Subject: U++ 2023.2 "pre-alpha"
Posted by mirek on Tue, 29 Aug 2023 12:29:00 GMT
View Forum Message <> Reply to Message

Hi,

having reached end of my personal todo whishlist for theide, I thing it a good opportunity to make another release. Please make a note here if you feel like there is something important to FIX for 2023.1.

What is new so far:

ide:

- New C++ sources indexer mode - previoualy, after each file change, all depended files were reindexed. That often resulted in very heavy CPU load after each header change. In the new mode, only the changed file is being reindexed - while less accurate, real problems in this new mode are actually rare and it makes for much more responsive experience using theide.

- Many Assist cornercases fixed

- Result lists of Find in Files and Usage now have "Replace" button which effectively brings the "rename the entity" functionality (e.g. to rename method, find its Usage and then use Replace)

- Bottom pane was redesigned allowing for more Find in Files and Usage result lists.

- Many improvements in source comparsions
"Show git history of file" now allows for choosing branches
"Compare with" now suggests files with the same name from other upp nests
"Compare directories" now allows comparison with git commit

- Various windows like comparisons are now non-modal. It is now e.g. possible to open multiple e.g. Directory comparison windows

- Merge nests function got ability to specify target folder. Default mode is now to "import" packages missing in the first nest to "imported" subfolder of the first nest. The idea is to create an assembly of first nest and its "imported" subfolder to have imported packages fixed, while still allowing the use of the very same packages from original sources with another assembly.

- Visual improvements, UHD versions of many icons

- C++ Language standard for assist (e.g. C++ 17) can now be selected in Assist setup

- Flow breaking commands (return, break, continue, goto, co_return, throw) now have specific syntax highlighting

- TheIDE now displays out-of-page function/class headers in the first line of editor. This is helpful for quick orientation when jumping into long function body.

- External changes for files now getting proper undo records (previously undo history was lost when file was edited outside of ide session)

- Repo sync git pull mode changed to "merge" (from "rebase")

- U++ Builder now defines bmGIT_HASH (with git hash of first nests HEAD, if it is git repo).

umk:

- -h option to automatically download missing packages from UppHub if possible

RichEdit, RichText:

- "Paste without formatting"
- Clipboard 'HTML format' support
- GetPlainText allow_tabs parameter; CtrlLib Prompt now sometimes uses '\t' to separate table cells.

Draw, CtrlLib:

- StdDisplay (if CtrlLib is present) now supports \1 qtf escape code

Core:

- CParser::ReadDoubleNoE (to ignore E part of double)
- CParser::ReadIdh (with '-' in id), RichText: EncodeHTML strikeout support

---

Subject: Re: U++ 2023.1 "pre-alpha"
Posted by omari on Thu, 31 Aug 2023 16:09:17 GMT
View Forum Message <> Reply to Message

You mean 2023.2

please consider investigating this issue: PostgreSql does not handle BOOL correctly

and thise feature:[FEATURE REQUEST] Add Jsonize() to S_ structs

Many thanks!

Subject: Re: U++ 2023.1 "pre-alpha"
Posted by jjacksonRIAB on Thu, 31 Aug 2023 18:22:48 GMT
View Forum Message <> Reply to Message

Very well done Mirek.

My personal wishlist right now is only to fix the bug that causes TheIDE to occasionally erase the entire undo history and go back to the top of an edited document when building a program. It sometimes happens 10-15 times during a session now. It's making me want to use a separate editor since I can't confidently test new changes to a program without worrying that they'll become permanent because a bug kills the undo buffer.

Subject: Re: U++ 2023.1 "pre-alpha"
Posted by Oblivion on Thu, 31 Aug 2023 21:17:00 GMT
View Forum Message <> Reply to Message

Quote:My personal wishlist right now is only to fix the bug that causes TheIDE to occasionally erase the entire undo history and go back to the top of an edited document when building a program. It sometimes happens 10-15 times during a session now. It's making me want to use a separate editor since I can't confidently test new changes to a program without worrying that they'll become permanent because a bug kills the undo buffer.

I agree. This is probably the most annoying, persistent and hard to catch bug in TheIDE.
It happens randomly (from the user POV, at least) when build starts and sometimes leave you with a half undo-ed, broken source code.

Best regards,
Oblivion

Subject: Re: U++ 2023.1 "pre-alpha"
Posted by Klugier on Sat, 02 Sep 2023 19:20:45 GMT
View Forum Message <> Reply to Message

Hello jjacksonRIAB,

jjacksonRIAB wrote on Thu, 31 August 2023 20:22Very well done Mirek.

My personal wishlist right now is only to fix the bug that causes TheIDE to occasionally erase the entire undo history and go back to the top of an edited document when building a program. It sometimes happens 10-15 times during a session now. It's making me want to use a separate editor since I can't confidently test new changes to a program without worrying that they'll become permanent because a bug kills the undo buffer.

