when removing duplicates.</p> <p>RemKey4 Sort columns specified for key 4 used when removing duplicates.</p> <p>Numeric Sort based on numeric value, not character order. Cannot be used with Alternate or Locale sort.</p> <p>Alternate Specifies that the sort should use the alternate sort method. With the alternate sort method only a single sort key may be used.</p> <p>Locale Specifies that the sort should be locale specific. Cannot be used together with the Alternate parameter.</p> <p>Pairs of sort keys (numeric values), each value is separated by a space.</p>	
SpacesToTabs		Convert (leading) spaces within the file to tabs
SpacesToTabsAll		Convert (all) spaces within the file to tabs
StartSelect		Start selection. This turns the selection mode on. Any cursor movement or positioning will be with selection on and the text is selected. EndSelect will stop the selection mode. The selected text will remain selected until another command causes it not to be selected as with normal editing.
TabsToSpaces		Convert all tabs within the file to spaces
Template	Template number (#)	Insert the specified template into the file
TimeDate		Insert the time and date into the file at the current location
ToCaps		Capitalize each word in the selected text

ToEBCDIC		Convert text to EBCDIC format
ToggleBookmark		Set or remove a bookmark at the current line
ToLower		Convert the selected text to lower case
Top		Jump to top of file
ToUpper		Convert the selected text to upper case
TrimTrailingSpaces		Trim trailing spaces from each line of current file
UltraEditReOn		Switch regular expressions to UltraEdit style regular expressions (not Unix)
UnicodeToASCII		Convert file from Unicode to ASCII
UnixMactoS		Convert the active file (line terminators) from Mac/Unix to DOS format
UnixReOn		Switch regular expressions to Unix style regular expressions
WrapToReturn	Column number (#) at which wrap is performed. Zero for the column number indicates the wrap is to occur at the window boundary.	Convert the file from word wrap to hard returns
XMLConvertToCRLF		Converts single-line XML text to indented XML format
	* Where (#) exists in the parameter column a numeric value is to be entered for that parameter. Likewise, where (") exists in the parameter column a string is to be entered between quotes. The text only describes the meaning of the parameter. All string parameters must be the last parameter.	

Delete All command (Macro menu)



This command deletes all the resident macros. If any of the macros have been created or modified since the file was saved, the user is prompted allowing the macros to be saved prior to deleting them.

Delete Macro/Modify Properties command (Macro Menu)



This command allows the user to delete a macro or modify the hot key associated with any macro. A dialog box is presented that shows all of the macros, and allows any macro to be selected and its hot key modified (or the macro may be deleted). No support is provided to modify the contents of a macro. For more details on the hot key assignments see Macro [Recording](#).

Record command (Macro menu)



The Macro Record command starts 'recording' keystrokes and actions for later playback. Recording is stopped when the [Stop Recording command](#) is selected.

This command presents the Macro Definition dialog allowing you to define the macro name and hot key if required. When the dialog is dismissed with the OK button the macro recording starts.

Notes for recording macros:

The Macro Record function records all keystrokes and characters typed while recording is selected. All keystrokes are recorded, including key combinations that move to end of line, top or bottom of document etc. Care should be taken to ensure that when the macro is to be [played again](#) the same keystrokes are appropriate. When a macro is being recorded, a **Macro Recording** notification will be shown in the second cell of the status bar.

Mouse motion and scrolling is NOT recorded while recording is selected. All positioning should be performed with the keyboard.

Macros will record and playback the [Next Window](#) and [Previous Window](#) commands.

[Find](#) and [Replace](#) are supported while recording is selected (see [Find](#) for special characters that are useful with macros). Any find or replace functions will be recorded directly as performed, unless using special characters that are replaced when executing with the contents of the clipboard or the highlighted text, see [Find](#). If all, or any occurrences of "FIND" are replaced with "REPLACE" in a file, the same action will be performed when the macro is [played again](#).

[Cut](#), [Copy](#) and [Paste](#) are also supported while macro recording is selected. Any of these operations will be performed dynamically as the macro is [played again](#). That is to say that if an item is cut while recording the macro, an item will be cut while playing back the macro based on the selected text at the time the macro is being played back. If the selected text is different while playing back the macro, from when the macro is recorded, the text cut while playing back the macro will also be different.

As multiple macros are now supported, this command is available when macros already exist.

[Hex editing](#) is now supported within macros.

The following commands are supported when recording a macro:

Keyboard entry - typing characters

Cursor positioning

Selecting text with keyboard commands

Cut, copy, paste

Find/replace

HEX mode switching and editing

File open, close, save

Bookmarks

Playing macros - nesting macros

Window switching - next/previous window

Insert/Overstrike selection

Unix/Mac to DOS

DOS to Unix

DOS to MAC

Convert Wrap to CR/LF's

Convert CR/LF's to Wrap

Stop Recording command (Macro menu)



The Macro Stop Recording command stops capturing keystrokes and actions that was started when the Macro [Record command](#) was selected. Once a macro has been recorded and the session has been stopped with this command, the macro may be played back by selecting the Macro [Play Again](#) or Macro [Play Any/Multiple Times command](#).

After a macro has been recorded, it may be saved using the Macro [Save All command](#).

Play Again command (Macro menu)



The Macro Play Again command allows a macro that has been recorded with the Macro [Record command](#) to be replayed. The macro is replayed one time for each invocation of the command. To replay the macro multiple times use the Macro [Play Any/Multiple Times command](#).

This command is not available if a macro record has not been recorded or loaded.

Play Any/Multiple Times command (Macro menu)



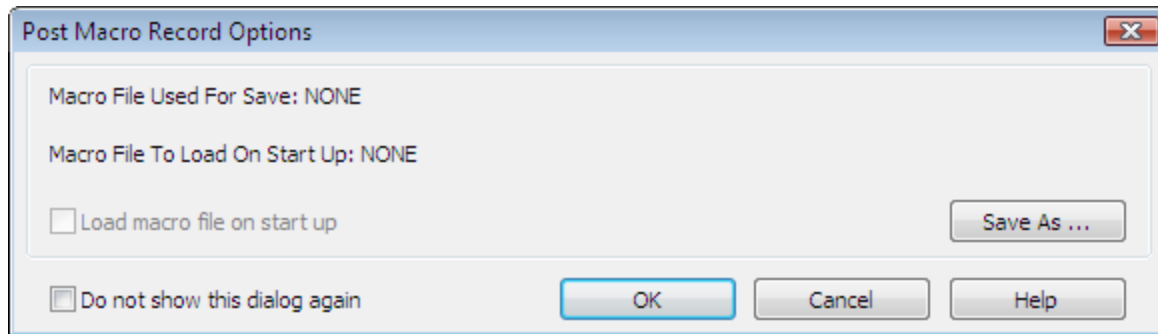
The Macro Play Any/Multiple Times command allows a macro that has been recorded with the Macro [Record command](#) to be replayed multiple times. A dialog box is presented that prompts the user for a number between 1 and 100000 that specifies the number of times the macro should be replayed. To replay the macro a single time use the Macro [Play Again command](#).

This command is not available if a macro record has not been recorded or loaded.

Post Macro Record Options dialog



The **Post Macro Record Options** dialog facilitates the use of options available to the user after a macro has been recorded.



Macro File Used For Save

If a macro is loaded while recording a macro, the currently loaded macro path and name will be reflected here. In this case, a **Save** button would be displayed in the dialog. If no macro is loaded, pressing the **Save As** button will present a **Save As** dialog where the user can specify the path and name the macro just recorded should be saved under. Once a macro save file is specified, the option to **Load macro file on start up** will be active.

Macro File To Load On Start Up

If a macro is currently specified under **Set Auto Load** to be loaded for use on startup, the path and name of the macro will be indicated here. If the user desires, the macro being saved may be specified to be loaded for use on startup. This would replace the previously specified macro with the macro just recorded.

Load macro file on start up

If this option is selected, the macro just recorded would be loaded when the editor is started.

Do not show this dialog again

If this option is selected, the **Show Post Macro Record Options** option will be unchecked in the Macro menu and the dialog would not be shown when future macros are recorded.

Warn on Exit if Macro not Saved command (Macro menu)



This indicates and determines if a warning will be issued when UltraEdit closes, or a new macro file is loaded and the existing macros have been modified but not saved.

If this is not checked, no warning will be given.

Scripting Menu

Scripting menu



The Scripting menu offers the following commands:

Scripts	opens the Script dialog where scripts may be added for execution
Script List	shows or hides listing of all currently specified scripts
Run Active Script	initiates JavaScript file currently being edited. If the active file is a JavaScript file, this option will dynamically be shown in the Scripting submenu. A keyboard shortcut may also be defined for this under Key Mapping in Configuration.

Advanced Menu

Advanced menu



The Advanced menu provides access to some of the advanced capabilities of UltraEdit and UESudio:

Configuration	presents Configuration dialog with index tree to access categorized configuration options
Scripting	presents submenu with "Scripts..." heading (Basic menu). Clicking on this opens the Configuration dialog to the Scripting topic. A list of currently added scripts is dynamically shown below this.
Run Active Script	initiates JavaScript file currently being edited (Basic menu). If the active file is a JavaScript file, this option will dynamically be shown in the Advanced submenu. A keyboard shortcut may also be defined for this under Key Mapping in Configuration.
Import Settings	presents dialog allowing user to select INI file from which to import settings
Export Settings	presents dialog allowing user to select folder to export settings to
Backup/Restore User Customizations	presents dialog allowing user to select menu/toolbar customization settings to backup or restore
Open New Instance of UltraEdit/UESudio	launches a separate instance of UltraEdit/UESudio
Set Code Page/Locale	sets code page or locale
DOS Command	runs a DOS command and captures its output
Last DOS Command	repeats the last DOS command

Run Windows Program	runs a Windows program
Alternate Capture Method	toggles alternate method for capturing output
Display/Modify Templates	presents dialog for displaying/modifying user-definable templates
Tool Configuration	presents dialog for user-definable tool configuration
Project Tool Configuration	presents dialog for user-definable tool configuration for active project

Configuration

Configuration Summary



When Configuration is selected in the [Advanced menu](#), a Configuration dialog is presented with configuration options divided by category. These may be easily navigated using the index tree to the left of the dialog. Clicking on any of the major headings will expand that section and subheading may be selected to access related options.

To read about what each setting does, please click on the Contents page in the Help and double click on Menu Summary and then double click on Advanced menu and double click on Configuration. This will open a mirror of the index tree in the configuration dialog in the Help files. Users may expand the major sections in the Help:

Editor

Search

Spell Checker

File Handling

File Associations

File Types

Editor Display

Application Layout

Toolbars/Menus

Directories

Key Mapping

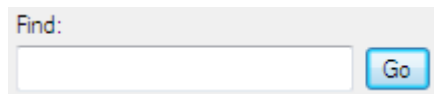
FTP/SFTP

IDE

Double-clicking on a heading and clicking on the topics included below will provide summaries of the associated options.

Configuration Navigation

All of the configuration dialogs include a natural language navigation textbox to the lower left of the dialog as shown below:



Users may enter search strings here for items they hope to find in configuration. The editor will then search for a match and if one is found, the associated configuration dialog will automatically be displayed.

* Please note: The IDE configuration items are only supported in UESstudio, and not in UltraEdit.

Editor

Advanced (Configuration - Editor)



The Advanced branch under Configuration - Editor has the following options:

Allow low ASCII values to be entered (usually control codes)

This setting allows the input of "low values" to be entered such as control codes while editing. Normally these would be ignored.

Allow editing of text files with HEX 00's without converting them to spaces

When a file is loaded the editor checks the first 7K of the file and if it contains more than two NULLS (HEX 00) the file is considered a binary/HEX file. NULLS are not valid in an ASCII file and will be converted to spaces if you edit the file in text mode if this option is not selected (default). If you view a HEX file in text mode but make no changes the editor should not convert the NULLS to spaces, but if you do make changes in TEXT mode to a HEX file the change will occur.

Selecting this option overrides this conversion of NULLS (HEX 00) to spaces but we still recommend NOT editing binary files in TEXT mode. NULLS are normally considered string terminators and not generally valid in ASCII files. Using this setting may cause some functions not to work past the NULL although this should not effect general editing.

Load/Restore printer settings

The editor tries to communicate with the printer driver when it loads and shuts down (to preserve settings). If a network printer is specified as the default printer and not available this may cause a delay on startup while the editor tries to find the printer on the network. If this option is not selected then the editor does not communicate with the printer.

ESC key hides output window when visible

If this option is selected the output window may be dismissed by pressing the Escape key if the output window has focus.

Grouped undo

If this option is selected undo will be executed on a word-by-word rather than character-by-character basis. If this setting is toggled during the editing of a document (and the undo buffer is partially created with both settings) the undo functionality will reflect the settings in place at the time the document was created. This could cause part of the text in a file to be undone as grouped characters and part to be undone character-by-character.

Auto-Save modified documents when application loses focus

If this option is selected all modified documents will be saved when focus shifts to another application.

Auto-complete (Configuration - Editor)



The Auto-complete branch under Configuration - Editor has the following options:

Do not auto-complete structure members (items following "." or "->")

This setting causes the editor not to auto-complete keywords following a "." or "->". This was configured as the default due to issues faced particularly by Java programmers using Auto Complete. If this setting is not selected keywords following these items would be auto corrected.

Show auto-complete dialog automatically

This setting causes the auto-complete popup-dialog to be shown automatically as the users types instead of waiting for it to be invoked specifically by the user. The default setting is for this not to be shown (unchecked). If desired, the user may specify a minimum number of characters that must be typed before the auto-complete dialog is displayed.

Bookmarks (Configuration - Editor)



The Bookmarks branch under Configuration - Editor has the following options:

Prompt for name when adding bookmark

If this option is selected the **Add Bookmark** dialog is presented when a bookmark is added, prompting the user to enter a name for the added bookmark.

Display name when hovering over bookmark

If this option is selected a tooltip containing the bookmark name and the line number on which the bookmark is located will be displayed when the cursor hovers over the bookmark.

Highlight entire line

If this option is selected, setting a bookmark in a file will cause the entire line to be highlighted rather than just the associated line number. If desired, users may select this option along with **Disable line numbers** under **Editor Display -> Miscellaneous** to remove the line number gutter.

Bookmark column with line

If this option is selected the column the cursor is located in at the time the line is bookmarked is stored. Jumping to a bookmark will position the cursor accordingly. If this option is not selected, the cursor will be positioned at the beginning of the line when moving to a bookmark.

Save Bookmarks

This option allows users to save bookmarks for any file that has active bookmarks when a file is closed. By default the feature is not enabled. If this option is disabled the editor will delete any saved bookmarks the next time the editor is closed.

Column Mode (Configuration - Editor)



The Column Mode branch under Configuration - Editor has the following options:

No overstrike of unselected columns

If you never want overstrike mode to be used with column operations, you may select this option.

Overstrike if in overstrike mode

If this option is selected, the column operations operate either in insert mode (as default) or overstrike/overpaste mode based on the insert/overstrike keyboard setting. This is toggled by the Insert Key and shows up on the status bar. If in overstrike mode, cut/delete operations will remove the contents but leave spaces so that column data is not shifted. Paste operations will paste over the current position data and not shift columns either.

Always overstrike unselected columns

If overstrike mode is what you require permanently for column mode, you may select this option.

Delimiters (Configuration - Editor)



The Delimiters branch under Configuration - Editor has the following options:

Delimiter character for line/column designator on command line: /

This setting determines the character following a filename that allows the user to add / xxx to go to a line number or column number. If this character conflicts with a file or path name it can be modified to a character that is not likely to appear in a file or path name.

Left delimiters for CTRL+Dbl Click: {}<>"

Right delimiters for CTRL+Dbl Click: {}<>"

Selection of text by double clicking the primary mouse button (usually the left button) is supported by default. Normally a single word is selected made up of any alphanumeric character or underscore '_'.

If the control key is pressed at the time the primary mouse button is double clicked, a string is selected rather than just a single word. This can be very useful in selecting filenames, email addresses, etc. The characters that delimit the string are configurable with the two options above. This function works on a single line only.

These entries, one for the left edge of the text to be selected and one for the right edge, specify what characters should be used to stop any further text from being selected. If a space only is specified in each field all text either side of the cursor position will be selected until a space is encountered.

If text is selected when the right mouse button is pressed, a new menu item "Open File Highlighted" is added to the top of the pop-up menu and if selected the file is opened if it exists.

Miscellaneous (Configuration - Editor)



The Miscellaneous branch under Configuration - Editor has the following options:

Disable drag & drop

Drag and drop editing allows the user to select text in the normal manner and then copy or move the selected text by dragging it to a new location with the mouse. No keyboard interaction is required. If this option is selected it will disable the [Drag and Drop Editing](#) feature.

Lock insert mode key

This setting causes the editor to lock the Insert key in Insert mode and not allow a toggle to Overstrike mode. If this is set this will be true for normal text editing as well as Column mode. The default setting is for this option not to be active.

Home key always goto column 1

When checked, the home key always positions to the first column. If not checked, the home key will go to the first non-space character unless in that position, in which case it will go to the first column

Disable left cursor from going to previous line

This setting causes the editor NOT to move up to the previous line if the cursor is at the start of a line and the left arrow key is pressed. If this setting is not selected (default) the editor will allow the cursor to move to the end of the previous line.

Cursor off selection is from current position

If this option is selected, using an arrow key to move off of a selection will dismiss the selection and move the cursor based on its current position. If this option is not selected, the cursor would always be moved from the end of the selection.

Allow selection of line with mouse in margin

This setting determines if the editor will allow line selection by clicking/dragging the mouse in the left margin of the edit window. The default setting is checked, allowing the selection.

Enable copy/cut of current line when no selection is active

This setting causes the editor to copy/cut the active line to the active clipboard when the appropriate menu item is selected or the keyboard shortcut is used if there is no selected text. The default setting is for this option to be active.

Automatically copy to clipboard when selection is made

If this option is selected, any selected text will automatically be copied to the clipboard upon selection, overwriting the current contents of the active clipboard.

Don't move focus to function list on refresh

If this option is selected, focus will not be moved from the active document to the function list automatically when the function list is refreshed (e.g. when F8 is pressed).

New File Creation (Configuration - Editor)



The New File Creation branch under Configuration - Editor has the following options:

Create a new file if file specified on command line does not exist

This setting instructs the editor to create a new file if the file specified to be opened on the command line does not exist. The default setting is to create the new file.

Create new Edit file when opening with no other files

This setting instructs the editor to create a new empty document (EDIT1) file when it starts up if no other files are being opened.

Encoding Type:

Create new files as ANSI

Create new files as UTF-8

Create new files as UTF-16

This setting causes the editor to create all new files in the selected encoding regardless of the file type (DOS/UNIX/MAC) configured for new files specified under [DOS/UNIX/MAC Handling](#) in the **File Handling** section.

Word Wrap/Tab Settings (Configuration - Editor)



The Word Wrap/Tab Settings branch under Configuration - Editor has the following options:

Select extension for settings or default

The editor allows the tab and Word-Wrap settings to be based on the file extension of the file. Initially the defaults will be used. To add an extension type click on Change List and a dialog will be displayed allowing you to add or edit up to 10 extension types. Users may select or add a file extension (or Default) to which all settings specified on this page will be applied. The tab and Word Wrap settings will change to reflect the extensions shown. The "default" settings are used for any files that do not match the user defined extensions. Periods **must not** be included when specifying extensions for word wrap settings. For example, "c cpp h" is correct, but ".c .cpp .h" is not.

Auto-complete file

If desired, the path to an auto-complete file specific to the selected file extension may be specified here. Users may enter the path directly or browse to the desired file.

Use spaces in place of tabs

If this item is selected the editor will replace TAB characters with spaces and the Indent Spaces value specified below will determine the number of spaces to use.

The editor allows for selection of TAB stops and for TAB characters to be replaced by spaces automatically while editing. The Tab Stop value specifies the TAB stops, while the Indent spaces specifies the number of spaces to use in place of a TAB character. This allows the user to have one setting for tabs and another for indentation.

The Indent spaces is how many spaces are used for an indent when the TAB key is used **ONLY WHEN** Use spaces in place of tabs is set. In this case the Tab Stop value tells the editor to expand actual tab characters on the appropriate tab stop boundary.

If Use spaces in place of tabs is **NOT** checked, pressing the TAB key will always insert a TAB character and it will be expanded per the Tab Stop value as indicated earlier.

The user may define multiple TAB stop values (up to 12) in comma-delimited format, i.e. "4,4,8". The last value defined will be used for subsequent TABS. In the example above, the first two TABS would be equal to four spaces and all subsequent TABS would be equivalent to 8 spaces.

If the option to replace TABs with spaces is chosen, the editor inserts spaces in place of TABS but maintains the TAB stop spacing using the number of spaces specified. This applies to TAB characters entered after this setting is changed. Existing TAB characters in the document are not affected by these settings.

Font

Users may specify a font to be used for files with the currently specified extension by pressing the Font button and selecting the desired font.

Wrap method

Word wrap may be used to automatically wrap words to the next line at the vertical edge of the active window or at a specified column.

There are four selectable modes of operation for word wrap:

Wrap at window boundary, without inserting a hard return.

Wrap at a specified column number without inserting a hard return.

Wrap at a specified column number and insert a hard return.

With this option, an additional option specifies if the file should be automatically wrapped on input. If selected, the file is wrapped on input, otherwise text is only wrapped as it is modified.

Note - The column wrap above wraps on a word boundary. This may mean that the wrap does not occur on the specified column but on a lesser column number.

Absolute wrap after specified column. This will wrap exactly at the specified column.

If the option to wrap at a specified column AND insert a hard return is made, the file is automatically modified to wrap at the column and insert appropriate hard returns.

Additionally, the [Reformat Paragraph](#) feature may be used to wrap a paragraph to the specified column number.

If the editor active window is in [Hex mode](#) this command is not available.

Search

Advanced (Configuration - Search)



The Advanced branch under Configuration - Search has the following options for Find in Files and Replace in Files:

Use active file path for default directory

This setting causes the editor to use the active file path for setting the Find/Replace in Files default directory. By default the editor uses the previous path used in the dialog for this.

Miscellaneous (Configuration - Search)



The Miscellaneous branch under Configuration - **Search** has the following options:

Show slim find dialog after start

This option determines if the slim Find dialog with Find Next and Find Previous buttons should be shown after the find command is initially started. If this is selected, the dialog is shown at the top of the screen.

Continue find at end of file

This option determines if the search for a string is to be continued at the opposite end of the file if the string is not found. If this option is not set, the editor does NOT wrap around at the end of file when searching for a string. The default behavior is for the editor to continue the search from the opposite end of the file when a target is not found.

Find dialog defaults to word under cursor

If this option is selected the editor automatically initializes the find/replace dialogs with the word under the cursor position. If not set, it uses selected text if present or the last find string.

Find next/prev finds selected text (not last searched for text)

If this option is selected the editor will find the selected text when a FIND NEXT/PREVIOUS command is performed instead of the previous search string UNLESS the Regular Expressions option is selected.

Set focus to edit window when going to result of lines containing string dialog

This setting causes the editor to set focus to the edit window rather than the List Lines Containing String dialog when the result is double clicked in the List Lines Containing String dialog. By default this is set but some users requested that focus remain on the List Lines Containing String dialog in this case.

Automatically bookmark current line on find

If this option is selected a bookmark will automatically be inserted at the current cursor location when a matching string is found and moved to so that the user may return to the original cursor position.

Fill Search In Columns with selection points

If this option is selected, and text is selected in column mode, the text fields for the "Search In Column" option will automatically be populated based on the boundaries of the existing selection.

Special character used in Find/Replace strings (normally ^): ^

In the find/replace strings a caret "^" (by default) has special meaning (see help on Find). This setting allows a different character to be used in Find/Replace routines from the caret "^". It is recommended a pipe character "|" or similar uncommon character be used.

Auto Reset Settings (Configuration - Search)



The Auto Reset Settings branch under Configuration - **Search** has the following options:

Items selected in this dialog will be reset (or unselected) after **Find** or **Replace** operation is completed if they were selected for the last executed Search. The items that may be specified to be automatically reset are:

Highlight All Items Found

This option causes all items matching the search string to be highlighted (shown as selected text) when the **Find Next** button is pressed.

List Lines Containing String

This option causes UltraEdit to search for all occurrences of the string in the active document and to list them in a dialog box. By double-clicking on one of the result lines (or selecting it and clicking on Goto) UltraEdit will position to that line in the file.

Match Case

By default, the search will not be case sensitive. The option to make the search case sensitive is given. If a search has already occurred in the document, the default selection for case sensitivity is that of the previous search.

Match Whole Word Only

By default, the search will not be restricted to matching whole words only. This option allows the search to be restricted to whole words only. This is useful for searching for 'a', or 'A' without finding all words including 'a'. If a search has already occurred in a document, the default selection will match the previous search.

Regular Expressions

If this option is selected the search will be executed using the regular expression type selected under the Advanced section.

Search in Column

If this option is selected the search will be restricted to the specified start and end columns. To search an entire line, "0" and "-1" should be used.

Always set 'Find/Replace Where' to 'Current File' in Find/Replace dialog

If this option is selected, the 'Find/Replace Where' option in the Find/Replace dialogs will always be reset to 'Current File' when the dialogs are invoked. If this option is not selected, the option selected the last time the dialog was used will be remembered.

Set Find Output Format (Configuration - Search)

The Set Find Output Format branch under Configuration - **Search** allows the user to configure the format to be used to display the output of a [Find in Files](#) command.

The following variables may be used to specify the output items described below:

\$P path to file

\$F string to find

\$L line number

\$S found line

\$C number of occurrences of searched string found

\$N number of files scanned

Output per File

These options control data that will be displayed for each file included in the Find in Files output:

Header

If this option is selected the specified format will be used as the header for content regarding each file found containing string(s) matching the Find in Files search string.

Found Line

If this option is selected the specified format will be used as the to describe the line in which a match occurred during the Find in Files operation.

File Summary

If this option is selected the specified format will be used as to summarize matched strings for each file included in the Find in Files output.

Final Output

These options control data that will be displayed in the final line of the Find in Files output:

Find Summary

If this option is selected the specified format will be used to display an overall summary of all matches in all files during the Find in Files operation.

Pressing the **Default** button will restore the default options for the Find in Files output format.

Spell Checker

Dictionary (Configuration - Spell Checker)



The Dictionary branch under Configuration - Spell Checker allows selection of the main spelling dictionaries to be used:

Aspell is installed along with UltraEdit and UESstudio. By default the executable for Aspell is located in the \ Aspell\bin\ subdirectory below the installation directory. Periodically an updated version of Aspell is released. Users may select any of the languages shown, or if preferred may select none of the languages in which case only the user-defined dictionary will be used.

Adding/Removing Words

Users may add/remove words as desired. A dictionary must first be selected in the Dictionary pane. If a word is typed into the Added Words field and the Add button is pressed the word will be added to the Added Words list. Words may be removed from the list by selecting them and pressing the Remove button. The Added Words list is maintained on a per language basis.

Dictionary Directories (Configuration - Spell Checker)



The Dictionary Directories branch under Configuration - Spell Checker supports specification of directories to be used for storing information related to spell checking:

Personal data directory

The directory specified here will be used to store words added to user dictionaries.

Aspell parent directory

The directory path specified here will be used to store the data/dictionary directories related to Aspell. By default this will point to the application install directory. When used from the editor, Aspell looks for dictionaries under the GNU/Aspell directory.

Filters (Configuration - Spell Checker)



The Filters branch under Configuration - Editor- Spell Checker allows specification of filters to be used to ignore certain items in documents being checked by Aspell. If the **Use Filters** checkbox is selected the following options are available. (Only one filter may be used at a time.):

Automatic detection

This option automatically detects if a file is an **HTML** or **Tex** file and applies the appropriate filter.

Email

The email filter skips over quoted text. It currently does not support skipping over headers.

HTML

The HTML filter is like the SGML Filter Mode but specialized for HTML. By default, 'script' and 'style' are members of the skip list in HTML mode.

SGML

The SGML filter allows you to spell check SGML, HTML, XHTML, and XML files. In most cases everything within a tag will be skipped by the spell checker. The SGML/HTML/XML that Aspell supports is a slight superset of most DTDs (Document Type Definitions) and can spell check the often non-conforming HTML found on the web.

URL

The URL filter skips over URLs, host names, and email addresses.

Nroff

The Nroff filter allows you to check the spelling of Nroff documents. This filter mode skips following nroff language elements:

- Comments
- Requests
- Names of nroff registers (both traditional two-letter names and GNU nroff long names)
- Arguments to the following requests: ds, de, nr, do, so.
- Arguments to font switch (\f) and size switch (\s) escapes
- Arguments to extended charset escape in both traditional (\() and extended (\[comp1 comp2 ...]) form.

TeX

The TeX filter skips over TeX commands and parameters and/or options to certain commands. It also skips over TeX comments by default.

C/C++

The C/C++ filter allows you to check the spelling of text in strings or comments. This filter mode skips checking for spelling errors in source code.

Ignore Options (Configuration - Spell Checker)



The Ignore Options branch under Configuration - Spell Checker allows configuration of options to be used during spell checking operations. Options are available to ignore case, accents, all-cap words, capitalized words, words with numbers, and words with mixed case. All of these options may be toggled using the appropriate checkbox.

