Syntax Highlighting

Syntax highlighting is the ability to recognize predefined words and display them in a different color(s). 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.

UltraEdit provides support for twenty languages or word sets that may be recognized. 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.

The **Auto-Correct Keywords** checkbox enables or disables the Auto-Correct feature.

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 a configuration file (the default file is "WORDFILE.TXT"), to configure the syntax highlighting. This file is read each time the editor starts up. The size of this file may be up to 128KB. The syntax for this file is as follows:

Language Definition
The word set, or language is specified by a /Ln where n has a value of 1 to 20. 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 18 characters.

Line Comments
The comment characters used for line comments is specified by the string "Line Comment = " followed by the comment characters. Three characters are allowed, if there is 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. Three characters are allowed, if there is 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").

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 **five** 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 **five** 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.

File Extensions/Types
The file extensions that are applicable for this language are specified by the string "File Extensions = " and each extension is separated by a space.

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 18 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 = CPP C 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 = CPP C 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 = CPP C 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 = CPP C 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 a 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 line 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>
        <B> </B> <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 = ~!@$%^&*()_-+=|\/{}[]:;"'<> ,.?/

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 =".

With release 5.0 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.

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.

Note - The regular expression string must use UltraEdit style regular expressions. This ensures that the function list will always work irrespective of the regular expression settings.

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 out-dent 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 ""). For the word to match it must be the LAST non space characters of the line. 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 ""). For the word to match it must be the FIRST non space characters of the line. 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 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.

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.

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 is configurable. To specify the color, add the two characters under the appropriate color section as if they were a word such as "ab", "cd" etc.

This file may be modified by the user.