Subject: New Assist features
Posted by mirek on Mon, 10 Oct 2022 12:43:12 GMT
View Forum Message <> Reply to Message

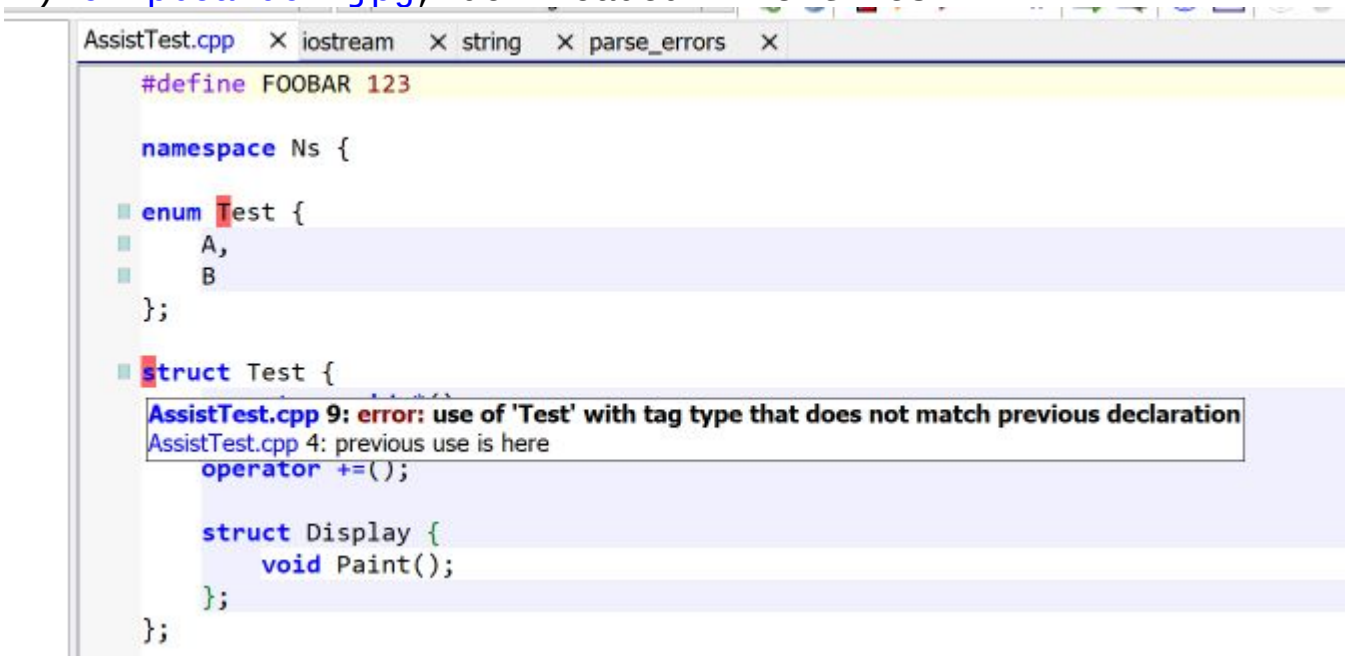
(now showing errors as you type...)

Also I have added Alt+Shift+U - similar to Alt+U (show all references), but shows all references for symbol at cursor instead of current function.

Mirek

File Attachments

1) Clipboard01.jpg, downloaded 2125 times



Subject: Re: New Assist features
Posted by Klugier on Mon, 10 Oct 2022 15:31:01 GMT
View Forum Message <> Reply to Message

Hello Mirek,

Glad that this landed. It is game changer for TheIDE! Thank you for your great work!

Klugier

Subject: Re: New Assist features

Posted by Lance on Tue, 11 Oct 2022 00:29:05 GMT
View Forum Message <> Reply to Message

Glad to see that theide is making new process. Thanks for the dedication!

Subject: Re: New Assist features
Posted by peterh on Tue, 11 Oct 2022 12:25:22 GMT
View Forum Message <> Reply to Message

mirek wrote on Mon, 10 October 2022 14:43
Also I have added Alt+Shift+U - similar to Alt+U (show all references), but shows all references for symbol at cursor instead of current function.
Mirek

Alt+Shift+U also works with overloaded operators.
This is especially useful if I read code, that I have not written and do not understand.

Thank you very much!

Subject: Re: New Assist features
Posted by Tom1 on Mon, 17 Oct 2022 07:51:38 GMT
View Forum Message <> Reply to Message

Hi Mirek,

While 'now showing errors as you type...' feature first appears great, I have noticed that my focus gets distracted by this feature especially when slowly writing and intensely thinking about some more complex programming challenges. So, could you make this feature optional in Assist Settings?

Thanks and best regards,

Tom

Subject: Re: New Assist features
Posted by peterh on Mon, 17 Oct 2022 08:13:39 GMT
View Forum Message <> Reply to Message

Yes, or introduce larger time delay after last keypress or less brightness and saturation or adjustable color.

Subject: Re: New Assist features

Posted by mirek on Mon, 17 Oct 2022 09:15:31 GMT
View Forum Message <> Reply to Message

Tom1 wrote on Mon, 17 October 2022 09:51Hi Mirek,

While 'now showing errors as you type...' feature first appears great, I have noticed that my focus gets distracted by this feature especially when slowly writing and intensely thinking about some more complex programming challenges. So, could you make this feature optional in Assist Settings?