Miscellaneous (Configuration - Spell Checker)



The Miscellaneous branch under Configuration - Spell Checker allows configuration of various options to be used during spell checking operations:

Enable spell check as you type

If this option is selected spell checking will occur automatically as text is typed. **Please note:** By default UltraEdit and UESudio will not perform this option on any syntax-highlighted files. To change this setting and support spell check as you type for a syntax-highlighted file, add the following to the language definition line (e.g., /L3"HTML" ...) of the wordfile:

```
EnableSpellasYouType
```

Display suggestions on right mouse click over misspelled word

If this option is selected when **Enable spell check as you type** is enabled, right-clicking on a word indicated as misspelled will result in a list of possible spelling options. If preferred, the option **Only if CTRL is pressed** may be selected to restrict the display of these suggestions to the combination of pressing the CTRL key along with right clicking on a misspelled word. Users may also specify a maximum number of suggestions to be displayed using the **Display suggestions (max)** control. If this is set to "-1" then all available suggestions will be displayed.

Suggestion mode

The suggestion mode may be set to "ultra", "fast", "normal", or "bad-spellers".

Ultra	This method will use the fastest method available to come up with suggestions. This currently means that it will look for homonyms within one edit distance apart without doing any typo-analysis or using replacement tables. In this mode Aspell gets about 87% of the words from the small test kernel of misspelled words. (Go to http:// aspell.net/test for more info on the test kernel as well as comparisons of this version of Aspell with previous versions and other spell checkers.)
Fast	This method is like "ultra" except that it performs typo-analysis and uses replacement tables. The typo-analysis brings words which are likely to be due to typos to the beginning of the list. This mode should get around the same number of words that the ultra method does.
Normal	This method normally looks for homonyms within two edit distance apart and performs typo-analysis unless it is turned off. Is much slower than "fast" mode but it returns better suggestions. In this mode Aspell detects approximately 93% of the misspelled words.
Bad-Spellers	This method also looks for homonyms within two edit distances apart but is tailored more for the bad speller, where as "fast" or "normal" are tailored more to strike a good balance between typos and true misspellings. This mode never performs typo-analysis and returns a huge number of words for the really bad spellers who can't seem to get the spelling anything close to what it should be. If the misspelled word looks anything like the correct spelling it is bound to be found somewhere on the list of 100 or more suggestions. In this mode Aspell detects approximately 98% of misspelled words.

Encoding

Users may select the encoding the be used with the active dictionary from this dropdown.

Exclude from checking words shorter than (number) characters

Users may specify a minimum number of characters (from 1 to 9) required for a word to be checked for spelling errors.

File Handling

Advanced (Configuration - File Handling)



The Advanced branch under Configuration - File Handling has the following options:

Check short names against files already loaded to prevent loading of two copies of same file

This setting forces a check when opening a file for existing files with a matching short name to prevent opening the same file twice. On some servers the short name is the same causing the editor not to open the file. Disabling this (unchecking it) will cause the check not to be made.

Delete original file before saving the file

When a file is saved the original file is overwritten with the new file. In some cases on some servers it's necessary to first delete the file and create a new file. Setting this option causes the file to be deleted and replaced by the new file.

Copy file instead of using rename when creating backup file

This setting affects how the editor generates the backup file. By default the original file is renamed to the backup file. The reason for the setting is that on some systems, if the original file is renamed to create the backup, the recreated original file does not maintain the file permissions that previously existed. Setting this causes the rename not to occur, preserving the permissions.

Always create new file when copying to/from temporary file

When copying to/from the temporary file to the user's file the editor does not create a new file, truncating the old file, it copies the file and sets the file length. A few servers appear not to support this and do not honor the setting of the file length correctly, causing remnants of the old file to remain if the users deletes a section of a file and saves it. Setting this causes the file to be truncated to 0 length prior to writing the file.

Open from Explorer also opens files list

When this option is set and a file is selected in Explorer to be opened in the editor, the last-used workspace (and all associated files) are reloaded when the file selected in Explorer is loaded.

Skip FTP close command when connecting to FTP server

When the editor connects to an FTP server it issues a close command first to clear any previous connection that may have been aborted. In some cases with products that use FTP Tunneling this has caused a crash within the tunneling software. Setting this options causes the editor not to issue this FTP close command.

Use DDE for file associations (changing this requires a change to be made to the file associations for this to be active)

When File Associations are set from within the editor the setting to use DDE is normally set in the Registry for the specific file type. On some systems this may cause an error message when the file is double clicked from Explorer. In this case ensuring that this option is not selected may resolve this. The file associations will need to be modified for this to take effect.

Save Project information for use on multiple systems

If this option is selected, all project information is stored together to facilitate using project settings/workspaces on multiple systems. For example, if a project named "Project.prj" is created, the workspace information will be stored in "Project.pui" and this is saved in the same location as the .prj file so that it is accessible for all users.

If this option is **not** selected, as per default settings, then the project workspace information is stored separately for each user. This allows users to share project settings with another user without the associated workspace information. In this case, if "Project.prj" is the project, the user's personal workspace information would be stored in "Project.[Workstation].[UserName].pu" so that this personal workspace would only be loaded when this user again loads the referenced project.

Backup (Configuration - File Handling)



The Backup branch under Configuration - File Handling has the following options:

Backup Files on Save

When a file is saved the option to generate a backup file is provided. Three choices are available:

No backup - When a file is saved the original file is overwritten and a backup file is not created.

Replace extension with . BAK - When the file is saved, the original file is copied or renamed to a backup file with the same filename and a ". BAK" extension in place of the existing extension. The modifications are saved to the original filename/extension.

Append . BAK to the existing extension - When the file is saved the original file is copied or renamed to a backup file with the same filename and extension and ".BAK" is added to the filename (filename.ext.BAK). The modifications are saved to the original filename/extension.

Default backup directory

Normally the directory used for the backup file is the same directory as the source file. This parameter allows all backup files to be copied to the same backup directory. If this item is left blank the directory used will be that of the source file.

Automatic Save

The option is provided to automatically save all open files at the desired interval. The interval is configurable in minutes. If the interval is set to zero (0) no automatic save is performed.

If Do not auto-save unnamed (new) files is selected, the automatic save will ignore files that do not have a name (i.e. files that were created with the New command). This stops the Save As dialog from appearing if an unnamed file exists and has been modified.

If Do not auto-save FTP files is selected, the automatic save will ignore modified FTP files that are open for edit.

If a file is saved due to the automatic save command a . BAK file is only created for the first save of the session. For subsequent saves with this feature the . BAK file is not created or overwritten.

Version Backup

The options here allow users to specify settings for (sequentially numbered backups of active files.

On save

If this option is selected a backup file with a version number will be saved every time a save of an active file is executed.

On automatic save

If this option is selected a backup file with a version number will be saved every time an automatic save of an active file is executed.

Format

This text field may be used to specify the format to be used for version backups. The following items may be defined:

Path Variables:

\$b - Default Backup Directory

\$p - File Path

\$n - File Name

\$e - File Extension

Time/Date Variables:

\$H - Hour

\$M - Minute

\$d - Day

\$m - Month

\$y - Year

Miscellaneous:

\$c - Counter

\$u - User Name

Maximum number of backups

This field may be used to specify the maximum number of version backups to be saved for each file. A value of "-1" would indicate an infinite number of version backups.

Code Page Detection (Configuration - File Handling)

The Code Page Detection branch under Configuration - File Handling has the following options:

Auto Code Page Detection

This setting forces the editor to evaluate if the content of the active file should be displayed with a code page other than the system default code page.

Prompt to accept code page (If not checked accepts highest rated match)

This setting is only active when **Auto Code Page Detection** is enabled. If this option is selected and the editor detects that a new code page is required, a dialog will be presented where the user may select the code page to be used for the active document. If this is not selected, but **Auto Code Page Detection** is enabled the highest rated matching code page will automatically be used.

Prompt to change font and script if different from default system code page

If this option is selected and a new code page is selected that requires a different font and script for proper display a dialog will be displayed to prompt this change.

When converting from Unicode to ASCII always use system default code page

If this option is not selected and a document is converted from Unicode to ASCII a dialog will be presented for the user to select the code page to change to as Unicode and ASCII files require different code pages. If this is selected, the system default code page will be used for the conversion.

Conversions (Configuration - File Handling)

The Conversions branch under Configuration - File Handling has the following options:

Use IBM-500 standard for EBCDIC conversion

This setting forces the editor to use the IBM-500 standard for EBCDIC conversion. The default conversion did come from published EBCDIC tables but does not provide a direct one-to-one comparison.

Do not convert returns (CR/LF) during EBCDIC conversion

This setting overrides the conversion of EBCDIC returns when converting files to ASCII format. The EBCDIC table does call for conversion of hard returns (CR/LF) but some users requested that this not occur.

On paste convert line ending to destination type (UNIX/MAC/DOS)

If this option is selected the editor will check the format of the destination document when pasting from the clipboard and modify the line terminators of the clipboard content if necessary to match the format of the destination file.

DOS/UNIX/MAC Handling (Configuration - File Handling)



The DOS/UNIX/MAC Handling branch under Configuration - File Handling has the following options:

Default file type for new files

DOS

UNIX

MAC

This setting determines the default file type for NEW files created within the editor. The default is for DOS files. Alternatively a default of UNIX or MAC file types may also be selected.

UNIX/MAC file detection/conversion

Disable

Detect file type and prompt to change

Automatically convert to DOS format

These options allow the user to determine if the editor should automatically detect and convert UNIX/MAC files to DOS format when they are opened. Optionally the user may select to be prompted each time a UNIX/MAC file is detected before it is converted. If no detection/conversion is desired this may be disabled.

Only recognize DOS terminated lines (CR/LF) as new lines for editing

This setting causes the editor to only recognize DOS terminated lines (CR/LF) as new lines for editing. By default the editor automatically recognizes UNIX/MAC terminated files for display purposes.

Save file as input format (UNIX/MAC/DOS)

This option instructs the editor to automatically save the file as the original format (UNIX/MAC) when saved unless it was manually converted after opening.

Encryption (Configuration - File Handling)

The Encryption branch under Configuration - File Handling has the following options:

Delete original file(s) after encrypting

This setting determines if the original plain-text file should be deleted automatically after it is successfully encrypted.

Secure delete original files using UltraSentry (Requires UltraSentry v2.00a+1 or later)

If this option is selected, UltraSentry will be silently invoked to securely delete the original file after it is successfully encrypted. Rather than just deleting the original file, this will permanently remove and make unrecoverable the content of the original plain-text file. If UltraSentry is not installed this option will be subdued and it will not be possible to select this option.

For further information regarding the integration of UltraSentry with UltraEdit, please [click here](#).

Replace encrypted/decrypted file if it exists

If this option is selected, any files that already exist in the specified path with names and extensions matching the files selected for encryption/decryption would automatically be overwritten.

Automatically re-encrypt file on close

If this option is selected, files that are decrypted prior to editing will automatically be re-encrypted when saved and closed. This option will respect other encryption options set in Configuration as well which pertain to deletion/overwriting of files after encryption.

File Change Detection (Configuration - File Handling)



The File Change Detection branch under Configuration - File Handling provides options that control the behavior of the application when the active file is updated by another user (in the case of a file being edited by more than one user at a time) or an external process (i.e. an application or log file, for example). This branch has the following options:

File change detection

Disable

Prompt for update when files change

Automatically update changed files

The editor automatically detects any changes to a file that is loaded and is changed outside of the editor. By default **Prompt for update when files change** is selected and when the changes are detected the editor will prompt the user with the option to reload the file (in which case any unsaved changes to the file in the editor will be lost). The user may choose to reload the file or ignore the changes.

Optionally, the editor provides the ability to ignore any file changes by setting the **Disable** option. In this case the editor will not detect or prompt for any file changes.

If preferred, the editor provides the ability to automatically update the loaded files with any changes that have been made outside of the editor by setting the **Automatically Update Changed Files** option. In this case, the file will be loaded and any unsaved changes made within the editor will be overwritten automatically without prompting.

Poll for file changes at forced interval in seconds (0 disables)

This setting determines the interval in seconds at which the editor checks for changes to files for which polling has been activated through the File Change Polling command in the View menu. If this is set to "0" this feature is disabled.

When a file marked for polling is detected as changed scroll to the last line in the file (used for tailing log files)

This setting forces the editor to scroll such that the last line in a polled file is always visible in the edit window when the file is active. This is similar to the UNIX tail command.

When a file is detected as changed and reloaded, make it active

This setting instructs the editor to make any file it detects as changed and reloads to be the active file. The default setting is not to make the file active.

Ignore file changes if the file was deleted

This setting instructs the editor not to notify the user when an opened file has been detected as changed. If this is not set (default) the editor will notify the user that a file has been deleted if the file change detection is enabled.

Check files for changes on application focus change

If this option is selected, files will only be checked for changes when UltraEdit receives focus (when switching from another application or from a dialog within UltraEdit) rather than checking automatically at predefined intervals.

Load (Configuration - File Handling)



The Load branch under Configuration - File Handling has the following options:

Reload files previously open on startup

If this option is checked the editor saves information about the open files when the editor is closed. This information is used to automatically reload the files that were open the next time the editor is started. This allows easy reloading of a set of files that are edited on a regular basis. If the editor is invoked automatically for printing the files are not reopened.

Reload FTP files when reloading a project or last open files

This setting determines if the editor should reload FTP files when it reloads files for a project/workspace. Typically FTP are not reloaded files automatically due to the potential time delay in connecting and load the files. Selecting this to option will ensure FTP files are treated as normal files in this regard.

Use default directory from shortcut on initial file open

Normally the directory used for the File Open command is the last directory that the editor opened a file from. This directory information is retained when the editor is shutdown and restarted. When the Use default directory from shortcut on initial file open option is set the normal behavior is overridden and the process default directory is used for the initial open. This allows the user to specify the default directory from the icon or shortcut that invokes the editor.

Use active file directory for file open dialog default

When this option is set, the normal behavior is overridden and the directory of the currently active file is used. This allows the user to easily open a file from the same directory as an already open file by activating the file and then use the File Open command.

Please note: Only one of the two items described above should be selected.

Open link (.lnk) files, not the file the link points to

This setting causes the editor to open the link for modification. The default setting (not setting this option) causes the editor to expand the link to open the root file.

Open files as read-only by default

The Read Only command toggles the state of the read-only status of the file. If the read-only status is set the file cannot be modified. If this is set, files will automatically be opened as read-only. This does not change the status of the file itself, just the ability to modify it within the editor.

Remember the read-only setting when reloading a file currently open

This setting causes the editor to maintain the Read-Only status of a file when this file is reloaded. By default when a file is reloaded the status of Read-Only does revert to the status of the underlying file. This is because a lot of users use this feature with version control systems and the only thing that changes is the Read-Only status and the users want/expect the status within the editor to change also.

Miscellaneous (Configuration - File Handling)



The Miscellaneous branch under Configuration - File Handling has the following options:

Change the read-only attribute when changing menu setting

This setting instructs the editor to change the read-only status of the file when the menu setting for read-only is changed. The default setting (unchecked) allows the user to change the editor's state preventing or allowing modification of the file but causes no change to the actual file status when the read-only status is changed from menu.

Lock file for write while editing

This setting causes the editor to keep the lock on any file that is opened. Additionally, if the file is already locked, it allows for open by read-only in this case prompting the user with a dialog box to confirm the open as read-only. This feature prevents modification of a file from outside of the edit session preventing loss of changes made.

Close all files when opening or switching projects

If this option is selected, all currently open files will be closed without a warning prompt when opening, closing, or switching projects. If this is not selected the user will be prompted as to whether or not the open files should be closed.

Save (Configuration - File Handling)



The Save branch under Configuration - File Handling has the following options:

Write UTF-8 BOM header to all UTF-8 files when saved

This option instructs the editor to write the UTF-8 BOM (byte order marker) to all files when they're saved. If this option is not set the BOM will not be written to the file when it is written to the disk. This will not remove a BOM that already exists in the file.

Write UTF-8 BOM on new files created within this program (if above is not set)

This option instructs the editor to write the UTF-8 BOM only to new files created within the editor when they're saved. If this option is not set the BOM will not be written to the file when it is written to the disk. This will not remove a BOM that already exists in the file.

Trim trailing spaces on file save

This function causes the editor to automatically remove the trailing white space (spaces/tabs) from the end of every line in the file and positions back to the original cursor position when it is complete each time a file is saved. The functionality of Trim Trailing Spaces is such that it does require a newline to operate on a given line of text. If the last line of text in your file is terminated with a hard return, Trim Trailing Spaces will work on that line as well.

Temporary Files (Configuration - File Handling)



The Temporary Files branch under Configuration - File Handling has the following options:

Use of temporary file for editing (Provided for very large file handling)

Use temporary file for editing (normal operation)

Open file without temp file but prompt for each file (CAUTION: Edits are permanent, decreases load time for large files)

Open file without temp file but NO prompt (CAUTION: Edits are permanent, decreases load time for large files)

Threshold for above (KB): 15000

UltraEdit/UESstudio allow editing of very large files. This is accomplished with relatively small amounts of memory being used. UltraEdit/UESstudio is a disk-based editor, and only loads a small portion of the file into memory at one time. Normally the editor copies a file that is being edited to a temporary file, and at the end of the editing session deletes the file after copying back to the appropriate original file.

To speed up editing of very large files, this configuration branch allows you to specify editing without using temporary files. This removes the time necessary to copy the file to a temporary file. When editing files of many megabytes this can be an advantage.

WARNING - If this option is selected, all changes are permanent!

A dialog box is shown warning the user this setting is enabled each time a file is opened. This may be disabled by selection the option to open the file without a temp file with NO prompt.

A "Threshold" field is provided to allow the user to specify files that are equal or larger than this value to be loaded without the creation of the temporary file. Files smaller than this would be loaded normally, creating a temporary file. Setting the value to zero means that all files would be loaded without the creation of the temporary file. The "Threshold" field is only applicable if the option to open files without the temporary file is set.

Additionally, the Disable Line Number option improves the performance when navigating around large files as less disk access are required to track the line numbers.

UltraEdit/UEStudio allows editing of files greater than 4GB (up to a size of $2^{64}-1$).

Use UltraSentry to wipe temporary file when closing document (requires UltraSentry v2.00a+1 or later)

If this option is selected, UltraSentry will automatically be invoked to wipe the temporary file when a document is closed. Rather than just deleting the handle for the temporary file, this will permanently remove and make unrecoverable the content of the temporary file. If UltraSentry is not installed this option will be subdued and it will not be possible to select this option.

Unicode/UTF-8 Detection (Configuration - File Handling)



The Unicode/UTF-8 Detection branch under Configuration - File Handling has the following options:

Auto detect UTF-8 files

This option instructs the editor to automatically detect UTF-8 files when opened.

Detect Unicode(UTF-16) files without BOM

This option instructs the editor to automatically detect UTF-16 files when opened even if they don't contain the standard UTF-16 BOM (HEX FF FE).

Disable automatic detection of HEX file format on reload

This setting overrides the default check for NULL characters in the first 64KB of a file when a file is reloaded. If this is selected a file which normally would be displayed in HEX/binary format on reload is displayed using whatever mode is selected for that file regardless of its content.

Detect ASCII/ANSI files with Escaped Unicode

This option instructs the editor to automatically detect ASCII/ANSI files with Escaped Unicode (e.g. \u0020).

File Associations (Configuration)



The File Associations branch under Configuration allows the user to associate the editor with selected file types:

This dialog shows a list of current associations for the editor, and a list of file types that the user may wish to associate with the editor. The user may add or remove file associations using this dialog.

A file association is used by Windows Explorer to determine which application is to be used to open the file when the file is double clicked or opened in the Explorer.

The list showing Associations shows all file types that are currently associated with the editor. The user may select any item in this list and remove the association with the Delete button.

To add an association, the user may select an item from the File types/extensions dropdown list or type in a new extension and add it with the Add button. The Description may be modified for any file type that is added.

When an association is added, the editor saves the old association if it exists and restores it if the association is later removed using this dialog.

Integrate with Explorer

If this option is selected UltraEdit/UEStudio options are added to the context (right-click) menu in Windows Explorer.

File Types (Configuration)



The File Types branch under Configuration allows the list of file types that appear in the File Open and File Save As dialog boxes to be configured as follows:

Up to 10 file types may be configured. These file types will be shown in the file operation dialog boxes in the same order they are shown on this page.

Each file type configured consists of a file name and file description. The file name may (and probably does) include wildcard characters and may include multiple file extensions each separated by a semicolon ";"(i.e. "*.c;*.cpp"). The file description is a text field describing the file types.

This dialog also provides a DEFAULTS button that resets the file types to the initial settings.

Editor Display

Advanced (Configuration - Editor Display)



The Advanced branch under Configuration - Editor Display has the following options:

Setting this may improve display issues with ClearType fonts on Windows XP

This setting does not actually enable ClearType but it does compensate for some issues that may show up when using ClearType with UltraEdit/UESudio, specifically with Windows XP. UltraEdit/UESudio should be restarted when this setting is changed.

Override default edit window dimensions

When a new document is created, it is automatically sized to fit the dimensions of the editor. This option overrides this autosizing with specific values. Note that if the dimensions of the editor are smaller than the specified height and width, the new document will be created with the old autosizing method.

Font Quality

Using this option the user may set the quality level Windows will use to render fonts in the editor. Information regarding the quality selected is displayed as the slider is moved. Following is an OS-specific guide for choosing font qualities:

For **Windows XP** and greater all options are available on a scale from **Default** to **Best**:

- Default Quality
- Draft Quality
- Proof Quality
- Non Antialiased Quality
- Antialiased Quality
- ClearType Quality
- Natural Quality

For **Windows NT** and **2000** the following options are available:

- Default Quality
- Draft Quality
- Proof Quality
- Non Antialiased Quality
- Antialiased Quality
- ClearType Quality

For **Windows 98** and **Me** the following options are available:

- Default Quality
- Draft Quality
- Proof Quality
- Non Antialiased Quality
- Antialiased Quality

UltraEdit/UEStudio should be restarted when the font quality is changed.

Code Folding (Configuration - Editor Display)



The Code Folding branch under Configuration - Editor Display has the following options:

Save folded lines

If this option is selected, information regarding folded lines will be stored when a file is closed so that it will be opened in the same state the next time the file is edited. If this option is not selected, all lines will be unfolded the next time a file is loaded regardless of their fold state at the time a file is closed.

Enable show/hide lines and code folding

This setting permits hiding/showing selected text and folding/expanding code blocks. If this is not selected the folding gutter will not be visible in the edit window and folding is disabled. If a file is opened that exceeds the threshold specified for using temporary files under the **Temporary Files** branch in the **File Handling** section of configuration, and the option is set to open without a temporary file in this case, code folding will not be displayed.

Enable show/hide lines in non-syntax-highlighted files (above must also be set)

This setting permits text to be hidden in non-syntax-highlighted files and expanded/collapsed using the View menu commands. If this is not selected the folding gutter will not be visible in the edit window in non-syntax-highlighted files and folding is disabled.

Show last line of fold in syntax highlighted files

If this option is selected, the last line of a folded section will be displayed in files with syntax highlighting. If this option is not selected, only the initial line of a folded section will be displayed.

Automatically unfold hidden areas on Find and Goto

If this option is selected, and a Find or Goto command positions the cursor within folded text, that folded level will be unfolded automatically to correctly display the cursor in context.

Detect XHTML from DTD on file load

If this option is selected, the editor will scan files for a DOCTYPE declaration. If a DTD for XHTML 1.0 Strict or XHTML 1.1 is specified, the rules for elements associated with this document type will be respected for code folding and indentation.

Draw graphical lines

If this option is selected and Enable show/hide lines and code folding is selected above, lines are drawn in the folding gutter to the right of the line numbers to indicate where folding nodes begin and end.

Cursor/Caret (Configuration - Editor Display)



The Cursor/Caret branch under Configuration - Editor Display has the following options:

Normal Cursor/Caret type

Vertical bar

Underscore cursor

This setting determines the type of cursor/caret that will normally be displayed in the editor.

Overstrike Cursor/Caret type

Block caret for overstrike mode, normal caret for insert mode

Inverted - Normal caret for overstrike mode, block caret for insert

Block caret disabled, always use normal caret

This setting determines how the type of cursor/caret will be affected by Overstrike Mode settings within the editor.

Allow positioning beyond line end

This option allows the cursor to be positioned beyond the end of text in a line.

Underscore is word stop for Ctrl left/right arrow

If this option is selected the cursor will stop at occurrences of "_" in text when CTRL+LEFT/RIGHT ARROW are pressed. The default is for this option not to be selected. Please note that the setting chosen for this option can affect the functionality of macros/scripts that include navigation operations.

CamelCase naming stops for Ctrl left/right arrow

If this option is selected the cursor will stop at uppercase letters in mixed case words when CTRL+LEFT/RIGHT ARROW are pressed.

Formatting (Configuration - Editor Display)



The Formatting branch under Configuration - Editor Display has the following options:

Auto indent new lines

This item toggles the state of automatic line indentation. When automatic line indentation is enabled, the editor automatically indents a line to match the indentation of the previous line. The indentation is performed by inserting spaces into the document.

Additionally language specific indenting and out-denting is supported. This provides for additional indenting after a certain character/string to indent a block of code and out-denting after a closing character/string. See the indentation section under [Syntax Highlighting](#) for details.

This option is only active in [INSERT mode](#), not in [OVERSTRIKE mode](#).

Auto indent wrapped lines

If this option is selected the editor automatically indents wrapped lines to match the indentation of the previous line.

Break on hyphen (-) when converting/reformatting paragraphs

This setting determines if the editor should break words at hyphens when reformatting a line. The default is for UltraEdit to break lines at hyphens. If this setting is not selected the editor will not break the line when a hyphen is encountered.

Reformat paragraph after paste when wrap settings set to add hard returns at specific column number

This setting causes the editor to automatically reformat a paragraph after a paste command if the Word Wrap settings are set to wrap at a specific column number with the insertion of hard returns (CR/LF). This is the default behavior and not selecting this option will disable this feature.

Do not convert spaces and tabs inside comments and strings

If this option is selected, spaces and tabs inside comments and strings (as defined in the associated wordfile) will not be converted by formatting operations.

Hex Edit Display (Configuration - Editor Display)



The Hex Edit Display branch under Configuration - Editor Display has the following options:

Number of HEX characters per line

This setting defines the number of HEX characters to be shown in each line of the display. The editor must be restarted once this value is changed.

Miscellaneous (Configuration - Editor Display)



The Miscellaneous branch under Configuration - Editor Display has the following options:

Editor color preferences

Clicking on the Set Colors button launches the [Set Colors dialog](#).

Maximum columns before line wraps

This setting sets the maximum columns allowed within the editor before it wraps the line (soft wrap). The default is 4096 columns but a maximum of 20,000 is allowed. Some video drivers have a problem when this is set too high.

Count wrapped lines as new lines in number display

This setting causes the editor to show wrapped lines with different line numbers. By default this is not set.

Disable line numbers

When this option is selected, line numbers will not be displayed in the editor and the Display Line Numbers option will be subdued in the View menu.

Start ruler numbering at column 1

This setting causes the editor to start the ruler at 1. If this option is not selected (default) the editor starts the ruler at 0.

Display full paths in project list

If this option is selected the project list in the File Tree View (Workspace Manager in UEStudio) will display the full paths for project files rather than just the file names.

Enable auto-brace matching

When this option is selected, and the cursor is positioned beside an open or close brace, the matching brace will automatically be highlighted.

Brace match in comments

If this option is selected the **Match Brace** function in the Search menu and **auto-brace matching** (if active) will match braces in commented text as well as in source code.

Auto-correct keywords

This option toggles the state of the auto-correct feature. If this option is selected, keywords detected will be corrected to match their case in the active wordfile. For example, "javaarray" would automatically be changed to "JavaArray" for JavaScript files.

Line Change Indicator

If this option is selected, modified lines will be indicated with a special colored shading between the line numbers and the first character on the line. The colors used to indicate saved and unsaved changes to a file within an editing session are configured under [Set Colors](#) in the **View** menu. If the undo buffer is cleared during an editing session, the line change indicators would be reset as well.

If a file is opened that exceeds the threshold specified for using temporary files under the **Temporary Files** branch in the **File Handling** section of configuration, and the option is set to open without a temporary file in this case, the line change indicator will not be displayed.

This setting is a global toggle for the function. If users wish to toggle this option on/off on a file-by-file basis, this may be done using the option in the View menu. If files are open for editing when this option is changed, they would need to be reloaded or the **Line Change Indicator** setting would need to be toggled in the **View** menu to affect these files.

Syntax Highlighting (Configuration - Editor Display)



The Syntax Highlighting branch under Configuration - Editor Display allows the user to configure syntax highlighting options:

Syntax highlighting is the ability to recognize predefined words and display them in different colors. This is particularly useful for programmers and can also be useful to other users who may want certain words in a document to show up in a different color.

Support for many languages or word sets that may be recognized is provided. Each word set may be configured to have a different color for the following groups of words:

Normal Text (Words that are not recognized)

Words that appear in a comment line, or block comments

Alternate set of block comments

Strings

Number values (must start with a digit (0-9))

Up to eight groups of different recognized words

The Configuration Dialog allows the colors to be selected for the different language and color groups. This also allows the user to specify the location of the "wordfile" used for highlighting.