The occasional undo history overwrites was fixed by this commit. It fixes the issue when you open licenses and statistics dialog. This will be the part of the next release. If you can provide reproduction scenario we will be grateful. You can also look for "ClearClipboard()" and "AppendClipboard()" usage within U++ code base and especially TheIDE (ide packages) and tell us if you are using one of options that are calling this function.

Mirek good work on the next release. It should be 2023.2 not 2023.1.

Klugier

Subject: Re: U++ 2023.1 "pre-alpha"
Posted by jjacksonRIAB on Sat, 02 Sep 2023 22:54:47 GMT
View Forum Message <> Reply to Message

Klugier wrote on Sat, 02 September 2023 21:20Hello jjacksonRIAB,

jjacksonRIAB wrote on Thu, 31 August 2023 20:22Very well done Mirek.

My personal wishlist right now is only to fix the bug that causes TheIDE to occasionally erase the entire undo history and go back to the top of an edited document when building a program. It sometimes happens 10-15 times during a session now. It's making me want to use a separate editor since I can't confidently test new changes to a program without worrying that they'll become permanent because a bug kills the undo buffer.

The occasional undo history overwrites was fixed by this commit. It fixes the issue when you open licenses and statistics dialog. This will be the part of the next release. If you can provide reproduction scenario we will be grateful. You can also look for "ClearClipboard()" and "AppendClipboard()" usage within U++ code base and especially TheIDE (ide packages) and tell us if you are using one of options that are calling this function.

Mirek good work on the next release. It should be 2023.2 not 2023.1.

Klugier

Klugier,
I think we're talking about two different issues. I noticed that the stats dialog was fixed so it doesn't overwrite the copy/paste buffer without pressing the button now, thanks for doing that.

Subject: Re: U++ 2023.1 "pre-alpha"
Posted by mirek on Tue, 05 Sep 2023 11:19:53 GMT
View Forum Message <> Reply to Message

jjacksonRIAB wrote on Thu, 31 August 2023 20:22Very well done Mirek.

My personal wishlist right now is only to fix the bug that causes TheIDE to occasionally erase the entire undo history and go back to the top of an edited document when building a program. It

sometimes happens 10-15 times during a session now. It's making me want to use a separate editor since I can't confidently test new changes to a program without worrying that they'll become permanent because a bug kills the undo buffer.

Well, I would be happy to help, but it does not happen to me AT ALL.

Which means that you must be doing something that I am not. Any clues?

Mirek

Subject: Re: U++ 2023.1 "pre-alpha"
Posted by jjacksonRIAB on Tue, 05 Sep 2023 20:07:52 GMT
View Forum Message <> Reply to Message

Mirek,

It happens to me in Linux Mint, Arch Linux and FreeBSD with two different computers, three different filesystems (ext4, ext3, zfs). I don't use Windows much anymore so I cannot confirm its existence there. I don't see what I could be doing different, I just press build and after the build is done boom - edit window refreshes, latest changes disappear, undo/redo buffer is erased, cursor is moved to the top of the document. This happens with both clang and gcc. I cannot reproduce it on fresh projects, it appears to manifest on larger projects particularly after theIDE has been open for a while and when quick changes to a document have been made in succession.

It's as if theIDE is losing sync with the files/timestamps on disk, letting you continue working on an unsynced ghost copy of the file and then reloading the last saved version during build but I cannot be entirely sure what's going on, being unable to reliably reproduce it. I don't know if any of this information helps.

I kind of wonder whether the autosave routine is failing to flush to disk or losing track of the handles and timestamps it is observing.

Subject: Re: U++ 2023.1 "pre-alpha"
Posted by mirek on Wed, 06 Sep 2023 12:16:28 GMT
View Forum Message <> Reply to Message

jjacksonRIAB wrote on Tue, 05 September 2023 22:07Mirek,

It happens to me in Linux Mint, Arch Linux and FreeBSD with two different computers, three different filesystems (ext4, ext3, zfs). I don't use Windows much anymore so I cannot confirm its existence there. I don't see what I could be doing different, I just press build and after the build is done boom - edit window refreshes, latest changes disappear, undo/redo buffer is erased, cursor is moved to the top of the document. This happens with both clang and gcc. I cannot reproduce it on fresh projects, it appears to manifest on larger projects particularly after theIDE has been open for a while and when quick changes to a document have been made in succession.

It's as if theIDE is losing sync with the files/timestamps on disk, letting you continue working on an unsynced ghost copy of the file and then reloading the last saved version during build but I cannot be entirely sure what's going on, being unable to reliably reproduce it. I don't know if any of this information helps.

I kind of wonder whether the autosave routine is failing to flush to disk or losing track of the handles and timestamps it is observing.

Is this a new problem with current master or was it doing that in say 2022 versions too? (There is a new logic intended to add unco records when files changes outside of theide which could be related, but it is just in master, since ~ 6/2023).

Do you edit files after starting the build?

To what point files revert?

