xretrace documentation - updated for xretrace V2.01

Thursday, 12 August 2021 3:22 pm

Using xretrace

xretrace allows you to retrace your footsteps where the edit cursor has been. It also allows you to retrace through lines that you have modified. The parameters in the xretrace control panel allow you to specify how long the edit cursor must remain on a line or in a region before that line/ region becomes a tracked location. Hence if you page up/ down or line up/ down quickly, you don't get a pile of unwanted tracked locations in the list.

xretrace provides the following commands that you can bind to keys

- *xretrace_cursor* - go to the most recent cursor location/region.
- *xretrace_modified_line* - go to the most recently modified line/region.
- *xretrace_modified_line_steps* - step through modified regions using an event loop.
- *xretrace_cursor_steps* - step through visited regions using an event loop.
- *xretrace_cursor_back* - go to the previous cursor location in the retrace list.
- *xretrace_cursor_fwd* - go to the next cursor location in the retrace list.
- *xretrace_show_control_panel* - set xretrace options (alternative is *xretrace_options*)

The *xretrace_cursor_back* command, when used repeatedly, steps through the list of tracked locations from latest to oldest. The *xretrace_cursor_fwd* command is the reverse of this. The *xretrace_cursor* command behaves like *xretrace_cursor_fwd* if *xretrace_cursor_back* has just been used, otherwise it alternates between the two last visited locations.

The "steps" macros run an event loop and can optionally show a popup dialog. When the event loop is running, the slick status bar shows the current line and position in the list. The popup dialog can be turned off/ on using the F5 key when the event loop is active. The screen position of the popup dialog can be selected by "right click" at the location you want the popup to appear - it's always on the main slickedit monitor, you can't choose which monitor.

ESC	Quit
ENTER/UP	Quit here
LEFT	Prev item
C-LEFT	Prev item, see all
A-LEFT	Prev buffer
RIGHT	Next item
C-RIGHT	Next item, see all
A-RIGHT	Next buffer
PAD STAR	Less <<<
R-CLICK	Set popup position
INS	Settings
F1	Help
F2	Xretrace source
F5	Hide/show popup
F6	Switch lists
F7	Restore mod lineflgs
F8	Toggle same/ALL buf
C-F4	Reset xretrace
A-F4	Disable xretrace
PGDN	Toggle bookmark
C-PGDN	Set bookmark

When the event loop is active, the pad-plus and pad-minus keys can also be used to step forward/ back - as well as the arrow keys. Hence if you bind Ctrl-pad-minus to *xretrace_cursor_steps,* you can navigate quickly.
See the *retrace_steps_event_loop2* function if you want to customise the key actions.

<u>Granularity options</u>
In the xretrace control panel, the "*line distance recording granularity*" determines the size of a region which if the cursor steps outside, an entry is added to the retrace list. The smaller the value, the more often entries are added. The *line distance viewing granularity* is used by *xretrace_cursor_steps* to allow you to skip over some entries in the retrace history. i.e. you can set a smallish value for *line*

distance recording granularity to capture lots of history and a larger value to jump in bigger steps when viewing. In the popup window above you can see that the left-arrow and right-arrow keys are used for stepping and the ctrl-left and ctrl-right keys are used for seeing everything in the retrace list.

Configuring xretrace

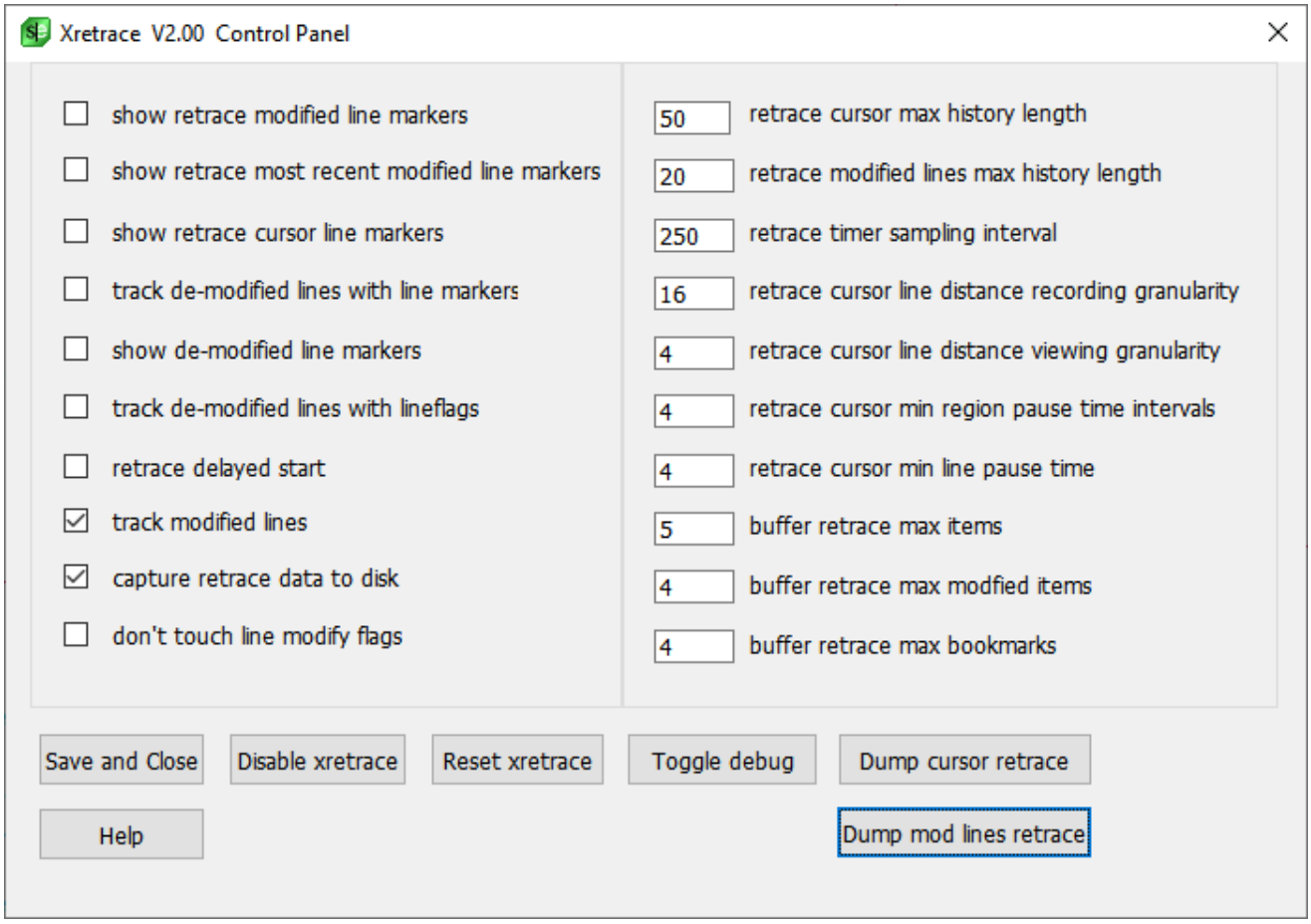
The control panel for xretrace is shown below. The parameters on the right hand side can be adjusted if you find that xretrace doesn't work exactly as you would like. xretrace control panel settings are stored in a file *xretrace_config.ini* in your configuration folder.