Thanks and best regards,

Tom

Yes, in TODO already.

That said, theide was showing red behind wrong parenthesis for ages now...

Also, yesterday I have changed "error at the end of line" (usually missing semicolon) to show just as single character.

Mirek

Subject: Re: New Assist features
Posted by mirek on Mon, 17 Oct 2022 10:04:32 GMT
View Forum Message <> Reply to Message

So

- you can now switch it off in settings
- you can setup the color
- it might be now be little less invasive as I have removed some 'false' moments to do the check
- some related bugs (like error at the end of some includes) fixed
- there is now status icon "OK" / "Errors" in the left corner; it can be switched off as well

Mirek

Subject: Re: New Assist features
Posted by Tom1 on Mon, 17 Oct 2022 12:12:26 GMT
View Forum Message <> Reply to Message

mirek wrote on Mon, 17 October 2022 13:04So

- you can now switch it off in settings
- you can setup the color
- it might be now be little less invasive as I have removed some 'false' moments to do the check

- some related bugs (like error at the end of some includes) fixed
- there is now status icon "OK" / "Errors" in the left corner; it can be switched off as well

Mirek
Thanks, Mirek!

BR, Tom

Subject: Re: New Assist features
Posted by mirek on Tue, 18 Oct 2022 12:32:23 GMT
View Forum Message <> Reply to Message

It now makes red dots in scrollbar to show where errors are...

Subject: Re: New Assist features
Posted by Klugier on Sat, 22 Oct 2022 14:47:24 GMT
View Forum Message <> Reply to Message

Hello,

I discussed this already with Mirek, but I am not happy how the new OK state is presented within TheIDE. I know that I can turn it off, but it is a default option, so all users will be affected by this change and I do not want to read that our UI feels like 90s. For me it is too invasive, it is colliding with line numbers, breakpoints and documentation annotation. TheIDE should be more minimal and less visually heavy. Screenshot with the new OK state below (top left corner of CodeEditor):

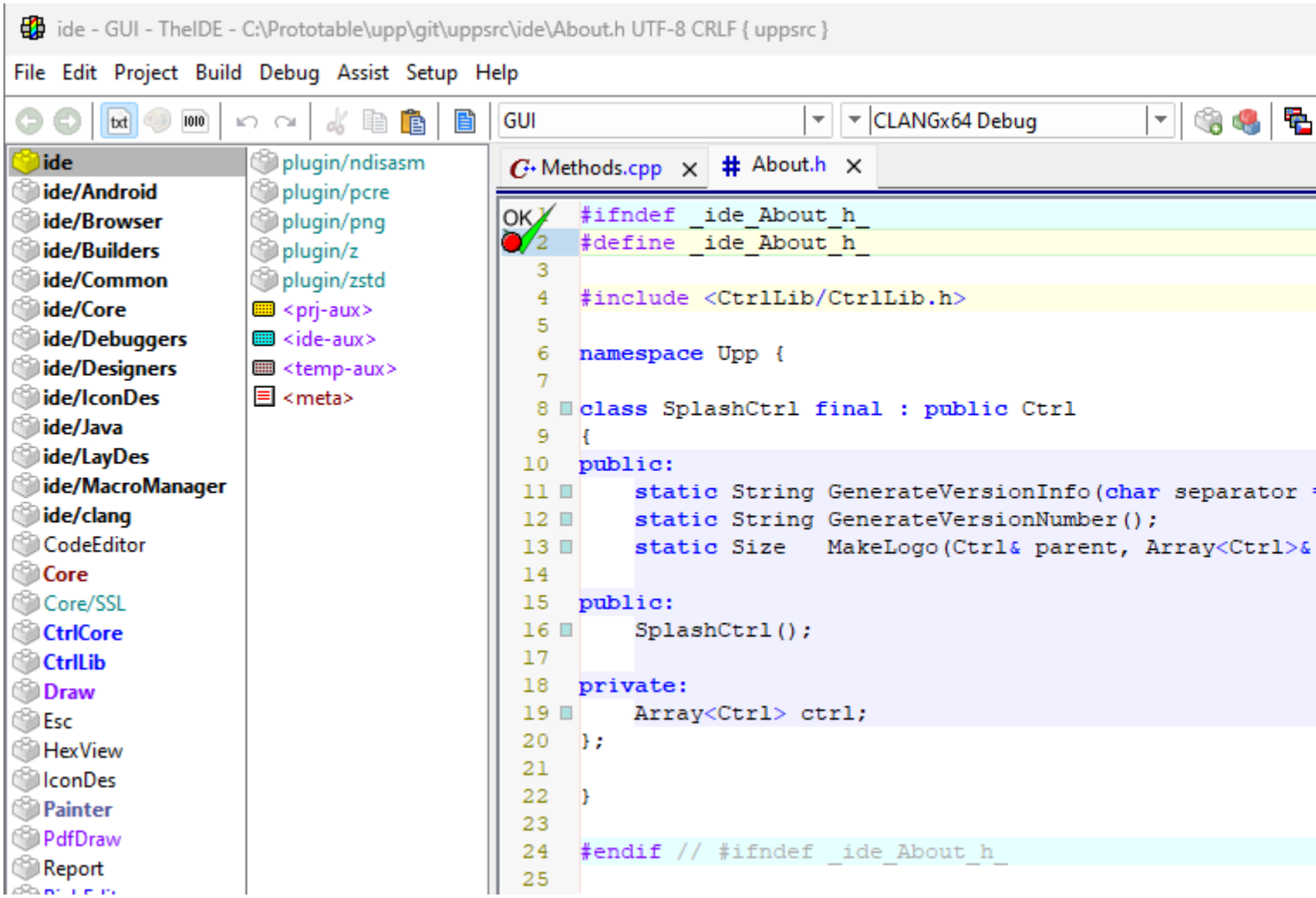
Can we just do not show nothing and display only errors only in scrollbar? I know that Mirek might be affraid that the check doesn't run, but it means that we have a bug that needs to be eliminated. We can show this OK/Errors state in verbose mode for debug purpose. Also, most of IDE's doesn't display this state, so precedence in this case is risky and unnecessary. The final option is to do not show this icon by default and make it optional.