Mirek

Subject: Re: U++ 2023.1 "pre-alpha"
Posted by jjacksonRIAB on Wed, 06 Sep 2023 13:11:06 GMT
View Forum Message <> Reply to Message

mirek wrote on Wed, 06 September 2023 14:16
Is this a new problem with current master or was it doing that in say 2022 versions too? (There is a new logic intended to add unco records when files changes outside of theide which could be related, but it is just in master, since ~ 6/2023).

Well that's why I'm a bit confused. I brought it up in 2022
https://www.ultimatepp.org/forums/index.php?t=msg&goto=5 8303&#msg_58303
and you said you have the same problem

It's never been solved and it's gotten more frequent since theIDE started using a more resources to accommodate the new autocomplete.

Quote:
Do you edit files after starting the build?

Nope, it typically happens right after the build is finished. I can just leave it there doing nothing and right after build has completed I experience an immediate loss of changes - and the changes weren't built into the program, if one was built at all. It usually fails to build because I'm making a fixing change which gets lost.

Quote:
To what point files revert?

That one is harder to quantify because I don't know when the last autosave occurred or how deep the undo buffer has gotten. I noticed theide is set up to reload on losing focus so I'd have to count things like that to know when it stopped tracking.

So all I'm left with is mainly frustrating symptoms and guesses.

Subject: Re: U++ 2023.1 "pre-alpha"
Posted by Oblivion on Thu, 07 Sep 2023 04:54:20 GMT
View Forum Message <> Reply to Message

Hi Mirek,

This is not a new issue. To my knowledge, the problem is there since at least ~2020.
The only two reliable patterns I get is that (1) it happens when I start the build, but this buggy behavior seems completely random, so I can't predict exactly when (on which build, I mean) it'll happen. (2) This happens when theIDE is running for a long time.

The editor sometimes removes the last changes to the source code completely, or in some cases removes the changes partially and renders the source code in a broken and truncated state.

This happens on Linux.

Best regards,

Oblivion

Subject: Re: U++ 2023.1 "pre-alpha"
Posted by mirek on Thu, 07 Sep 2023 06:36:50 GMT
View Forum Message <> Reply to Message

I think I have found possible culprit, I mean error that is quite consistent with what we see:

https://github.com/ultimatepp/ultimatepp/commit/b16e199ae61a3290d5e4cb53206d502eb98c4700

Please recompile theide and check...

Mirek

Subject: Re: U++ 2023.2 "pre-alpha"
Posted by luoganda on Thu, 07 Sep 2023 20:42:18 GMT

...this is somehow an extension to this post.

Quote:- New C++ sources indexer mode - previoualy, after each file change, all depended files were reindexed. That often resulted in very heavy CPU load after each header change. In the new mode, only the changed file is being reindexed - while less accurate, real problems in this new mode are actually rare and it makes for much more responsive experience using theide.
I think i see your point here, altough the thing is far from perfect(checked both included clang on w7 and msbt17 and no go).
But i think you forgot to see both point of view:
* from developing upp point of view it's probably better(since new mode recompiles "needed" - mostly developing "system" header sources faster),
* from consumer point of view(just using the ide out of the box for a quick project) - original is magic - it works!
Speed even on slow computers is not bad at all.
It works out of the box also what concerns Navigator window - it affects it. In an attached img it's obvious what i mean,
this is why i have proposed Navwnd in the first place(you don't get that from Navigator bar) - in the pic is what i mean with "panoramic view"(example is simple Ctrl).
With "panoramic view" here - i mean having all parsed structs/etc handy at hand all the time(eg to quickly check it's sources,usage,etc).
Of course - new indexer could use on demand "TriggerIndexer" to show that Ctrl thing - but i think this is out of the scope of this msg.

With this i incline that original and new modes of indexer should be selectable in Settings,
original mode is a must for people who want: "do it's thing at opening project and then just use it", for "consumer" usage - this will always be better(from just using theide - also for just a quick project).
Even on old computer like mine, original indexer is pretty fast(from "consumer" point of view) mostly doing it's thing when a project is opened(with this i mostly mean constant system headers), more problems what concerns speed are quite slow compilers.
(in my oppinion c++11 is an absolute must - all new features c++23/etc should be optimized to not become an bloatWare).