For any of the color groups the foreground/text color and the background color may be set. By default the background will revert to the background specified for normal text (Background color automatic will be checked). This must be unchecked to activate the background color.

Each color may be changed by clicking on the colored box beside the text description. When clicked, a dialog box will appear that allows the new color to be selected.

Additionally for each color group (except Normal Text) font styles of Bold, Italic and Underline are available. These may be selected individually for each color group. With some fonts the underline may not always show correctly, and with bold, the spacing may not be correct for non-fixed pitch fonts.

Note - The highlighting is determined by the file extension (details below). The file extension of the file being modified must be defined in the "wordfile" for one of the languages.

The predefined words may be configured by the user as follows:

UltraEdit reads configuration files (the default file extension is "*.UEW") from the directory specified under Advanced -> Configuration -> Editor Display -> Syntax Highlighting, to configure the syntax highlighting. These files are read each time the editor starts up. Each file may be up to 372KB in size. The syntax for these files is as follows:

Language Definition

The word set, or language is specified by "/Ln". This must be at the beginning of the line. A description or name of the language may be specified immediately following the / Ln in quotes. This description if present is displayed when setting up the colors for the language. The description may be up to 24 characters.

Line Comments

The comment characters used for line comments are specified by the string "Line Comment = " followed by the comment characters. Five characters are allowed; if there are less than five then the last character must be followed by a space.

A second set of line comments may be specified by the string "Line Comment Alt = " followed by the comment characters. Five characters are allowed; if there are less than five then the last character must be followed by a space.

As some instances may require that a space is a part of the line comment an alternative method to describe the line comment is available. Using the alternative method, the number of characters used is specified by the user with the following syntax:

"Line Comment Num = xCC "

where x specifies the number of characters (1 to 5) and IMMEDIATELY following are the characters to be used as line comments. In the example above, x would be 3, and the line comment would be "CC " (note the space after "CC").

Additionally, there are times when qualifiers are required for line comments such that the comments are only valid if they occur at certain columns, or after certain characters (well, more often they are not valid if they follow certain characters). To cater for this, two additional line comment commands are provided:

"Line Comment Preceding Chars = [...]"

and

"Line Comment Valid Columns = [1-7,10]"

The default for "Line Comment Preceding Chars" is that all characters are valid. Therefore this entry would include the characters that are not valid following a tilde character as in ...[~a-z]. This would say that the comment is not valid if it immediately follows a character in the range a-z. Any characters are valid between the brackets.

The default for "Line Comment Valid Columns" is that all columns are valid if this is not defined. If this is defined, then only the columns specified are valid. There can be up to 10 column ranges, or columns specified separated by a comma as in ...[1-7, 10]. This indicates that the comment characters are valid if they occur at columns 1 through 7, or at column 10.

Block Comments

The characters used for block comments may also be configured (i.e. /* ... */ for 'C'). These are in the form "Block Comment On = " and "Block Comment Off = " followed by up to nineteen characters each that define the comment designators. For compatibility with previous releases, the '/' and '*' are used for file types that have '.C' as one of the extensions. This may be overridden. The first character of the block comments may be a space.

If a "Block Comment On" is defined but the "Block Comment Off" is not defined the commenting will stop at the end of the line. This effectively allows the block comments to be used as line comments also.

Additionally, a second set of block comments may be defined for languages that require it. This is particularly useful for ASP allowing HTML comments to be maintained with the addition of adding highlighting for ASP blocks.

The second set of block comments are in the form "Block Comment On Alt = " and "Block Comment Off Alt = " followed by up to nineteen characters each that define the comment designators. The first character of the block comments may be a space.

If a "Block Comment On Alt" is defined but the "Block Comment Off Alt" is not defined the commenting will stop at the end of the line. This effectively allows the block comments to be used as line comments also.

UltraEdit/UEStudio by default will end any block comment whenever a single block comment end string is encountered. To instruct UltraEdit/UEStudio to explicitly match all block comment start strings with all block comment

end strings, the nest block comment command may be specified. This is accomplished by adding the **NestBlockComments** command to the language definition line (i.e., /L3"HTML" NestBlockComments ...).

File Extensions/Types

Syntax Highlighting is determined either by the name of the file or its extension. More commonly the extension is used and to specify the extensions for which this language is applicable the following string should be used: "File Extensions = " and each extension is separated by a space. File extensions specified here are NOT case sensitive.

The file extension is everything AFTER last dot in the name of a file. Up to 97 bytes may follow the "File Extensions = " declaration. If a list of file extensions exceeding 97 bytes is specified here, they will not all be parsed.

Users may specify a language as the default for syntax highlighting by placing an asterisk "*" as the last file extension specified. This would also be used as the highlighting file type for unsaved files or files saved without an extension.

To specify that a filename rather than file extension is to be used to determine the language the following string should be used: "File Names = " and each name is separated by a space. This must include the extension, i.e.: "File Names = ThisFileName.xml". Up to 125 bytes may follow the "File Names = " declaration.

File Names/Extensions Considerations

The following items must be considered if using both File Names and File Extensions options in the same wordfile:

1. Names including spaces may not be specified with the **File Names** option.
2. The **File Names** option may not be used in conjunction with the **File Extensions** option.
3. If a language is defined using the **File Names** option, and a different language within the same wordfile is specified using the **File Extensions** option and the definitions overlap, the language defined with the **File Names** option must have a lower language number, i.e.:

Example 1 - won't work:

```
/L6"XML General" XML_LANG Noquote ... File Extensions = XML XUL XSD XSL XSLT
```

```
/L12"XML Special" XML_LANG Noquote ... File Names = ThisIsJustAFile.xml
```

Example 2 - will work:

```
/L6"XML Special" XML_LANG Noquote ... File Names = ThisIsJustAFile.xml
```

```
/L12"XML General" XML_LANG Noquote ... File Extensions = XML XUL XSD XSL XSLT
```

Color Selection

Color codes may be specified by adding a line with /Cn at the beginning of the line, where n is the color index of 1 to 8. A description or name of the section may be specified immediately following the /Cn in quotes. This description if present is displayed when setting up the colors for the language. The description may be up to 24 characters.

All information specified remains in effect until overridden with new command information.

The following example specifies the first language to be used with files with the extensions 'C', 'CPP', 'H' or 'HPP'. The color used for the words is the first selectable color and the comment characters are //.

```
/L1"C/C++" Line Comment = // Block Comment On = /* Block Comment Off = */ File Extensions = CPPC  
H HPP
```

```
/C1
```

```
auto
```

```
break
```

```
case char const continue chr$
```

```
default do double
```

Case Sensitivity

If the language is not case sensitive, the keyword "Nocase" may be added to the command line i.e.:

/L1 Line Comment = // Nocase File Extensions = CPPC H HPP

Strings

When using many programming languages, characters in single quotes and double quotes are treated as literal strings, and word and comment recognition should be ignored. This is the default behavior for UltraEdit. There are some languages, (i.e. HTML and others) where this behavior is undesirable. To facilitate such languages the keyword "Noquote" may be added to the command line to override the default behavior i.e.:

/L1 Noquote File Extensions = HTM

UltraEdit now supports configurable characters for quotes strings. The default characters for strings are single and double quotes (') and ("). These may be overridden with the keywords "String Chars = " followed by up to two characters. This is only required if you wish to use different characters from the default. If for example you wish to use the double quote only for strings you would have a line similar to the following:

/L1"C/C++" Line Comment = // Block Comment On = /* Block Comment Off = */ String Chars = " File Extensions = CPPC H HPP

Note the "String Chars" portion.

Additionally, if you have two characters for the strings defined (or using the defaults) and you wish to have a different color for each type of string, you may now include the character (i.e. double quote) in one of the color groups in a line by itself. This will override the configurable color for the strings that are encapsulated by this character.

/L1"C/C++" Line Comment = // Block Comment On = /* Block Comment Off = */ String Chars = "" File Extensions = CPPC H HPP

/C1

auto

break

"

...

In the above example, all strings that have double quotes (") around them will be colored with the color group C1 while strings with a (') around them will use the default color for strings.

Escape Character

In many languages there is a special designated character that is referred to as an Escape Character. This is used to override the normal string characters (and other characters). If a string is defined that itself contains a quote character normally used to define the string, the syntax highlighting would interpret this quote as the end of the string. If however this quote character is preceded with an Escape Character the quote character would be appropriately treated as part of the string and not the end of it.

The Escape Character is defined as follows:

/L1"C/C++" Escape Char = \ ...

In this case the Escape character is defined as a backslash.

Keywords

Note that ALL words starting with the same character may be on the same line or spread across multiple lines, however if they are spread across multiple lines the lines must be one after the other with no empty lines or other lines between them.

If the language is case sensitive, the letter 'A' is different from 'a' and so words starting with 'A' MUST be on a different line from words starting with 'a'. If the language is case insensitive words starting with the letter 'A' must be on the same line as words starting with the letter 'a'.

Keywords beginning with a Sub String

There are instances in some languages where it is desirable to highlight keywords that begin with a particular sub-string, however the complete word is not known. UltraEdit provides the ability to define sub-strings that are used to determine if a word should be highlighted. If such sub-strings are defined for

a particular language under a color group UltraEdit will determine if a word begins with one of the sub-strings. If it does, it will be highlighted accordingly.

The sub-strings should be defined as with any other set of keywords however the line containing the sub-strings should start with "*** " and all sub-strings should be on the same line as in:

```
** aaa bbb
```

The example above would highlight any words beginning with "aaa" or "bbb".

Words Starting with '/'

As UltraEdit uses '/' as a command character words to be highlighted require special handling. To highlight words beginning with a '/' the line should begin with '//' followed by the keywords as in:

```
// / mykeyword / anotherkeyword
```

HTML Specific

HTML is considerably different from other languages, and to better facilitate the use of UltraEdit for editing HTML files the "HTML_LANG" keyword was added. When this keyword is present, the special characters '<' and '/' may be placed in front of any keyword as desired without all keywords with the special ('<' and '/') characters all having to be on the same line. In this case, words starting with the same letter (a-z etc) must be on the same or contiguous lines as is normally required.

Below is an example portion of a word file for HTML:

```
/L3"HTML" Line Comment = // HTML_LANG Block Comment On = <!-- Block Comment Off = --> File  
Extensions = HTM HTML
```

```
/C1
```

```
<A </A> <ADDRESS> </ADDRESS> <APPLET </APPLET>
```

 <BASE <BASEFONT <BGSOUND <BIG> </BIG> <BLINK> </BLINK> <BLOCKQUOTE>
 </BLOCKQUOTE>
 <CAPTION> </CAPTION> <CENTER> </CENTER> <CITE> </CITE> <CODE> </CODE>

Additional enhancements for HTML will be added in the future.

FORTRAN Specific

FORTRAN is quite different from other languages regarding comments, and to better facilitate the use of UltraEdit for editing FORTRAN files the "FORTRAN_LANG" keyword was added. When this keyword is present, UltraEdit treats a 'C', 'c' or '*' in the first column as a line comment indicator and the rest of the line is highlighted as if it were commented out.

Below is an example portion of a word file for FORTRAN:

```
/L4"Fortran" FORTRAN_LANG File Extensions = FOR FTN
/C1
...
```

Note: Any of the normal comment indications may also be used (line comments, block comments).

LaTeX/Tex Specific

TeX/LaTeX is quite different from other languages and each command starts with a '\'. To better facilitate the use of UltraEdit for editing TeX/LaTeX files the "LATEX_LANG" keyword was added. When this keyword is present, UltraEdit has special handling for syntax highlighting to allow words to be appropriately handled and highlighted with the '\', and with consecutive words.

This also allows the recognized words to be placed in the wordfile without all of them being on the same line. If the word begins with '\' then the second character is used to determine which line the word should be on. All words beginning with '\a' should be on the same line as other words beginning with '\a' or 'a'. In the same way, all words beginning with '\b' should be on the same line as other words beginning with '\b' or 'b' but on a different line from those starting with '\a' etc.

Below is an example portion of a word file for TeX/LaTeX:

```
/L4"TeX/LaTeX" LATEX_LANG File Extensions = TEX LATEX
```

```
/C1
```

```
...
```

Note: Any of the normal comment indications may also be used (line comments, block comments).

Delimiters

UltraEdit has built in delimiters that are used to determine when a new word starts and when a word finishes so that it may be matched against the set of words for a given language. With the exception of the '<' and '>' characters in HTML a character that is a delimiter may not also be part of a word; that is, you can not say the @ symbol is to be considered part of a word, and it is a delimiter between words. (Future enhancements may allow this).

With release 4.1 and later, UltraEdit allows the delimiter characters to be configurable by the user. For compatibility, the existing delimiters are retained if the user does not specify the delimiters for a given language. Each language may have its own set of delimiters. It is not necessary to configure the delimiters for languages if you are using the defaults.

To specify the delimiters, add a NEW line similar to the following to your wordfile:

```
/Delimiters = ~!@$%^&*()_+={|V{}[]:;'"<> ,. ?/
```

Note that the Delimiters should include a space and a TAB character if you want them to be considered delimiters. The line must begin with "/Delimiters =".

It is possible to assign the delimiter characters to the color sections. If you have a character that is a delimiter, such as a '+', and you wish this to be colored with one of the group colors you may add this character to a line of its own under the color section, and this will retain its recognition as a delimiter and be highlighted with the appropriate color. A delimiter may be included at the beginning of a keyword and be highlighted accordingly but may not be included in the middle of a keyword. If a "compound" keyword, or a keyword that includes a delimiter character between two sections is desired, the delimiter character

would need to be removed from the Delimiters list, or the two portions of the keyword would need to be defined separately to highlight correctly.

Function Definition Strings

UltraEdit allows the user to show a list of functions in the active document, or all project documents. As the definition of a function may be different for different languages it is necessary to allow this to be configured based on the language.

This is accomplished by modifying the "wordfile" and defining the string UltraEdit uses for each language. The string will be used by UltraEdit as a Regular Expressions search string to find the functions.

By default the regular expression string uses UltraEdit style regular expressions. If desired, Perl compatible Regular Expressions may be used for the function strings by adding the following:

```
/Regexp Type = Perl
```

Up to six functions strings per language may now be defined allowing more function/procedure formats to be specified.

To specify the function string add a line similar to the following for the specific language:

```
/Function String = "%[ a-zA-Z]*")"
```

or

```
/Function String 1 = " ..."
```

...

```
/Function String 5 = " ..."
```

This string to be searched for MUST be in quotes. If required, you can specify that only part of the resultant string is displayed in the function list. To do this, use the tagged expressions as defined under the Regular Expressions and enclose the portion of the expression that is to be displayed between "^(" and "^)" as in:

```
/Function String = "%[ a-zA-Z]+^(*^)"
```

This example would ignore the first word and display the rest of the line.

Indentation

UltraEdit provides for automatic indentation based on the specific language to indent a block of code or to unindent a block of code.

For indentation for a specific language add a line similar to the following line under the particular language section:

```
/Indent Strings = "{"
```

Any number of words may be specified in quotes (each word/string must be in a separate set of quotes ""). If an indent string occurs anywhere on a line it will be used for indenting (except when it occurs in quoted/commented text). The indentation is the next TAB stop from the indentation of the preceding line (same as if a TAB key was pressed).

For out-denting for a specific language add a line similar to the following line under the particular language section:

```
/Unindent Strings = "}"
```

Any number of words may be specified in quotes (each word/string must be in a separate set of quotes ""). If an indent string occurs anywhere on a line it will be used for indenting (except when it occurs in quoted/commented text). If the line that contains the string is indented LESS than the preceding line an indentation does not occur, otherwise the matching character/word is out-dented to the preceding TAB stop and the new line is indented to the same point.

For reindenting of files it may be undesirable to indent certain lines that are commented out, or compiler directives. This can be avoided by specifying lines that should not be indented. To do this add a line similar to the following to the appropriate language section in the wordfile:

```
/Ignore Strings SOL = "#" "/"
```

Any number of words may be specified in quotes (each word/string must be in a separate set of quotes ""). For the word to match it must be the FIRST character(s) of the line. If a line matches this, it would not be indented, however the indenting of the next line would be performed as if this line was not present.

Brace Matching

Open/Close brace strings may be for each language defined in the wordfile. This extends and enhances the functionality of the [Match Brace command](#) in the [Search menu](#) and auto-brace matching which is enabled in the Syntax Highlighting dialog.

To define an open brace add a line similar to the following under the particular language section:

```
/Open Brace Strings = "{" "(" "["
```

or

```
/Open Brace Strings = "If" "For" "Select Case" "Else" " Elself"
```

To define a close brace add a line similar to the following under the particular language section:

```
/Close Brace Strings = "}" "}" "]"
```

or

```
/Close Brace Strings = "End If" "Next" "End Select" "End If" " Elself"
```

Please note: Open and close brace strings must be positionally matched in their lists for this to work as desired. If a "{" is defined as the first Open Brace String then "}" should be defined as the first Close Brace String. They must also be unique. For example, " endcase" cannot be used as the close brace for " casex" and " casez".

If open/close brace strings are not defined UltraEdit uses a standard group of characters when performing brace matching functions as in previous versions.

Marker Characters

There are times when it is desirable to highlight all characters between two characters. UltraEdit provides for "marker characters" that mark the first and last part of a string that UltraEdit highlights between. All characters between the two characters are highlighted.

To define marker characters for a specific language add a line similar to the following line under the particular language section:

```
/Marker Characters = "ab"
```

where 'a' is the first character of the string to be highlighted and 'b' is the last character. Note that all characters on a line will be highlighted including spaces. If the line is a comment or string this is ignored. Alphanumeric characters may be used, but whitespace characters (space/tab) may not be used for marker characters.

Additionally, you may define up to 4 pairs of characters to highlight between as in:

/Marker Characters = " abcdefgh"

where strings starting with 'a' and ending with 'b' are highlighted as are strings starting with 'c' and ending with 'd' etc.

The color of the highlighted string must be configured. To specify the color, add the two characters under the appropriate color section as if they were a word such as "ab", "cd", etc.

Open/Close Fold Strings

UltraEdit has defaults for Open and Close folding strings for many languages. If no strings are specified in the Wordfile, the defaults are used. Otherwise, the specified strings are used for folding.

/Open Fold Strings = "{"

/Close Fold Strings = "}"

Ignore Fold Strings

In specific cases it is necessary for the Fold logic to ignore certain lines which contain a string. If an Ignore Fold String is found on a line of source

code (outside a line comment or block comment) the fold logic will ignore any other open or close fold strings on that line. No defaults are provided

for Ignore Fold Strings.

/Ignore Fold Strings = "Exit Function"

Open/Close Comment Fold Strings

UltraEdit provides a means to specify unique Open and Close Fold Strings which are recognized in line or block comments only. No defaults are provided for Open and Close Comment Fold Strings.

/Open Comment Fold Strings = "#Region"

/Close Comment Fold Strings = "#End Region"

Multi-line Strings

UltraEdit has defaults for multi-line string highlighting capability for many languages. If no strings are specified in the wordfile, the defaults are used. Otherwise, one of the following strings may be added to the language definition line (i.e., /L8 PHP ...)

EnableMLS

DisableMLS

String Literal Prefix

UltraEdit supports specification of the string literal format by specifying a string prefix character in the language definition line. For example, in C# .NET 2.0, the string literal prefix would be defined as follows:

String Literal Prefix = @

In this case, the @ before a string literal indicates that a backslash is not an escape character which is really useful for entering path literals. To illustrate, the following two statements are equivalent:

"c:\\data\\"

@ "c:\data\"

The only special character in a @ "..." literal is the ", which is simple doubled if you want to embed one.

Language Markers

UltraEdit has improved syntax highlighting and it now supports multiple languages within a single file. This is specifically for HTML type files. To help facilitate this, we have added additional language indicators that should be added to the wordfiles (*.uew) to indicate the type of language for any languages that may be included within another. Our default wordfiles have these modifications.

Example:

If an HTML file includes PHP then the syntax highlighting section must exist in the main wordfile and the PHP section should include in the definition line: PHP_LANG

Unambiguous language markers have been added to the default wordfile for the following languages:

C_LANG	// C/C++
COBOL_LANG	// Cobol
FORTTRAN_LANG	// Fortran
PASCAL_LANG	// Pascal
PERL_LANG	// Perl
PLB_LANG	// Plb
VB_LANG	// Visual Basic
VBSCRIPT_LANG	// Vb Script
ASP_LANG	// ASP
CSHARP_LANG	// C Sharp
CSS_LANG	// CSS
LATEX_LANG	// TeX / LaTeX
HTML_LANG	// HTML
JAVA_LANG	// Java
JSCRIPT_LANG	// Javascript
ECMA_LANG	// Ecma / EcmaScript
PHP_LANG	// PHP
MATLAB_LANG	// MATLAB
PUREBASIC_LANG	// PureBasic

UltraEdit_UESstudio_Help

PYTHON_LANG	// Python
SQL_LANG	// SQL
XML_LANG	// XML
XSL_LANG	// XSL
MASM_LANG	// Microsoft Assembler
AASM_LANG	// AT&T Assembler
NASM_LANG	// Netwide Assembler

Currently, UltraEdit uses the above language markers to correctly syntax highlight multiple languages within a file. In the future UltraEdit may make further use of these languages markers in the Wordfile.

NOTE: If your wordfile has multiple occurrences of the HTML_LANG language marker, this must be corrected or some languages may not highlight correctly. UltraEdit allows only one instance of a language marker (i.e., C_LANG, COBOL_LANG, HTML_LANG) in a wordfile.

The following file extensions were moved to the HTML section of the default wordfile: php, asp, and jsp. This was done to facilitate correct HTML highlighting at the outer language level of the file even when no <HTML> language tag is specified within the file. All php, asp, and jsp sections of the files are highlighted correctly based on language start and end tags (<? and ?> for example), however a "View As" will show the file as HTML type.

IntelliTip support

Style keywords have been added to the wordfile to identify color groups for IntelliTip support. Currently the use of the style elements is limited, but will find expanded use in future enhancements. A complete list of style elements follows:

STYLE_KEYWORD	// Keyword
STYLE_FUNCTION	// Function
STYLE_EXTENSION	// Extensions
STYLE_IDENTIFIER	// Identifier

STYLE_OPERATOR	// Operator
STYLE_METHOD	// Methods
STYLE_EVENT	// Events
STYLE_STATEMENT	// Statements
STYLE_TAG	// Tag
STYLE_VARIABLE	// Variable
STYLE_ATTRIBUTE	// Attributes
STYLE_ELEMENT	// Element
STYLE_COMMAND	// Command

Highlight new file as

Using the dropdown for this feature the user may set a default language for syntax highlighting from the list of languages defined in the active wordfile. This language will be used as the default for unsaved files.

Spell checking

By default UltraEdit and UESstudio will not perform "spell check as you type" on any syntax-highlighted files. To change this setting and allow "spell check as you type" for a syntax-highlighted file, add the following to the language definition line (e.g., /L3"HTML" ...) of the wordfile:

```
EnableSpellasYouType
```

The appropriate spell check option must also be selected in [Configuration](#) to support this.

The wordfile(s) may be modified by the user.

Application Layout

Advanced (Configuration - Application Layout)



The Advanced branch under Configuration - Application Layout has the following options:

Old-style window list

If this option is selected, open files will be dynamically listed at the bottom of the Window menu based on the order in which they were opened. If this option is not selected, open files will be listed here, but their order will vary based on a last active file order.

Disable all Workspace "Close" buttons

This option toggles the state of close buttons for peripheral windows (File Tree View, ASCII Table, Function List, Macro List, Tag List, Template List, Clipboard History, Script List, Bookmark List, Output Window, and SSH/Telnet Console).

Reset all "Don't Show Again" checkboxes

If this option is selected, all dialogs with a "Don't show this window again" option are reset to an unchecked state so they will be visible the next time the associated option is invoked.

Confirm removing File/Directory from Favorites

If this option is selected, a confirmation dialog will be shown whenever a file or directory is selected for removal from the **Favorites** list shown under the **Lists** tab in the Workspace Manager.

Display prompt opening read-only project

If this option is selected, a confirmation dialog will be shown when a project is loaded indicating that the project file is read-only. The user will be prompted to decide if they wish to make the project file writeable or not. If the project file is not changed to be writeable, then changes made to the project will not be saved when the project is closed.

Use Registry for settings (not INI file) - This will NOT move the existing items to the registry

This setting causes the editor to save and retrieve all settings from the Registry. After the editor determines the INI file in use it checks for this value before reading any other values. If this is set the editor stores all settings in the Registry.

Store windows/menu/toolbars placements and configurations for multiple instances of application

If this item is checked and a second instance of the editor is open and window sizes/positions or menu/toolbar configurations are changed, the next time a second instance of application is opened, it will use the separately saved sizes and positions.

INI File Location

This block displays the full path to the INI file being used by the current instance of UltraEdit or UESudio. This text may not be edited by the user, but is provided to facilitate locating the INI file when necessary.

File Tabs

Miscellaneous (Configuration - Application Layout - File Tabs)



The Miscellaneous branch under Configuration - Application Layout - File Tabs has the following options:

Double-click tab will close file

If this option is selected double clicking on an open file tab will close the associated file. This option is selected by default.

Middle-button-click will close file

If this option is selected clicking the middle mouse button on an open file tab will close the associated file. This option is selected by default.

Move to nearest left tab after current tab is closed

If this option is selected focus will shift to the nearest left tab when the active file is closed. If this option is not selected, focus will shift to the tab associated with the last active file when the active file is closed.

Sort Tabs on File Open

If this option is selected file tabs will automatically be sorted alphabetically as each file is opened.

Operation (Configuration - Application Layout - File Tabs)



The Operation branch under Configuration - Application Layout - File Tabs has the following options:

Dockable tabs

If this option is selected, the file tabs will be displayed in a dockable control that may be moved outside of the main GUI and arranged and resized as desired.

Types

Single line - scrollable

If this option is selected, file tabs will be displayed in a single row. When the file tabs for the number of files opened exceeds what can be displayed on a single line, left and right scroll arrows will be displayed at opposite ends of the tab row.

Single line - drop down file list

If this option is selected, file tabs will be displayed in a single row. In this mode a list of currently open files may be accessed by clicking on the down arrow at the end of the tab row.

Multiline

If this option is selected, the number of lines used to display file tabs for the open files will automatically expand and contract as necessary based on the number of open files.

Close Tab Button

Users may select one of three options related to the display of an "x" close button for file tabs. If **None** is selected, a close button is not displayed. If **On tab bar** is selected, an "x" will be displayed at the end of the file tab row and users may click on this to close the active file. If **On active tab** is selected, an "x" will be displayed on the active file tab.

Colors (Configuration - Application Layout - File Tabs)



The Colors branch under Configuration - Application Layout - File Tabs has the following options:

Tab colors by file specification

If this option is selected the background color of file tabs be changed to indicate the file type. Users may assign file types to the colors available in the table by double-clicking on a color line and adding an extension (i.e. "*.html"). Additional extensions should be listed immediately following the preceding extension and a semi-colon ";" (i.e. "*.php;*.php3;*.html;*.js"). If preferred, users may specify colors for individual files rather than extensions, i.e. "ResolveLine.cpp".

File Tree View (Configuration - Application Layout)



The File Tree View branch under Configuration - Application Layout has the following options:

Show volume labels in Explorer view

If this option is selected, volume labels rather than just drive letters will be displayed under the **Explorer** tab in the File Tree View.

Network drive label format

If the option above is selected, the format defined in this field will determine how the volume label is displayed under the **Explorer** tab in the File Tree View.

Show registered file type icons in Explorer view

If this option is selected, system registered icons will be used under the **Explorer** tab in the File Tree View rather than generic file type icons.

Scan for empty folders in Explorer view

If this option is selected, the editor will detect empty folders in the Explorer view in the File Tree View. Those folders determined to be empty will NOT have a "+" shown beside them (indicating that they may be expanded). If this option is not selected, the "+" may or may not be accurate, but will not be updated until the editor attempts to update/expand the selected folder.

Cache volume info in Explorer view

When the mouse hovers over a drive in the Explorer view, the editor attempts to get the latest information about that drive and display it. If this option is selected, drive information is cached and the drive is not accessed so often.

Activate file in Open file list on single click

If this option is selected, a single click in the Open file list in the File Tree View will cause the referenced file to become the active file in the editor. By default this option is not selected and a double click is required.

Resolve Link files and directories in Explorer view when opened

If this option is selected, linked files will be opened when the lnk file is clicked on in the Explorer view, and the Explorer view will navigate to the linked directory when the lnk directory is clicked on in Explorer.

Dockable Windows (Configuration - Application Layout)



The Dockable Windows branch under Configuration - Application Layout has the following options:

Time-out before Auto-hide (in ms)

This field is used to specify the number of milliseconds of delay that will pass before an auto-hide occurs when a child window which is configured to auto-hide loses focus.

Auto-hide scrolling

The slider here can be used to adjust the speed used to close a child window when set to auto-hide. If the slider is moved as close as possible to **Disabled**, the scrolling will be very fast. As the slider is adjusted more towards **Smooth**, the scrolling will seem to slide out of the edit window space more gradually.