Moreover, the default Navigator position was bring back from right to left. I understand that it is harder to read expanded symbols, but the overall default layout of TheIDE suffers from it. It is much easier to navigate with the mouse from CodeEditor to list of files when Navigator is not between Files/Packages and CodeEditor. Also, right now you can not just do not use Navigator, it is integrated part of expirience, because it shows the progression of parsing. Maybe we can move information about parsing somwhere else for example to status bar, which is hidden by default?

Klugier

File Attachments

1) OKState.png, downloaded 1426 times



Subject: Re: New Assist features
Posted by Tom1 on Thu, 27 Oct 2022 09:59:36 GMT
View Forum Message <> Reply to Message

Hi Mirek,

I have noticed, that Assist autocomplete does not show all methods for e.g. String. If I type "String::" I would expect to see a whole bunch of methods (including e.g. 'StartsWith'), but for some reason I only get about 20 results or so. Is this issue depending on the way String inherits from other classes? Pre-libclang versions of TheIDE returned quite a lot of options for String:: .

Further on, if I may add to the wish list...

```
typedef struct{
 int a;
 int b;
 union{
  int i;
  dword u;
```

```
  float f;
 };
}tt;
```

In addition to a and b, would it be possible to see also i, u and f in the assist autocomplete listing for the type tt above?

Thanks and best regards,

Tom

Subject: Re: New Assist features
Posted by mirek on Thu, 27 Oct 2022 10:29:16 GMT
View Forum Message <> Reply to Message

Tom1 wrote on Thu, 27 October 2022 11:59Hi Mirek,

I have noticed, that Assist autocomplete does not show all methods for e.g. String. If I type "String::" I would expect to see a whole bunch of methods (including e.g. 'StartsWith'), but for some reason I only get about 20 results or so. Is this issue depending on the way String inherits from other classes? Pre-libclang versions of TheIDE returned quite a lot of options for String:: .

I am aware that this problem was in Linux before yesterday's fix. Could that be the case?

(It is always helpful to specify OS...)

Subject: Re: New Assist features
Posted by Tom1 on Thu, 27 Oct 2022 10:46:22 GMT
View Forum Message <> Reply to Message

Sorry, stupid of me: Today, I'm on Windows. Just upgraded TheIDE and the problem still persists.

Best regards,

Tom

UPDATE: Just looked on the Linux size, and there assist shows all the bells and whistles for String:: ! So this seems to be on the Windows side only.

Subject: Re: New Assist features
Posted by mirek on Thu, 27 Oct 2022 11:04:56 GMT

Tom1 wrote on Thu, 27 October 2022 12:46Sorry, stupid of me: Today, I'm on Windows. Just upgraded TheIDE and the problem still persists.

Best regards,

Tom

UPDATE: Just looked on the Linux size, and there assist shows all the bells and whistles for String:: ! So this seems to be on the Windows side only.

Weird, probably somehow related to something, I just did quick test

```
#include <Core/Core.h>

using namespace Upp;

CONSOLE_APP_MAIN
{
 String h;
 h.
}
```

and I can see StartsWith.

That said, I have noticed there are yet problems when build method is MSVC (I have it in my TODO list). Can that be the problem? (if that is the case, test by switching to CLANGx64).

Mirek

Subject: Re: New Assist features
Posted by Tom1 on Thu, 27 Oct 2022 11:12:46 GMT

Now this is interesting!

With CLANG/CLANGx64 and also with MSVS22/MSVS22x64 it works fine! (No need to compile, just switching the build method is enough.) However, with MSBT19/MSBT19x64/MSBT22/MSBT22x64 it fails with only a short list of items.

BR, Tom

UPDATE: To be more accurate, I'm on "Windows 11 Pro, Version 21H2, OS build 22000.1098"

Subject: Re: New Assist features
Posted by mirek on Thu, 27 Oct 2022 11:47:49 GMT
View Forum Message <> Reply to Message

Tom1 wrote on Thu, 27 October 2022 13:12Now this is interesting!

With CLANG/CLANGx64 and also with MSVS22/MSVS22x64 it works fine! (No need to compile, just switching the build method is enough.) However, with
MSBT19/MSBT19x64/MSBT22/MSBT22x64 it fails with only a short list of items.