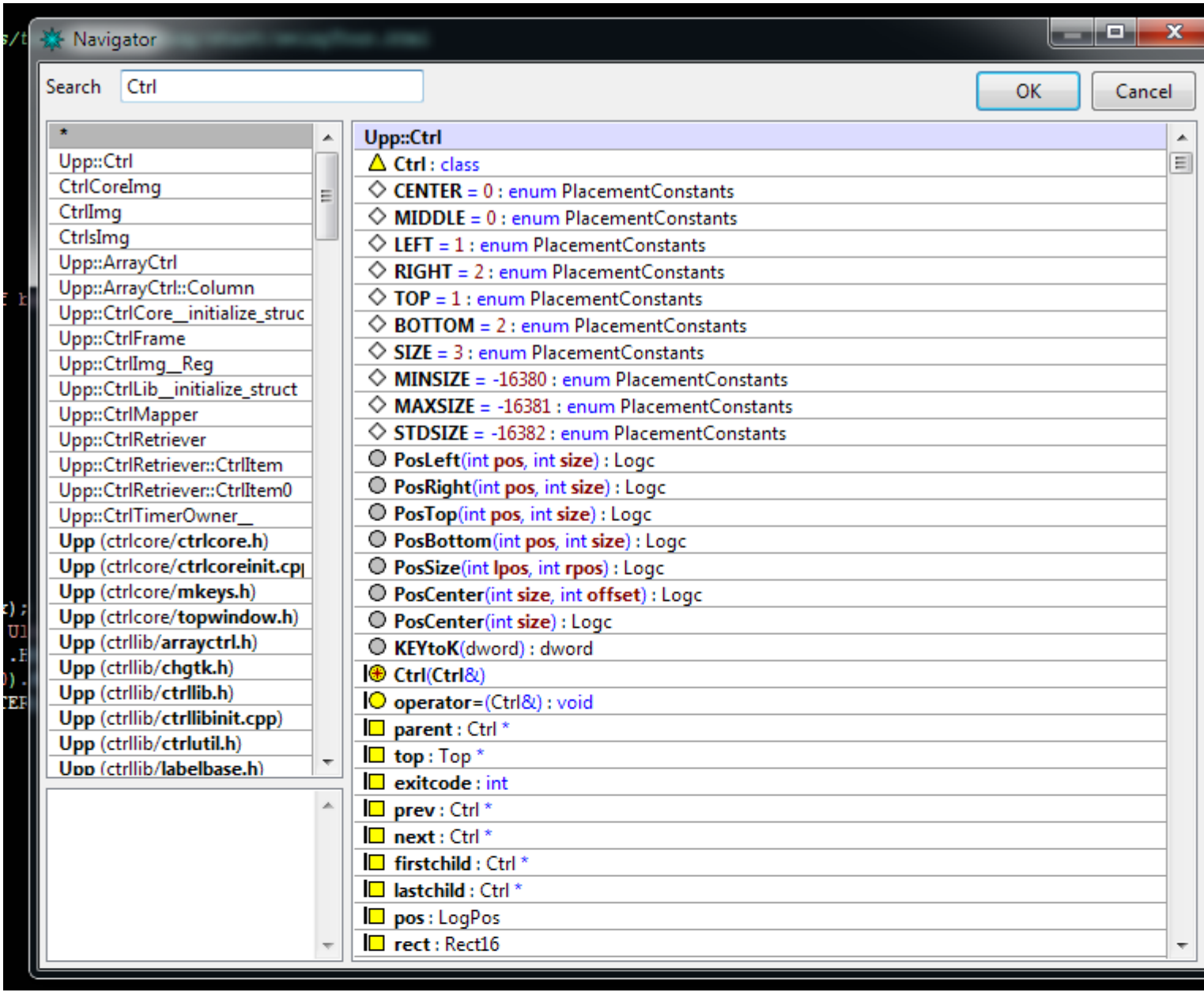
Note: this was written after testing v16969.