Miscellaneous (Configuration - Application Layout)



The Miscellaneous branch under Configuration - Application Layout has the following options:

Minimize on last file close

If this option is checked the editor will minimize when all open files are closed.

Minimize on System Tray (not Task Bar)

If this option is checked the editor will minimize to the System Tray and NOT to the task bar.

Always open editor maximized

This setting determines if the editor should always open its main window in a maximized state. The default is for the previous settings to be remembered. If this setting is checked the editor will open the main window in a maximized state.

Display file name only in document title

This setting allows the user to specify that the editor should only display the filename only and not the path in the title bar for the editing window. If this is not selected the filename and path will be displayed (default). If this setting is selected only the filename will be displayed.

Show CHM user help files always on top

This setting allows users to force the editor to always show user defined CHM Help files as the top-level window when opened.

Alternate method to open CHM files (if cached file opens)

If two user CHM files with the same name are created on different paths it is possible that the last one used will be opened (rather than the selected CHM file). If this is the case, this option should be selected which will flush the cache when user CHM files are opened, thus avoiding this issue.

Animate configuration dialogs

This setting animates configuration dialogs both in the Advanced menu and (in UEStudio) under Project Settings.

Check and show file associations on startup

This setting causes the file association dialog to be shown each time the editor is started. If this is not selected File Associations may only be changed under the File Associations branch under Configuration - File Handling in the Advanced menu.

Allow multiple instances

The Multiple Instances item determines if another instance of the editor is opened if the editor is invoked when an instance is already active. If this option is enabled a new instance is opened each time the editor is invoked. If this option is not enabled only a single instance of the editor is opened, and any additional requests result in the first instance becoming active with the new file loaded in addition to any files already open.

Files selected in Explorer will open in a new single instance

If this option is selected, and multiple files are selected at once in Windows Explorer, and UltraEdit is selected from the context menu, all selected files will be opened in a single new instance. If this option is not selected, each file selected may be opened in a separate instance.

Toolbars/Menus

Customization (Configuration - Toolbars/Menus)



The Customization branch under Configuration - Toolbars/Menus includes the following items:

Customize Toolbar

Pressing this button presents the Toolbar Configuration dialog which consists of two panes labeled Toolbars and Commands along with a checkbox for Large Buttons and text entry fields for New Toolbar and Ext. Bmp. Also included in the dialog are up/down and right/left arrows. These items and their functionality are all described below:

Toolbars/Commands Configuration Panes

The toolbars which have been specified are displayed in the Toolbars pane with the items currently assigned to them in an Explorer interface. Clicking on the "-" to the left of a toolbar name will collapse that toolbar and clicking on the "+" will expand the toolbar to display the items included in it. If an item is selected in a toolbar, pressing the up/down arrow will move the item accordingly in the toolbar list. It is not possible to use the up/down arrows to move an item from one toolbar to another adjacent toolbar.

The Commands pane displays all the functions currently available for use in the toolbars. To move an item from the Commands pane to the Toolbars pane, click on the desired Commands item and press the left arrow in the center of the dialog to add the selected item. Selected toolbar items will be added directly below the currently selected item in the Toolbars pane. If there is no selection in the Toolbars pane, the item will be added at the top of the first toolbar listed. Once items are added they may be repositioned within a toolbar as desired using the up/down arrows.

New Toolbars

To create a new toolbar, type the toolbar name (i.e. "Test1") in the New Toolbar field and press the left arrow button to the right of the New Toolbar field. This will add a new, empty toolbar to the bottom of the Toolbars list. Toolbars may be removed by selecting them and pressing the right arrow button in the center of the dialog.

Ext. Bmp

If desired, new bitmaps may be associated with toolbar items (such as User Tools) so that this is displayed in the toolbar. To do so, select the toolbar item in the Commands pane and add it to the appropriate toolbar as described above. Select the item in the Toolbars pane and specify the path to the desired bitmap in the Ext. Bmp field. If you know exactly where this is located you may directly specify this and press Return to assign this bitmap to the selected toolbar item. If you're not sure where this is located, you may press the "..." button to browse to the desired bitmap and then press Return to assign this bitmap to the selected toolbar item.

Large Buttons

Selecting the Large Buttons checkbox will set ALL toolbars to use large buttons.

Manage Toolbars

Pressing this button presents the Toolbar Configurations dialog.

Customize Menus

Pressing this button presents the Menu Configuration dialog which consists of two panes labeled Menus and Commands along with a text entry field for New Menu/Submenu. If Use check marks on menus is selected, check marks will be shown for selected menu items rather than the depressed menu icons. This option helps resolve some issues seen with screen readers for disabled users. Also included in the dialog are up/down and right/left arrows and an Edit button. These items and their functionality are all described below:

Menus/Commands Configuration Panes

The currently active menus are displayed in the Menus pane with the items currently assigned to them in an Explorer interface. Each menu or submenu heading is indicated by a line terminated in a square to the left of the menu/submenu name. Double clicking on the heading indicator will cause the included items to be expanded/collapsed appropriately.

If a menu heading or item is selected and the Edit button is pressed the name for the menu/item may be modified. Pressing Return on the keyboard will save the menu item change and pressing the OK button will commit the change so that this is reflected in the menus.

When a menu heading is selected the entire menu may be moved to a new location using the up/down and right/left arrows. When a menu item is selected that item may be moved to a new location within the current menu by pressing the up/down buttons. A menu item may be moved to the extreme top or bottom of its current menu. If a menu item is part of a submenu, it cannot be moved outside the submenu. If a menu item is part of a main menu, moving it up or down will cause it to "skip" any submenus it encounters.

The Commands pane displays all the functions currently available for use in the menus. To move an item from the Commands pane to the Menus pane, click on the desired Commands item and press the left arrow in the center of the dialog to add the selected item. Selected menu items will be added directly below the currently selected item in the Menus pane. If there is no selection in the Menus pane, the item will be added at the top of the first menu listed. Once items are added they may be repositioned within a menu as desired using the up/down arrows.

New Menus

To create a new menu/submenu, type the menu/submenu name (i.e. "Test1") in the New Menu field and press the left arrow button to the right of the New Menu field. This will add a new, empty menu to the bottom of the Menus list. Menus may be removed by selecting them and pressing the right arrow button in the center of the dialog.

The data for the configurable menus is stored in the % APPDATA% directory by default under IDM Comp\UltraEdit in the uedit32.mnu file. (For UESstudio this would be under the UESstudio directory rather than the UltraEdit directory).

Customize Popup Menu

Pressing this button presents the Popup Menu Configuration dialog which functions exactly as the Menu Configuration dialog described above except that this allows the user to configure what is displayed in the right-click context menu.

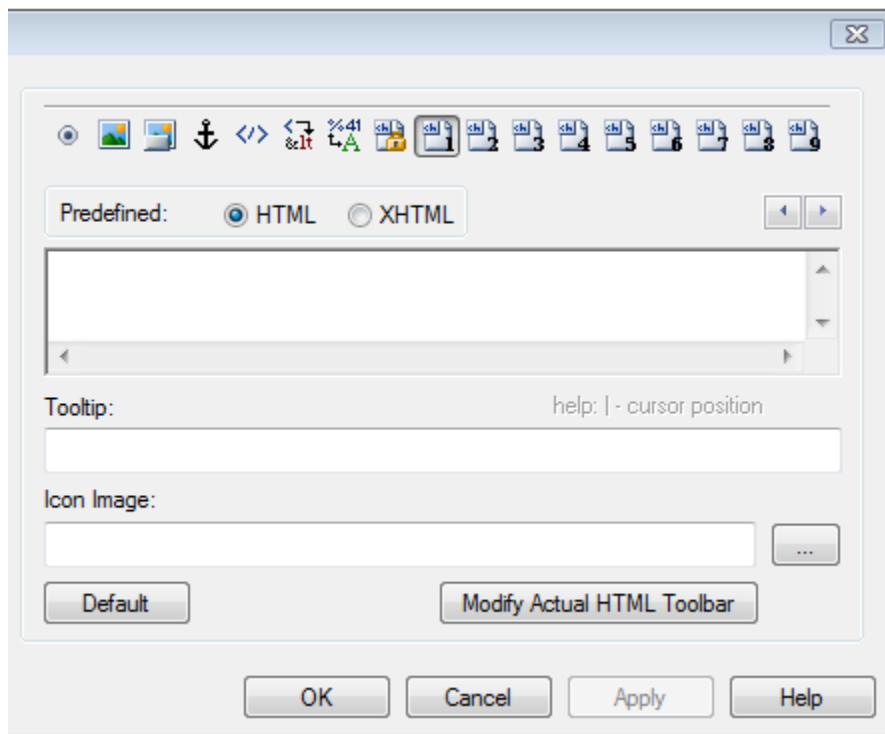
Manage Menus

Pressing this button presents the Menu Configurations dialog.

HTML Toolbar (Configuration - Toolbars/Menus)



The HTML Toolbar branch under Configuration - Toolbars/Menus provides an interface where users may customize the HTML text inserted when HTML buttons are selected in the [HTML toolbar](#).



When a button is selected from the toolbar row, the text associated with this button will be displayed in the edit window and users may modify this text and save it to customize what is inserted by pressing the selected HTML button. By default, the **HTML** option will be selected in the **Predefined** section and the default buttons will insert text conforming to the HTML 4 standard. If the **XHTML** option is selected, the default buttons will insert text conforming to the XHTML standard.

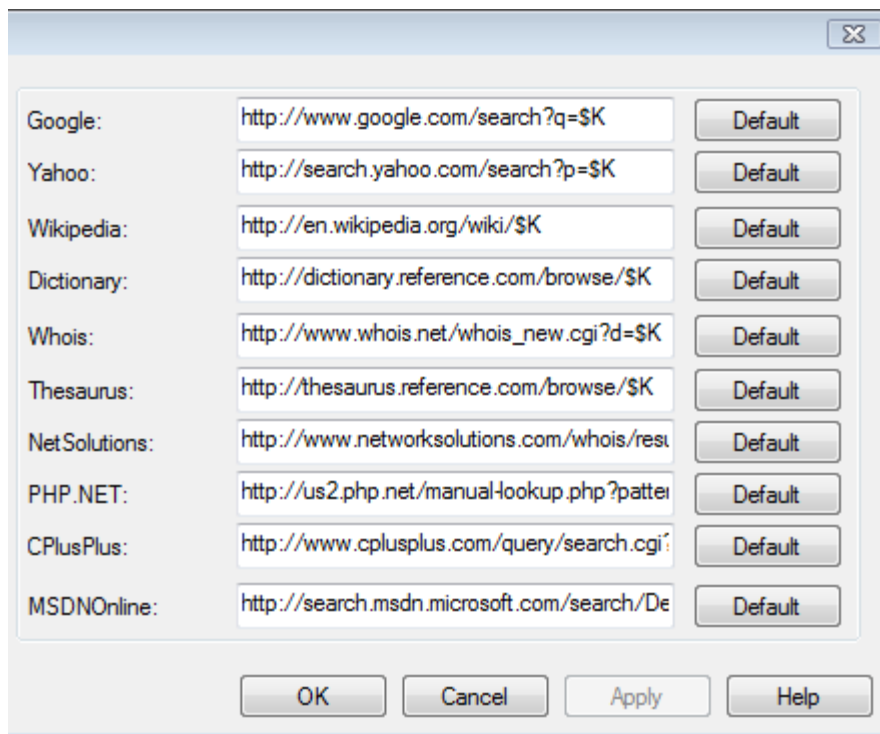
Users may also associate custom bitmap images with new "blank" toolbar buttons and specify what text should be inserted when pressing these buttons. Users may specify where the cursor should be presented when an element is inserted into the active document by using "|" in the text area, as is supported for templates. For example typing "< td>|</ td>" would present the cursor or any currently selected text between the specified HTML elements. "\n" should be used in the template text to represent a newline. If a tab is desired, users should copy an actual tab character to the Windows clipboard prior to opening the HTML toolbar configuration dialog and paste the actual tab character into the template text.

If the **Modify Actual HTML Toolbar** button is pressed, the [Toolbar Configuration dialog](#) will be presented with the HTML Toolbar selected for customization.

Web Search (Configuration - Toolbars/Menus)



The Web Search branch under Configuration - Toolbars/Menus provides an interface where users may customize the search strings used when a search for text selected in the application is executed through the [Web Search toolbar](#).



Through this interface users may localize (if desired) the "portal" through which their search is executed. For example, a user located in Germany may wish to search on "www.google.de" rather than the U.S. portal - "www.google.com". This may be accomplished by changing the "Google" search string to:

`http://www.google.de/search?q=$K`

In all cases, the "\$K" represents the string selected in the editor at the time the search is executed. If desired, the default search strings may be reinstated for each option by pressing the associated **Default** button and then pressing **Apply** or **OK** to save the change.

Users may also define custom search buttons to be added to toolbars as desired. For further information on how to do this, please go to http://www.ultraedit.com/support/tutorials_power_tips/ultraedit.html and check the **Integrate Yahoo!, Google, Wikipedia and more with UltraEdit** topic.

Management (Configuration - Toolbars/Menus)



The Management branch under Configuration - Toolbars/Menus includes the following items:

Manage Toolbars

Pressing this button presents the Toolbar Configurations dialog. This consists of a list of available toolbars, a text entry field for naming a new toolbar as well as the following buttons:

Save As - saves active toolbar with newly specified name

Delete - deletes currently selected toolbar

Reset - resets the active toolbar to its default state

Reset All - resets all toolbar to their default state

Close - closes dialog

Manage Menus

Pressing this button presents the Menu Configurations dialog. This consists of a list of available menus, a text entry field for naming a new menu as well as the following buttons:

Save As - saves active menu with newly specified name

Delete - deletes currently selected menu

Reset - resets the active menu to its default state

Reset All - resets all menus to their default state

Close - closes dialog

Miscellaneous (Configuration - Toolbars/Menus)



The Miscellaneous branch under Configuration - Toolbars/Menus has the following options:

Project menu on main menu

This option configures the editor to include the Project menu on the main menu bar of the editor. If this option is not checked the Project menu appears as a submenu of the File menu. If this setting is changed, the application will have to be shut down and restarted for the change to take effect.

Recent files list on file menu

This option configures the recent file list to appear on the File menu rather than as a submenu of the File menu. If this option is not checked the recent file list appears as a submenu of the File menu. If this setting is changed, the application will have to be shut down and restarted for the change to take effect.

Number of recent files on file Menu

This setting sets the number of recent files that appear on the File menu for quick opening of files.

Clear recent document/project and search/replace histories on exit

If this option is selected the [Recent Files](#) list and [Recent Projects/Workspaces](#) lists in the File menu and the stored search/replace strings will automatically be cleared upon exit.

Clear History (Remove recent document/project and search/replace histories)

When this button is pressed the [Recent Files](#) list and [Recent Projects/Workspaces](#) lists in the File menu and any stored search/replace strings are immediately cleared.

Directories (Configuration)



The Directories branch under Configuration allows the default directories used by the editor to be configured. The following options are supported:

Default save directory	sets the default directory for saving files with the Save As Command
Macro default directory	sets the default directory for loading and saving macros
Template directory	sets the directory for template storage
Default open directory	sets the default directory for opening files with the Open command
Default project file directory	sets the default directory for creating new projects with the New Project/Workspace command

Default Save Directory

Normally the directory used for the [Save As command](#) is the default directory which may vary depending on which directory was last used for a file open and may be any directory.

This may be overridden by specifying the default directory that is always presented when the [Save As command](#) is used. If this item is left blank then the default directory is the default directory for the process.

Key Mapping (Configuration)



The Key Mapping branch under Configuration allows the user to reassign the default key mapping provided by UltraEdit/UEStudio to suit their particular needs:

This dialog shows a list of commands within the editor that may be mapped by the user to a key combination of their choice. Clicking on any of the commands will show the existing key assignment (if any) and the command description below the list. If the **Sort** checkbox is selected the commands will be sorted alphabetically. If this is not selected, commands will be sorted logically based on menu order.

To add or change a command key assignment:

- 1) Select the command you wish to modify from the Commands list.
- 2) Clicking the mouse in the Press New (Multi-)Key entry window area, or TAB to give it focus.
- 3) Type the new key assignment just as you would if you were invoking the command - This key combination will be captured and displayed. Users may use up to three keys (i.e. "Ctrl+Shift+T") for multi-key assignments in each field.
- 4) Click on the Assign button and the new key will be assigned to the command. If an existing key is assigned you will be asked if you wish to delete it and replace with the new assignment.

Chords

Users may use both text fields under **Press new (multi-)key** to define "chords". For example:

Ctrl+R

M

could be used to toggle a bookmark when "CTRL+R, and M" are pressed in quick succession.

The **Multi-key delay** value may be set to allow the user time to enter multi-key shortcuts rather than having the first portion of a multi-key shortcut processed on its own.

To remove a key assignment:

- 1) Select the command you wish to modify from the Commands list.
- 2) Select the existing key assignment from the Existing keys list.
- 3) Click on the Remove button and the key assignment will be removed.

To reset the key assignments to the default:

- 1) Click on the Defaults button and the key assignments will be reset to the defaults. The users will be prompted to confirm the action.

File Containing User Keyboard Mapping

The keyboard mapping file is stored (and searched for when the editor starts up) in the same directory as the INI File. The filename is the same as the INI with the . UEK extension, i.e. UEDIT32.UEK or UESTUDIO.UEK.

Tab Key moves focus from "New Key" entry

If this option is selected pressing the TAB key while focus is on the "New Key" entry field will move focus out of this field rather than including "TAB" in the mapping for the selected command.

Show key mapping in editor

When this button is pressed a listing of all currently configured keyboard shortcuts for use with UltraEdit/UEStudio will be written to a file with the format "[Editor Name].shortcuts". This includes standard keyboard shortcuts, menu shortcuts, and macro hotkeys. If **Sort by Keys** is selected this list will be sorted based on the shortcut keys for each function. If **Sort by IDs** is selected this list will be sorted based on the command names. If desired, this list may be printed for reference.

FTP (Configuration)



The **FTP** branch under Configuration has the following options:

Store FTP accounts and settings in user selected file

This option may be used to allow the user to specify a file which will be used to store FTP account options. The user may define this directly in the provided textbox or browse to select a file that already exists. If this option is not selected, FTP account information will be stored in the INI file as usual.

Display FTP accounts in explorer view

If this option is selected, FTP accounts will be shown in the File Tree View/Workspace Manager under the Explorer tab. The root of the account will be listed as a top level directory and may be expanded to show files/folders under this account.

Show FTP log in output window

If this option is selected an FTP log showing all commands sent to and data received from the FTP server will be displayed in the output window. This can provide helpful diagnostic information and the contents of this window may be copied using the clipboard.

Scripting (Configuration)



The **Scripting** branch under Configuration has the following options:

Show status information in output window

If this is selected all status information including script name, whether the script succeeded or failed, and error information is printed to the output window. If the output window is closed but this option is selected or if the showOutput property is true (which it is by default) then data is still written to the output window but the output window is not automatically displayed..

IDE

IntelliTips

Auto-complete (Configuration - IDE - IntelliTips)



The Auto-complete branch under Configuration - IDE - IntelliTips has the following options related to auto-complete in UESstudio:

Show Symbol List automatically

If this option is selected the auto-complete box (including symbols) will be displayed automatically after the specified number of characters have been saved.

Insert Smart Space after completed word

If this option is selected a space will be inserted automatically after the accepted word is inserted.

Auto-complete One-Symbol List

If this option is selected and the auto-complete list contains only one unique item which matches the criteria specified for typed characters, then the item from the auto-complete list will automatically be inserted without prompting the user.

Delete "trailing" chars on Auto-complete

If this option is selected and an auto-complete string is immediately followed by a non-matching string, the non-matching string will be deleted from the active document when the auto-completed word is selected. For example ("|" represents cursor) the user is typing:

ThisIs|Enum

and presses **Enter**, the auto-complete function would update the string to **ThisIsVariable** and the **Enum** portion of the active string would be deleted.

Disable this feature for

If the users wishes, the settings selected above may be disabled for strings occurring within comment blocks or strings (as defined by the **String Chars** setting in the associated wordfile).

Function Tips (Configuration - IDE - IntelliTips)

The Function Tips branch under Configuration - IDE - IntelliTips has the following options:

Show automatically (Redisplay Tip: Ctrl + Shift + Space)

Function Tips for parameter list display are automatically displayed upon typing a function name followed by the parameter list open character (such as open parenthesis). It can be displayed again by typing Ctrl+Shift+Space.

Use Function Tips data (if available) for Function List

If this option is selected the built-in parser in UESTudio will be used to identify functions in the active file rather than the **Function Strings** defined in the wordfile. The file to be parsed must be part of the active project and the file extension must be defined under **Configuration - IDE - Parser** for this to work properly. If the file type and extension aren't specified under the **Parser** branch then UESTudio wouldn't be able to automatically parse the functions and the **Function String** definitions from the wordfile would be used.

Miscellaneous (Configuration - IDE - IntelliTips)

The Miscellaneous branch under Configuration - IDE - IntelliTips has the following options:

Insert ")" automatically after "["

If this option is selected, a close brace, i.e. [] and (), will be inserted immediately after the open brace is typed with the cursor displayed between the two braces.

Insert "}" automatically after "{"

If this option is selected, a close brace, i.e. { }, will be inserted immediately after the open brace is typed with the cursor displayed between the two braces.

Smart placement of "}" when inserted automatically

If this option is selected, UESudio will automatically insert a "}" after a "{" is typed. This function will first check to see if the "{" is typed to define an initialization set, or a function (based on C/C++ syntax). If it is a function, a line terminator will be inserted and the "}" will be inserted on the next line at the current indent level. If the "{" is typed to initialize a data set, it will add the "}" as the next character and place the cursor between the curly braces exactly as the Insert "}" automatically after "{" option does.

Insert () when complete function

If this option is selected, "()" characters will be inserted automatically when a function name is completed. A function is recognized based on the namespace as shown under Globals -> Functions under the Tags tab in the Workspace Manager.

Change ".." with "->"

If this option is selected, then typing ".." will produce "->" in any file. This is typical "C/C++" syntax for a pointer to a structure member rather than referencing the structure/structure member directly (i.e. "struct.member").

Automatically insert line-comment at new line

If this option is selected and the user edits content in a commented line and this wraps to a new line, the new line will automatically be commented. This allows the user to write multi-line comments without stopping to insert the line comment character.

Parser (Configuration - IDE - IntelliTips)



The Parser branch under Configuration - IDE - IntelliTips has the following options:

Auto re-parse documents when saved

When this option is selected UESTudio will automatically re-parse the saved document for items to be provided in the IntelliTip list.

Parse extensionless files as

Users may specify a file extension to be used by default for parsing for files that are saved without an extension.

Do not use cache for symbols

This option will force UESTudio not to use the cache for symbols in the active project. By default this option is not selected. Check this only if you have very low disk space.

The cache is in the % APPDATA% directory. If the cache is not used then the parser has to re-parse everything every time rather than working based on dependency checks (when files are updated or changed). Therefore the use of the cache is recommended for performance reasons.

Clear Cache

This button may be used to force UESTudio to clear the cache.

Quick Tips (Configuration - IDE - IntelliTips)



Quick Tips display a suggestion for the word being typed, based on words previously found in the document that begin with the same characters. The first word found, scanning back from the cursor towards the top of the document, is the word that is suggested.

The **Quick Tips** branch under Configuration - IDE - IntelliTips has the following options:

Show Tip "x" ms

This option is set in milliseconds and controls how long an IntelliTip tip will be displayed without user interaction.

Scan back "x" chars (set chars=0 to disable quicktips)

This option is set in characters and controls how many characters back from the current cursor position UESstudio will read to populate the IntelliTip list. This is set to 100,000 characters by default.

Complete with Tab/Enter

This option controls what keys may be used to accept the suggested word (TAB or ENTER).

Debugger

WinDbg (Configuration - IDE)



The **WinDbg** branch under Configuration - IDE includes options for downloading **Debugging Tools for Windows**. The items presented for download in this dialog are downloaded directly from microsoft.com. For further information please see <http://www.microsoft.com/whdc/devtools/debugging/default.mspx>.

Download debugging tools for Windows

WinDbg for Windows, 32bit version 6.9.3.113 [17.5MB] **1**

From the list select 32/64bit debugger and press button. Download

Download Windows symbol packages

Windows Server 2003 symbols with no Service Pack **2**

Itanium checked symbols, all languages [123MB] Download

WinDbg

Path: WinDbg.exe Browse

Cmd:

☒ Use WinDbg as default debugger **3**

☐ Enable postmortem debugging

OK Cancel Apply Help

Debugging Tools for Windows is a set of extensible tools for debugging device drivers for the Microsoft® Windows® family of operating systems. Debugging Tools for Windows supports debugging of:

- Applications, services, drivers, and the Windows kernel.
- Native 32-bit x86, native Intel Itanium, and native x64 platforms.
- Microsoft Windows NT 4.0, Windows 2000, Windows XP, Microsoft Windows Server 2003, and Windows codenamed "Longhorn/Vista".
- User-mode programs and kernel-mode programs.
- Live targets and dump files.
- Local and remote targets.

System Requirements

The following are system requirements for the 32-bit version of Debugging Tools for Windows:

- 32-bit or 64-bit Microsoft Windows Server 2003, 32-bit or 64-bit Windows XP, Windows 2000, Windows NT 4.0, or Windows codenamed "Longhorn/Vista".
- Microsoft Internet Explorer 5.0 or later.
- Approximately 25 MB of hard disk space.

In addition to **Debugging Tools for Windows**, effective debugging also requires:

- Special debugging routines, macros, and global variables in the Windows Driver Development Kit (DDK). You can use these routines in your driver code to send messages to a debugger and set breakpoints to aid in debugging.
- Access to Windows symbol files. If you have access to the Internet while debugging, you can set your debugger's symbol path to point to the Windows symbol server. If you do not have access to the Internet while debugging, you can download symbols in advance from the Microsoft website.

Debugging Tools for Windows includes **WinDbg**, a powerful debugger with a graphical interface and a console interface, as well as the console-based debuggers NTSD, CDB, and KD.

Download Debugging Tools for Windows [1]

Please note: The 32-bit version of **Debugging Tools for Windows** is the best choice, unless you are debugging an application on a 64-bit processor. In that case, you should use a 64-bit package. Please select the item appropriate to your needs and press the **Download** button.

To Prepare for a Debugging Session

1. If you want to debug kernel-mode code, set up a host computer to run the debugger and a target computer to run the code being debugged. Connect the two computers using either an IEEE 1394 cable or a null-modem cable between two serial ports.
2. Download and install the latest version of Debugging Tools for Windows on the host computer. The tools can also be installed from the Windows DDK, the Platform SDK, or the Customer Support Diagnostics CD.
3. Use Windows Symbol Server to access the symbol files from the host computer.

Download Windows symbol packages [2]

If you want the entire set of symbols for the next version of Windows, codenamed "Longhorn/Vista," Windows Server 2003, Windows XP, or Windows 2000, then you can download a symbol package and install it on your computer.

The symbol download packages are listed by processor type (x86, Itanium, and x64) and build type (retail and checked). Almost all customers require the symbols for the retail version. If you are debugging a special version of Windows with extra debugging information, then you should download the symbols for the checked version.

Windows XP and Windows Server™ 2003 do not require localized symbols in order to debug localized versions of the product. Each Windows XP and Windows Server 2003 symbol download package works for debugging all localized versions.

Each x86 symbol package may require 750 megabytes (MB) or more of hard disk space, each Itanium symbol package may require 560 MB or more, and each x64 symbol package may require 640 MB or more. Because of the size of the download package and required temporary files, it is recommended that you have at least 1 GB of disk space free before downloading and installing each symbol package.

To use the Microsoft Symbol Server

1. Make sure you have installed the latest version of Debugging Tools for Windows.
2. Start a debugging session.
3. Decide where to store the downloaded symbols (the "downstream store"). This can be a local drive or a UNC path.
4. Set the debugger symbol path as follows, substituting your downstream store path for DownstreamStore.

SRV*DownstreamStore*http:// msdl.microsoft.com/download/symbols

For example, to download symbols to c:\ websymbols, you would add the following to your symbol path:

SRV*c:\ websymbols*http:// msdl.microsoft.com/download/symbols

WinDbg [3]

WinDbg provides source-level debugging through a graphical user interface and a text-based interface. WinDbg uses the Microsoft Visual Studio debug symbol formats for source-level debugging. It can access any public function's names and variables exposed by modules that were compiled with Codeview (. pdb) symbol files.

WinDbg can view source code, set breakpoints, view variables (including C++ objects), stack traces, and memory. It includes a command window to issue a wide variety of commands not available through the drop-down menus. For kernel-mode debugging, WinDbg typically requires two computers (the host machine and the target machine). It also allows you to remotely debug user-mode code.

Use WinDbg as default debugger

If this option is selected, **WinDbg** will be used as the default debugger.

Enable postmortem debugging

This option may be selected to debug failing programs that are already in production.

Xdebug

Download (Configuration - IDE - Debugger - Xdebug)



The **Download** branch under Configuration - IDE - Debugger - Xdebug includes options for downloading items related to the functionality of the integrated PHP debugger (Xdebug) in UEStudio. It is important to note that, once activated, the debug client works like a server by creating a port to which the debug engine can connect. Therefore, just starting the debugger does not necessarily mean a debug session will begin; the user must next run a script with debugging enabled.