For the options on the left

- *show line marker* - options are normally disabled.
- *retrace delayed start* - is normally disabled.
- *capture retrace data to disk* - should be enabled if you want to use the xretrace scrollbar.
- *track modified lines* - should be enabled if you want xretrace to track modified lines.
- *don't touch line modify flags* - if this is enabled, xretrace won't clear the line modify flags when tracking modified lines and consequently, the order of entries in the modified line history may not match the order that the changes were actually made. If this is disabled, xretrace clears the "modified line flag" generated by slickedit so that it can detect when a line has been newly modified and allow the modified line history to have the correct order. If you have "colour modified lines" enabled (this is per language, selected on the "view" tab), slickedit changes the colour in the gutter at the left of each line to indicate if the line has been modified. <u>If you allow xretrace to clear the line modified flags the colouring in the gutter will be lost</u>. In slickedit options => "File options" => "save", if you enable "reset modified lines", the line modified flags are cleared when you save a file.

For the parameters on the right

- See the "using xretrace" section further below for an explanation of the granularity options.
- The "*retrace sampling interval"* is in milliseconds and determines how often the retrace cursor tracking code checks for a change in the edit cursor line number.
- The "*min region pause time interval*" selects the minimum amount of time that the edit cursor must be within a region before that region gets added to the retrace list - units of "*retrace sampling interval".* The "*min line pause time*" selects the minimum amount of time that the edit cursor must remain on a line for that line to be added to the retrace list.



The *toggle debug* button enables debug info to be output using the slick "say" function. The dump commands are also for debugging. You will probably need to widen the "say" debug window to see the output properly.

The reset button in the xretrace control panel can be used to reset xretrace (clear the lists) and start it running.

Installation from zip file

Unzip the supplied zip file *xretrace-slick-v-xx-x-x.zip* into your configuration folder. This will create a UserMacros folder (if it's not already there) containing an xretrace folder.

e.g.
…/SlickConfig/25.0.2/UserMacros/xretrace/

1. Open xload-macros.e in slickedit and load it using the "Load module" command in the "Macro" menu.

2. Run the command *xxutils_xretrace_load* to load xretrace and xxutils.

If xretrace loads successfully it will show the xretrace control panel for setting xretrace options. By default, "retrace delayed start" is enabled. You should uncheck this if you want xretrace to be always enabled. You can disable xretrace at any time using the xretrace control panel or the *xretrace_disable* command.

If for some reason you don't want to use the default path for the xretrace macros, you can modify the XRETRACE_PATH setting in xretrace.sh

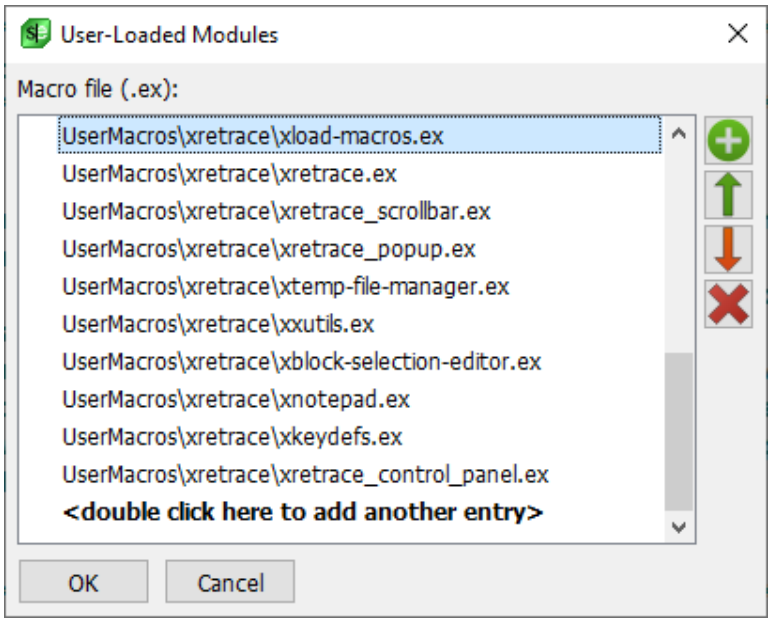
```
#define XRETRACE_PATH _ConfigPath() :+ 'UserMacros' :+ FILESEP :+ 'xretrace' :+ FILESEP
```