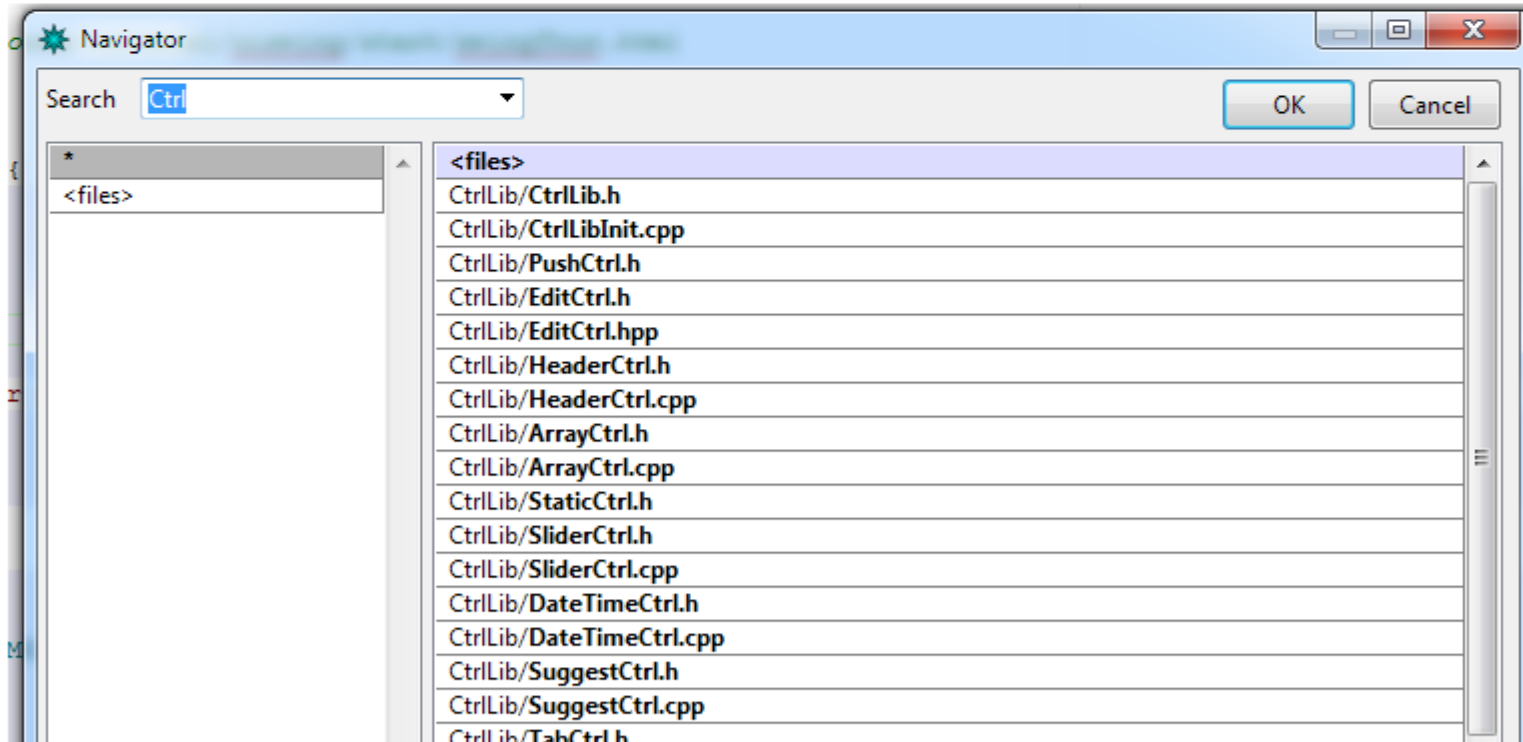
By the way, personally - i don't know why borland(now embarcadero) has
used new llvm infrastructure - because old bcc compiler with it's turbolinker was quite fast.

By the way2:
where is an example(or doc) using customCompiler
in BuilderSettings since i once used it and it worked out of the box?

File Attachments

1) indexerAndNavWnd.png, downloaded 779 times


Navigator
Search
Ctrl
OK
Cancel
*
Upp::Ctrl
CtrlCoreImg
CtrlImg
CtrlsImg
Upp::ArrayCtrl
Upp::ArrayCtrl::Column
Upp::CtrlCore__initialize_struc
Upp::CtrlFrame
Upp::CtrlImg__Reg
Upp::CtrlLib__initialize_struct
Upp::CtrlMapper
Upp::CtrlRetriever
Upp::CtrlRetriever::CtrlItem
Upp::CtrlRetriever::CtrlItem0
Upp::CtrlTimerOwner__
Upp (ctrlcore/ctrlcore.h)
Upp (ctrlcore/ctrlcoreinit.cp
Upp (ctrlcore/mkeys.h)
Upp (ctrlcore/topwindow.h)
Upp (ctrllib/arrayctrl.h)
Upp (ctrllib/chgtk.h)
Upp (ctrllib/ctrllib.h)
Upp (ctrllib/ctrllibinit.cpp)
Upp (ctrllib/ctrlutil.h)
Upp (ctrlib/labelbase.h)
Upp::Ctrl
Ctrl : class
CENTER = 0 : enum PlacementConstants
MIDDLE = 0 : enum PlacementConstants
LEFT = 1 : enum PlacementConstants
RIGHT = 2 : enum PlacementConstants
TOP = 1 : enum PlacementConstants
BOTTOM = 2 : enum PlacementConstants
SIZE = 3 : enum PlacementConstants
MINSIZE = -16380 : enum PlacementConstants
MAXSIZE = -16381 : enum PlacementConstants
STDSIZE = -16382 : enum PlacementConstants
PosLeft(int pos, int size) : Logc
PosRight(int pos, int size) : Logc
PosTop(int pos, int size) : Logc
PosBottom(int pos, int size) : Logc
PosSize(int lpos, int rpos) : Logc
PosCenter(int size, int offset) : Logc
PosCenter(int size) : Logc
KEYtoK(dword) : dword
Ctrl(Ctrl&)
operator=(Ctrl&) : void
parent : Ctrl *
top : Top *
exitcode : int
prev : Ctrl *
next : Ctrl *
firstchild : Ctrl *
lastchild : Ctrl *
pos : LogPos
rect : Rect16


