

Scripting commands

Article Number: 1245 | Last Updated: Mon, Oct 24, 2011 12:05 PM

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 + " ");
}

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 externalScript\externalScript.js
```

The include command must be preceded by a line comment. If a user 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
```

var_dump

This function is included outside the other objects defined for scripting in UltraEdit. It displays structured information about the referenced information including type and value. Arrays and objects are explored recursively with values indented to show structure.

Example:

```
var_dump(UltraEdit.frInFiles);
```

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 adI = 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:temptest.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	<div> <div>.directoryStart</div> <div>string</div> <div>full path of directory where search should begin</div> </div> <div> <div>.encoding</div> <div>int</div> <div>value: Encoding value to be used for search</div> </div> <div> <div>.filesToSearch</div> <div>int</div> <div>value: 0 - Files Listed 1 - Open Files 2 - Favorite Files 3 - Project Files 4 - Solution Files</div> </div> <div> <div>.ignoreHiddenSubs</div> <div>boolean</div> <div>true/false</div> </div> <div> <div>.matchCase</div> <div>boolean</div> <div>true/false</div> </div> <div> <div>.matchWord</div> <div>boolean</div> <div>true/false</div> </div> <div> <div>.openMatchingFiles</div> <div>boolean</div> <div>true/false</div> </div> <div> <div>.regExp</div> <div>boolean</div> <div>true/false</div> </div> <div> <div>.searchSubs</div> <div>boolean</div> <div>true/false</div> </div> <div> <div>.unicodeSearch</div> <div>boolean</div> <div>true/false</div> </div> <div> <div>.useEncoding</div> <div>boolean</div> <div>true/false</div> </div> <div> <div>.searchInFilesTypes</div> <div>string</div> <div>files/types to search in</div> </div> <div> <div>.find("string to find");</div> <div>.replace("string to find", "string to replace");</div> <div>** Only used with replace **</div> <div> <div>.logChanges</div> <div>boolean</div> <div>true/false</div> </div> <div> <div>.preserveCase</div> <div>boolean</div> <div>true/false</div> </div> <div>** Only used without replace **</div> <div> <div>.displayLinesDoNotMatch</div> <div>boolean</div> <div>true/false</div> </div> <div> <div>.reverseSearch</div> <div>boolean</div> <div>true/false</div> </div> <div> <div>.useOutputWindow</div> <div>boolean</div> <div>true/false</div> </div> </div>	<p>Searches through specified files to find the string in quotes "" based on the parameters specified.</p> <p>Find in Files Example:</p> <pre>UltraEdit.frInFiles.directoryStart = "c:temp"; UltraEdit.frInFiles.searchInFilesTypes = "*.txt"; UltraEdit.frInFiles.useOutputWindow = true; UltraEdit.frInFiles.find("3939");</pre> <p>Replace in Files Example:</p> <pre>UltraEdit.frInFiles.directoryStart = "c:temp"; UltraEdit.frInFiles.searchInFilesTypes = "*.txt"; UltraEdit.frInFiles.useOutputWindow = true; UltraEdit.frInFiles.replace("3939", "7878");</pre>

getString	<p>String in quotes ("") used in the prompt dialog</p> <p>Optional flags :</p> <p>0 int</p> <p>1 int return value</p>	<p>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:</p> <p>UltraEdit.getString("What is your name?");</p> <p>or</p> <p>var str = UltraEdit.getString("What is your name?",1);</p> <p>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.</p>
getValue	<p>String in quotes ("") used in prompt dialog</p> <p>Optional flags:</p> <p>0 int</p> <p>1 int return value</p>	<p>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:</p> <p>UltraEdit.getValue("How old are you?");</p> <p>or</p> <p>var str = UltraEdit.getValue("How old are you?",1);</p> <p>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.</p>
insertMode		Change text editing mode to insert mode for characters typed.
insOvrMode	** This is a READ ONLY property.	<p>Returns BOOLEAN value indicating if insert mode is on. Example:</p> <p>var insertActive = UltraEdit.insOvrMode;</p>
messageBox	<p>Message text in quotes ("")</p> <p>Title text in quotes ("") (optional)</p>	<p>Presents message dialog with "OK" button. Example:</p> <p>UltraEdit.messageBox("Can't complete process", "Process Abort");</p>
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>