Choosing the path for xretrace source/ Installing a SlickEdit upgrade

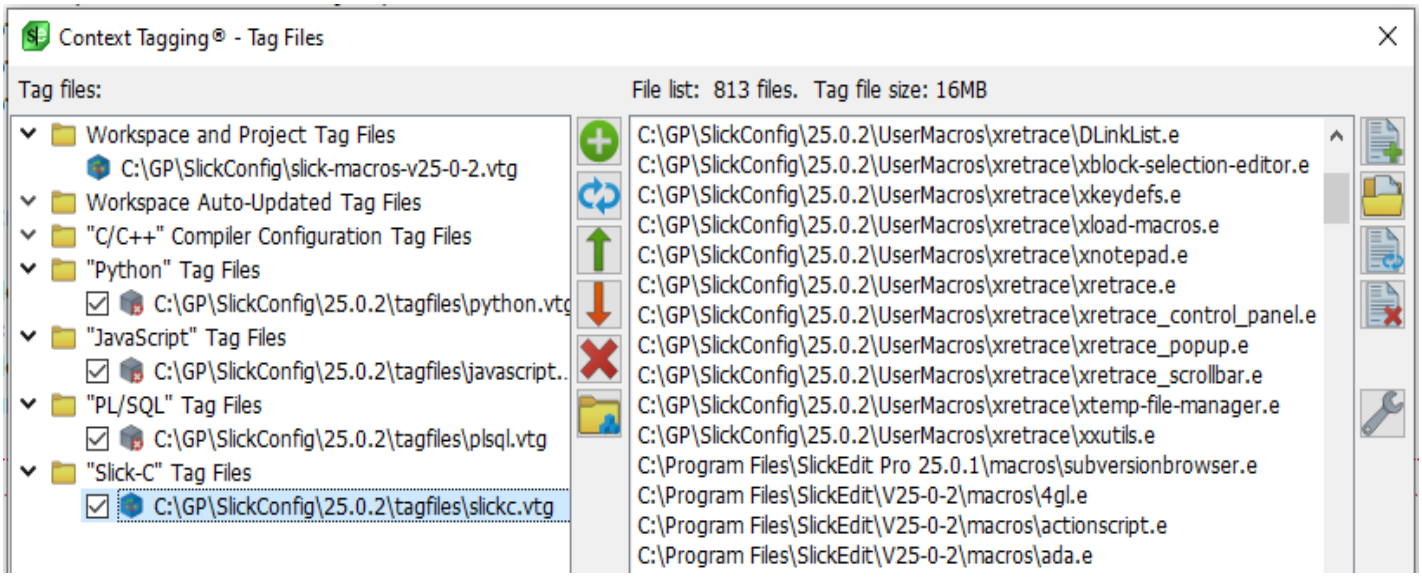
The default path for xretrace is set by the XRETRACE_PATH define in xretrace.sh

```
#define XRETRACE_PATH _ConfigPath() :+ 'UserMacros' :+ FILESEP :+ 'xretrace' :+ FILESEP
```

Older versions of xretrace had the xretrace folder directly in your configuration folder. This is now <config folder>/UserMacros/xretrace. If you previously had xretrace with the original path setting, you may end up with two copies of the xretrace source files in your "user loaded modules" list. This is undesirable. You should use the "list user loaded modules" command in the SlickEdit macro menu to remove the unwanted files. Select the file then click the "cross" to delete it. After deleting files from the list you should reload xretrace by loading the xload-macros.e file.



You may also want to remove unwanted duplicate xretrace source files from the slickC tag file. Click "tools" on the menu, then "tag files".



Older versions of slickedit may not copy and recompile xretrace source files when you install a slickedit upgrade.

When you install a SlickEdit upgrade and you allow slick to upgrade your configuration folder, slick may copy and rebuild all of the xretrace macros.