Navigator
Search
Ctrl
OK
Cancel
*
<files>
<files>
CtrlLib/CtrlLib.h
CtrlLib/CtrlLibInit.cpp
CtrlLib/PushCtrl.h
CtrlLib/EditCtrl.h
CtrlLib/EditCtrl.hpp
CtrlLib/HeaderCtrl.h
CtrlLib/HeaderCtrl.cpp
CtrlLib/ArrayCtrl.h
CtrlLib/ArrayCtrl.cpp
CtrlLib/StaticCtrl.h
CtrlLib/SliderCtrl.h
CtrlLib/SliderCtrl.cpp
CtrlLib/DateTimeCtrl.h
CtrlLib/DateTimeCtrl.cpp
CtrlLib/SuggestCtrl.h
CtrlLib/SuggestCtrl.cpp

Subject: Re: U++ 2023.2 "pre-alpha"
Posted by mirek on Fri, 08 Sep 2023 06:58:10 GMT
View Forum Message <> Reply to Message

luoganda wrote on Thu, 07 September 2023 22:42...this is somehow an extension to this post.

Quote:- New C++ sources indexer mode - previoualy, after each file change, all depended files were reindexed. That often resulted in very heavy CPU load after each header change. In the new mode, only the changed file is being reindexed - while less accurate, real problems in this new mode are actually rare and it makes for much more responsive experience using theide.
I think i see your point here, altough the thing is far from perfect(checked both included clang on w7 and msbt17 and no go).
But i think you forgot to see both point of view:
* from developing upp point of view it's probably better(since new mode recompiles "needed" - mostly developing "system" header sources faster),
* from consumer point of view(just using the ide out of the box for a quick project) - original is magic - it works!

Now I am not sure whether whether you are speaking about pre-libclang homegrown parser or about difference between 2023.1 and 2023.2. If about later, I can tell you that from consumer point of view, "precise behaviour" was problematic - any change to deep header in large project choked CPU for minute. Pretty bad if you edit header and source simultanously.

Quote:
With this i incline that original and new modes of indexer should be selectable in Settings,

If you are speaking about new indexer mode introduced since 2023.1, the yes, it is selectable.

Quote:
original mode is a must for people who want: "do it's thing at opening project and then just use it",

But it does not seem so... Maintaining both old heuristic home-brew parser and libclang is far beyond our resources, sorry. libclang provides some new features that I find quite essential (as in "how could I live without Usage for so long"), so there is no way back.

I am more interested with users that have 12+ core CPUs and 32+GB of RAM (e.g. "me")...

Subject: Re: U++ 2023.1 "pre-alpha"
Posted by jjacksonRIAB on Fri, 08 Sep 2023 07:01:03 GMT
View Forum Message <> Reply to Message

mirek wrote on Thu, 07 September 2023 08:36I think I have found possible culprit, I mean error that is quite consistent with what we see:

https://github.com/ultimatepp/ultimatepp/commit/b16e199ae61a3290d5e4cb53206d502eb98c4700

Please recompile theide and check...

Mirek

I've been trying it for a day and no problems to report yet. Will let you know if that changes. Thanks, Mirek!

Subject: Re: U++ 2023.2 "pre-alpha"
Posted by luoganda on Sat, 09 Sep 2023 08:49:41 GMT
View Forum Message <> Reply to Message

Quote:Now I am not sure whether whether you are speaking about pre-libclang homegrown parser or about difference between 2023.1 and 2023.2.
I meant pre-clang(homegrown) parser - since it's probably the only one who had "panoramic view" of all classes in Navwnd at hand all the time.

I tested latest upp-win-16974 on w7 - both clang and msbt17 and no go.

12+ cores?
hah, if anything does not work well with 4 - you can throw it away right away...

Ok, if there will be changes about this i'll post it in TheIDE forum.

Subject: Re: U++ 2023.2 "pre-alpha"
Posted by Tom1 on Mon, 11 Sep 2023 13:33:22 GMT
View Forum Message <> Reply to Message

Hi Mirek!

Looks good.

Can you change the 'Main configuration' drop list to not grab keyboard focus. After changing the Main configuration, trying to start compiling with F7 will not work until I click the code editor area. This issue was introduced in 2023.1, I think, but I discovered it only after release...

Best regards,

Tom

Subject: Re: U++ 2023.2 "pre-alpha"
Posted by Tom1 on Mon, 11 Sep 2023 13:38:39 GMT
View Forum Message <> Reply to Message

And another thing...

When using "Show git history for file..." for "plugin/tif/tifupp.cpp" and comparing the previous and current version, very large region of code seems to be changed at the first glance. However, in reality, the comparison function does not match the lines very well and there are just few changes to display. Please take a look.

Best regards,

Tom

Subject: Re: U++ 2023.2 "pre-alpha"
Posted by coolman on Mon, 11 Sep 2023 18:00:27 GMT
View Forum Message <> Reply to Message

Hi,

When I execute TheIde from the cmd, I can see /usr/bin/valgrind output when I exit TheIde.