Both **WampServer** (which includes Apache, PHP, and MySQL) and **Xdebug** are necessary to this process and should be downloaded and installed in the order referenced below:

Download WampServer for Windows

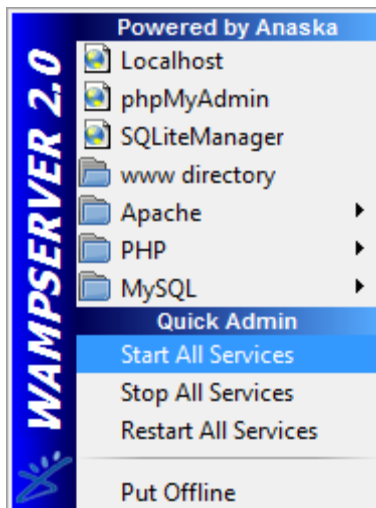
Currently this dropdown only offers "WampServer 2.0 for Windows". Pressing the **Download** button will begin the download of the associated files to your system. This is necessary as this includes the Apache web server, PHP scripting engine and MySQL which allow you to run PHP scripts on your local system for testing. For more information please see: <http://www.wampserver.com/en/>.

If desired, other packages such as XAMPP (<http://www.apachefriends.org/en/xampp.html>) may be used instead of WampServer for Windows, or users may simply install the Apache web server separately, but we have found WampServer to be exceptionally easy to install and maintain on Windows OS's and will describe configuration of Xdebug for browser-based debugging based on the assumption that WampServer is installed.

Installing WampServer

WampServer is installed by invoking the downloaded EXE. The only information that must be provided during installation is the email address you would like to be used by the PHP mail commands. The server name would normally be left as "localhost".

When the installation completes, launch WampServer and select **Start All Services** from the control menu:



This will start Apache, PHP and MySQL on your system. To test this, launch a browser and type this URL:

<http://localhost/>

This should populate the browser with the active server configuration for WampServer and will indicate the versions of Apache, PHP and MySQL that are active and the related extensions.

Download PHP for Windows

This download link should be used by those who only work with command line based PHP scripts. For these users, installing the Apache server and MySQL is not strictly necessary and this link will download the PHP scripting engine only.

Installing PHP

To install PHP on your system, please do the following:

1. Unzip the downloaded files to **C:\php**.

2. Copy php.ini-recommended php.ini

3. Open C:\php\php.ini

4. Modify the following items as shown:

```
upload_tmp_dir="C:\php\upload"
```

```
session.save_path="C:\php\session"
```

```
cgi.force_redirect=0
```

```
extension_dir = "C:\php\ext"
```

```
display_errors = Off
```

```
log_errors = On
```

```
error_log = "C:\php\error_log"
```

5. Enable the extensions you need by deleting the semicolon at the beginning of the line. These will depend on your application.

Example:

```
extension=php_bz2.dll
```

```
extension=php_curl.dll
```

6. Right-Click on Start > Computer.

- a. Choose "Properties", then "Advanced".
- b. Click on "Environment Variables".
- c. Click on PATH, then Edit... and add ";C:\php;c:\php\ext" (without quotes) to the end of the variable.

Download Xdebug for Windows

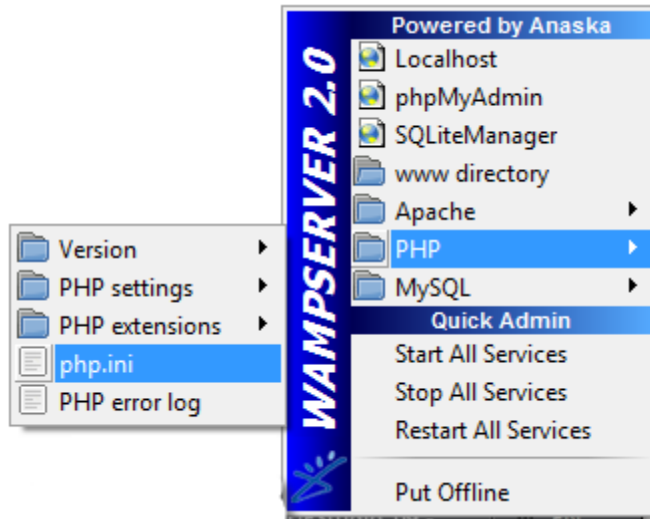
There are several build options for Xdebug offered in the associated dropdown. It is important to note which version of PHP is supported by the version of Xdebug you are downloading. You **MUST** select the version of Xdebug that supports the currently installed version of PHP or they will not work together. Pressing the **Download** button will begin the download of the associated files to your system.

Installing Xdebug

Once the dll for Xdebug is downloaded, it should be copied to the appropriate PHP extension directory on your system. If WampServer was installed to the default path, this would be:

```
C:\wamp\bin\php\php5.2.6\ext
```

Once the dll has been copied to the appropriate location, the php.ini file must be edited to configure support for Xdebug. The easiest way to do this is to access the php.ini file through the WampServer menu:



There are several php.ini files in the WampServer directories and this is the easiest way to be sure you are editing the correct one. Once you have opened this php.ini file, add the following lines at the end of the file:

```
;Xdebug location
```

```
zend_extension_ts="C:/wamp/bin/php/php5.2.6/ext/php_xdebug-2.0.3-5.2.5.dll"
```

Please note: The path to the dll must be modified as appropriate for your system. The following settings for the debugger also need to be included at the end of the php.ini file:

```
;Xdebug settings
```

```
xdebug.remote_autostart = 0
```

```
xdebug.remote_enable = 1
```

Please note: If the **remote_autostart** option is set to "1", debugging would start automatically every time a PHP script is detected as running after Build -> Start Debugger is selected.

Once you've made these changes, save the file and close it. Select the option to **Restart All Services** in the WampServer menu to activate the changes made to the php.ini file. After restarting the services, if you open a browser and navigate to **http://localhost/** you should see Xdebug listed as one of the loaded extensions.

For more on using Xdebug see: <http://devzone.zend.com/article/2803-Introducing-xdebug>.

Miscellaneous (Configuration - IDE - Debugger - Xdebug)



The **Miscellaneous** branch under Configuration - IDE - Debugger - Xdebug includes the following options related to the functionality of the integrated PHP debugger (Xdebug) in UESudio:

Startup

Start Xdebug using the active document

If this option is selected, Xdebug will automatically begin debugging the PHP document active in UESudio when **Start Debugger** is selected from the **Build** menu. This setting will cause UESudio to automatically check to see if the active script exists on or below the **Root web directory** defined below. If it does, then the default browser for the OS will be launched with the active script being run from **http://localhost/** and **?XDEBUG_SESSION_NAME=phpTest** will automatically be appended to the script URL to enable detection by Xdebug for debugging.

PLEASE NOTE: The Apache server must already be running when debugging begins for the script to be launched automatically in the browser. If the server isn't running, the URL provided to the browser won't be recognized as a valid URL.

If the active script does **NOT** exist on or below the **Root web directory**, it will automatically be launched and debugged as a console (command line) application using the [environment variable](#) defined during download and installation.

If this option is NOT selected, the script to be debugged would need to be launched in a browser after selecting **Start Debugger** from the **Build** menu, with the **?XDEBUG_SESSION_NAME=phpTest** flag appended or from a command prompt (which would automatically force the use of the environment variable defined for Xdebug interaction) so that UESudio can detect the running process on the specified port. Alternatively, the Xdebug settings in the php.ini file could be configured so that the `remote_autostart` option:

```
xdebug.remote_autostart = 1
```

is set to "1". If the **remote_autostart** option is set to "1", debugging would start automatically every time a PHP script is detected as running after Build -> Start Debugger is selected, and the **?XDEBUG_SESSION_NAME=phpTest** flag would not have to be appended to the script URL to enable debugging.

Root web directory

This field is subdued unless the option above is selected. In order for the active script to be recognized as a browser-based script (and to be launched automatically on the localhost) the root web directory for the localhost must be defined here. With the default WampServer installation this would be at **C:\wamp\www** but users should browse to the appropriate directory on their system if unsure of the exact location.

Port

By default, the port the Xdebug listens on is set to 9000. This port is configured in UESTudio by default, and if there is no compelling reason to do otherwise, you should keep the default port. If you want to change the port, keep in mind that you have to configure the same port number in UESTudio and the related php.ini.

Parser (Configuration - IDE)



The **Parser** branch under Configuration - IDE presents a table for defining the extensions associated with recognized parsers.

Default	Find (Parser/Ext):	Default
Parser	Handled Extensions	
Asm	.asm.nasm.s	
Asp	.asa.asp	
Awk	.awk.gawk.mawk	
BETA	.bet	
C	.c	
C#	.cs	
C++	.c++.cc.ch.chf.chs.cp.cpp.cxx.h.h++.hh.hp.hpp.hxx	
Cobol	.cbl.cob	
Eiffel	.e	
Erlang	.erl.hrl	
Fortran	.f.f77.f90.f95.for.ftn	
Java	.java	

To change list of extensions attached with the parser, double-click on the line.

Find (Parser/ Ext)

The **Find (Parser/Ext)** textbox may be used to search for matches among the recognized **Parsers** or **Handled Extensions**. If searching for a parser, this should be typed as expected, i.e. "Fortran". If searching for a supported extension, this should be preceded by ".", i.e. ". jav". The text field is interactive in that as soon as a matching item is found, the appropriate line in the Parser table will be highlighted. Users may double-click on a parser line to modify the extensions associated with that parser.

If desired, the **Default** button may be pressed to restore default extensions for all parsers.

Please note: It is not possible to add parsers not currently recognized by the IDE. This interface only supports defining the extensions associated with recognized parsers.

Integrated Applications

JavaScript Lint (Configuration - Integrated Applications)



The **JavaScript Lint** branch under Configuration - Integrated Applications includes the following option for JavaScript Lint:

Full path for JavaScript Lint configuration file

By default this field is populated with the path to a configuration file with default settings. If desired, the user may create a custom configuration file in another location and specify the path to that file here. Pressing the **Open** button will open the specified file for editing.

Set Code Page/Locale (Advanced menu)



This allows the user to specify the code page and the locale to be used by certain functions within the editor.

The initial selection shows the current settings. In all previous versions the "" C' Default" locale and code pages were used. These are the default values unless the user specifies something different.

If "" C' Default" is selected, this will apply to both the code page and locale.

If "System Default" is selected, the dialog box will show the actual system default if the dialog is invoked again.

The code page selection is used for conversion to/from Unicode. The locale selection is used specifically if selected with the [Sort Function](#).

Display/Modify Templates command (Advanced menu)



Use this command to display the dialog for editing the templates. A template is a predefined text that may be recalled automatically at a later time and inserted into a file.

Up to 50 templates may be defined. Each of the first 20 templates may be recalled at a later time using hot keys Alt+0 through Alt+9, and Shift+Alt+0 through Shift+Alt+9, or any of the templates may be recalled with the Insert Template dialog.

Each template may optionally have a name associated with it, that if defined will show up on the tool tips on the toolbar and in the list of templates when inserting a template.

The templates are stored in a file in the default directory or the directory specified under the [Directories](#) branch under Configuration in the Advanced menu.

A template may contain special text strings that are translated at the time the template is recalled. These special strings are:

[DATE_DMY]	Is translated into the current date with the format DAY/MONTH/YEAR.
[DATE_MDY]	Is translated into the current date with the format MONTH/DAY/YEAR.
[DATE_TEXT]	Is translated into the current date with the date displayed as text.
[TIME]	Is translated into the current time IN 24 HOUR FORMAT.
^	This indicates the position at which to place the cursor after the template has been inserted.
[FULL_FILE_NAME]	Is translated into the full path and filename of the active document.
[FILE_PATH]	Is translated into the drive and path of the active document.
[FILE_NAME]	Is translated into the root filename of the active document.
[FILE_EXTENSION]	Is translated into the file extension of the active document with the "point", i.e. ".txt".
[FILE_EXTENSION_NP]	Is translated into the file extension of the active document without the "point", i.e. "txt".

[\$REPLACE\$]	Is replaced with selected text in the active document. The selection will be converted to uppercase text when inserted with the template.
[\$replace\$]	Is replaced with selected text in the active document. The selection's current case will be maintained when it is inserted with the template.

Note - ^ refers to the character '^' NOT Control Key + value.

Additionally, for extended date and time formatting, UltraEdit supports the following enhanced functionality:

Time

To specify the USER TIME format use the following syntax in your template:

For local time use the following markers:

[TIME_USER]...time format string here...[TIME_USER_END]

For system time use the following markers:

[TIME_USER_SYS]...time format string here...[TIME_USER_SYS_END]

The time format string is defined as follows:

If you use spaces to separate the elements in the format string, these spaces will appear in the same location in the template. The letters must be in uppercase or lowercase as shown (for example, "ss", not "SS"). Characters in the format string that are enclosed in single quotation marks will appear in the same location and unchanged in the template.

String	Meaning
h	Hours with no leading zero for single-digit hours; 12-hour clock

hh	Hours with leading zero for single-digit hours; 12-hour clock
H	Hours with no leading zero for single-digit hours; 24-hour clock
HH	Hours with leading zero for single-digit hours; 24-hour clock
m	Minutes with no leading zero for single-digit minutes
mm	Minutes with leading zero for single-digit minutes
s	Seconds with no leading zero for single-digit seconds
ss	Seconds with leading zero for single-digit seconds
t	One character time marker string, such as A or P
tt	Multicharacter time marker string, such as AM or PM

For example, to get the time string

"11:29:40 PM"

use the following format string:

" hh':'mm':' ss tt"

This string **MUST** be between the markers, for example:

[TIME_USER] hh':'mm':' ss tt[TIME_USER_END]

Date

To specify the USER DATE format use the following syntax in your template:

For local date use the following markers:

[DATE_USER]...date format string here...[DATE_USER_END]

For system date use the following markers:

[DATE_USER_SYS]...date format string here...[DATE_USER_SYS_END]

The date format string is defined as follows:

Use the following elements to construct a format string. If you use spaces to separate the elements in the format string, these spaces will appear in the same location in the template. The letters must be in uppercase or lowercase as shown in the table (for example, "MM" not "mm"). Characters in the format string that are enclosed in single quotation marks will appear in the same location and unchanged in the template.

String	Meaning
d	Day of month as digits with no leading zero for single-digit days.
dd	Day of month as digits with leading zero for single-digit days.
ddd	Day of week as a three-letter abbreviation.
dddd	Day of week as its full name.
M	Month as digits with no leading zero for single-digit months.
MM	Month as digits with leading zero for single-digit months.
MMM	Month as a three-letter abbreviation.
MMMM	Month as its full name.
y	Year as last two digits, but with no leading zero for years less than 10.
yy	Year as last two digits, but with leading zero for years less than 10.
yyyy	Year represented by full four digits.

For example, to get the date string

"Wed, Aug 31 94"

use the following format string:

" ddd', ' MMM dd yy"

This string **MUST** be between the markers, for example:

[DATE_USER] ddd', ' MMM dd yy[DATE_USER_END]

DOS Command command (Advanced menu)



This command allows a DOS command to be run from within UltraEdit and the results of the command are captured and automatically displayed with UltraEdit at the completion of the command.

A dialog box is presented that allows the command to be specified (with a browse button) and the working directory to be specified (working directory only supported with UltraEdit).

The command line specified should be the full path of the file to be executed unless it is a system command (such as "DIR"), or it is in the Windows directory, or the Windows System directory.

The working directory should be the full path of the working directory, or left blank if not required.

In order to allow maximum flexibility of programs to be run, a special command line sequence is permitted that will cause the active filename to be inserted into the command line and passed to the program to be executed. For example:

You set up a command to run a compiler and you want to invoke the compiler with the file you are currently editing. If you change files you want the compiler to be invoked with the new file. The compiler is c:\comp\compiler.

Specify the following command in the dialog box:

c:\comp\compiler % F

When the command is executed, the % F will be replaced by the full directory path, filename and extension of the active window, The file will be saved before the command is executed.

The following commands may be used to pass a portion of the full qualified filename:

%P Path only (" C:\ project\test\")

% N fileName only ("test")

% E Extension only (". c")

% P%N%E is equal to % F which is (" C:\ project\test\test.c") in the example above.

NOTE - If the % f, %p, % n, % e are lower case the filenames are passed as long filenames and should be put in quotes, i.e. "% f" or "% p%n" etc. If the % F, %P, % N, % E are in upper case the filename and path will converted and passed as the "8.3" short filename specification for maximum compatibility with DOS programs.

Additionally the following commands are available:

%R This may be used for the full path/name of the project (% r for long name, %R for short name)

% RP This may be used for the full project path (% rp for long name, % RP for short name)

%RN This may be used for the project name only (% rn for long name, %RN for short name)

% modify% In some cases it may be necessary to modify the command line parameters each time the tool is run. To do this, add % modify% in the command line and when the tool is run a dialog box will be shown allowing the user to modify the command line or cancel and run without modification.

% sel% This instructs UltraEdit to add the highlighted/selected text from the active document into the command line in place of the % sel%.

% Env: This allows the user to use an environment variable in the command. The environment variable immediately follows the % Env: and UltraEdit replaces this with the contents of the matching environment variable.

% line% This instructs UltraEdit to add the line number of the cursor into the command line in place of the % line% - 1 is the first line.

% col% This instructs UltraEdit to add the column number of the cursor into the command line in place of the % col% - 1 is the first column.

The output from the command will be captured and displayed in a window when the command has completed.

As the output from a DOS program is sometimes buffered by the operating system, this process may not work correctly with a program that requires interaction while it is executing, the prompts may not appear on the screen at the correct time and the user may wonder why there is no activity.

As the output will always be captured. If the program generates no console screen output the captured output may be empty and UltraEdit will show an empty window.

If the Show DOS Box checkbox is checked, a DOS window will be shown as the command is being executed, otherwise the DOS window will be hidden.

Note - If the capturing the output does not work correctly, or the command does not function correctly you may try the [Alternate Capture Method](#) command from the [Advanced Menu](#).

The Last DOS Command repeats the last command specified with the DOS command dialog without presenting the dialog box.

Run Windows Program command (Advanced menu)



This command allows a Windows program to be executed from within UltraEdit. The output is not captured.

A dialog box is presented that allows the command to be specified (with a browse button) and the working directory to be specified (working directory only supported with UltraEdit).

The command line specified should be the full path of the file to be executed, unless it is in the Windows directory, or the Windows System directory.

The working directory should be the full path of the working directory, or left blank if not required.

In order to allow maximum flexibility of programs to be run, a special command line sequence is permitted that will cause the active filename to be inserted into the command line and passed to the program to be executed. For example:

You set up a command to run a compiler and you want to invoke the compiler with the file you are currently editing. If you change files you want the compiler to be invoked with the new file. The compiler is c:\comp\compiler.

Specify the following command in the dialog box:

```
c:\comp\compiler % F
```

When the command is executed, the % F will be replaced by the filename of the active window, The file will be saved before the command is executed.

The following commands may be used to pass a portion of the full qualified filename:

%P Path only (" C:\ project\test\")

% N fileName only ("test")

% E Extension only (". c")

% P%N%E is equal to % F which is (" C:\ project\test\test.c") in the example above.

NOTE - If the % f, %p, % n, % e are lower case the filenames are passed as long filenames and should be put in quotes, i.e. "% f" or "% p%n" etc. If the % F, %P, % N, % E are in upper case the filename and path will be converted and passed as the "8.3" short filename specification for maximum compatibility with DOS programs.

Additionally the following commands are available:

%R This may be used for the full path/name of the project (% r for long name, %R for short name)

% RP This may be used for the full project path (% rp for long name, % RP for short name)

%RN This may be used for the project name only (% rn for long name, %RN for short name)

% modify% In some cases it may be necessary to modify the command line parameters each time the tool is run. To do this, add % modify% in the command line and when the tool is run a dialog box will be shown allowing the user to modify the command line or cancel and run without modification.

% sel% This instructs UltraEdit to add the highlighted/selected text from the active document into the command line in place of the % sel%.

% Env: This allows the user to use an environment variable in the command. The environment variable immediately follows the % Env: and UltraEdit replaces this with the contents of the matching environment variable.

% line% This instructs UltraEdit to add the line number of the cursor into the command line in place of the % line% - 1 is the first line.

% col% This instructs UltraEdit to add the column number of the cursor into the command line in place of the % col% - 1 is the first column.

To run a DOS program use the DOS Command.

Alternate Capture Method command (Advanced menu)



This command changes the method used to capture the output from a DOS or Console application run from within UltraEdit. Under Windows 95 some applications do not run correctly, or do not capture output correctly using the standard method.

The primary difference is that this alternate method does not show the output of the command on the DOS Window as it is executing.

This method does however resolve issues seen by some users.

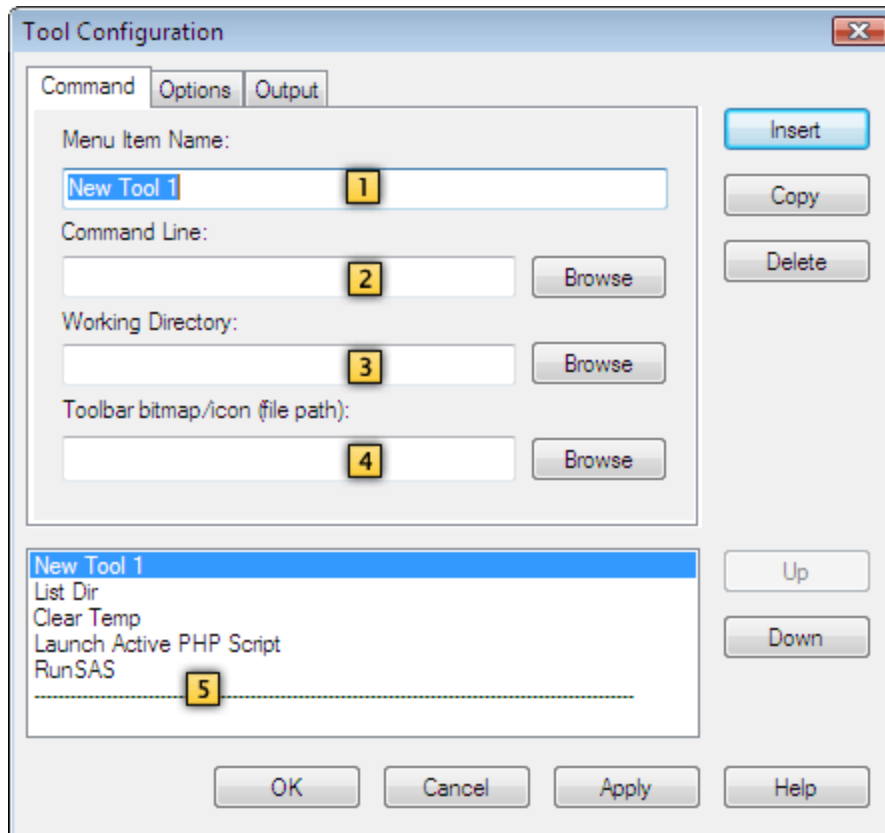
Tool Configuration command (Advanced menu)



UltraEdit/UESudio allows the user to configure up to ten menu items to commands of the user's choice. Additionally up to ten additional tools may be configured for the active project by selecting the Project Tool Configuration menu item.

These commands may be any DOS or Windows commands, and when set up appear in the lower portion of the [Advanced menu](#).

When the Tool Configuration menu item is selected, a dialog is presented allowing the user to specify the command line and action for each of the users choices. Any tool changes may be saved by pressing the **OK** or **Apply** buttons. The Tool Configuration dialog arranges different options for user-defined tools under the **Command**, **Options**, and **Output** tabs for easier management. Each tab presents different settings related to configuration of the currently specified tool. All three tabbed views include the **Insert**, **Copy**, and **Delete** buttons as well as the **Tool List pane [5]** which may be used to rearrange the list of currently defined tools.



Command Tab

This tab in the Tool Configuration dialog is used to create new tools or modify the defined options for existing tools. Users may also rearrange the tool list here and changes made here will be reflected in the tools listed at the bottom of the Advanced menu.

Creating a New Tool

If any tools are currently listed in the **Tool List pane [5]** you may select a tool and press the Insert button to create a new tool. This will cause the new tool to be added to the list above the selected item. The following options may be defined for a tool under the **Command** tab:

Menu Item Name [1]

This is the name of the item as it appears on the [Advanced menu](#) and will be reflected in the Tool List pane [5] as well.

Command Line

This is the actual **command line [2]** that will be invoked when the tool is run. If desired, it may contain a "% f" that will be replaced by the full directory specification and filename of the active file. The file will be saved first if changes have been made.

For example, you might set up a command to run a compiler and you want to invoke the compiler with the file you are currently editing. If you change files you want the compiler to be invoked with the new file. The compiler is c:\comp\compiler.

Specify the following command in the dialog:

```
c:\comp\compiler "% f"
```

When the command is executed, the % F will be replaced by the full directory path, filename and extension of the active window, The file will be saved before the command is executed.

The following commands may be used to pass a portion of the full qualified filename:

%P Path only (" C:\ project\test\")

% N fileName only ("test")

% E Extension only (". c")

% P%N%E is equal to % F which is (" C:\ project\test\test.c") in the example above.

NOTE - If the % f, %p, % n, % e are lower case the filenames are passed as long filenames and should be put in quotes, i.e. "% f" or "% p%n" etc. If the % F, %P, % N, % E are in upper case the filename and path will converted and passed as the "8.3" short filename specification for maximum compatibility with DOS programs.

Additionally the following commands are available:

%R This may be used for the full path/name of the project (% r for long name, %R for short name)

% RP This may be used for the full project path (% rp for long name, % RP for short name)

%RN This may be used for the project name only (% rn for long name, %RN for short name)

% modify% In some cases it may be necessary to modify the command line parameters each time the tool is run. To do this, add % modify% in the command line and when the tool is run a dialog box will be shown allowing the user to modify the command line or cancel and run without modification.

% sel% This instructs UltraEdit to add the highlighted/selected text from the active document into the command line in place of the % sel%.

% Env: This allows the user to use an environment variable in the command. The environment variable immediately follows the % Env: and UltraEdit replaces this with the contents of the matching environment variable. **Please note:** this command does not require the "closing" percent sign "%".

% line% This instructs UltraEdit to add the line number of the cursor into the command line in place of the % line% - 1 is the first line.

% col% This instructs UltraEdit to add the column number of the cursor into the command line in place of the % col% - 1 is the first column.

Working Directory

This is the **working directory [3]** for the specified command. If applicable, the process will be started with this as its working directory. As above, the %p may be used to specify the path. It should be noted that UltraEdit has special handling for the working directory and quotes should not be used when specifying the working directory (i.e. %p rather than "%p" should be used in this case).

If quotes are used to specify the working directory users will likely see an "Error Creating Process 267" which indicates that the specified working directory is not valid.

Toolbar/bitmap icon (file path)

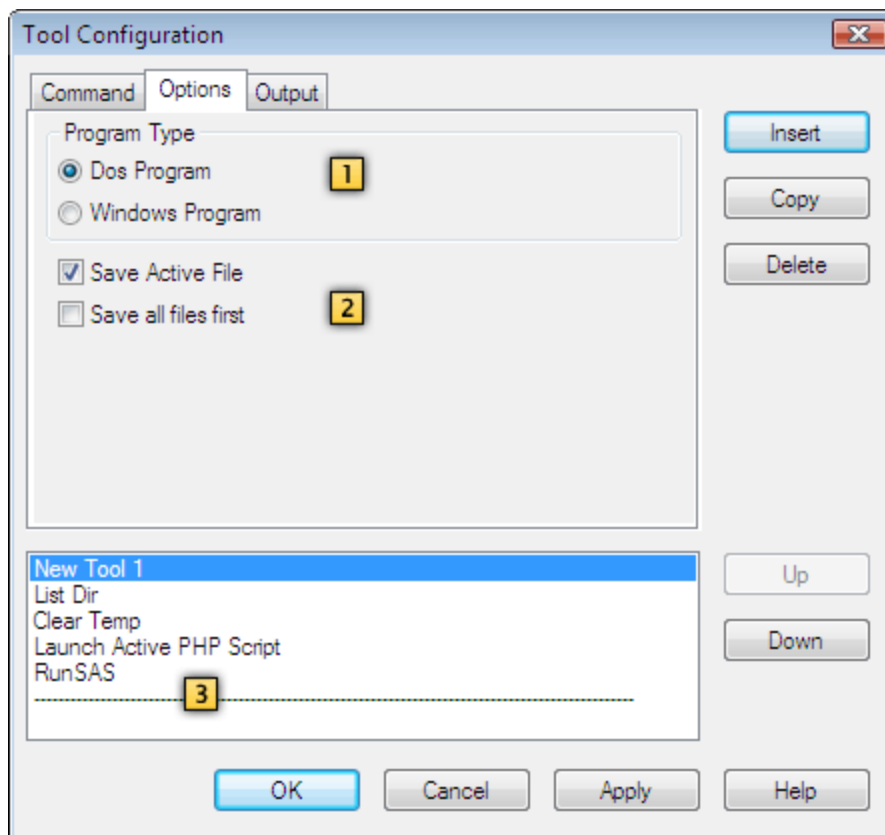
This [4] allows the user to specify a bitmap or icon file to be used on the toolbar for the current selected tool. The browse button allows the user to browse for the bitmap or icon. The bitmap or icon will be scaled to fit on the toolbar. Each file should contain a single bitmap or icon. If this is not defined, the default tool icon will be used.