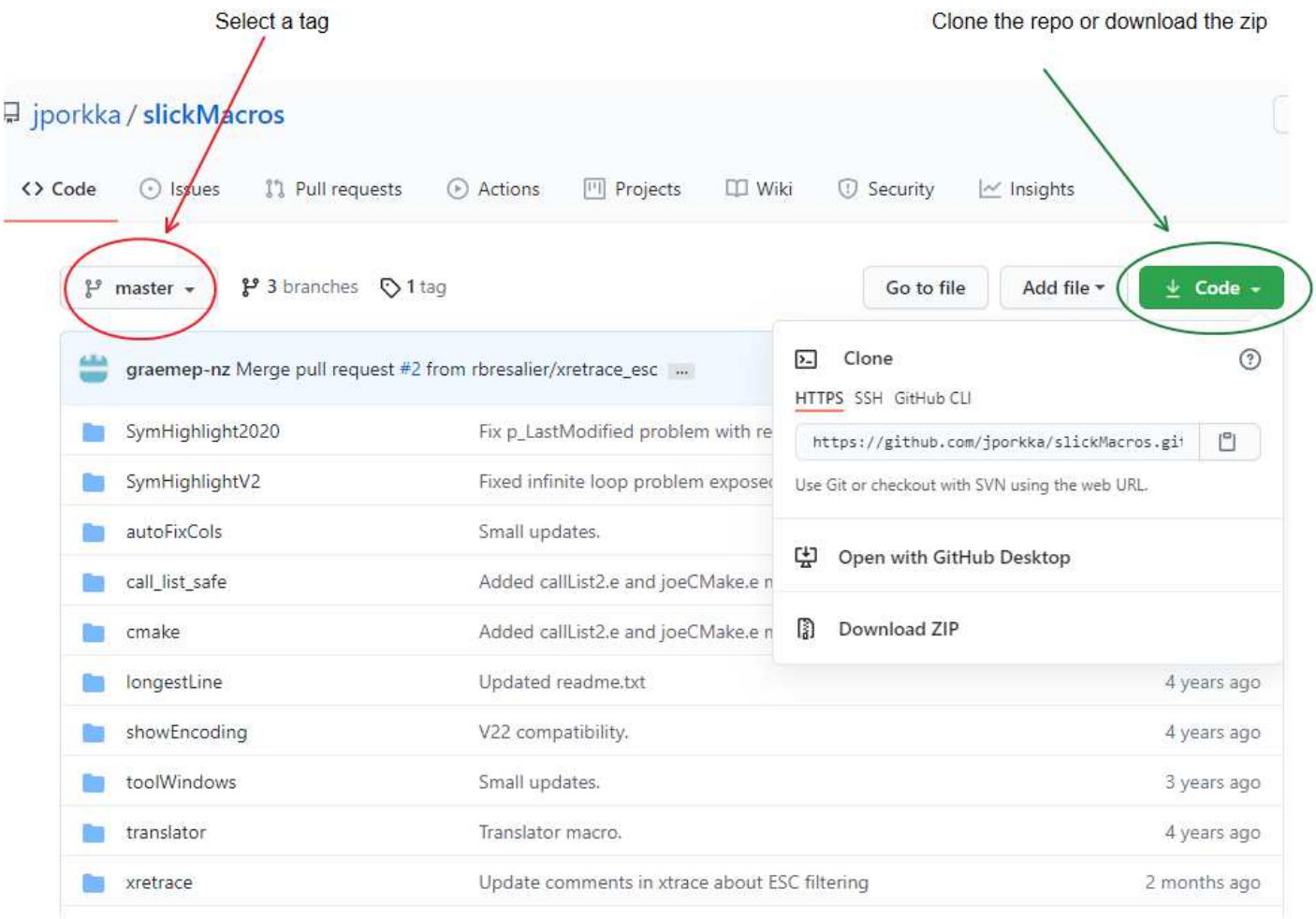
It is recommended that after installing any upgrade, you manually load xload-macros.e using the "Load Module" command on the Macro menu and run *xxutils_xretrace_load* to load xretrace and xxutils.
<config folder>/UserMacros/xretrace/xload-macros.e

Installation from github

Browse to https://github.com/jporkka/slickMacros

Select a tag according to the version of slickedit you have, then either clone the repo or download the zip file.
Create a *UserMacros* folder in your configuration folder, then copy the xretrace folder into the *UserMacros* folder.

e.g.
…/SlickConfig/25.0.2/UserMacros/xretrace/



1. Open xload-macros.e in slickedit and load it using the "Load module" command in the "Macro" menu.
2. Run the command *xxutils_xretrace_load* to load xretrace and xxutils.

If xretrace loads successfully it will show the xretrace control panel for setting xretrace options. By default, "retrace delayed start" is enabled. You should uncheck this if you want xretrace to be always enabled. You can disable xretrace at any time using the xretrace control panel or the *xretrace_disable* command.

If for some reason you don't want to use the default path for the xretrace macros, you can modify the XRETRACE_PATH setting in xretrace.sh

```
#define XRETRACE_PATH _ConfigPath() :+ 'UserMacros' :+ FILESEP :+ 'xretrace' :+ FILESEP
```