OS: Ubuntu 20.04

BR, Radek

Subject: Re: U++ 2023.1 "pre-alpha"
Posted by jjacksonRIAB on Wed, 13 Sep 2023 07:13:59 GMT
View Forum Message <> Reply to Message

Oblivion,

It looks like Mirek really squashed it this time with his latest bugfix. I've been running it for several days now without a single issue but it would be nice to have a second confirmation when you have time.

Subject: Re: U++ 2023.2 "pre-alpha"
Posted by mirek on Wed, 13 Sep 2023 09:04:39 GMT
View Forum Message <> Reply to Message

luoganda wrote on Sat, 09 September 2023 10:49Quote:Now I am not sure whether whether you

are speaking about pre-libclang homegrown parser or about difference between 2023.1 and 2023.2.
I meant pre-clang(homegrown) parser - since it's probably the only one who had "panoramic view" of all classes in Navwnd at hand all the time.

Navigator window is back now - I hoped you will check it out.

Quote:
12+ cores?
hah, if anything does not work well with 4 - you can throw it away right away...

Make a distinction between development machine and average target user machine. U++ apps work just fine on single core 1Ghz ARM CPUs. But time is money and powerful machines are for pennies today. Moreover, C++ programmers are rare and expensive. Usually one month salary of entry level C++ programmer is worth 5+ multicore workstations...

That said, my "vacation" machine is now quite old 2GHz double core notebook with Linux installed (I rarely need it anyway). It was still OK-ish experience.

Anyway, Win7 probably does not work anymore - blame clang.

Mirek

Subject: Re: U++ 2023.2 "pre-alpha"
Posted by mirek on Wed, 13 Sep 2023 09:17:08 GMT
View Forum Message <> Reply to Message

Tom1 wrote on Mon, 11 September 2023 15:33Hi Mirek!

Looks good.

Can you change the 'Main configuration' drop list to not grab keyboard focus. After changing the Main configuration, trying to start compiling with F7 will not work until I click the code editor area. This issue was introduced in 2023.1, I think, but I discovered it only after release...

Best regards,

Tom

Hopefully fixed, thanks.

Subject: Re: U++ 2023.2 "pre-alpha"
Posted by mirek on Wed, 13 Sep 2023 09:34:03 GMT

View Forum Message <> Reply to Message

Tom1 wrote on Mon, 11 September 2023 15:38And another thing...

When using "Show git history for file..." for "plugin/tif/tifupp.cpp" and comparing the previous and current version, very large region of code seems to be changed at the first glance. However, in reality, the comparison function does not match the lines very well and there are just few changes to display. Please take a look.

Best regards,

Tom

There is a change in indentation. Should we ignore it? (That is not rethrical question, I really wonder whether we should... it is doable)

Mirek

Subject: Re: U++ 2023.2 "pre-alpha"
Posted by Tom1 on Wed, 13 Sep 2023 10:42:09 GMT
View Forum Message <> Reply to Message

mirek wrote on Wed, 13 September 2023 12:17Tom1 wrote on Mon, 11 September 2023 15:33Hi Mirek!

Looks good.

Can you change the 'Main configuration' drop list to not grab keyboard focus. After changing the Main configuration, trying to start compiling with F7 will not work until I click the code editor area. This issue was introduced in 2023.1, I think, but I discovered it only after release...

Best regards,

Tom

Hopefully fixed, thanks.
Thanks! // BR, Tom

Subject: Re: U++ 2023.2 "pre-alpha"
Posted by Tom1 on Wed, 13 Sep 2023 10:49:31 GMT
View Forum Message <> Reply to Message

mirek wrote on Wed, 13 September 2023 12:34Tom1 wrote on Mon, 11 September 2023 15:38And another thing...

When using "Show git history for file..." for "plugin/tif/tifupp.cpp" and comparing the previous and current version, very large region of code seems to be changed at the first glance. However, in reality, the comparison function does not match the lines very well and there are just few changes to display. Please take a look.

Best regards,

Tom

There is a change in indentation. Should we ignore it? (That is not rethrical question, I really wonder whether we should... it is doable)

Mirek
Hi,

Yes, I see there are several indentation changes, but the real issue begins on line 904 "break;", where it completely loses sync. Then it re-synchronizes to "break;" on line 1001, within another switch. Do you get this same behavior?

I do not mind seeing the indentation changes, especially as they are so neatly presented inline. It is actually good to see why a file was updated, even if it just indentation. But the large portions of code (such as 904...) getting out of sync is a problem.

Best regards,

Tom

Subject: Re: U++ 2023.2 "pre-alpha"
Posted by mirek on Wed, 13 Sep 2023 11:40:59 GMT
View Forum Message <> Reply to Message

Tom1 wrote on Wed, 13 September 2023 12:49mirek wrote on Wed, 13 September 2023 12:34Tom1 wrote on Mon, 11 September 2023 15:38And another thing...