		UltraEdit.open("c:temptest.txt"); or UltraEdit.open('^c'); or UltraEdit.open("c:temp^c"); or UltraEdit.open("FTP::myserver.com/home/mypath/public_html index.html");
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.	Returns a number indicating active regular expression type. 0 = UltraEdit expressions 1 = Unix expressions 2 = Perl expressions Example: var regexType = UltraEdit.regexMode;
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. Example: UltraEdit.runTool("Script Tool");
save		Save active file.
saveAll		Save all active files.
saveAs	File name 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. UltraEdit.saveAs("c:temptest.txt"); or UltraEdit.saveAs("^s"); or UltraEdit.saveAs("^c");
selectClipboard	Clipboard number (0-9)	Select the specified clipboard 0 = Windows clipboard

		and 1-9 specifies user clipboards. Example: UltraEdit.selectClipboard(2);
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.
ASCIIToUTF8		Convert file from ASCII to UTF-8.
bottom		Jump to end of file.
cancelSelect		Clears any selection in active document. Example: <code>UltraEdit.activeDocument.cancelSelect();</code>
clearAllBookmarks		Clears all bookmarks in active document. Example: <code>UltraEdit.activeDocument.clearAllBookmarks();</code>
codePage	** This is a property of the active/specified document.	Returns value of code page for active document. Example: <code>var cp = UltraEdit.activeDocument.codePage;</code> May be used to set code page to be used for active document. Example: <code>UltraEdit.open("C:\temp\korean_file.txt")</code> <code>UltraEdit.activeDocument.codePage = 949;</code>
collapseAll		Collapse all foldable text in active file. Example: <code>UltraEdit.activeDocument.collapseAll();</code>
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	<div>Start Number int</div> <div>Increment int</div> <div>LeadingZero boolean true/false</div> <div>HEX boolean true/false</div>	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: <code>UltraEdit.activeDocument.columnInsertNum(2, 3 , false, true);</code>
columnLeftJustify		Left justify selected columns.

columnRightJustify		Right justify selected columns.
commentAdd		Insert the line comment string as defined in the syntax highlighting language used to highlight the current file at start of every selected line or the current line if there is no selection. This command has no affect on a file not highlighted with a syntax highlighting language or when the language has no line comment definition. The cursor position does not change and the selection remains.
commentRemove		Remove the line comment string as defined in the syntax highlighting language used to highlight the current file at start of every selected line or the current line if there is no selection. This command has no affect on a file not highlighted with a syntax highlighting language or when the language has no line comment definition. The line comment string must be at start of the line (column 1) without preceding whitespace. A line comment string after one or more spaces or tabs is ignored and not removed. The cursor position does not change and the selection remains.
CommentSelectionAdd		Insert the "block comment on" string at start and the "block comment off" string at end of a selection as defined in the syntax highlighting language used to highlight the current file. If nothing is selected on execution of the command, both strings are inserted at current cursor position. This command has no affect on a file not highlighted with a syntax highlighting language or when the language has no block comment definition. If the selected block contains already a block comment and the language does not support nested blocks comments, command CommentAdd is automatically executed instead of this command for inserting the line comment string at start of every selected line if the language supports a line comment. The cursor moves to start of the inserted "block comment on" string and the selection is removed after execution when the block comment strings are inserted.
CommentSelectionRemove		Remove the "block comment on" string at start and the "block comment off" string at end of a selection as defined in the syntax highlighting language used to highlight the current file. The command has no affect if nothing is selected on execution of the command. And the command works only if the current selection starts with the "block comment on" string and ends with the "block comment off" string. Whitespace inside the selection before "block comment on" string or after "block comment off" string are not ignored and result in not removing the block comment strings. The cursor moves to start of the selection and the selection is removed after execution when the block comment strings are removed.

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</div> <div>boolean</div> <div>true/false</div> </div> <div> <div>.matchWord</div> <div>boolean</div> <div>true/false</div> </div> <div> <div>.mode</div> <div>int</div> <div>value:</div> <div>0 - Current File</div> <div>1 - In Selection</div> <div>2 - All Open Files</div> </div> <div> <div>.regExp</div> <div>boolean</div> <div>true/false</div> </div> <div> <div>.searchAscii</div> <div>boolean</div> <div>true/false</div> </div> <div> <div>.searchDown</div> <div>boolean</div> <div>true/false</div> </div> <div> <div>.searchInColumn</div> <div>boolean</div> <div>true/false</div> </div> <div> <div>.fromCol</div> <div>int</div> <div>default: 0</div> </div> <div> <div>.toCol</div> <div>int</div> <div>default:- 1</div> </div> <div> <div>.find("string");</div> </div>	<p>Find the string in quotes "" based on the parameters specified. Example:</p> <p>UltraEdit.activeDocument.findReplace.matchWord = true;</p> <p>UltraEdit.activeDocument.findReplace.find("3939");</p> <p>or</p> <p>UltraEdit.document[0].findReplace.matchWord = true;</p> <p>UltraEdit.document[0].findReplace.matchCase = true;</p> <p>UltraEdit.document[0].findReplace.replace("Copper", "Silver");</p>