xretrace troubleshooting

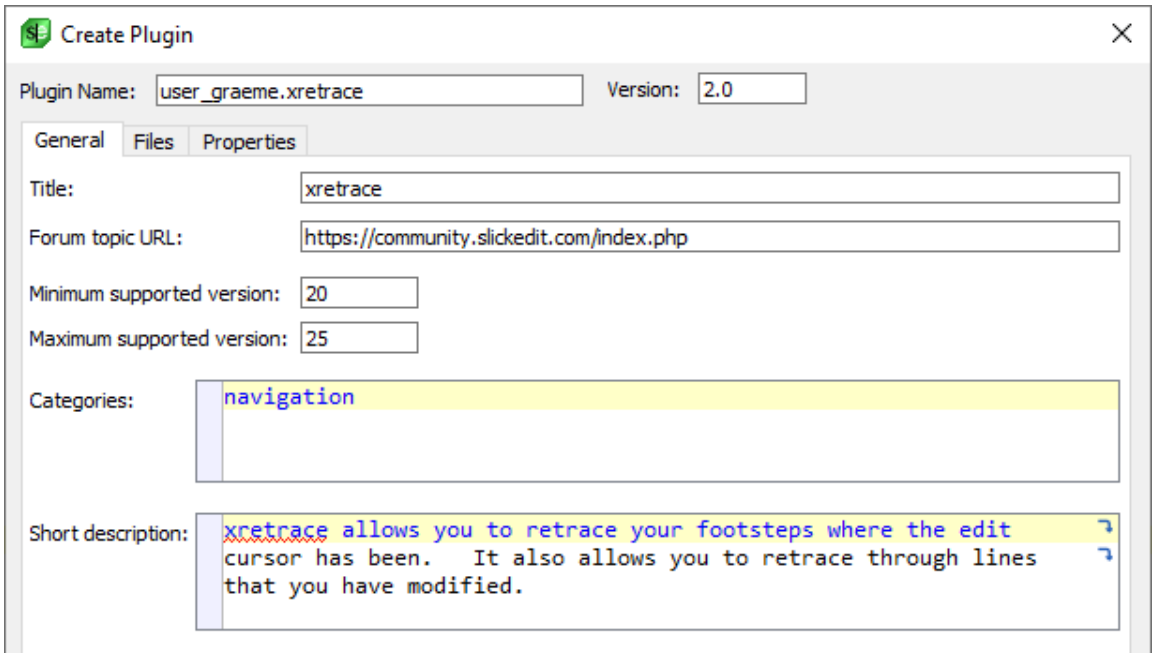
If xretrace doesn't seem to be working you can use the "toggle debug" button in the xretrace control panel to enable some debug output. xretrace will log information to a debug window when it adds or removes tracked locations from the retrace lists. You can also use the "dump" buttons to output the current content of the two retrace lists. Restarting slickedit may also help if xretrace is not working properly.

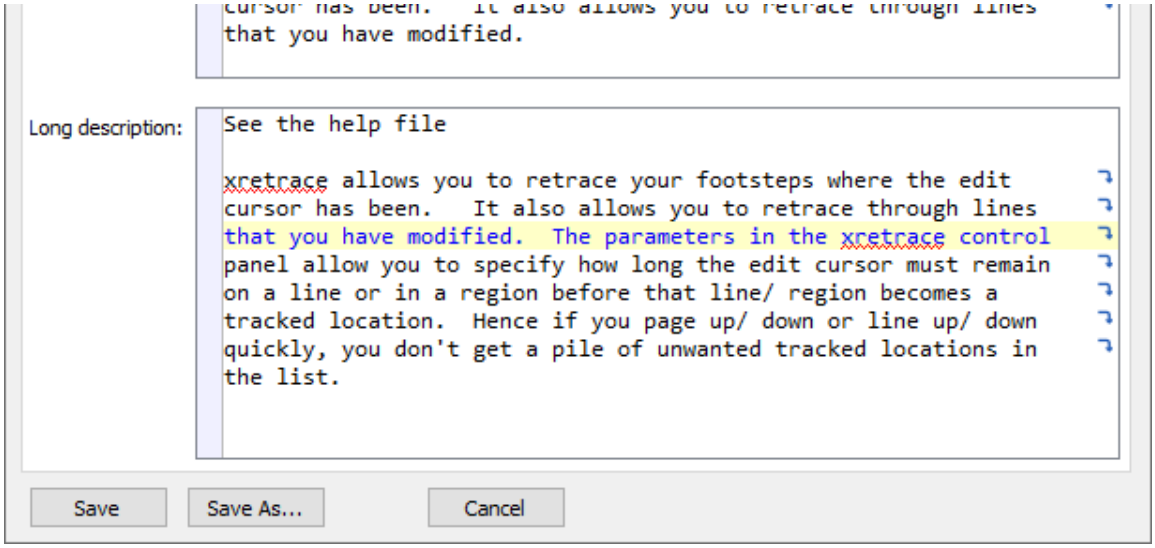
If you get repeated SlickC "stack errors", you may want to enable "retrace delayed start" in the xretrace control panel to prevent xretrace from running when you start slick. If slickedit won't run for long enough to let you use the xretrace control panel, you can prevent xretrace from running when slick starts by creating a file in your configuration folder.
`DontRunMyMacros.txt`
When xretrace sees this file at startup, it won't start running.

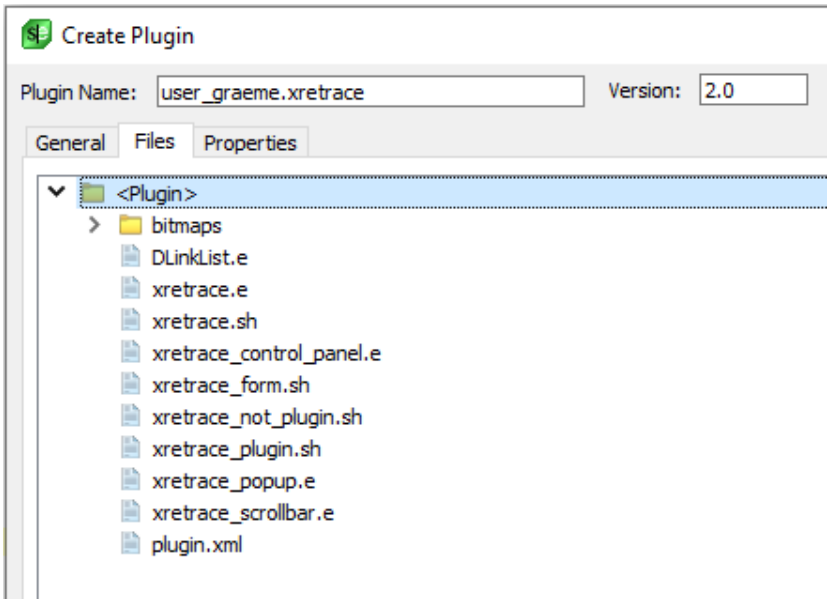
The reset button in the xretrace control panel can be used to reset xretrace (clear the lists) and start it running.