BR, Tom

UPDATE: To be more accurate, I'm on "Windows 11 Pro, Version 21H2, OS build 22000.1098"

Fixed, at least for common usage (I am now always using CLANGx64 method for assist).

Mirek

Subject: Re: New Assist features
Posted by Tom1 on Thu, 27 Oct 2022 11:59:42 GMT
View Forum Message <> Reply to Message

Thanks!

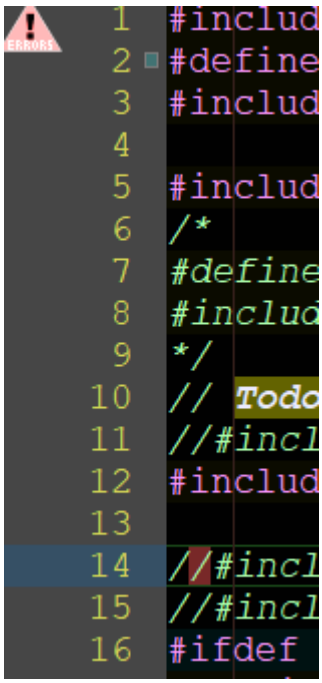
BR, Tom

Subject: Re: New Assist features
Posted by Wermann33 on Thu, 03 Nov 2022 09:29:44 GMT
View Forum Message <> Reply to Message

Hi!

I realized some weird behavior of the new error-detection, like in this example. In some files errors are detected, that definitly arent.
Another issue is, that a file i see without errors, a collegue of me, has errors shown, while i havent, using CLANG both on Windows.
In all these cases, compilation shows no problem.

File Attachments
1) Screenshot 2022-11-03 102632.png, downloaded 766 times



Subject: Re: New Assist features
Posted by Klugier on Thu, 03 Nov 2022 17:54:16 GMT
View Forum Message <> Reply to Message

Hello Wermann33,

Could you attached the problematic code with the error. From your screenshot I see that commented include was detected as an error, but nothing more. To fix this issue we need to have broader context.

I know that your code might be secret. If it is, please rework it to not be. We need only reproduction scenario, everything else is not needed.

Klugier

Subject: Re: New Assist features
Posted by mirek on Thu, 03 Nov 2022 22:05:24 GMT
View Forum Message <> Reply to Message

Klugier wrote on Thu, 03 November 2022 18:54Hello Wermann33,

Could you attached the problematic code with the error. From your screenshot I see that commented include was detected as an error, but nothing more. To fix this issue we need to have broader context.

I know that your code might be secret. If it is, please rework it to not be. We need only reproduction scenario, everything else is not needed.

Klugier

It is OK. I have seen those "phantom" errors as well, now in my internal TODO.

Subject: Re: New Assist features
Posted by Wermann33 on Fri, 04 Nov 2022 06:29:54 GMT
View Forum Message <> Reply to Message

Hello Mirek. Hello Klugier!

Sry, for the late reply. Wasnt at work anymore yesterday

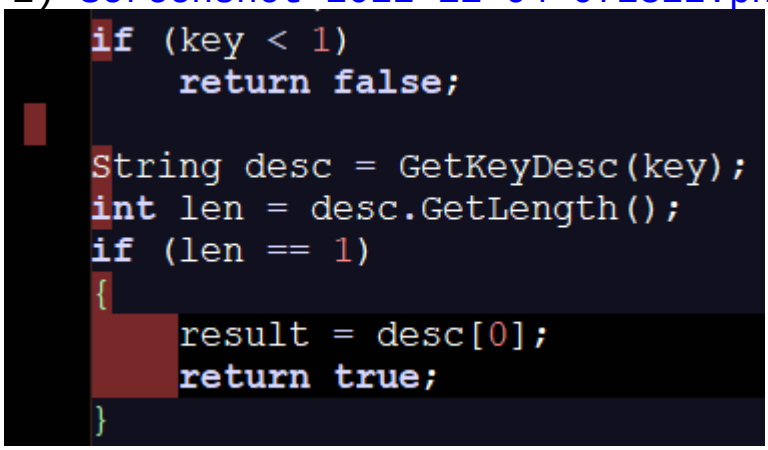
Yes, the code is secret and its really hard to reproduce those fake-errors by intention.

Just added another screenshots of an occurence.

Best Regards

File Attachments

1) Screenshot 2022-11-04 072822.png, downloaded 748 times



```
if (key < 1)
    return false;

String desc = GetKeyDesc(key);
int len = desc.GetLength();
if (len == 1)
{
    result = desc[0];
    return true;
}
```

Subject: Re: New Assist features
Posted by mirek on Fri, 04 Nov 2022 07:25:37 GMT
View Forum Message <> Reply to Message

Wermann33 wrote on Fri, 04 November 2022 07:29Hello Mirek. Hello Klugier!

Sry, for the late reply. Wasnt at work anymore yesterday

Yes, the code is secret and its really hard to reproduce those fake-errors by intention.

Just added another screenshots of an occurence.

Best Regards

Just to be sure we see the same thing: There is no information when you move mouse over it, right?