	<pre>.replace("string to find", "string to replace");</pre> <p>** Only used with replace **</p> <pre>.preserveCase boolean true/false</pre> <pre>.replaceAll boolean true/false</pre> <pre>.replaceInAllOpen boolean true/false</pre> <p>*supersedes .mode in replace</p> <pre>.selectText boolean true/false</pre> <p>*replaces only in selected text</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> <pre>UltraEdit.activeDocument.gotoBookmark(0);</pre>
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> <pre>UltraEdit.activeDocument.gotoBookmarkSelect(0);</pre>
gotoEndOfNextWord		<p>Jump to end of next word. Example:</p> <pre>UltraEdit.activeDocument.gotoEndOfNextWord();</pre>
gotoEndOfNextWordSelect		<p>Jump to end of next word and select all text from current caret position. Example:</p> <pre>UltraEdit.activeDocument.gotoEndOfNextWordSelect();</pre>
gotoEndOfPrevWord		<p>Jump to end of previous word. Example:</p> <pre>UltraEdit.activeDocument.gotoEndOfPrevWord();</pre>
gotoEndOfPrevWordSelect		<p>Jump to end of previous word and select all text from current caret position. Example:</p> <pre>UltraEdit.activeDocument.gotoEndOfPrevWordSelect();</pre>
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> <pre>UltraEdit.activeDocument.gotoLine(1,5);</pre>
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.

		<p>Use line number 0 to jump to the specified column in the current line while selecting text. Example:</p> <pre>UltraEdit.activeDocument.gotoLineSelect(1,5);</pre>
gotoPage	Numeric value of page to jump to	<p>Jump to specified page number. Example:</p> <pre>UltraEdit.activeDocument.gotoPage(5);</pre>
gotoPageSelect	Numeric value of page to jump to	<p>Jump to specified page number and select text from cursor position to page number. Example:</p> <pre>UltraEdit.activeDocument.gotoPageSelect(5);</pre>
gotoPos	Numeric value specifying position in number of bytes from beginning of file	Jump to specified position
gotoPosSelect	Numeric value specifying position in number of bytes from beginning of file	Jump to specified position (passed as parameter in number of char from beginning of file) while making a selection
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 into the file. This will insert spaces (HEX 20).
hexMode	** This is a READ ONLY property of the active/specified document.	<p>Returns a BOOLEAN value indicating if Hex mode is active. Example:</p> <pre>var hexActive = UltraEdit.activeDocument.hexMode;</pre>
hexOff		Turn Hex mode off - switch to text mode.
hexOn		Turn Hex mode on.
hideOrShowLines		Hide the selected lines, or if hidden, show the lines hidden at the line of the cursor.
insertLine		Inserts blank line below current cursor position.
insertPageBreak		Insert a form feed/page break character at the current cursor position in the file.
insertTemplate	Index of template	<p>Insert the specified template into the file. Example:</p> <pre>UltraEdit.activeDocument.insertTemplate(0);</pre>
invertCase		Invert the case of selected text.

isCharGt	"character"	<p>This checks if the current character at the cursor position is greater than the specified character. Example:</p> <pre> if (UltraEdit.document[1].isCharGt('k')){ //do these commands... } else { //do these commands... } </pre>
isChar	"string"	<p>This checks if the current character at the cursor position is the character specified. Example:</p> <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"	<p>This checks if the file extension of the 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()){ //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 { </pre>

		<pre>//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>
isReadOnly		<p>This command checks if the active document is set to read only. Example:</p> <pre>if (UltraEdit.activeDocument.isReadOnly()){ //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>
isWordWrap		<p>This command checks the word wrap state of the active document. Example:</p> <pre>if (UltraEdit.activeDocument.isWordWrap()){ //do these commands ... } else { //do these commands ... }</pre>
key	BACKSPACE DEL DOWN ARROW END HOME LEFT ARROW	<p>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:</p>