Tool List pane

All defined tools will be listed in the **Tool List pane [5]** and users may rearrange tools by selecting them and using the **Up** and **Down** buttons to move the selected tool in the list.

Options Tab

This tab in the Tool Configuration dialog is used to define options for the currently selected tool. The following options are included.



Program Type Selection

The **Program Type [1]** may be configured here. It is important that the selection of DOS Program or Windows Program match the type of program being invoked by the tool.

Save Active File

If **Save Active File [2]** is checked, the editor will save the active file before running the tool, otherwise it will not (unless Save all files first is checked).

Save all files first

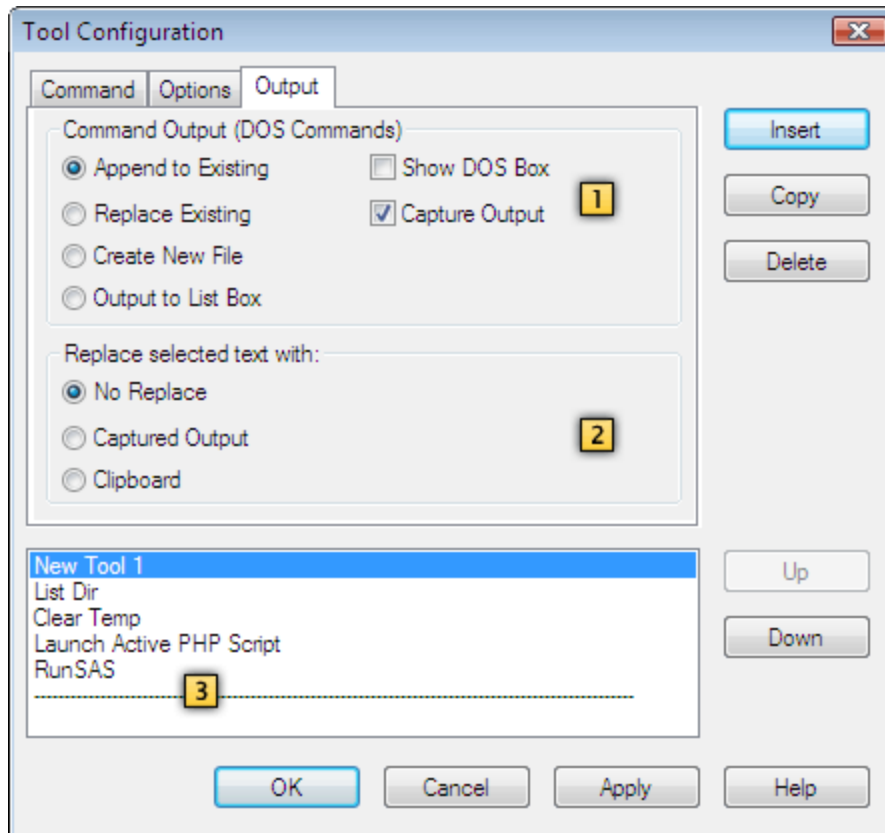
If Save all files first [2] is checked, the editor will save all open files before running the tool.

Tool List pane

All defined tools will be listed in the **Tool List pane [3]** and users may rearrange tools by selecting them and using the **Up** and **Down** buttons to move the selected tool in the list.

Output Tab

This tab in the Tool Configuration dialog is used to define output options for the currently selected tool. The following options are included.



Command Output (DOS Commands ONLY) [1]

DOS command output may be optionally captured to a file. The four options for DOS command output are:

- 1) Append to Existing (create new file if one does not exist)
- 2) Replace Existing
- 3) Create New File
- 4) Output to List Box ([Output Window](#)).

This list box allows the user to double-click on a line of the captured output and the editor attempts to open the referenced file and position to the specified line if found.

If the Show DOS Box checkbox is checked, a DOS window will be shown as the command is being executed, otherwise the DOS window will be hidden.

If the **Capture Output** checkbox is not selected then neither the output of the tool or any associated error output will be captured.

As several commands may be configured, it is quite possible the user may wish for some commands to always create a new file, while other commands should replace the existing file (or append to it). For this reason, each capture mode (append, replace, new) will create a new file for that mode if one does not exist, and each time a command is run, the file for the given capture mode will be append to or replaced if appropriate, or a new file created if selected.

Replace Selected Text With

This **[2]** indicates that at the completion of the tool command, the captured output or contents of the clipboard will be used to replace the current selection (or inserted at the cursor position if no selection), or if no replace is selected, the output will be displayed as normal.

Note - If capturing the output does not work correctly, or the command does not function correctly you may try the [Alternate Capture Method](#) command from the [Advanced Menu](#) (32-Bit version only).

Tool List pane

The **Tool List pane [3]** shows the currently configured commands. Clicking on a command in this list will cause the associated parameters of this command to be loaded and they may then be modified. To keep the changes, the user MUST press OK or Apply to save any tool configuration changes.

If you click on an item, you may delete it by selecting the **Delete** button.

Output Window

If a tool is configured that returns errors to the output window (such as a compiler), UltraEdit and UESstudio will open the referenced file to the specified line if the user double-clicks on the error listing. For this to be supported the file must be referenced in the following format [FULL PATH(Column Number): Error Message] i.e.:

```
C:\Development Path\ProjectDev\EditWindow.c(341): Undefined variable " GeneralError"
```

Requirements

DOS commands require that certain files exist depending on the operating system:

Windows 95, 98, Me require that COMMAND.COM file be on the Windows directory.

Windows NT, 2000, XP require that CMD.EXE be in the path.

Window Menu

Window menu



The Window menu offers the following commands which enable the user to arrange multiple views of multiple documents in the application window:

Duplicate Window	creates a duplicate window of the active document
Cascade	arranges windows in an overlapped fashion
Tile Horizontal	arranges windows in non-overlapped horizontal tiles
Tile Vertical	arranges windows in non-overlapped vertical tiles
Next Window	makes the next window active
Previous Window	makes the previous window active
Arrange Icons	arranges icons of closed windows
Next File Tab	promotes next file to top based on left-right tab order
Previous File Tab	promotes previous file to top based on left-right tab order
Split Window	creates a duplicate pane within the active document allowing the user to edit the active document in two separate locations at the same time
Minimize All Windows	minimizes all opened documents
Close All Files	closes all opened documents
Output Window	shows or hides the output window
Show File in Browser	passes active file to browser
Synchronized Scrolling	synchronizes vertical scrolling for all non-minimized windows
Next Panel	moves focus sequentially through next active windows
Previous Panel	moves focus sequentially through previous active windows
Windows	goes to specified window

Duplicate Window command (Window menu)



Use this command to open a new duplicate window with the same contents as the active window. You can open multiple document windows to display different parts or views of a document at the same time. If you change the contents in one window, all other windows containing the same document reflect those changes. When you open a duplicate window it becomes the active window and is displayed on top of all other open windows.

Cascade command (Window menu)



Use this command to arrange multiple opened windows in an overlapped fashion.

Tile Horizontal command (Window menu)



Use this command to vertically arrange multiple opened windows in a non-overlapped fashion.

Tile Vertical command (Window menu)



Use this command to arrange multiple opened windows side by side.

Next Window command (Window menu)



Use this command to switch to the next open document window. UltraEdit determines which window is next according to the order in which you opened the windows.

Shortcut: CTRL+F6 or CTRL+TAB

Previous Window command (Window menu)



Use this command to switch to the previous open document window. UltraEdit determines which window is previous according to the order in which you opened the windows.

Shortcut: SHIFT+CTRL+F6 or CTRL+SHIFT+TAB

Arrange Icons command (Window menu)



Use this command to arrange the icons for minimized windows at the bottom of the main window. If there is an open document window at the bottom of the main window, then some or all of the icons may not be visible because they will be underneath this document window.

Close All Files command (Window menu)



Use this command to close all files. UltraEdit suggests that you save changes to your document before you close it. If you close a document without saving you lose all changes made since the last time you saved it. Before closing an untitled document UltraEdit displays the Save As dialog and suggests that you name and save the document.

Show File in Browser command (Window menu)



This feature allows the active file to be displayed in the default WWW browser. For this feature to work, the user must have installed a WWW browser and the file types/extensions must be registered in Explorer/File Manager.

When selected, UltraEdit creates a temporary copy of the file and invokes the default browser with this file. The temporary file is deleted when the original file is closed, or if the user invokes the browser again with the same file.

This function uses the Shell Extensions and the default application associated with the extension will be used to show the file. If the extension is not associated with the browser the file will not be shown in the browser.

Synchronized Scrolling command (Window menu)



If this feature is enabled UltraEdit will scroll all windows that are not minimized when any of the windows are scrolled vertically. This allows viewing two (or more) files together without the need to scroll each one individually.

When the user clicks on the vertical scroll bar to provide scrolling, or on the up/down arrows on the vertical scroll bar the windows will be scrolled together.

Additionally if the user uses the Ctrl+Up/Down for scrolling the windows will be scrolled together.

If the user uses the cursor keys to reposition the file the windows will not be scrolled together allowing the user to change position in one file without affecting another file.

Scrolling by dragging the "Thumb" on the vertical scroll bar will attempt to scroll the windows together, however this method on large files may not keep the files synchronized.

Windows command (Window menu)



UltraEdit displays a list of currently open document windows at the bottom of the Window menu. A check mark appears in front of the document name of the active window. Choose a document from this list to make its window active.

Help Menu

Help menu



The Help menu offers the following commands:

Index	presents an index to help topics
Using Help	provides general instructions on using help
Quick Start	presents dialog with links to online Power Tips content

Guide	
Add Help Files	adds additional help files to Help menu
Online Forums	opens browser to user forums at ultraedit.com
Check for Updates	checks to see if an upgrade/hotfix is available for the version currently being used
IDM Products	presents local page in browser describing all IDM products and providing links for downloads
Register	presents registration dialog for application
About	displays the version number of this application

Index command (Help menu)



Use this command to display the opening screen of Help. From the opening screen you can jump to step-by-step instructions for using UltraEdit and various types of reference information.

Once you open Help you can click the Contents button whenever you want to return to the opening screen.

Add Help File command (Help menu)



This menu item allows Windows help files to be added to the UltraEdit Help menu for direct reference from within the editor.

Up to 5 help files may be added and will appear directly above this item in the help menu. When this item is selected, a dialog box appears that allows the user to enter the help filename and path, or browse for a help file. Additionally, a field exists for the name of the help file that will be used to identify it in the help menu.

If some text is selected within an open document within the editor, the user may press F1 to see a list of user configured help files that may be searched for the selected string. Alternatively, with text selected in an open document, the user may select the appropriate help file from the help menu and that help file will be searched for the corresponding topic that matches the selection.

The topic will be shown if found in the help file. If not, the search dialog with closest match of topics will be shown.

The user help files may be Windows Help files (. HLP) or compiled HTML files (. CHM) files.

Using Help command (Help menu)



Use this command for instructions about using Help.

Check for Updates command (Help menu)



This option launches your browser upon selection and searches our site for respective update availability. The check for updates process passes obfuscated registration details from your computer to the IDM website such as product name, version, and so on, which enables our site to direct you to the proper update information. Note: No information is collected by IDM Computer Solutions, Inc.

About command (Help menu)



Use this command to display the copyright notice and version number of your copy of UltraEdit.

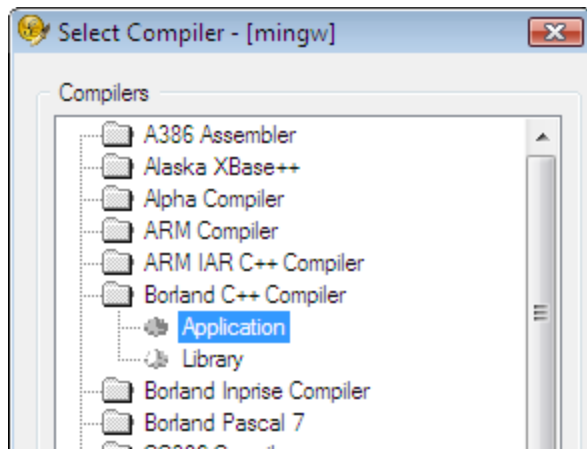
Compiler Configuration

Select Compiler dialog

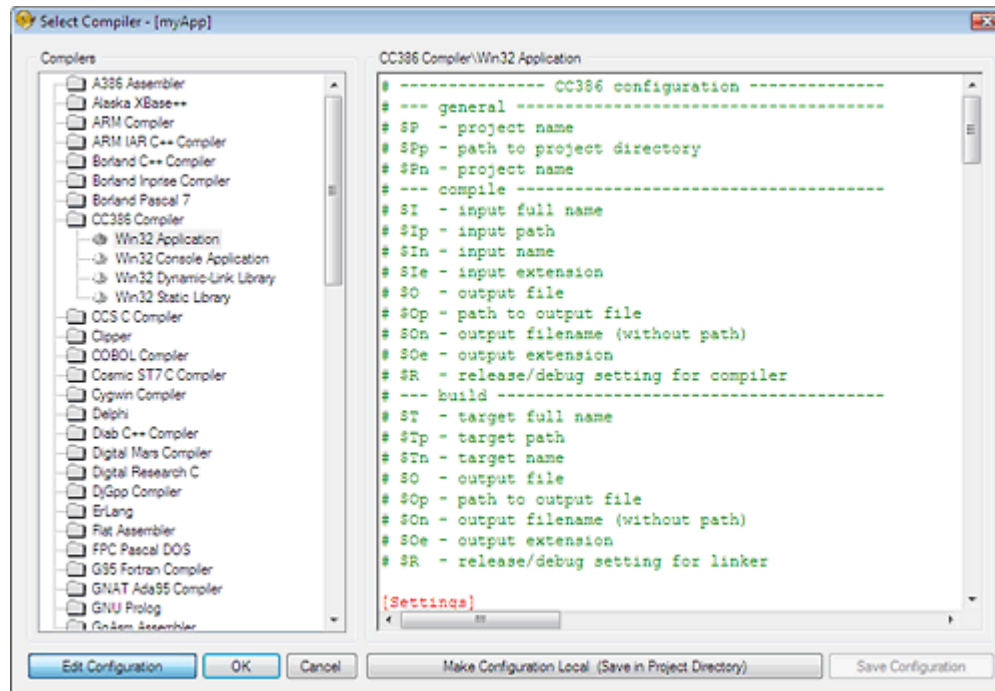


Please note: The items and options described below are only available in UESTudio and the following information doesn't apply to UltraEdit.

This dialog may be used to select the compiler to be used for the active project. The Select Compiler dialog may be invoked from the [Build menu](#) or by pressing the Select Compiler button on the [Build toolbar](#). Please note: This feature is only supported in UESTudio.



To configure the options to be used with this compiler, press the Edit Configuration button. This will expand the active dialog to display the Compiler Configuration Editor.



In the editor comments begin with a "#", sections are enclosed in "[" and "]" and MUST be placed at the beginning of the line. To save the modified file, press the Save Configuration button. If you have opened a project, the Make Configuration Local button will be enabled and is used to store the configuration with the project.

When the tool is executed, UESstudio first searches in the project directory for a local configuration and if one is not found, UESstudio will search in the \ configs directory. If you need to use a modified configuration file, but only for some projects, the Make Configuration Local button may be used to save the modified configuration to disk. If you no longer need the local configuration, deselect this button and the file with configuration information will automatically be removed from disk.

In the configuration files there are BASE SECTIONS that MUST be there because without them the Build functions will not work. These are: Settings, General, Build and MakeCommands. As sub-settings of the Settings section there can be SettingsInfo and SettingsReps. Probably the Variables, Environment and FileGroups sections will be used as well as some sections for tools attached to source files. When you write a configuration file, you will use some of the PROJECT VARIABLES. Project variables begin with "\$" or with another char that is set in the General section.

Project variables are: \$P, \$Pp, \$Pn.

\$P - is a full path to project (e.g. C:\ Projects\Application\Application.prj), \$Pp - is path without name and \$Pn is the name of project without its extension. \$Pe is not used, because extension of a project is always ". prj". These are GLOBAL variables and can be used in any build/compile command.

Following are the variables that can be used in the BUILD section. These are: \$T, \$Tp, \$Tn,

\$T - is a full path to target file (e.g. C:\ Prjs\App\Out\App.exe), \$Tp is a path and \$Tn is a name as in previous case.

\$O, \$Op, \$On, \$Oe.

\$O is the full path to an output file. In the BUILD section it maybe equal with target. It's the value of the "Out" key in a Build section. \$ Op is a path, \$On is a filename without extension, \$ Oe is an extension. The last variable that can be used is \$R. It's the value of the key DebugFlag or ReleaseFlag, depending on the MODE you use to build the project.

Following are the variables that can be used in the COMPILE sections or the sections for tools that work with single files.

These are: \$I, \$ Ip, \$In, \$ le,

\$I is a full path to input file. \$ Ip is a path, \$In is a filename without extension, \$ le is an extension.

\$O, \$ Op, \$On, \$ Oe.

\$O is a full path to output file. It's a value of a key "Out" in a section. \$ Op is a path, \$On is a filename without extension, \$ Oe is an extension. The last variable that can be used is \$R. It's the value of the key DebugFlag or ReleaseFlag, depends on MODE you use to build project.

There are 12 different sections in the configuration files:

Section	Description
[Settings]	where all the keys that you can see when you open Project Settings dialog are stored
[SettingsInfo]	contains short information about Project keys
[SettingsReps]	SettingsReps means "settings replacements". SettingsReps may be used to simplify configuration creation
[Variables]	may be used to simplify the content of configuration files
[Environment]	variables declared in this section are also accessible from other sections as \$(name), but will be stored in the environment
[General]	MAIN compiler's section. It includes BASE keys that MUST be set and OPTIONAL keys that may or may not be set.
[MakeCommands]	where keys for tools such as PACKER, PROFILER or PETOOLS that you want to use to work with target files are specified (Most compilers include two keys in this section: RUN and DEBUG)
[InsertFiles]	files and groups of files specified here will be added to the BUILD process
[FileGroups]	groups of files used for the same intent are specified here. Most compilers have groups of OBJECTS generated with compilers specified here and these objects are used in linker commands.
[GroupFormats]	where the FORMAT for objects in a generated group is set
[Build]	where commands that are executed when you click the Build button are specified
[. EXT]	where commands are specified that are executed if a source file is open and the Compile button is clicked

[Settings] Compiler Section



All the keys that you can see when you open Project Settings dialog are stored in this section. The format of the keys entry is: Key=value or Key=value1|value2|value3.

If you follow the first example you see in the "Project settings" dialog that there will only be one value attached to the key. If you follow the second example, values will be added to the list for the key, the first value will be set as value of the key, and when you click on the little button with the down pointing arrow, you will be able to select one of the other values from the list:

Values that **MUST** be in this section are:

Target	this key MUST be empty if you don't want to create projects with the same default name
Working Directory	default is "." as in current directory. This directory will be set when the target is run.
Command Line Arguments	default is empty and it is recommended to leave this key empty if you don't want to use the same arguments for all projects

All these variables may be used in other sections by typing \$(KeyName). When the compiler or linker is executed \$(KeyName) will be replaced with its value. This section is linked to the two next sections, [SettingsInfo](#) and [SettingsReps](#).

[SettingsInfo] Compiler Section



This section contains short information about the keys for the active Project. If you select a key in the Project Settings window this short help will be displayed at the bottom of the window. For key names use the same name as in [Settings](#), e.g. key = if you set "value1" short memory model will be used... then this text will be displayed in the Project Settings window below the list of keys.

[SettingsReps] Compiler Section



SettingsReps (Settings Replacements) may be used to make creating configurations easier. For example, with Turbo C compiler if you want to set the memory model, you must use the options - mt, -ms, etc. You can create a key in [Settings](#) like this: MemoryModel=-mt|-ms|-mm, but it is much more understandable to create a key in Settings like this: MemoryModel=Small|Tiny|Medium and in SettingsReps create a key like this: MemoryModel=Tiny=-mt|Small=-ms|Medium=-mm. Then if you select "Tiny" in "Project Settings" \$(MemoryModel) will be replaced with "- mt" in the compiler command.

If a value starting with '@' is used here the value from the [Settings] section is output like in printf where "%s" is replaced with the value.

If the value doesn't start with '@' then the resultant value is a string substitution based on the input value (like "ini" format, i.e.: *val1=replacer1|val2=replacer2*) The key is passed in as the input value.

[Variables] Compiler Section



Variables may be used to simplify the content of configuration files. All keys set here are of the format: key=value. You can use them in other sections as \$(key). It is possible to get values from registry keys here. The format is: \$(HKEY_???\PathToKey\KeyName). You may use LOCAL_MACHINE, CURRENT_USER and other hives of the registry to replace "???".

[Environment] Compiler Section



Variables declared in this section are accessible from other sections as \$(name), but all of these variables will be stored in the environment. Most compilers use environment variables to store paths such as INCLUDE, LIB, etc. If you need to use the value of other environment variables in this section, use %NAME% e.g.: PATH=C:\COMPILER\BIN;%PATH%. Use of the format %VARIABLE% is valid only for this section. In other sections you must access environment variables as from the [Variables](#) section.

[General] Compiler Section



The [General] section is the MAIN compiler section. It includes BASE keys that MUST be set and OPTIONAL keys that may or may not be set.

BASE keys are:

Key	Usage
TargetExt = . EXT	used when "Project Settings" is run the first time. UESstudio uses this key to generate the target name
ReleaseOut = Release	sets the name of the directory that will be used to store all object and target files if you set build mode to RELEASE
DebugOut = Debug	sets the name of the directory that will be used to store all object and target files if you set build mode to DEBUG.
Excludes = \$(Excluded Files)	sets the name of a project group. All files stored in this group will be excluded from build.

OPTIONAL keys are:

Key	Usage
UseFullPaths = 0/1	if set to "1" UESstudio will use full paths for all source files
UseDosNames = 0/1	should be set to "1" if your compiler works in DOS mode and doesn't recognize Long File Names
RemoveDot = 0/1	if set to "1" paths that begin with "." will have the "." removed
ConvertBS = 0/1	should be set to "1" to enable use of "/" instead of "\" in file paths. May be used for GCC compilers if you will be using GCC's MAKE utility.
MakeTool = \ Compiler\Bin\make.exe	UESstudio uses its own make utility to build projects but if you want to use another make utility you can. If the full path to the desired make tool is set here it will be used. UESstudio generates makefiles that are compatible with most make tools, so the makefile should work.
GrabOut = 0/1	if set, UESstudio uses it instead of option (a.) from "GLOBAL SETTINGS"
GrabErr = 0/1	if set, UESstudio uses it instead of option (b.) from "GLOBAL SETTINGS"
HideTools = 0/1	if set, UESstudio uses it instead of option (c.) from "GLOBAL SETTINGS"
EasyCapture = 0/1	if set, UESstudio uses it instead of option (d.) from "GLOBAL SETTINGS"
UseStatusBarFont = 0/1	if set, UESstudio uses it instead of option (e.) from "GLOBAL SETTINGS"
ShowCmdLine = 0/1	if set, UESstudio uses it instead of option (f.) from "GLOBAL SETTINGS"
SetCurSel = 0/1	if set, UESstudio uses it instead of option (g.) from "GLOBAL SETTINGS"

The remaining optional keys are for example . CPP=.C. and are explained in the [\[. EXT\] section](#).

[MakeCommands] Compiler Section



Most compilers include in this section two keys: RUN and DEBUG. Keys for tools such as PACKER, PROFILER or PETOOLS that are to be used with target files should be specified here. All keys defined in the MakeCommands section will be displayed when you click on the Commands button on the [toolbar](#).

The format of this section is: Key = NameOfSection. If a key is set, a section with the name NameOfSection MUST be added to the configuration file.

[NameOfSection]

The following keys may be set in the new [NameOfSection] section:

Keys	Usage
Title = TitleDisplayedInMenu	Text that is here will be used instead of " NameOfSection" text in MENU COMMANDS. Global variables such as \$ T. E.g. Title = Execute \$ T may also be used here.
Depends = \$ T	Filename(s) or global variables representing filenames that must be created if this command is executed should be set here.
CmdX = tool [options] [files]	Commands that should be executed should be defined here. Many tools may be executed. The first must be Cmd0 and if more tools are desired, they should be added as Cmd1, Cmd2, etc.
ShowWindow = 0/1	if set to "0", tools will be executed in HIDE mode.
DisplayConsole = 0/1	should be set to "1" if a DOS/CONSOLE to work in is desired.

[InsertFiles] Compiler Section



All files and groups of files specified here will be added to the BUILD process. You can use TWO types of keys here: One to add a single file to the BUILD fileX= and one to add a group of files to the BUILD groupX= where in both cases X = 0~n.

The content of groupX can include files separated with "\ n" (new line), "\ t" (tab) or with " " (white space). If the key "Files" is used in the [Settings section](#) and the content of this key is "file1.lib file2.lib file3.lib" (separated with whitespace) and the key incFiles = \$(Files) is also set in the InsertFiles section then files "file1.lib", "file2.lib" and "file3.lib" will be added as single files to the BUILD process (as source files from projects). Together these files "appear" as project files that are in the project group with name groupX. This feature may be used in the [FileGroups section](#).

[FileGroups] Compiler Section



Groups of files used for the same intent may be specified here. Most compilers have groups of OBJECTS generated with compilers specified here and these objects are used in linker commands. The format of keys specified here is: FGNAME= .EXT1;.EXT2;Group; groupX; group-r; group-o;.

The name of a group can be anything, e.g. FGOBJ, FGRES, FGASM, etc. The contents of the key are separated by a semicolon ";" and are:

Keys	Usage
.EXT1, .EXT2	all files with the extensions .EXT1 and .EXT2 will be added in the group specified with the name of the key.
Group	a project group, e.g. "Resource Files". All files independent of their extensions will be added to this group.
groupX	e.g. group0, group1, etc. Groups as specified in the InsertFiles section . All files from this group will be added to the group specified here (FGXX).
group-r	if this key is added to FG group, all files from the root (i.e. that are not in groups) will be added to the group defined in this section
group-o	all generated files (files that are created during BUILD process) will be added to this group if this is added to FG group content.

The content of all groups specified here may be used by typing \$ FGNAME in one of the Cmd commands in the build or compiler sections. For example, if the group FGOBJ=.obj; is added here and \$ FGOBJ is written in the linker command, then a group of objects will be created named FG??? in the makefile and this group will be used in the linker command as \$(FG???) where ??? is a number generated automatically by UESTudio when the makefile is generated.

[GroupFormats] Compiler Section



This section is used to set the FORMAT for a generated group. Some compilers will not allow the specification of files separated with spaces and a prefix must be used. For Borland's compilers if you want to create a library you must add the objects to the library with the prefix "+" as in: TLIB LIBRARY.LIB +FILE1.OBJ +FILE2.OBJ +FILE3.OBJ.

EXAMPLE: If you create group FGO=.obj in section " FileGroups" and here you set key FGO=+%s, a group will be generated with the content FGO+=file1.obj +file2.obj, etc.

In special cases you may need to set one format for the first file and another format for the next file(s).

EXAMPLE: Your linker expects object files separated with "," but "," must only be between object files as in: "LINK OBJ1.A,OBJ2.A,OBJ3.A EXE". If you need to do this add to [FileGroups](#) a section FGA = .A; and to the GroupFormats section add these two keys: FGA0=%s FGA=,%s where %s represents a filename.

These settings ensure that if you have only one file in the project first the FGA0 format will be used (format without ",") and if you have other files, FGA will be used. You can use it also for cases where a prefix must be set before the first file, as in: "LINK / objs:ob1,obj2,obj3". In this case you would set FGA0=/ obj:%s and FGA=,%s. If some group has no format set here the default (%s) is used.

[Build] Compiler Section



The Build section is an EXECUTIVE section - a section that does something and works with groups of files. Commands typed here are executed when you click BUILD.

The format of this section is the same as the format of the [MakeCommands section](#). In the BUILD sub-sections you must set five BASE keys Out, Depends, ReleaseFlag, DebugFlag and Cmd0. You may also set some optional keys (Cmd1 ~ CmdN) and keys that tell UESstudio to create "response" files:

Key	Usage
Out = \$ T	the "output filename" is generated from this value. UESstudio expects that this file will be generated after execution of the commands defined here.
Depends = \$ FGO	the name of a group with the files that will be used to make the target MUST be set here. For most compilers I set here \$ FGO, where \$ FGO is a group of files generated by compiler (. OBJ)
CmdX = LINK \$O \$ FGO	Cmd0 ~ CmdN - are commands that generate the target of the project. You can use the GLOBAL VARIABLES described for use in the Build section under the Select Compiler dialog topic
file@0=[respname.ext] Options1 Options2 Options3 Options4 etc.	some compilers use RESPONSE files. These are files in which are typed all the compiler options. The name of this file is passed as a parameter to the compiler like LINK @ RESP, where RESP is the text file that includes parameters to the linker. UESstudio can make these response files for you, if you set this section key as shown here. If you create a key like this UESstudio will create a response file with the name " respname.ext" and this text file will include lines with options, where Options1 will be on first line, Options2 will be on second line and so on. As options you can use all the options that you can use in classic CmdX commands.