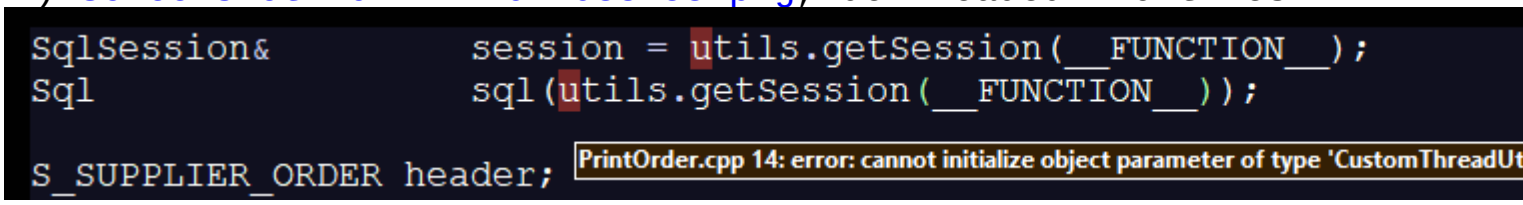
Subject: Re: New Assist features
Posted by Wermann33 on Fri, 04 Nov 2022 07:33:21 GMT
View Forum Message <> Reply to Message

Most of the time, there are no informations.

But there are some rare occurencies, information is present.
I attached a screen-example.

Code compiles anyway.

File Attachments
1) Screenshot 2022-11-04 083258.png, downloaded 770 times



Subject: Re: New Assist features
Posted by mirek on Fri, 04 Nov 2022 17:38:07 GMT
View Forum Message <> Reply to Message

Wermann33 wrote on Fri, 04 November 2022 08:33Most of the time, there are no informations.

But there are some rare occurencies, information is present.
I attached a screen-example.

Code compiles anyway.

I believe that I have found the reason (those errors were in included headers; there was missing test that the error is from the current file). Hopefully fixed in master.

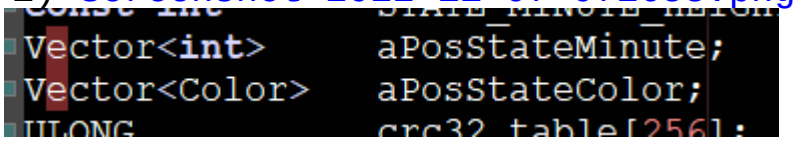
Subject: Re: New Assist features
Posted by Wermann33 on Mon, 07 Nov 2022 06:27:21 GMT
View Forum Message <> Reply to Message

Hi Mirek.

Thank you, but while some fake-errors are gone now, others appear.

File Attachments

1) Screenshot 2022-11-07 072658.png, downloaded 771 times



Subject: Re: New Assist features
Posted by mirek on Tue, 08 Nov 2022 09:36:12 GMT
View Forum Message <> Reply to Message

Wermann33 wrote on Mon, 07 November 2022 07:27Hi Mirek.

Thank you, but while some fake-errors are gone now, others appear.

Any chance you are in debug mode? (Currently, in debug, errors are not displayed - easy fix coming soon).

Subject: Re: New Assist features
Posted by Wermann33 on Tue, 08 Nov 2022 09:38:15 GMT
View Forum Message <> Reply to Message

it doesn´t matter what mode im in...

Subject: Re: New Assist features
Posted by mirek on Wed, 09 Nov 2022 13:36:11 GMT
View Forum Message <> Reply to Message

Wermann33 wrote on Tue, 08 November 2022 10:38it doesn´t matter what mode im in...

OK, another thing to try: In Assist setup, activate Assist diagnostics. Then when phantom errors appear again, in Assist menu there should be "Current file errors" item - check it out and

copy/paste here...

Mirek

Subject: Re: New Assist features
Posted by Wermann33 on Wed, 09 Nov 2022 13:45:44 GMT
View Forum Message <> Reply to Message