Building the xretrace plugin zip file

SlickEdit V25 includes a plugin mechanism. A plugin is a set of macro files packaged in a zip file with an xml file describing it. Slick has a plugin builder that can be invoked using the plugin-new or plugin-open commands.





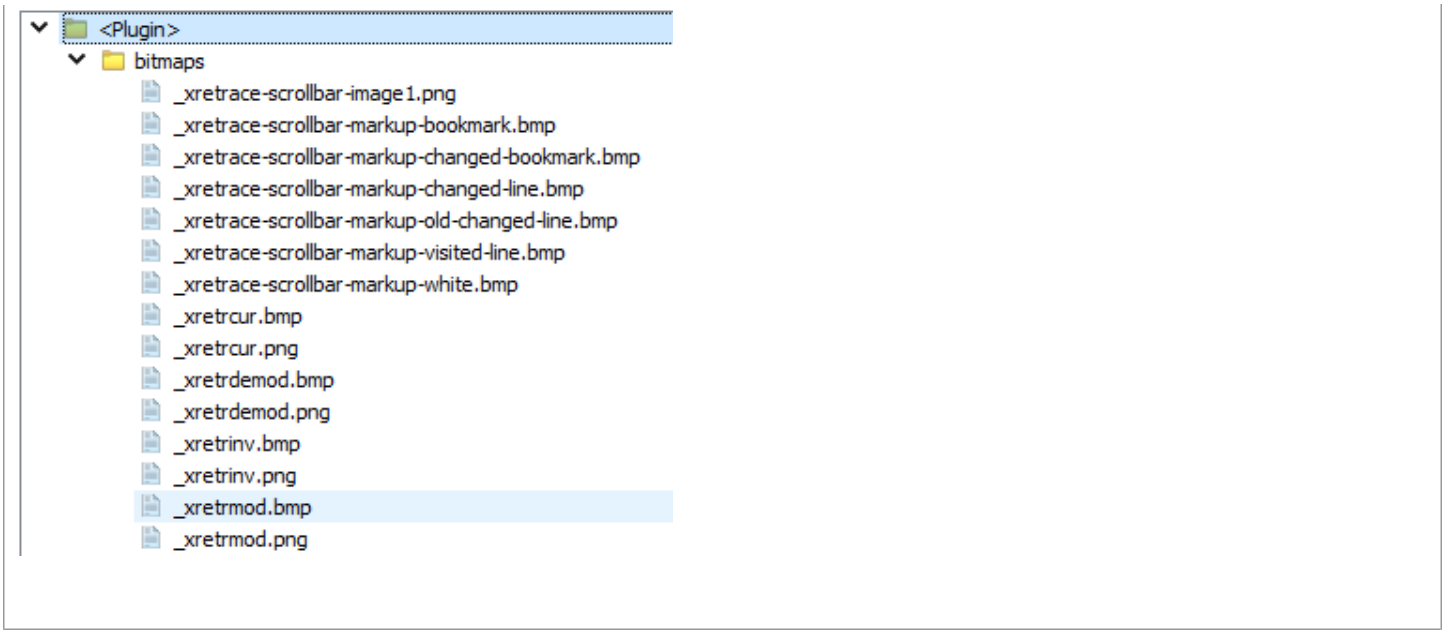


The content of xretrace.sh is different for the plugin than for the non plugin build of xretrace. For the plugin build, xretrace_plugin.sh must be copied to xretrace.sh. It has the following defines.

```
#define XRETRACE_IS_PLUGIN yes
#define XRETRACE_VERSION '2.0'
#define XRETRACE_PATH "plugin://user_graeme.xretrace.ver." :+ XRETRACE_VERSION :+ "/"
```

In GitHub, xretrace.sh is the version needed by the non plugin build.

The bitmaps folder has the following bitmaps. These are the same as for the non plugin build.


<Plugin>
bitmaps
_xretrace-scrollbar-image1.png
_xretrace-scrollbar-markup-bookmark.bmp
_xretrace-scrollbar-markup-changed-bookmark.bmp
_xretrace-scrollbar-markup-changed-line.bmp
_xretrace-scrollbar-markup-old-changed-line.bmp
_xretrace-scrollbar-markup-visited-line.bmp
_xretrace-scrollbar-markup-white.bmp
_xretrcur.bmp
_xretrcur.png
_xretrdemod.bmp
_xretrdemod.png
_xretrinv.bmp
_xretrinv.png
_xretrmod.bmp
_xretrmod.png

xxutils

Thursday, 12 August 2021 3:16 pm

xxutils popup menu and commands

The command ***show_xmenu1*** produces the menu below.

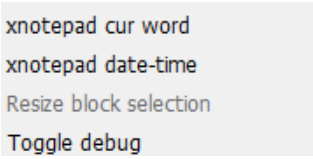
Accelerator keys : M - More, 1,2,3 - xfloat1/2/3, S - Selections, O - Open, C - Case, B - Bookmarks, K - keybindings

Names in **purple** are the names of _commands that can be called or bound to keys.