	PGDN PGUP RIGHT ARROW UP ARROW CTRL+END CTRL+HOME CTRL+LEFT ARROW CTRL+RIGHT ARROW	UltraEdit.activeDocument.key("BACKSPACE"); UltraEdit.activeDocument.key("CTRL+RIGHT ARROW");
length	** This is a READ ONLY property of the active/specified document.	Returns number of active documents. Example: var num_of_docs = UltraEdit.document.length;
lineTerminator	** This is a READ ONLY property of the active/specified document.	Returns a numeric value indicating line terminator type in active document. Example: var lt = UltraEdit.activeDocument.lineTerminator; Supported values: -2 = MAC but content of file contains currently DOS line terminators -1 = UNIX but content of file contains currently DOS line terminators 0 = DOS 1 = UNIX 2 = MAC
matchBrace		Find next matching brace and select the text between them.
moveLineDown		Move current line downward one line in active document. Example: UltraEdit.activeDocument.moveLineDown();
moveLineUp		Move current line upward one line in active document. Example: UltraEdit.activeDocument.moveLineUp();
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();
readOnlyOff		Sets active document as writeable
readOnlyOn		Sets active document as read only
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 .col2End int end column key 2 .col3Start int start column key 3 .col3End int end column key 3	Sort the file, or selected text according to specified parameters. Example: UltraEdit.activeDocument.sort.ascending = true;> UltraEdit.activeDocument.sort.ignoreCase = false; UltraEdit.activeDocument.sort.removeDuplicates = 1; UltraEdit.activeDocument.sort.remKey1 = true; UltraEdit.activeDocument.sort.remKey2 = true;

	<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>0 - false</div> <div>1 - all keys</div> </div> <div> <div>.removeDuplicates</div> <div>int match</div> <div>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</div> <div>0 - character order</div> <div>1 - numeric sort</div> <div>2 - use locale</div> <div>3 - alt. sort</div> </div>	<pre>UltraEdit.activeDocument.sort.type = 0; UltraEdit.activeDocument.sort.col1Start = 1; UltraEdit.activeDocument.sort.col1End = 15; UltraEdit.activeDocument.sort.col2Start = 35; UltraEdit.activeDocument.sort.col2End = 50; UltraEdit.activeDocument.sort.sort();</pre>
<div>sortAsc</div> <div>sortDes</div>	<div>Sort Type</div> <div>0 - Sort based on character order.</div> <div>1- Sort based on numeric value, not character order.</div> <div>2 - Specifies that the sort should be locale specific.</div> <div>3 - Specifies that the sort should use the alternate sort method. With the alternate sort method only a single sort key may be used.</div> <div>Ignore Case boolean true/false</div> <div>Remove Duplicates boolean true/false</div> <div>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> <pre>UltraEdit.activeDocument.sortAsc(0, true, true, 1, -1);</pre> <div>Example:</div> <pre>UltraEdit.activeDocument.sortDes(1, true, false, 4, 8);</pre>
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.
UTF8ToASCII		Convert file from UTF-8 to ASCII.
wordWrapOff		Turns off word wrap for active document
wordWrapOn		Turns on word wrap for active document
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: UltraEdit.activeDocument.write("This is a test."); or

		<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		<p>Clears contents of output window. Example:</p> <pre>UltraEdit.outputWindow.clear();</pre>
copy		<p>Copies contents of output window to active clipboard. Example:</p> <pre>UltraEdit.outputWindow.copy();</pre>
showOutput	boolean true/false	<p>Determines visibility of user specific data that is written to the output window. Currently that only includes outputWindow.write(). Example:</p> <pre>UltraEdit.outputWindow.showOutput=false;</pre>
showStatus	boolean true/false	<p>Determines visibility of all status information in output window (script name, success/failure of script, and error information). Example:</p> <pre>UltraEdit.outputWindow.showStatus=true;</pre>
showWindow	boolean true/false	<p>Toggles visibility of output window. Example:</p> <pre>UltraEdit.outputWindow.showWindow(true);</pre>
visible	** This is a READ ONLY property.	<p>Returns a BOOLEAN value indicating if output window is visible. Example:</p> <pre>UltraEdit.outputWindow.visible;</pre>
write	Text to write in quotes ("")	<p>Write specified text to output window. This will only support one line at a time and may not include line terminators. Example:</p> <pre>UltraEdit.outputWindow.write("This is a test.");</pre>

Posted - Thu, Oct 13, 2011 6:11 PM. This article has been viewed 5049 times.

Online URL: <http://www.ultraedit.com/help/article/scripting-commands-1245.html>