just an example of an older code, compiling, but showing me errors

ERROR: C:\source\Gks_trunk\Upp\uppsrc_2010\Core/Core.h (305): 'abs' is missing exception specification 'noexcept'
 NOTE: C:\source\Gks_trunk\Gks\POS\Pos.cpp (1): in file included from C:\source\Gks_trunk\Gks\POS\Pos.cpp:1:
 NOTE: C:\source\Gks_trunk\Gks\POS\Pos.cpp (1): in file included from C:\source\Gks_trunk\Gks\POS\Pos.cpp:1:
 NOTE: C:\upp\bin\clang\include\c++\v1\stdlib.h (114): previous declaration is here
ERROR: C:\source\Gks_trunk\Gks\tapi/sysTapi.h (2829): unknown type name '__out_ecount_opt'
 NOTE: C:\source\Gks_trunk\Gks\POS\Pos.cpp (1): in file included from C:\source\Gks_trunk\Gks\POS\Pos.cpp:1:
 NOTE: C:\source\Gks_trunk\Gks\POS\Pos.cpp (5): in file included from C:\source\Gks_trunk\Gks\POS\Pos.cpp:5:
 NOTE: C:\source\Gks_trunk\Gks\POS/pos.h (22): in file included from C:\source\Gks_trunk\Gks\POS/pos.h:22:
 NOTE: C:\source\Gks_trunk\Gks\tapi\tapi.h (7): in file included from C:\source\Gks_trunk\Gks\tapi\tapi.h:7:
ERROR: C:\source\Gks_trunk\Gks\tapi/sysTapi.h (2830): expected ')'
 NOTE: C:\source\Gks_trunk\Gks\tapi/sysTapi.h (2826): to match this '('
ERROR: C:\source\Gks_trunk\Gks\tapi/sysTapi.h (2843): unknown type name '__out_ecount_opt'
ERROR: C:\source\Gks_trunk\Gks\tapi/sysTapi.h (2844): expected ')'
 NOTE: C:\source\Gks_trunk\Gks\tapi/sysTapi.h (2840): to match this '('
ERROR: C:\source\Gks_trunk\Gks\tapi/sysTapi.h (2856): unknown type name '__out_ecount_opt'
ERROR: C:\source\Gks_trunk\Gks\tapi/sysTapi.h (2857): expected ')'
 NOTE: C:\source\Gks_trunk\Gks\tapi/sysTapi.h (2853): to match this '('
ERROR: C:\source\Gks_trunk\Gks\C_Db/C_Db.h (1426): non-const lvalue reference to type 'Upp::String' cannot bind to a temporary of type 'Upp::String'
 NOTE: C:\source\Gks_trunk\Gks\C_Db/C_Db.h (1426): passing argument to parameter 'message' here
ERROR: C:\source\Gks_trunk\Gks\C_Db/C_Db.h (1427): non-const lvalue reference to type 'Upp::String' cannot bind to a temporary of type 'Upp::String'
 NOTE: C:\source\Gks_trunk\Gks\C_Db/C_Db.h (1427): passing argument to parameter 'message' here
ERROR: C:\source\Gks_trunk\Gks\POS/pos.h (8490): invalid parameter name: 'default' is a keyword
 NOTE: C:\source\Gks_trunk\Gks\POS\Pos.cpp (1): in file included from C:\source\Gks_trunk\Gks\POS\Pos.cpp:1:

NOTE: C:\source\Gks_trunk\Gks\POS\Pos.cpp (5): in file included from C:\source\Gks_trunk\Gks\POS\Pos.cpp:5:
ERROR: C:\source\Gks_trunk\Gks\POS\Pos.cpp (385): no viable conversion from 'Upp::String' to 'Upp::StringBuffer'
NOTE: C:\source\Gks_trunk\Upp\uppsrc_2010\Core\String.h (342): candidate constructor (the implicit copy constructor) not viable: no known conversion from 'Upp::String' to 'const Upp::StringBuffer &' for 1st argument
NOTE: C:\source\Gks_trunk\Upp\uppsrc_2010\Core\String.h (386): candidate constructor not viable: expects an lvalue for 1st argument
NOTE: C:\source\Gks_trunk\Upp\uppsrc_2010\Core\String.h (387): candidate constructor not viable: no known conversion from 'Upp::String' to 'int' for 1st argument
NOTE: C:\source\Gks_trunk\Upp\uppsrc_2010\Core\String.h (337): candidate function
NOTE: C:\source\Gks_trunk\Upp\uppsrc_2010\Core\String.h (51): candidate function
NOTE: C:\source\Gks_trunk\Upp\uppsrc_2010\Core\String.h (53): candidate function
NOTE: C:\source\Gks_trunk\Upp\uppsrc_2010\Core\String.h (54): candidate function
ERROR: C:\source\Gks_trunk\Gks\POS\Pos.cpp (475): conditional expression is ambiguous; 'const char *' can be converted to 'Upp::String' and vice versa
ERROR: C:\source\Gks_trunk\Gks\POS\Pos.cpp (4607): taking the address of a temporary object of type 'PrintParams'
ERROR: C:\source\Gks_trunk\Gks\POS\Pos.cpp (4855): conditional expression is ambiguous; 'const char *' can be converted to 'Upp::String' and vice versa
ERROR: C:\source\Gks_trunk\Gks\POS\Pos.cpp (4891): conditional expression is ambiguous; 'const char *' can be converted to 'Upp::String' and vice versa
ERROR: C:\source\Gks_trunk\Gks\POS\Pos.cpp (5051): conditional expression is ambiguous; 'Upp::String' can be converted to 'const char *' and vice versa
ERROR: C:\source\Gks_trunk\Gks\POS\Pos.cpp (8303): no matching function for call to 'PosPromptArray'
NOTE: C:\source\Gks_trunk\Gks\TsGui\TsGui.h (1960): candidate function not viable: expects an lvalue for 2nd argument
ERROR: C:\source\Gks_trunk\Gks\POS\Pos.cpp (8314): no matching function for call to 'PosPromptArray'
NOTE: C:\source\Gks_trunk\Gks\TsGui\TsGui.h (1960): candidate function not viable: expects an lvalue for 2nd argument
ERROR: C:\source\Gks_trunk\Gks\POS\Pos.cpp (8328): no matching function for call to 'PosPromptArray'
NOTE: C:\source\Gks_trunk\Gks\TsGui\TsGui.h (1960): candidate function not viable: expects an lvalue for 2nd argument
ERROR: C:\source\Gks_trunk\Gks\POS\Pos.cpp (8864): non-const lvalue reference to type 'Vector<...>' cannot bind to a temporary of type 'Vector<...>'
NOTE: C:\source\Gks_trunk\Gks\POS\Pos.cpp (8748): passing argument to parameter 'data' here
ERROR: C:\source\Gks_trunk\Gks\POS\Pos.cpp (8902): non-const lvalue reference to type 'Vector<...>' cannot bind to a temporary of type 'Vector<...>'
NOTE: C:\source\Gks_trunk\Gks\POS\Pos.cpp (8748): passing argument to parameter 'data' here
ERROR: C:\source\Gks_trunk\Gks\POS\Pos.cpp (8908): non-const lvalue reference to type 'Vector<...>' cannot bind to a temporary of type 'Vector<...>'
NOTE: C:\source\Gks_trunk\Gks\POS\Pos.cpp (8748): passing argument to parameter 'data'