Set diff region	xset_diff_region xcompare_diff_region
Compare diff region	
Beautify project	xbeautify_project
Diff last two buffers	diff_last_two_buffers
--	
New temporary file	xtemp_new_temporary_file
More ▸	
--	
Transpose chars	search_cpp_ref, search_devdocs_cpp, xnotepad, xnotepad_word, start_xtemp_files_manager etc.
Transpose words	transpose-chars transpose-words
Transpose lines	transpose-lines
Append word to clipboard	xappend_word_to_clipboard
Copy names ▸	xcurbuf-name-to-clip etc.
Key bindings ▸	find_key_binding gui_keybindings
--	xkey_binding_trainer, xkey_bindings_show
Alternate last 2 buffers	alternate_buffers
Float 1	xfloat1 xfloat2 xfloat3
Float 2	
Float 3	
Set float ▸	xset_float1 xset_float2 xset_float3
Save app layout	xsave_named_toolwindow_layout xload_named_toolwindow_layout
Restore app layout	
Save session	save_named_state load_named_state (slick V23 or later)
Restore session	
--	
Bookmarks ▸	xsave_bookmarks xrestore_bookmarks
Complete ▸	complete_prev_no_dup complete_next_no_dup etc.
Select / Hide ▸	select_code_block hide_code_block etc.
Open / Explore ▸	xopen_from_here xopen_from_config explore_cur_buffer etc.
Case conversion ▸	lowcase-selection etc.

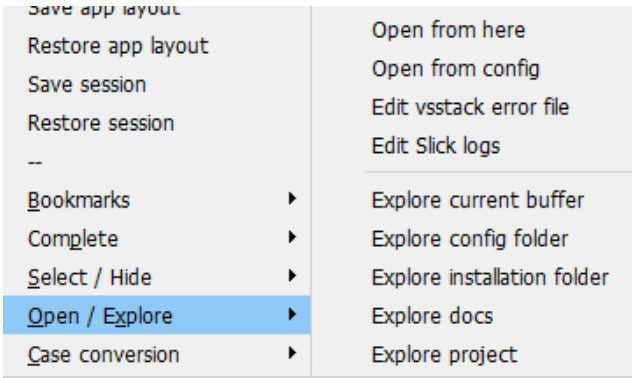
The "More" menu

Search cplusplus.com C	search_cpp_ref
Search devdocs C	search_devdocs_cpp
New temporary file no keep	xtemp_new_temporary_file_no_keep
Start xtemp file manager	start_xtemp_files_manager
Stop xtemp file manager	stop_xtemp_files_manager
xnotepad cur line or selection	xnotepad
xnotepad cur word	xnotepad_word
xnotepad date-time	xnotepad_create_time_date_string




```
xnotepad
xnotepad_word
xnotepad_create_time_date_string
xblock_resize_editor
toggle_xxutils_debug
```

The "Open/ Explore" menu



`xopen_from_here`	open from current buffer folder
`xopen_from_config`	open from configuration folder
`xopvss`	open vsstack slick error file
`xopen_logs`	open vs.log, stack.log, pip.log

Command descriptions

`xset_diff_region` `xcompare_diff_region`

These commands allow you to compare sets of lines. Use the xset command first, to specify the first region, then the xcompare command to specify the second region and run the diff tool. Both commands use the currently selected set of lines if there is a selection, otherwise they select 50 lines from the current cursor location. i.e.

1. Select a set of lines (optionally) - or, place the edit cursor at the start of the lines to be compared.
2. Run the xset_diff_region command
3. Select a second set of lines - or, place the edit cursor at the start of the lines to be compared.
4. Run the xcompare_diff_region command

`xbeautify_project`

This command beautifies all files in the current project. By default it will ask you for each file, whether to beautify that file - you can respond with

1. Yes
2. Yes to all
3. Cancel (quit the beautify)
4. No

Modified files are automatically saved. The file about to be beautified will appear in the preview window. You can write a command of your own that calls xbeautify_project with different options.

```
xbeautify_project(boolean ask = true, boolean no_preview = false, boolean autosave =
true);
```

`diff_last_two_buffers`
Runs diff on the last two visited buffers

`search_cpp_ref`
Opens google.com in a browser for the word at the cursor, specifying a search
<search word> site:cplusplus.com

`search_devdocs_cpp`
Opens devdocs.io/cpp in a browser with <search word> on the clipboard ready for pasting into the search box on devdocs.

`xblock_resize_editor`

Allows you to resize a block selection, it works only with block selections. It has an option for finding the longest line in the selection.

`xnotepad`
places selected text in a floating 'notepad' window. If a notepad window already exists, current selection is appended. If no selection, the current line is copied.

`xnotepad_word`
places word at the cursor in a floating 'notepad' window. If a notepad window already exists, current word is appended.

`xnotepad_create_time_date_string`
Creates a string yyyy-mm-dd-hh-mm-ss in the notepad.

`xkey_binding_trainer, xkey_bindings_show`
The trainer prompts on the command line for a key to be pressed. It then shows all the key bindings associated with that key. The "show" command shows all (most) keybindings in an editor control grouped by key family e.g. F5, C+F5, A-F5 etc. are the F5 family.

`xfloat1` `xfloat2` `xfloat3` (accelerator keys 1,2,3)
The current edit window is floated at a pre-set location, window size and layout.