[. EXT] Compiler Section



The . EXT section is an EXECUTIVE section - a section that does something and works with single files. Commands typed here are executed if you have a source file open and you click the COMPILE button.

This is not a section with the name . EXT but rather a section with the extension of your source file(s), e.g. ". C". In one config file there can be none, one or multiple sections like this. It depends on the number of types of source files that your compiler can use.

Format of the keys used here is the same as in the [Build section](#) with a few additions and one difference. The difference is that in the Out key you may not set \$ T, but for example \$ In.obj, where \$In represents the name of a source file. In Cmd0~CmdN you can then use \$O, where \$O will be "somefile.obj".

Some of compilers like C & Asm use INCLUDE files to store declarations of functions and variables. If this include file is modified the source file must be recompiled so you must tell UESTudio how to search for include files in sources. There are some keys that can help you to set this:

Keys	Usage
IncFiles = 0/1	if set to "1", UESTudio will search for included files in your source file
CaseSensitive = 0/1	indicates if keywords in your source file are case sensitive (1) or not
IncDirs = .;\$ (INCLUDE);	Most compilers store system include files in special directories like INCLUDE, or INC so you must tell UESTudio where it must search for these files. I recommended you to set as the first path "." as this provides that the directory where your source file is will be searched first
IncKeyWords=#include;	one or more keywords that are used by the preprocessor of your compiler to include files may be listed here
Comments = /*.*//. eol.	<p>multi-line comment brackets and single line comment brackets are set here</p> <p>Most compilers support comments. The part of your source code that is enclosed by comments is not compiled so it must not be searched for include files.</p> <p>Comment brackets MUST be in pairs. Single line comments must appear as: "/*.*//. eol." or "/*.*//. eol." where ". eol." stands for "to the end-of-line". Comment brackets are delimited with points ".". Do NOT put spaces between comment brackets and "." !</p>

If your compiler can work with more than one type of file with the same options, do not create a section for all these extensions. Instead create only one section, e.g. [C] and in the [\[General\] section](#) add keys like this: . CPP=.C or . ASM=.C and so on.

Adding A New Compiler

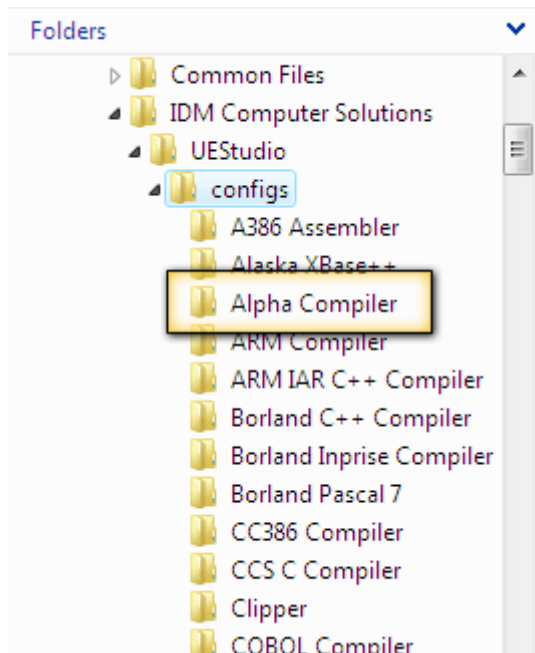


There are more than 50 compilers and assemblers supported for use with UESTudio by default. It may be necessary, however, to add a compiler that is not currently supported for use with UESTudio.

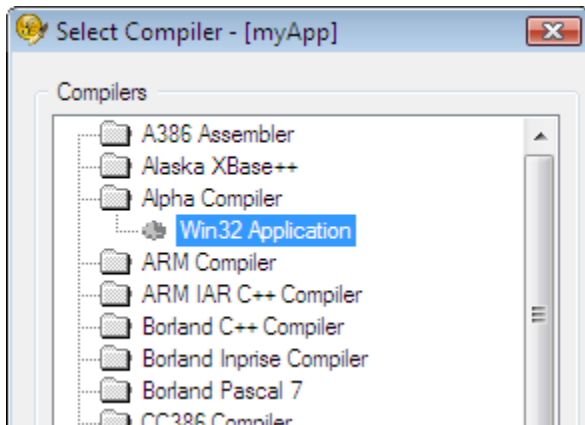
Adding a New Compiler

Compiler configuration files are like classic INI files, consisting of sections, keys and their values. Each configuration can be used to create only one "target" (application, library, dynamic-library), so in most cases compilers need multiple configurations and that's why every compiler has its own directory with configuration files inside.

The first step to adding a new compiler is to create a new directory under UES_PATH\Configs\. The best name for that directory is the name of the compiler to be added. For the purpose of illustration only we will create a configuration for Alpha Compiler.



To start creating a new configuration, open a new file in UESstudio. For the purpose of illustration we will use Win32 Application. Note that when the configuration file is completed, the added directory and file names will be used to populate the compiler list in UESstudio:



Some sections are required and some sections are optional and expand the functionality of the build system as a whole. The content of the configuration file will be used to populate the compiler configuration displayed in UESTudio when the Edit Configuration button is pressed in the [Select Compiler dialog](#). The first and most important section is General, so we need to create a new section - [General].

[General] Configuration Options

As previously mentioned, these are "target" configurations, so the first key that needs to be created is key TargetExt. For Windows/DOS applications the value for that key would be ".exe". For static libraries the value would be ".lib", for GCC compilers ".a", and for dynamic libraries ".dll". ANY extension may be set here and the compiler will build an application with that extension.

All "generated" files (object files, etc.) are stored in standalone directories to make the project environment clean. Every configuration supports two modes: release and debug. Two keys may be set here: ReleaseOut and DebugOut, but as a minimum ReleaseOut should be set. If DebugOut isn't set, it will use the same value as in ReleaseOut. The default values are Release and Debug but may be set as desired.

If the compiler/linker can only work with "full paths" (doesn't accept relative paths to the files), a new key UseFullPaths should be created and set the value to "1". If settings are being created for a DOS compiler that doesn't accept long file names, the key UseDosNames would also need to be created and the value should be set to "1".

Sometimes a project can contain files that are compilable but users may not want to compile them. (For example if they aren't finished, or there are errors that can't be fixed prior to a build.) These files can be moved to the special folder and UESTudio's build system will not pass them to the compiler/linker. The folder can have any name except "External Dependencies". This name is reserved for the UESTudio Build system and is created/deleted automatically during the build, so it is most recommended NOT to

use that project folder to store any user file(s). To specify the name of this special folder, create a new key Excludes and set the name of that special project folder, for example "Excluded Files".

Some compilers do need to have object files in the same directory as the source files, because they only accept a filename (name of source file) without the path as a parameter. When the build system generates relative paths of files that are passed to the compiler(s), these paths always start with directory names and when the output directory is the same as the source directory, these paths begins with ".\". But in the case where compilers accept only filenames, UESstudio can't pass the paths, so a new key RemoveDot needs to be created with a value of "1".

If a compiler is being used that accepts only Unix-like paths (with forward slashes "/") instead of classic DOS/WIN (with backslashes) "\", the key ConvertBS should be created with a value of "1". This will ensure that "/" will be used instead of "\" in paths. The example General section now looks like this:

[General]

TargetExt = .EXE

DebugOut = Debug

ReleaseOut = Release

Excludes = Excluded Files

Command Configuration Options

Command sections tell UESstudio how to build an application or compile a file. There are also other command sections that are not important, but they may be set to configure other tools to work with the project/source/target files. Most important of these sections is the [Build] section. This section must be a part of every configuration.

The base key for this section is Out. This tells UESstudio what has to be created using the defined commands. In ALL configurations the value for the key Out should be set to "\$T". All command sections support special values starting with \$. These values are described in the commented section at the beginning of every configuration file, but the most important ones are \$T as a target file, \$P as a project file, and \$O as an output file. Each command section may reference one of these files. In the [Build] section this is the value set for the Out key and in the [Compile] section it is the value for the input file - \$I.

The creation of the output file (in the [Build] section's target file) depends on other files, so the key Depends needs to be created with the appropriate value. For the [Build] section this can be a list of object files – for example \$FGO. As previously mentioned every configuration can "handle" two build modes - release and debug. If use of this feature is desired two special keys: ReleaseFlag and

DebugFlag must be created. These keys may be set for the [Build] section and all compiler sections. Most of the compilers support these two modes. In debug mode they will add all important information to the objects/executables for debugging code. In release mode (production use) this information is not included. The values of these keys are "expanded" in the compiler commands where the \$R special value (the release/debug setting for the linker) is used.

Now command keys should be created. Every command section must contain at least one command. They are set as keys Cmd0, Cmd1, Cmd2, etc. In most cases only one command needs to be set but more may be used if desired. The example [Build] section now looks like this:

[Build]

Out = \$T

Depends = \$FGO

DebugFlag = /debug

Cmd0 = linker \$R -out \$O -input \$FGO

Example: Given that the target file will be "Application.EXE", the DebugFlag is "/debug", the ReleaseFlag is not set, and the project contains only one source file "example.cpp", the Cmd0 command will look like this during the build:

```
"compiler /debug -out Application.EXE -input example.obj"
```

in debug mode and the same without "/debug" when release mode is set.

If the compiler being used doesn't support long command lines, but supports "response files" (files that contain command line options), UEStudio can generate these response files for you. More response files may be created just by changing the index after file@, so the keys that contain information regarding how to create response files will have names like "file@0", "file@1", etc. The value of these keys consist of information about the response filename to be created and parameters (compiler/linker options), so the value could be "[filename.ext] option1 option2". File groups and other variables may be specified here as well. If options need to be included in the response file on multiple lines, "|" should be used between options. If response files and objects need to be on separate lines (not with other compiler options), the example [Build] section could be expanded with the following keys:

file@0 = [myresp.file] -out \$O | -input \$FGO

Cmd0 = linker \$R @myresp.file

With the [General] and [Build] sections defined, other compiler sections may be defined. These sections are not required, but in most cases these sections are a part of compiler configuration, and are therefore described. For example when creating a configuration for a C compiler, at least one compiler's section named [.C] needs to be created. Multiple configurations may be defined as needed to describe how to deal with source files. If the compiler being used handles multiple source files the same way (i.e.: .C, .Cpp, .Cxx) a new section is not required for all these extensions ([.C], [.Cpp], [.Cxx]); instead keys may be set in the [General] section that will tell UESudio which compiler section to use. For example, the line

.CPP = .C

may be added to the [General] section and UESudio will use the [.C] section for all ".C" and ".CPP" files. The compiler section (in our example [.C]) will consist of all the keys described in the [Build] section, and only the values will be different. In these sections the special key \$I may be used, which is the full path to the source file (for example "application.cpp"). All the special keys that contain paths may also be used to get only names, paths, or extensions. For example, \$Ip contains only the path, \$In the name, and \$Ie the extension of the source file. (The same methodology may be applied to \$P, \$T and \$O.) So the example compiler section can consist of the following keys and values:

[.C]

Out = \$In.obj

Depends = \$I

DebugFlag = /debug

Cmd0 = compiler \$R -out \$O -input \$I

This section may also contain keys related to include files so that UESudio may use them and related symbols when parsing source files that reference the include files. The first of these keys is IncFiles which tells UESudio that this file can contain included files. By default, include files are searched only in the current directory. The key IncDirs may be defined next and the value for this would specify a list of directories to be searched. The same format should be used here as is used for the environment variable PATH (directories separated with ';') and the value "\$(INCLUDE)" may be set here. Values set as "\$(something)" are variables and reference keys set in the [Variables] section (described below). The next keywords to be specified are those used to define include files. In C/C++ languages this would be #include, in NASM this would be %include, and in Microsoft's assembler only include. If these keywords are case sensitive, the key CaseSensitive must be added with the value "1".

Most of the source files also support comments – line and/or block. Comments aren't compiled, so include files defined in comments are ignored. If comments are supported by the compiler, the key Comments should be added here. The format used should be:

StartingSymbolOfComment[dot]EndingSymbolOfComment[dot]

End of line (for line comments) must be specified as "eol". As every comment block has a start and end symbol, these must always be defined in pairs. So the example [.C] section may include the following keys:

IncFiles = 1

CaseSensitive = 0

IncDirs = .;\$(INCLUDE);

IncKeyWords = include;

Comments = //.eol./*.*/.

[FileGroups] and [GroupFormats] Configuration Options

The value \$FGO (a file group list) was used in the [Build] section. Users may define as many file groups as are desired – including all project files and files generated during the build (for example object files). To define file groups, create a new section named [FileGroups] and include the key FGO. File extensions of files to be used to create file groups from may be set as a value or project folder names or special groups. In the current example, the value could be ".obj". In most cases object files are passed to the linker just as a list of files. But sometimes a special format may be required to pass object (or source) files to the linker (other tool) and for these cases the section [GroupFormats] may be defined.

If the linker needs object files passed to the command-line as "a + Name1.obj + Name2.obj", etc., key should be added to the [GroupFormats] section that will tell UESTudio what format is required. For example the key FGO will have the value "+ %s". In rare cases a situation may exist where the first file of the list doesn't need a special format, or needs a different format from all other files. In this case a group of object files may be specified that should be passed to the linker's command-line separated by ",", between objects and started with "/obj". The new sections look like this:

[FileGroups]

FGO = .obj;

[GroupFormats]

FGO = , %s

FGO0 = /obj: %s (<= add zero to the FG definition, so this format will be used for first file from that group)

[InsertFiles] Configuration Options

Some compilers need to link special files every time they build an application. One solution is to add these files (for example objects will startup code) every time a new project is created. A simpler solution would be to define these files in configuration files that will always be added to the project "virtually" (only during the build). If this is needed, create a new section [InsertFiles]. Two types of keys may be used to define these files - file0 through fileX and the full/relative path to the standalone file should be set as a value, or "group0" through "groupX" and the value will be a list of files separated with whitespaces, for example "lib1.lib lib2.lib", etc. These "groups" may also be used in the [FileGroups] section as mentioned above. For example, if a list of libraries from "group0" defined in the [InsertFiles] section is needed and should be added to the list of object files, our modified FGO key from the [FileGroups] section will be:

[FileGroups]

FGO = .obj;group0;

[Variable] and [Environment] Configuration Options

Any variable may be defined and used in most keys in most sections – in most any place where it seems reasonable. (An exception is the TargetExt key in the [General] section – that value can't be "variable".) The primary section where you can define variables has the name [Variable]. For example, the key CPATH may be set here and the full path to the compiler may be set as the value. This variable may then be used in other sections as \$(CPATH). Most compilers need to also have some environment variables set like INCLUDE, LIB, etc. The [Environment] section is used for this purpose. Keys defined here will be added to the normal environment during the build. It may be desirable to use the original value of the environment variable in some cases, like PATH (to add the path to the compiler) while the "current" path should also be used. The value may then be set as "\$ (CPATH);%PATH%". Notice the original values (only in this section) may be specified as "%KEY%" as in classic environment variables. In both of these sections it may be desirable to include information from the registry and UESstudio supports this. The variable syntax described above should be used and should start with HKEY as in \$(HKEY_CURRENT_USER\MyCompiler\Path). This value will be expanded from the HKEY_CURRENT_USER key, "directory" MyCompiler and key Path. If it is desired to use the "Default key" in the registry path MyCompiler, \$(HKEY_CURRENT_USER\MyCompiler\@) should be used.

[Settings], [SettingsInfo], and [SettingsReps] Configuration Options

When changes are made directly to a compiler configuration file, and the Make Configuration Local button is not pressed, these changes will be global (for all projects). There are some special sections that allow modification of options using the GUI - [Settings], [SettingsInfo], and [SettingsReps].

[Settings] is the most important section and all keys defined here are also saved with the project, so these values may be modified in UESstudio without changing the compiler's configuration file. With use of these sections "common" configuration files may be created that will have the same options for all projects (for

example all Windows applications) and items that need to be configurable may be specified here. If the key Compiler Options is created in the [Settings] section, and as a value you use empty string, this key will be empty by default, but when some value is set here, this value will be used in any place where you want to specify \$(Compiler Options).

If there are some predefined values (for example the key Warning Level is created), they may be set here and the values for those keys may be specified as "w1|w2|w3". As shown, values are delimited with "|". The first value in the list will be used as a default value, in this case "w1". Every change made here from inside UESTudio (not directly modifying the configuration file) will be saved with the project. So now when you click the key Warning Level is clicked on from inside UESTudio, any of these three values may be selected.

It would be easier to choose from "Errors only", "Standard", "Warnings as Errors". For this purpose a [SettingsReps] section is defined. Now in the [Settings] section the key Warning Level may be set to "Errors only|Standard|Warnings as Errors" and in the [SettingsReps] section the key Warning Level may be set as "Errors only=w1|Standard=w2|Warnings as Errors=w3" and UESTudio will replace all these "word based" options with proper command-line options.

Options may also need to be passed to the compiler in a specified form. For example if the linker command-line needs to be added as a parameter resource script, a key ResourceScript may be created in the [Settings] section and in the linker cmdline it could look like "/rscript:\$(Resource Script)". No problems would occur if Resource Script always has to be set, but if this is not the case, an error may result when Resource Script is set as an empty string and this would be passed to the linker cmdline as "/rscript:".

Instead of this the key Resource Script may be set in the [SettingsReps] section with the value "@/rscript:%s". Starting with "@" will tell UESTudio that it's a "format", so if this key is not set, nothing will be passed to linker line and if it is set, "/rscript:Script.ext" or the value will be passed to that line.

Rather than the user remembering all possible options, a section may be created where information about the keys may be stored. This information will be displayed in UESTudio in the compiler's options dialog when key is selected. The section [SettingsInfo] may be created and keys should be listed there with the same name as they have in the [Settings] and [SettingsReps] sections and the values for the keys here would be descriptions of the keys. The new [Settings] section now looks like this:

[Settings]

Target =

Working Directory =.

Warning Level = Errors only|Standard|Warnings as Errors

[SettingsReps]

Warning Level = Errors only=w1|Standard=w2|Warnings as Errors=w3

[SettingsInfo]

Target = place for you to specify name of target file

Working Directory = this directory (if set) will be used to run application from

Warning Level = you can set here warning level for your compiler

Two of the keys in the [Settings] section above are part of every [Settings] section in compiler configurations and one of them - Target is required. It is recommended that this be kept empty, and if this is done, the target name will be generated from the project name and TargetExt defined in the [General] section.

CVS Support

CVS in UESTudio



UEStudio includes a CVS (Concurrent Versions System) interface in the [Project menu](#). Please note: This is not supported in UltraEdit.

Please note that UESTudio does not provide a CVS server itself, but only a way to interact with a CVS server that has already been configured.

CVS is an open-source, network-transparent program that allows developers to keep track of different development versions of source code. CVS does not maintain multiple versions of source code files but keeps a single copy and records of all of the changes that are made. When a developer wants a particular development version of a file, CVS will reconstruct that version based on its records.

Accounts for checking out source files through CVS may be created/modified using the [CVS Manager dialog](#). UESTudio includes support for the following CVS related functions:

* Checkout	checks out module from repository
* Commit	commits working copy to repository
* Diff	checks for differences between two versions of the same file
* Update	updates working copy from repository
* Import	imports new module into repository
* Export	checks out module excluding CVS information
* Accounts/Settings	launches CVS Manager dialog

FAQ

How can I disable the "no temp file" prompt?



By default a confirmation dialog is shown when editing files without a temporary file as there is no backup made and all changes are permanent. If you wish you may disable this under the [Temporary Files](#) topic in the File Handling branch under Configuration in the Advanced menu by selecting "Open file without temp file but NO Prompt".

How can I open UltraEdit without blank Edit files?



If you're not specifying any file, but only opening UltraEdit, check to ensure that "Create new EDIT file when opening with no other files" is NOT selected under the [New File Creation](#) topic in the Editor branch under Configuration in the Advanced menu. The default setting is to create a new file if no files are being opened automatically when UltraEdit starts up.

Also, when attempting to open a file that doesn't exist, the default setting is to create a new blank file with the name specified. If you don't wish this to happen please ensure that "Create a new file if file specified on command line does not exist" is NOT selected under the [New File Creation](#) topic in the Editor branch under Configuration in the Advanced menu.

How can I control the format of new files?



There is a [DOS/UNIX/MAC Handling](#) topic under the File Handling branch under Configuration in the Advanced menu to allow specification of the default format for new files created in UltraEdit. Under "Default file type for new files" you may select either "DOS", "Unix", or "MAC" and new files would be created with the appropriate line terminators.

How can I remove lines containing a particular string?



You can replace lines containing a string with a Regular Expression [Replace All](#).

To do this you would do a find of:

Find What: %*YOUR STRING*^p
Replace With: (literally nothing)

The Regular Expressions section in the Help file has further details.

How can I delete blank lines?



To delete blank lines you can use a regular expression Replace as follows:

Find What: ^p\$

Replace With: (literally nothing)

This will cause a hard return immediately followed by the beginning of a line to be replaced with nothing, effectively deleting them.

How can I remove separators from the toolbar?



To remove a separator, click with secondary (right) mouse button on the toolbar and select Customize Toolbar from the popup menu. Select the separator you want to delete in the left pane and click the right arrow button.

How can I print syntax highlighting?



From the [File menu](#), select "Print Setup/Configuration" and then "[Page Setup](#)". Here you will see an option to "Print Syntax Highlighting".

How can I modify the file types in the File Open dialog?



Click on Configuration in the [Advanced menu](#) and click on the [File Types](#) branch. This page allows the list of file types that appear in the File Open and File Save As dialogs to be configured. Up to 10 file types may be configured. The first extension for these file types will be shown in the file operation dialogs in the same order they are shown on this page.

How can I do a multi-line Replace?



You can do this using "^p" as the CR/LF - UltraEdit translates the ^p into a hard return. You can also use the contents of the clipboard or selected text in a search or replace string with "^c" and "^s" respectively. You can copy/select a string up to 30,000 bytes for use this way.

Where can I find additional wordfiles for UltraEdit?



Many wordfiles, macros, auto-complete files and tag lists are available for download from our site at <http://www.ultraedit.com/downloads/extras.html#wordfiles>.

Why does reformat break on hyphenated words?



This is the default behavior in UltraEdit for dealing with hyphenated words in a paragraph reformat situation. If you don't want hyphenated words to be broken you should ensure that the option "Break on hyphen (-) when converting/reformatting paragraphs" is not selected in the [Formatting](#) topic under the Editor Display branch under Configuration in the Advanced menu.

Why does UltraEdit reformat a pasted paragraph?



If you have the [Word Wrap](#) settings set to "Wrap after column #, insert CR/LF" a paste will reformat the paragraph after the paste to ensure the correct wrapping is maintained. If you ensure that "Reformat paragraph after paste when wrap settings set to add hard returns at specific column number" is NOT selected under the [Word Wrap/Tab Settings](#) topic in the Editor branch under Configuration in the Advanced menu this will not occur.

Why does my IntelliMouse not work with UltraEdit?



UltraEdit has built in IntelliMouse support. If you disable the IntelliMouse driver from running with UltraEdit this should help. The IntelliMouse driver allows you to do this and this will allow UltraEdit to control interaction with the mouse.

Why does Trim Trailing Spaces not work on last line?



The functionality of [Trim Trailing Spaces](#) is that it does require a newline to operate on a given line of text. If the last line of text in your file is terminated with a hard return Trim Trailing Spaces will work on that line as well.

Why does the notepad.exe get overwritten?



Some Win2K/ XP installations, such as those from Compaq, place an I386 subdirectory on the hard disk to make drivers available for plug and play. That directory also contains a notepad.ex_ which will mysteriously overwrite the new notepad.exe just as the CD ROM does. You can tell that this is happening if the unsecure driver dialog box does not appear as the readme says it will. Renaming the file to notepad.ex_.bak resolves the problem.

Also, XP users must copy the new notepad.exe to c:/windows and c:/windows/system32.

If it appears that you do not have a dllcache directory on your system, open Explorer and go to Tools -> Folder Options -> View and ensure that "Hide protected operating system files (Recommended)" is NOT selected and go to the WINNT\system32\dllcache directory and rename the old notepad.exe to notepad.exe.bak or something similar and then you can replace the notepad.exe in the WINNT and WINNT\system32 directories.

Finally, it might be helpful to note that some Win2K/XP installations, such as those from Compaq, place an I386 subdirectory on the hard disk to make drivers available for plug and play. That directory also contains a notepad.ex_ which will mysteriously overwrite the new notepad.exe just as the CD ROM does. You can tell that this is happening if the unsecure driver dialog box does not appear as the readme says it will. Renaming the file to notepad.ex_.bak resolves the problem.

Why does the ruler start with a "0"?



The ruler works as a normal ruler would, showing "0" as the start point, allowing the user to measure the number of columns prior to a given point. The status bar, on the other hand, is intended to show the

column in which you're about to make an entry. As with normal Windows applications, both of these modes of indication and measurement are, I believe, the standard way of doing things, however, most applications do not include both.

Why does copy and paste stop working?



UltraEdit supports multiple clipboards and I suspect you have activated one of the user clipboards.

The clipboards are activated from the [Edit menu](#), Right Click Menu or Ctrl+0 - Ctrl+9 with Ctrl+0 being the standard clipboard. The active clipboard is shown in the status bar following the line/ col separated by a comma. CW reflects the Windows Clipboard and C1-9 the user clipboards.

Another possible cause of confusion in relation to the clipboards is a setting under Advanced -> Configuration -> Editor -> Miscellaneous. If the option **Automatically copy to clipboard when selection is made** is selected, this would cause any selected text to automatically overwrite the current contents of the clipboard when selected.

Why does it take so long to open large files?



UltraEdit handles files up to and over 4GB. UltraEdit is disk-based. This means it only loads small portions of the file at once into memory so it does not use all the memory and stop other applications from running. However, it does make a temporary copy of the file to achieve this and this can take time for large files.

There are two options in the [Advanced menu](#) and Configuration to consider.

1) Disable line numbers - Check this to disable the line numbers on large files, it makes navigation a lot quicker. This may be set in the [Miscellaneous](#) topic under the Editor Display branch in Configuration in the Advanced menu.

2) Open file without temp file - This instructs UltraEdit not to create a temporary copy of the file. This does however mean it is editing the original file so any changes are permanent. Several users make use of this for large files. You can set a threshold for this so that it only happens on files over a certain size.

This may be set in the [Temporary Files](#) topic under the File Handling branch in Configuration in the Advanced menu.

Additionally, you may want to disable automatic conversion from Unix to DOS in configuration if it is enabled also.

Why is there a delay starting UltraEdit?



UltraEdit does try to communicate with the printer driver when it loads and shuts down. If you have a network printer as the default this may be the cause.

If you go to the **Advanced** menu and select the **Advanced** branch under **Configuration -> Editor** and ensure that **Load/Restore printer settings** is NOT selected and restart UltraEdit this may help.

License Agreement

License Agreement - Unregistered



LIMITED WARRANTY

THE PROGRAM IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT WILL THE AUTHOR or AUTHORS BE LIABLE TO YOU FOR ANY DAMAGES, INCLUDING INCIDENTAL OR CONSEQUENTIAL DAMAGES, ARISING OUT OF THE USE OF THE PROGRAM, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

YOU ACKNOWLEDGE THAT YOU HAVE READ THIS LICENSE, UNDERSTAND IT AND AGREE TO BE BOUND BY ITS TERMS AS THE COMPLETE AND EXCLUSIVE STATEMENT OF THE AGREEMENT BETWEEN US, SUPERSEDING ANY PROPOSAL OR PRIOR AGREEMENT, ORAL OR WRITTEN, AND ANY OTHER COMMUNICATIONS BETWEEN US RELATING TO THE SUBJECT MATTER OF THIS LICENSE.

UltraEdit/UltraEdit is a shareware program and is provided at no charge to the user for evaluation. The purpose of shareware software is to provide personal computer users with quality software on a "try before you buy" basis, however payment is still required for continued use of the product.

If you find this program useful and continue to use it after the trial period, you must make a registration payment (see the registration instructions for details). This registration fee will license one user to use one copy of UltraEdit and/or UltraEdit on any one computer at any one time. All users will receive a copy of the latest release when they register, or it will be made available for downloading, and free technical support.

All users must register and pay for their copies within 45 days of first use or their license is withdrawn. Site License and multiple user license arrangements may be made by contacting IDM Computer Solutions, Inc.

Anyone distributing this product for any kind of remuneration must first contact IDM Computer Solutions, Inc. for authorization.

You may distribute this software to friends and colleagues but you must include all files in the original distribution. Please encourage them to register their copy if they find that they make use of it.

License Agreement - Registered



This is a legal agreement between you (either an individual or an entity) and IDM Computer Solutions, Inc. By opening the sealed software packages (if appropriate) and/or by using the SOFTWARE, you agree to be bound by the terms of this Agreement.