here
ERROR: C:\source\Gks_trunk\Gks\POS\Pos.cpp (8964): non-const lvalue reference to type 'Vector<...>' cannot bind to a temporary of type 'Vector<...>'
 NOTE: C:\source\Gks_trunk\Gks\POS\Pos.cpp (8748): passing argument to parameter 'data' here
ERROR: C:\source\Gks_trunk\Gks\POS\Pos.cpp (8992): non-const lvalue reference to type 'Vector<...>' cannot bind to a temporary of type 'Vector<...>'
 NOTE: C:\source\Gks_trunk\Gks\POS\Pos.cpp (8748): passing argument to parameter 'data' here
ERROR: C:\source\Gks_trunk\Gks\POS\Pos.cpp (8998): non-const lvalue reference to type 'Vector<...>' cannot bind to a temporary of type 'Vector<...>'
 NOTE: C:\source\Gks_trunk\Gks\POS\Pos.cpp (8748): passing argument to parameter 'data' here
ERROR: C:\source\Gks_trunk\Gks\POS\Pos.cpp (9055): non-const lvalue reference to type 'Vector<...>' cannot bind to a temporary of type 'Vector<...>'
 NOTE: C:\source\Gks_trunk\Gks\POS\Pos.cpp (8748): passing argument to parameter 'data' here
ERROR: C:\source\Gks_trunk\Gks\POS\Pos.cpp (9080): non-const lvalue reference to type 'Vector<...>' cannot bind to a temporary of type 'Vector<...>'
 NOTE: C:\source\Gks_trunk\Gks\POS\Pos.cpp (8748): passing argument to parameter 'data' here
ERROR: C:\source\Gks_trunk\Gks\POS\Pos.cpp (9124): non-const lvalue reference to type 'Vector<...>' cannot bind to a temporary of type 'Vector<...>'
 NOTE: C:\source\Gks_trunk\Gks\POS\Pos.cpp (8748): passing argument to parameter 'data' here
ERROR: C:\source\Gks_trunk\Gks\POS\Pos.cpp (9226): non-const lvalue reference to type 'Vector<...>' cannot bind to a temporary of type 'Vector<...>'
 NOTE: C:\source\Gks_trunk\Gks\POS\Pos.cpp (8748): passing argument to parameter 'data' here

---

Subject: Re: New Assist features
Posted by mirek on Wed, 09 Nov 2022 15:33:44 GMT
View Forum Message <> Reply to Message

Are you using MSC to compile? In that case, does it compile with CLANG?

Perhaps you are using some higher C++ standard (specified in build method)? (libclang runs c++ 14 as default now, can be altered in Assist settings).

Mirek

Subject: Re: New Assist features
Posted by Wermann33 on Wed, 09 Nov 2022 15:41:44 GMT
View Forum Message <> Reply to Message

Hi Mirek!

Yes. you are right. This project is to compile with MSC.

Subject: Re: New Assist features
Posted by mirek on Wed, 09 Nov 2022 16:01:16 GMT
View Forum Message <> Reply to Message

Wermann33 wrote on Wed, 09 November 2022 16:41Hi Mirek!

Yes. you are right. This project is to compile with MSC.

So if you compile with clang?

The problem is that MSC tolerates some issues that are not strictly C++ standard... And of course, libclang does not...

Mirek

Subject: Re: New Assist features
Posted by Wermann33 on Thu, 10 Nov 2022 06:41:58 GMT
View Forum Message <> Reply to Message

understood, can´t compile it with clang

Subject: Re: New Assist features
Posted by Klugier on Fri, 11 Nov 2022 12:18:09 GMT
View Forum Message <> Reply to Message