`xset_float1` `xset_float2` `xset_float3`
Sets the location, window size and layout for the three xfloat commands

`xsave_named_toolwindow_layout` `xload_named_toolwindow_layout`
Saves/restores the current toolwindow layout using a name you specify.

`save_named_state` `load_named_state (slick V23 or later)`

Saves/restores toolwindow layout, edit windows and open buffers using a name you specify. A workspace also provides this capability (plus more), however, you might need this capability without switching workspaces. SlickEdit V23 or later also has save_named_files, load_named_files which saves/ restores a set of buffers without affecting tool-window layout etc.

`xsave_bookmarks` `xrestore_bookmarks`
Saves/restores bookmarks using a name you provide.

`xopen_logs`
Opens the three log files from <config folder>/log - pip.log, stack.log, vs.log

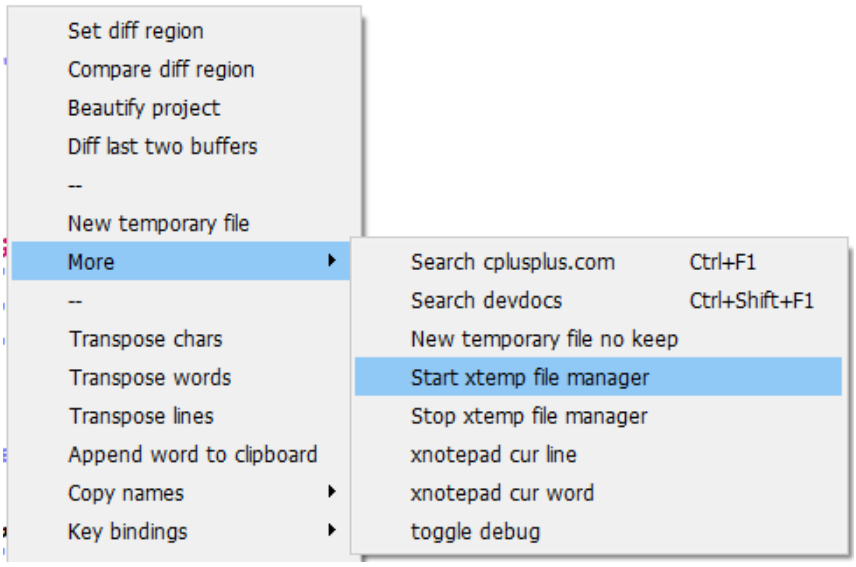
`xtemp_new_temporary_file`

Creates a new "temporary" text file with a name that includes an incrementing number - e.g DA-000045.txt. The file is created in a specific folder

```
// define an environment variable xtemp_files_path OR change the #define in xtemp-file-manager.e
below
#define TEMP_FILES_PATH  _ConfigPath() :+ 'xtemporary_files' :+  FILESEP
```

You can specify a different value for an environment variable per project on the open tab of the project properties dialog. e.g.
set xtemp_files_path=C:/project/temp/

There is a "temporary file manager" that you can start and stop - by default it is not running.



When the temporary file manager is running, it tries to keep any temporary files that you have open, as staying open when you switch workspaces. If you don't want this behaviour, then just don't start the temporary file manager. When you first create a temporary file, you are asked if you want to start the temporary file manager. The temporary file manager will try to keep open (when you switch workspaces) any files that are located in the specified "temporary" folder - not just .txt files. Refer to the start of xtemp-file-manager.e for even more detail.

xxutils.e provides the following cursor movement commands

`xcursor_to_next_token_stop_on_all`
moves edit cursor right, stops at start of a word, skips whitespace, stops on all other symbols

`xcursor_to_prev_token_stop_on_all`
moves edit cursor left, stops at start of a word, skips whitespace, stops on all other symbols

`xcursor_to_next_token`
moves edit cursor right, stops at start *and end* of a word

`xcursor_to_prev_token`
moves edit cursor left, stops at start *and end* of a word

`xselect_to_next_token`
move edit cursor right, selects to end of word or to start of next word

`xselect_to_prev_token`
move edit cursor left, selects to start of word or to end of previous word

xxutils.e provides the following cursor movement search commands

`xfind_next_whole_word_at_cursor`
moves the cursor forward to the next occurrence of the current word under the cursor and selects it. This wraps to the start of the file automatically.

`xfind_prev_whole_word_at_cursor`
moves the cursor back to the previous occurrence of the current word under the cursor and selects it. This wraps to the end of the file automatically.

`xquick_search`
if a selection is active, search forward for the next occurrence of the selected text - otherwise the same as xfind_next_whole_word_at_cursor.

`xquick_reverse_search`
if a selection is active, search back for the previous occurrence of the selected text - otherwise the

same as xfind_prev_whole_word_at_cursor.

Utility code for debugging - see xxutils.e

```
static boolean    xxutils_debug = false;
static void xxdebug(...)
{
   if ( !xxutils_debug )
      return;
   _str s1 = "xr: ";
   int k = 0;
   while ( ++k <= arg()) {
      s1 = s1 :+ arg(k) :+ ' ';
   }
   // https://www.epochconverter.com/
   say(_time('G') :+ s1);
}
_command toggle_xxutils_debug()
{
   if ( xxutils_debug ) {
      xxdebug("xxutils debug off");
      xxutils_debug = false;
   }
   else
   {
      xxutils_debug = true;
      xxdebug("xxutils debug on");
      say("Use F1 for help, Ctrl K to clear");
   }
}

_command test_xxutils_debug()
{
   xxdebug("a string", something, 123);
}
```