When using "Show git history for file..." for "plugin/tif/tifupp.cpp" and comparing the previous and current version, very large region of code seems to be changed at the first glance. However, in reality, the comparison function does not match the lines very well and there are just few changes to display. Please take a look.

Best regards,

Tom

There is a change in indentation. Should we ignore it? (That is not rethrical question, I really wonder whether we should... it is doable)

Mirek
Hi,

Yes, I see there are several indentation changes, but the real issue begins on line 904 "break;", where it completely loses sync. Then it re-synchronizes to "break;" on line 1001, within another switch. Do you get this same behavior?

Tom

Yes, and I believe it works correctly. That break has different indentation, so it "synchronizes" (not that it is how it really works) on the first break that has the same indent. Which is on line 1001. (Please check whether I interpret it correctly...)

Subject: Re: U++ 2023.2 "pre-alpha"
Posted by Tom1 on Wed, 13 Sep 2023 12:43:59 GMT
View Forum Message <> Reply to Message

mirek wrote on Wed, 13 September 2023 14:40Tom1 wrote on Wed, 13 September 2023 12:49mirek wrote on Wed, 13 September 2023 12:34Tom1 wrote on Mon, 11 September 2023 15:38And another thing...

When using "Show git history for file..." for "plugin/tif/tifupp.cpp" and comparing the previous and current version, very large region of code seems to be changed at the first glance. However, in reality, the comparison function does not match the lines very well and there are just few changes to display. Please take a look.

Best regards,

Tom

There is a change in indentation. Should we ignore it? (That is not rethrical question, I really wonder whether we should... it is doable)

Mirek
Hi,

Yes, I see there are several indentation changes, but the real issue begins on line 904 "break;", where it completely loses sync. Then it re-synchronizes to "break;" on line 1001, within another switch. Do you get this same behavior?

Tom

Yes, and I believe it works correctly. That break has different indentation, so it "synchronizes" (not that it is how it really works) on the first break that has the same indent. Which is on line 1001. (Please check whether I interpret it correctly...)
Well, I guess it's just so complicated problem that there is no easy nor perfect solution. In this specific case the indentation is the cause for this readability issue, but it could be any change really that can cause similar 'loss of sync'. Probably not worth spending tens of hours, to make it find the minimal set of changed characters in optimal sequences and visualize accordingly.

I just wanted to point out that the current comparison on this file makes up an illusion that the file is all different now. It is really hard to see what the real changes are.

Best regards,

Tom

Subject: Re: U++ 2023.2 "pre-alpha"
Posted by jjacksonRIAB on Wed, 13 Sep 2023 17:13:10 GMT
View Forum Message <> Reply to Message

Mirek,

Since I know you're checking this thread, you might want to look into this bug report/fix for pre-alpha. It gave me back my CPU on linux.

https://www.ultimatepp.org/forums/index.php?t=msg&th=121 58&start=0&

Subject: Re: U++ 2023.2 "pre-alpha"
Posted by zsolt on Thu, 14 Sep 2023 00:11:55 GMT
View Forum Message <> Reply to Message

Could you include this one line XmlParser bugfix?
https:// www.ultimatepp.org/forums/index.php?t=msg&th=12159&s tart=0&

Subject: Re: U++ 2023.1 "pre-alpha"
Posted by Oblivion on Sat, 16 Sep 2023 18:27:32 GMT
View Forum Message <> Reply to Message

Quote:Oblivion,

It looks like Mirek really squashed it this time with his latest bugfix. I've been running it for several days now without a single issue but it would be nice to have a second confirmation when you have time.

Sorry, I'm a bit late.
I've been continuously running the patched TheIDE since last week (compiling huge apps), and I think at this point it is safe to say this annoying and nasty bug is gone (on Linux).

Thanks and well done, Mirek!

Best regards,
Oblivion

Subject: Re: U++ 2023.2 "pre-alpha"
Posted by mirek on Mon, 18 Sep 2023 06:55:46 GMT
View Forum Message <> Reply to Message

coolman wrote on Mon, 11 September 2023 20:00Hi,

When I execute TheIde from the cmd, I can see /usr/bin/valgrind output when I exit TheIde.

OS: Ubuntu 20.04

BR, Radek

Should be fixed.

Subject: Re: U++ 2023.2 "pre-alpha"
Posted by Tom1 on Wed, 20 Sep 2023 18:09:00 GMT
View Forum Message <> Reply to Message

Hi Mirek,

How about "float Null"?

https://www.ultimatepp.org/forums/index.php?t=msg&th=120 82&goto=59874&#msg_59874

Best regards,

Tom

Subject: Re: U++ 2023.2 "pre-alpha"
Posted by Tom1 on Wed, 20 Sep 2023 18:15:52 GMT
View Forum Message <> Reply to Message

Hi Mirek,

CtrlCore/Win32DnD.cpp throws some RLOG()'s even with release. Is this necessary? :
#define LLOG(x) RLOG(x)
Could be:
#define LLOG(x) // RLOG(x)

Best regards,

Tom