SOFTWARE LICENSE

1. **GRANT OF LICENSE.** This License Agreement (License) permits you to use a single copy, or multiples copies if you are the only user of the software product identified above, which may include user documentation provided in on-line or electronic form (SOFTWARE). The SOFTWARE is licensed as a single product, to an individual user, or group of users for Multiple User Licenses and Site Licenses. This Agreement requires that each user of the SOFTWARE be Licensed, either individually, or as part of a group. A Multi-User License provides for a specified number of users to use this SOFTWARE at any time. This does not provide for concurrent user Licensing unless specifically agreed upon and ordered as such at an additional cost above the normal licensing fee. Each user of this SOFTWARE must be covered either individually, or as part of a group Multi-User License. The SOFTWARE is in use on a computer when it is loaded into the temporary memory (i.e. RAM) or installed into the permanent memory (e.g. hard disk) of that computer. This software may be installed on a network provided that appropriate restrictions are in place limiting the use to registered users only.

2. **COPYRIGHT.** All physical copies of the software are the property of the registered user(s) but use of the software is restricted according to the terms as specified above (GRANT OF LICENSE). The SOFTWARE is protected by United States of America copyright laws and international treaty provisions. All Intellectual Property Rights (IPR) remain the property of IDM Computer Solutions, Inc. You may not copy the printed materials accompanying the SOFTWARE (if any), nor print copies of any user documentation provided in on-line or electronic form unless for the specific use of a legally registered user as specified above (GRANT OF LICENSE). You must not redistribute the registration codes provided, either on paper, electronically, or as stored in the files ULTRAEDT.REG, UEDIT32.REG, ULTRAEDT.INI, ULEDIT32.INI or any other form.

3. **OTHER RESTRICTIONS.** The registration notification provided, showing your authorization code and this License is your proof of license to exercise the rights granted herein and must be retained by you. You may not rent or lease the SOFTWARE, but you may transfer your rights under this License on a permanent basis, provided you transfer this License, the SOFTWARE and all accompanying printed materials, retain no copies, and the recipient agrees to the terms of this License. You may not reverse

engineer, decompile, or disassemble the SOFTWARE, except to the extent the foregoing restriction is expressly prohibited by applicable law.

LIMITED WARRANTY

LIMITED WARRANTY. IDM Computer Solutions, Inc. warrants that the SOFTWARE will perform substantially in accordance with the accompanying printed material (if any) and on-line documentation for a period of 365 days from the date of receipt.

CUSTOMER REMEDIES. IDM Computer Solutions, Inc. entire liability and your exclusive remedy shall be, at IDM computer Solutions, Inc. option, either (a) return of the price paid or (b) repair or replacement of the SOFTWARE that does not meet this Limited Warranty and that is returned to IDM Computer Solutions, Inc. with a copy of your receipt. This Limited Warranty is void if failure of the SOFTWARE has resulted from accident, abuse, or misapplication. Any replacement SOFTWARE will be warranted for the remainder of the original warranty period or thirty (30) days, whichever is longer.

NO OTHER WARRANTIES. To the maximum extent permitted by applicable law, IDM Computer Solutions, Inc. disclaims all other warranties, either express or implied, including but not limited to implied warranties of merchantability and fitness for a particular purpose, with respect to the SOFTWARE and any accompanying written materials.

NO LIABILITY FOR CONSEQUENTIAL DAMAGES. To the maximum extent permitted by applicable law, in no event shall IDM Computer Solutions, Inc. be liable for any damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information, or other pecuniary loss) arising out of the use or inability to use the SOFTWARE, even if IDM Computer Solutions, Inc. has been advised of the possibility of such damages.

Sales and Support

Technical Support



You can obtain free support for UltraEdit if you've registered and paid for a license. You may email any questions/suggestions to:

support@idmcomp.com

Sales



For the most recent information regarding licensing costs and options, please contact us at sales@idmcomp.com.

Registration may be placed by one of the following methods:

1) SECURE order form at: <https://www.ultraedit.com/store/customer/home.php>

2) FAX to (513) 892 4915

Telephone to (513) 892 8600.

3) MAIL - send payment/order to:

IDM Computer Solutions, Inc.

5559 Eureka Dr.

Suite B

Hamilton, OH 45011

USA

4) E-Mail to sales@idmcomp.com

Index

.

.NET Framework Code Protector dialog 68

[

[.EXT] Compiler Section..... 382

[Build] Compiler Section..... 381

[Environment] Compiler Section 377

[FileGroups] Compiler Section 380

[General] Compiler Section..... 377

[GroupFormats] Compiler Section..... 381

[InsertFiles] Compiler Section 380

[MakeCommands] Compiler Section 379

[Settings] Compiler Section 375

[SettingsInfo] Compiler Section..... 376

[SettingsReps] Compiler Section..... 376

[Variables] Compiler Section 377

1

1, 2 ... command..... 369

3

32-bit..... 338

6

64-bit..... 338

A

About command 371

Accounts/Settings command..... 177

activeDocument..... 37

activeDocumentIdx..... 37

Add Help File command..... 370

Add Indents command 223

Add Project command 192

Add To Project command 160

Adding A New Compiler 383

Advanced..... 260

Advanced Build Options command ... 188

Advanced menu..... 258

Advanced Sort/Options 95

Advanced sorting..... 77

Allow Docking 28, 200

Allow Multiple Instances..... 322

Allow Positioning Beyond Line End... 294

Alternate Capture Method command 358

Alternate method to open CHM files . 322

Always On Top command..... 216

Animate Configuration Dialogs 322

ANSI	98	Auto Scroll Polled Files command.....	216
ANSI Character Set	217	Auto-complete.....	261, 334
ANSI to OEM command.....	98	Auto-Correct.....	298
APPDATA.....	65, 324	Auto-hide scrolling.....	322
Append.....	235	Automatically re-encrypt file on close	283
Application Layout.....	259, 317, 319	Automatically unfold hidden areas on Find and Goto	293
Application Wizard	5	Automatically Update Changed Files	284
Arrange Icons command.....	368	Autosizing	291
ASCII	59, 81, 98, 99, 100	B	
ASCII Table	193, 202	Back.....	148
ASCII Table command	202	Back/Forward command	148
ASCII to EBCDIC command.....	98	Background Color Automatic.....	298
ASCII to Unicode command	99	Backup	88, 160, 164, 278, 395
ASCII to UTF-8 command	100	Backup Files on Save.....	278
ASP_LANG	298	Backup Project/Workspace command	160
Aspell.....	119, 275	Backup/Restore User Customizations	258
Aspell parent directory.....	273	Base Class	156
Associations	290	Batch Build command.....	183
Auto Code Page Detection	281	Batch Builder.....	164
Auto Complete	261	Big Endian	77, 101
Auto Indent	295	Binary	81
Auto indent new lines	295	Binary/HEX file.....	260
Auto indent wrapped lines	295		

- Blank Edit files 395
- Block Comment Off 298
- Block Comment On 298
- Block Comments 298
- Blocking Text 2
- BOM 289
- Bookmark column with line 262
- Bookmarks 58, 120, 262
- Brace Highlight 208
- Brace Matching 298
- Bracket/brace 143
- Branch 175
- Branch/Tag 169
- Breakpoint 73
- Build 182, 381
- Build command 182
- Build menu 181, 187
- Build Options dialog 188
- Build Toolbar 69
- C**
- Call Stack 73
- Capitalize command 223
- Capture Output checkbox 358
- Carriage Return/Line Feed 126
- Cascade command 367
- Case Sensitivity 298
- Center Align 219
- Center Justify command 233
- Change Polling command 216
- Character Delimited command 233
- Character Properties 147
- Characters 146
- Check files for changes only when
UltraEdit receives focus 284
- Check for Updates 371
- Checkout command 169
- Chmod 81, 84
- Chords 331
- ClassViewer 2, 70
- Clean command 183
- Clear Cache 336
- Clear History 330
- Clear Output 28
- ClearType 291
- Clipboard History 205
- Clipboards 111, 112, 230, 231, 264, 399
- Close All Files command 80, 368

Close All Files Except This command	80	Command Line Options	71
Close Brace Strings	298	Command Line Parameters.....	237
Close command.....	80	Command Output.....	358
Close Fold Strings.....	3, 298	Comment Add/Remove commands ..	119
Close Project.....	155	Comment Selection.....	109
Close Project/Workspace command.	155	Commit.....	172
Close Solution command	192	Commit command	172
Close Tab Button.....	319	Compare	92
CMD.EXE	358	Compare Files command.....	92
COBOL_LANG	298	Compare Type	92
Code Folding.....	3, 293	Compilable.....	69, 181
Code Page Detection	281	Compile.....	182
Collapse All	3	Compile command	182
Collapse All command.....	215	Compiler.....	164
Color Selection	298	Compiler Configuration Editor	373
Color Selector dialog	68	Compiler Options.....	183, 383
Color Value.....	152	Compiler Options dialog.....	5
Column menu.....	228	Compiler Paths dialog	181
Column Mode.....	2, 193, 210, 229, 263	Compiler Section ..	376, 377, 379, 380, 381
Column Mode command	229	Compiler/linker	383
Column/block	228	Concurrent Versions System.....	393
Comma-delimited	266	Configuration.....	330
Command Line	358	Configuration Dialog	298

- Configuration Summary 259
- Confirm removing File/Directory from Favorites 317
- Console application..... 358
- Conversions 59, 99, 100, 231, 281
- Convert CR/LFs..... 220
- Convert Wrap..... 220
- Copy 111
- Copy command..... 111
- Copy Selected Line to Clipboard 28
- Copy to Clipboard 28
- Copy/append..... 264
- CR..... 126
- CR/LF 121, 126, 220, 281, 295
- Create Dir 81, 84
- Creating Your First Application 5
- CSE HTML Validator 217, 226
- CSE HTML Validator command..... 226
- CSHARP_LANG..... 298
- CSS_LANG 298
- CTAG Parameters..... 179
- Ctags 2, 17
- Ctags support..... 179
- CTRL+F4 80
- Cursor Arrows 10
- Cursor Arrows (Cursor Movement) 10
- Cursor Editor 17
- Cursor Movement..... 10, 23, 29
- Cursor/Caret..... 294
- Customization 324
- Customize Menu..... 63
- Customize Toolbar 65, 324
- Cut 111
- Cut Columns command..... 230
- Cut command..... 111
- CVS .. 17, 169, 170, 172, 174, 175, 177, 393
- CVS Checkout 177
- CVS Export dialog..... 175
- CVS Import dialog 174
- CVS in UEStudio '05..... 393
- CVS Manager dialog..... 393
- CVS Server 177
- CVS Settings..... 177
- CVS Support 170, 172
- D**
- Date/Time command 149
- DATE_USER..... 349

DATE_USER_END	349	Delete Line command.....	114
DATE_USER_SYS	349	Delete Macro/Modify Properties command	253
DDE	277	Delete original file(s) after encrypting	283
Debug.....	69, 338, 373	Delimiters.....	231, 263, 298
Debugger	338	Detect Unicode	289
Debugger key.....	183	Detect XHTML from DTD on file load	293
Debugger toolbar.....	73	Detection/conversion	282
Debugging Tools	338	Dictionary	273, 275
Decode Base64	109	Directories	331
Decrease Font Size command.....	210	Disable all Workspace	317
Decrypt Files	90	Disable Drag	264
Default Checkout Root Path	177	Disable Line Numbers	288, 296
Default Debugger	338	Disable Workspace Close Buttons	193
Default Open Directory	331	Disassembly	73
Default Project File Directory.....	331	Display Line Number	102
Default Tagfile.....	179	Display Line Numbers command.....	213
Delete	114, 115	Display prompt opening read-only project.....	317
Delete Active File command.....	101	Display Ruler command	213
Delete All command.....	253	Display/Modify Templates command	349
Delete all hidden lines	109	DLL	160
Delete blank lines.....	396	Do not convert spaces and tabs inside comments and strings	295
Delete Columns command	230	Dockable tabs	319
Delete command	114		

- Dockable Windows..... 322
- Don't move focus to function list on
refresh 264
- DOS..... 77, 97, 282, 358, 395
- DOS command 97
- DOS Command command..... 353
- DOS Commands ONLY 358
- DOS to MAC command..... 97
- DOS to UNIX command 97
- DOS Window..... 358
- DOS/UNIX/MAC Handling 282
- Drag and Drop Editing 16
- Drag and Drop Files..... 17
- Draw graphical lines..... 293
- Drive/File List 200
- Duplicate Line 115
- Duplicate Window command..... 367
- Dynamic Link Library 160
- E**
- EBCDIC 77, 98, 117, 281
- EBCDIC to ASCII command..... 98
- ECMA_LANG..... 298
- Ecmascript..... 298
- EDIT file 266, 395
- Edit Macro command..... 238
- Edit menu..... 109
- Editor color preferences 296
- Enable Postmortem Debugging 338
- Enable show/hide lines for non-syntax-
highlighted files..... 293
- Encode Base64 109
- Encrypt File 88
- Encryption..... 283
- End 23
- Environment Selector dialog 195
- Environments 193
- Escape Char 298
- Escaped Unicode 289
- Expand All command..... 215
- Export..... 175
- Export command 175
- Export Options 175
- Export Settings 258
- Extensions..... 298
- Exuberant Ctags..... 179
- F**
- F12..... 87
- F7 149

Favorite Files.....	107	Fill Search In Columns with selection points.....	269
Favorite Files command.....	107	Filter.....	81
FF FE	59	Find command	121
Field Widths.....	231, 233	Find in Files command	133
File Associations.....	277, 290, 322	Find Next button	121
File Change Detection.....	284	Find Next command.....	126
File Change Polling.....	216, 284	Find Prev command.....	126
File Change Polling command	216	Find Results	133
File Filter dialog	17	Find Symbol	120, 147, 179
File Handling 277, 278, 281, 284, 285, 287, 288		Find Symbol in Active Document.....	17
File Handling Items	395	Find Symbol in Project Files	17
File Manager	17	Find/replace.....	138
File menu	77	Fixed auto-adjust Tabs.....	319
File Open dialog	79, 396	Fixed Columns.....	231
File Tab	199	Flat List.....	203
File Tabs	319	Flat-Button tabs	319
File Tree View	17, 200, 321	Folder Options	398
File Tree View command	200	Format.....	295, 395
File Tree View Interface	17	Format menu.....	217
File Types	99, 208, 277, 290, 291	FORTTRAN_LANG	298
Files Listed	133	Forward.....	148
Files selected in Explorer will open in a new single instance.....	322	FTP	81, 84, 285
		FTP accounts file.....	333

FTP accounts in explorer view 333

FTP Tunneling 277

FTP/SFTP Browser 81

Full path for JavaScript Lint
configuration file 348

Full Screen Mode command..... 217

Function Definition Strings..... 298

Function List command 146

Function Strings..... 203

Function Tips..... 335

Functions 335

G

Getting Started..... 1

GLOBAL VARIABLES 381

Globals 335

Go To Definition..... 17

Go To Error/Warning 28

Goto 58, 144

Goto Bookmark..... 58, 144

Goto Line/Page/Bookmark command144

Group 160

GroupFormats..... 381

GUID Generator 68

H

Handled Extensions 347

Header/footer 102

Help menu 369

HEX 115, 117, 210

HEX 00..... 260

Hex Copy Selected View command.. 118

Hex Edit command..... 115

Hex Edit Display 296

Hex Edit/EBCDIC command..... 117

Hex Editing 117

Hex Insert/Delete command..... 117

HEX Mode 117, 118

Hex/EBCDIC 117

Hidden Lines 214

Hide Directories 81

Hide File List 200

Hide Lines..... 214

Hide Output Window(s) 28

Hide/Show Lines..... 3, 214

Hide/Show Lines submenu 215

Hide/Show Selection..... 214

Hide/Show Selection command..... 214

Highlight All Items Found	121	Ignore Fold Strings.....	298
Highlight new file as	298	Ignore Spaces.....	92
Home	23	Import	174
Home and End keys (Cursor Movement)	23	Import command.....	174
Home Key Always Goto Column	264	Import Settings.....	258
How can I control the format of new files?	395	In Files/Types.....	133
How can I delete blank lines?	396	Increase Font Size command.....	210
How can I do a multi-line Replace?...	396	Incremental Search.....	132
How can I print syntax highlighting?..	396	Indent Spaces	266
HTML Tags.....	204	Indent/Unindent Strings.....	3, 215
HTML Tidy	68, 224	Indents	223
HTML Tidy command	224	Index.....	369, 370
HTML Tidy Output.....	224	Index command	370
HTML Toolbar Commands	67	INI file	258, 317
HTML Toolbar Configuration	327	INI File Location.....	317
HTML_LANG.....	298	Insert.....	264
I		Insert Color command	152
IBM-500	281	Insert command.....	156
Icon Editor	17	Insert File command	149
IDE	334	INSERT key.....	24
Ignore Blank Lines	92	Insert Line	149, 230
Ignore Case.....	92	Insert Literal Character	151
		Insert Mode	294

Insert Number command..... 230
 Insert Page Break command..... 149
 Insert Resource dialog 17
 Insert Smart Space 334
 Insert String..... 150
 Insert Template command..... 151
 Insert/Fill Columns command..... 229
 Insert/Overstrike Modes 24
 Inserting Special Characters 24
 IntelliMouse 397
 IntelliTip..... 337
 IntelliTip Toolbar 70
 IntelliTips..... 70, 334, 335, 336, 337
 International Distributors 405
 Invert Case command 223
J
 JAVA_LANG 298
 JavaScript..... 37
 JSCRIPT_LANG..... 298
K
 Key Mapping 331
 Keyboard Commands..... 25
 Keywords List..... 70

L

Language Definition 298
 Language Markers 298
 Large Buttons..... 65, 324
 Last DOS Command..... 353
 Leading Zeros..... 230
 Left Align..... 219
 Left Justify command..... 233
 LF 126
 License Agreement 402
 License Agreement - Registered 402
 License Agreement - Unregistered.... 401
 LIMITED WARRANTY 401, 402
 Line 146
 Line Change Indicator 193, 296
 Line Comment Alt..... 298
 Line Comment Num..... 298
 Line Comment Preceding Chars..... 298
 Line Comment Valid Columns..... 298
 Line Comments..... 298
 Lines Containing String 121
 Linker Options..... 183
 List Box 358

Lnk	285	Mapping	331
Load.....	234, 235, 256, 260, 284, 285, 322	Marker Characters	298
Load command	235	MASM_LANG	298
Local Root Path	169, 175	Match Brace command	143
Locking the File Tabs	199	Match Case	121, 126, 133, 138
M		Match Whole Word Only	121, 133, 138
MAC.....	77, 97, 282, 395	Menu Configuration.....	63, 324
Macro	235, 236, 237, 253, 255, 256, 257	Menu Configurations dialog.....	63
Macro List	203	Middle-button-click will close file.....	319
Macro menu	234	Minimize All Windows.....	366
MAIL	101	Mirror Display Font command	107
Mainframe.....	98	Miscellaneous	264
Make Configuration Local button.....	373, 383	Module Name.....	174
Make Copy/Backup command	88	MONTH/DAY/YEAR.....	349
Make Folder Active File Path	17	Move Line Down.....	109
Make Folder Active Path	200	Move Line Up.....	109
MakeCommands	379	Multi-Byte.....	59
Manage Menu Configurations	63	Multi-key Delay	331
Manage Toolbar Configurations	65	multi-key shortcuts	331
Manage Toolbars	324	Multiline.....	319
Management	329	multiple instances.....	322
Many wordfiles.....	397	N	
MAPI.....	101	Namespace	2

NASM_LANG.....	298	Open command	79
Network drive label format	321	Open File Highlighted	34
New command.....	78	Open File Tabs command	199
New File Creation.....	266	Open Files	133
New Group	155	Open Fold Strings	3, 298
New Menus	63	Open from FTP.....	81
New Project/Workspace command ...	153	Open large files.....	399
New Solution command	191	Open Project	154
New Toolbar	65	Open Project/Workspace command .	154
New Toolbars.....	65	Open Selected File	200
Next Window command	367	Open Solution command.....	191
No Scroll Left Side	212	Output Window	28, 193, 207
Notepad.exe.....	398	Output Window command	207
NULLS.....	260	Overstrike	264
Number Converter.....	68	Overstrike Mode	294
O		P	
OEM	77, 98	Page Break.....	144
OEM Character Set.....	193, 217	Page Break Character	102
OEM to ANSI command.....	98	Page Down.....	29
Offset.....	115	Page Margins.....	102
One-Symbol List.....	334	Page Setup command.....	102
Online Forums	369	Page Up.....	29
Open Brace Strings.....	298	Page Up & Page Down keys (Cursor Movement).....	29

Paragraph Setup/Formatting dialog ..	219	Print Setup command	106
Parse extensionless	336	Print Setup/Configuration	396
Parser	336, 347	Print Syntax Highlighting	102, 396
PASCAL_LANG	298	Project	155
Paste command	111, 397	Project Converter command	163
Pasted paragraph	397	Project Converter dialog	163
PERL_LANG	298	Project Files	133
Persistent Selection	113	Project Files/Drive	200
Personal data directory	273	Project List	164
PHP_LANG	298	Project Manager	164
Play Again command	256	Project Manager command	164
Play Any/Multiple Times command ...	256	Project Manager dialog	152
PLB_LANG	298	Project menu	152
Poll	216	Project Path	164
Port	346	Project Settings	155, 322
Post Macro Record Options	256	Project Settings command	155
Postscript	102	Project Settings dialog	373
Preserve Case	126	Project Templates	164
Previous Window command	367	Project/workspace	152, 153, 160, 161
Print All Files command	102	Prompt to accept code page	281
Print command	102	Pserver	177
Print Line Numbers	102	PYTHON_LANG	298
Print Preview command	102		

Q

Quick Open command..... 79

Quick Record command..... 236

Quicktips 337

R

READ ONLY 109

Read Only command..... 119

Read-Only 285, 287

Rebuild All command..... 183

Recent Files 108

Recent Files command..... 108

Recent Projects 108

Record command..... 254

Redisplay Tip 335

Redo 110, 111

Redo command 111

Reformat 397

Reformat Paragraph command..... 218

Refresh Browser View 193

Regenerate Makefile..... 181

Registers..... 73

Regular Expressions... 121, 126, 133, 138,
146, 298, 395, 396

ReIndent Selection command 227

Relative Paths..... 155

Release Tag..... 174

Reload FTP 285

Remove Indents command..... 224

Remove lines 395

Remove Project command 192

Remove separators..... 396

Rename File command 91

Reparse 70

Replace..... 138

Replace All 126

Replace command 126

Replace Encrypted File If Exists 283

Replace in Files command 138

Replace Where 126

Reset all 317

Reset All..... 329

Reset Fonts command 210

Resource Editor 32, 71

Resource Editor Toolbar 71

Resource Files..... 380

Restore Project/Workspace command
..... 161

Revert to Saved command 87

Revision Options	170	Save Selection As command	88
RGB	152	Scan First Line Only	231
Right Align	219	Script List	206
Right Justify command	234	Script menu	258
Right Mouse Button Commands	34	Script Toolbar	71
Root web directory	346	Scripting	37, 258
Ruby	71	Scripting commands	37
Ruby EXE	71	Scripting Demo and Tutorial	35
Run Active Script	258	Scripts	258
Run App	69	Scroll bars	212, 368
Run Application command	187	Scroll Output	28
Run Commands command	187	Scroll tabs in single line mode	319
Run Windows Program command	355	Search menu	120, 126
S		Searching	269
Sales	405	Secure delete original files using UltraSentry	283
Save	87, 88, 257, 278, 287	Select All command	112
Save All command	88, 236	Select Case	298
Save As command	87	Select Compiler command	187
Save As to FTP	84	Select Compiler dialog	69, 181, 373
Save changes	80	Select Line command	113
Save command	87	Select Range command	113
Save folded lines	293	Select Word	109
Save project information for use on multiple systems	277	Selecting Text	2, 58, 223, 227, 229

Send Mail command	101	Show Spaces/Tabs command	211
Set Auto Load command	237	Show status information in output window	334
Set Breakpoint	73	Show Symbol List.....	334
Set Code Page/Locale	348	Show Tool tips	28
Set Color dialog	208, 213	Single line - drop down file list.....	319
Set Colors command	208	Single line - scrollable.....	319
Set Column Markers command	211	Smart Spacing	335
Set Compiler Options command.....	188	Solution	190, 192
Set Compiler Paths command	189	Solution dialog	192
Set Compiler Paths dialog	189	Solution file.....	191
Set Filter.....	17	Solution menu	190, 192
Set Font command.....	210	Sort File command	94
Set Hex/Column Mode Font	210	Sort Tabs on File Open	319
Set Macro for File Load/Save command	237	Spaces	221, 222
Set Printer Font command.....	106	Spacing commands	214
Shell Extensions.....	368	Special Characters.....	24
Show Column Markers	211	Special characters (Find/Replace)	121
Show Key Mapping in Editor	331	Spell Check command	119
Show last line of fold in syntax highlighted files	293	Spell Checker.....	109, 119, 275
Show Log.....	81, 84	Split Window	366
Show Names Only	200	Start Debugger	183
Show Page Breaks	211	Start Debugger command.....	183

Start Xdebug using the active document	346
Status Bar	59, 112, 193, 199, 398
Status Bar command	199
STDERR	28, 188
STDOUT	28, 188
Stop Debugging	73
Stop Recording command	255
Store windows/menu/toolbars placements and configurations for multiple instances	317
String Literal Prefix	298
String Table	17
Struct	2
Style Builder dialog	68
Sub Window Background	208
Sub Window Text	208
SUM button	231
Sum Columns/Selection command ...	231
Support	405
Symbols List	70
Synchronized Scrolling command	368
Syntax Highlighting	119, 298

T

TAB	221, 222
-----------	----------

Tab Colors by File types	319
Tab Indentation	58
TAB key	58, 266
Tab Stop Value	266
Tabs	221, 222
Tabs to Spaces command	221
Tag List	193, 204
Tag List command	204
Tags	70
Tailling	284
Technical Support	1, 405
Temp file	395
Template List	205
Temporary Files	277, 288, 368, 395
Text Selection/Highlighting	58
Threshold	288
Tile Horizontal command	367
Tile Vertical command	367
Tile Windows Horizontally	17
Tile Windows Vertically	17
TIME_USER	349
Time-out before Auto-hide	322
To Lower Case command	223

- To Upper Case command..... 222
- Toggle Bookmark 145
- Toggle Bookmark command..... 145
- Toggle Browser View..... 193
- Tool Configuration dialog..... 358
- Toolbar command 199
- Toolbar Configuration 65, 324
- Toolbars 324
- Tools Toolbar 68
- Transfer Type..... 81, 84
- Tree View..... 193
- Trim Trailing 222
- Trim Trailing Spaces..... 287, 398
- Typedef 2
- U**
- U3 60
- UE3..... 60
- UEDIT32.UEK..... 331
- UESTUDIO.UEK..... 331
- UltraCompare..... 92
- UltraCompare Lite 92
- UltraCompare Pro 92
- Uncomment Selection 109
- Undo 126
- Undo command 109, 110
- Undockable 199
- Unicode..... 77, 99, 100, 101, 133, 138
- Unicode and UTF-8 Support 59
- UNICODE Big Endian..... 101
- UNICODE Big Endian to UNICODE
command..... 101
- Unicode Editing..... 59
- Unicode Search 138
- Unicode to ASCII command 99
- UNICODE to UNICODE Big Endian
command..... 101
- Unicode/ASCII/UTF-8..... 77
- Unicode/UTF-8 59, 77, 100
- Unicode/UTF-8 Detection 289
- Unindent Strings..... 298
- UNIX..... 77, 97, 282, 395
- UNIX/Linux 81
- UNIX/MAC 97
- UNIX/MAC to DOS command 97
- UNIX/MAC/DOS 281
- Update command..... 170
- Updates..... 369

Use DDE	277	View menu.....	193
Use Encoding.....	133, 138	Views/Lists.....	193
Use Locale.....	95	Virtual Destructor.....	156
Use Spaces instead of Tabs	28	Virtual Memory.....	95
Use UltraSentry	288	Visual Studio Project.....	163
User adjustable dockable Tabs.....	319	W	
Using Help	1	Warn on Exit.....	257
Using Help command	371	Watch Window	73
UTF-16	289	Web Search Toolbar	328
UTF-16BE.....	101	Where can I find additional wordfiles for UltraEdit?	397
UTF-16LE	101	Why does copy and paste stop working?	399
UTF-8	77, 99, 100, 289	Why does it take so long to open large files?	399
UTF-8 BOM	287	Why does my IntelliMouse not work with UltraEdit?	397
UTF-8 to ASCII command	100	Why does reformat break on hyphenated words?	397
UTF-8 to Unicode command	99	Why does the notepad.exe get overwritten?	398
V		Why does UltraEdit reformat a pasted paragraph?	397
Validator.....	226	Why is there a delay starting UltraEdit?	400
Variables.....	373	Wildcards.....	133
VB_LANG	298	WinDbg	183, 338
VBSCRIPT_LANG	298		
Vendor Tag.....	174		
View As	208		
View As command	208		

Window menu	366	Wrap Text When Printing.....	102
Windows Clipboard	112, 399	Write UTF-8 BOM.....	287
Word Count command.....	146	X	
Word Wrap	217, 220, 266, 295	XML	226
Word Wrap/Tab Settings.....	266	XML Convert	217, 227
Wordfile.....	298	XML Convert to CR/LFs command ...	227
Words.....	146	XML_LANG	298
Workspace.....	163	XMLlint	217
Workspace Manager.....	2, 296	XMLlint Tool	217