Hello Mirek,

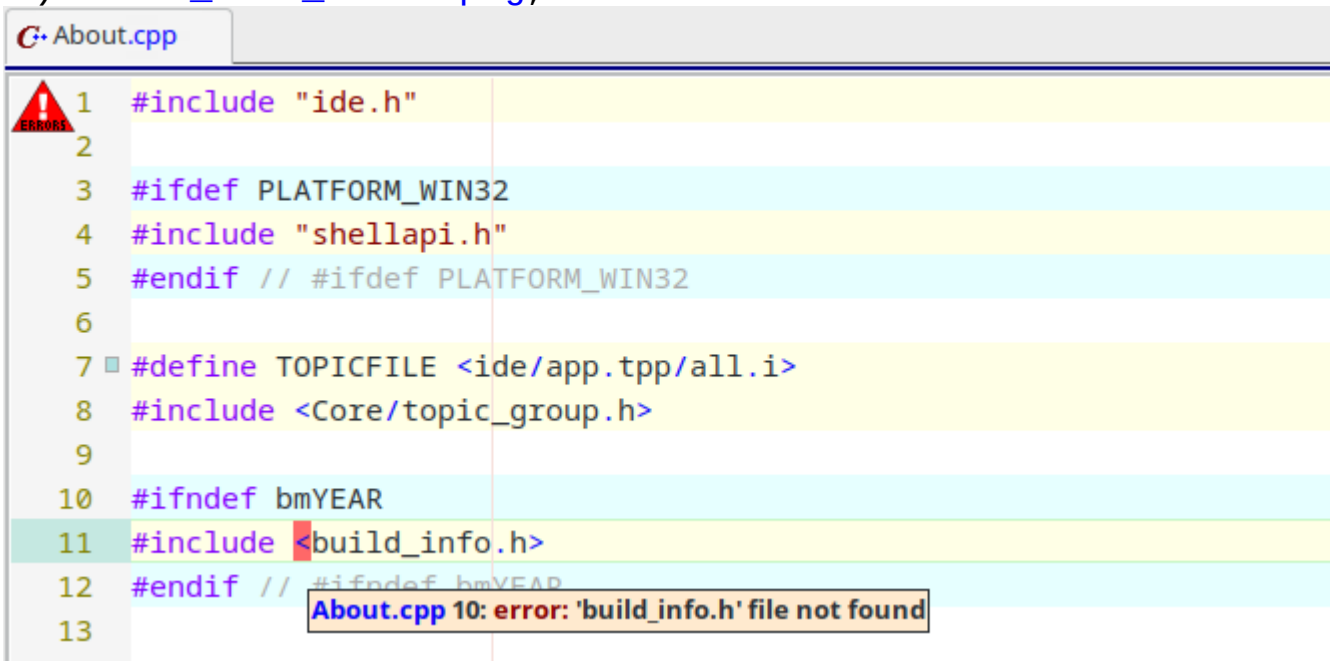
I found that Assist++ warns that build_info.h is not defined. It is true, but this file is automatically generated and attached during build process. We should do not warn about this issue or this file should be pregenerated and available for libclang to silcen it.

Here is the issue (ide/About.cpp file):

Klugier

File Attachments

1) build_info_error.png, downloaded 1268 times



Subject: Re: New Assist features
Posted by mirek on Fri, 11 Nov 2022 13:28:56 GMT
View Forum Message <> Reply to Message

Klugier wrote on Fri, 11 November 2022 13:18Hello Mirek,

I found that Assist++ warns that build_info.h is not defined. It is true, but this file is automatically generated and attached during build process. We should do not warn about this issue or this file should be pregenerated and available for libclang to silcen it.

Here is the issue (ide/About.cpp file):

Klugier

Good catch, thank you!

Mirek

Subject: Re: New Assist features
Posted by coolman on Tue, 15 Nov 2022 15:53:38 GMT
View Forum Message <> Reply to Message

Hi,

I tried new U++ assist and I found a problem with external headers. When Alt+J is executed on a function from an external header, the header is opened for the first time correctly. But then Alt+J doesn't work on this function or other functions from this external header file anymore.

BR, Radek

Subject: Re: New Assist features
Posted by mirek on Tue, 15 Nov 2022 16:19:38 GMT
View Forum Message <> Reply to Message

coolman wrote on Tue, 15 November 2022 16:53Hi,

I tried new U++ assist and I found a problem with external headers. When Alt+J is executed on a function from an external header, the header is opened for the first time correctly. But then Alt+J doesn't work on this function or other functions from this external header file anymore.

BR, Radek

Well, I am doing that all the time with libclang.h (I mean, during development of theide, I am often jumping into libclang.h, which is from this point of view 'external header').

So maybe can you be more specific?

Also, when I read "first time" - to me often happens that I edit the file, which introduces some compile errors, which is causing alt-j not working any more. Probably not your situation, but worth mentioning.

Subject: Re: New Assist features
Posted by coolman on Tue, 15 Nov 2022 17:00:08 GMT
View Forum Message <> Reply to Message

mirek wrote on Tue, 15 November 2022 17:19coolman wrote on Tue, 15 November 2022 16:53Hi,

I tried new U++ assist and I found a problem with external headers. When Alt+J is executed on a function from an external header, the header is opened for the first time correctly. But then Alt+J doesn't work on this function or other functions from this external header file anymore.

BR, Radek

Well, I am doing that all the time with libclang.h (I mean, during development of theide, I am often jumping into libclang.h, which is from this point of view 'external header').

So maybe can you be more specific?

Also, when I read "first time" - to me often happens that I edit the file, which introduces some

compile errors, which is causing alt-j not working any more. Probably not your situation, but worth mentioning.

OK,

E.g. at this code: Alt+J on X509_time_adj() open the x509.h header file at function in this header file. I switched back to the source, moved to the function X509_time_adj() (at different position using mouse) and Alt+J did not work. When I moved out of the U++ window and back in, Alt+J worked again until I did the previous steps.

ASN1_TIME *ans1Time = X509_time_adj(X509_getm_notBefore(crt.Get()), static_cast<long>(0), &time);

Edit:
I found that the symbols in the external file disappear after a while and then Alt+J doesn't work. And OK in the upper left corner is still visible.

Subject: Re: New Assist features
Posted by coolman on Sat, 17 Dec 2022 15:31:27 GMT
View Forum Message <> Reply to Message

coolman wrote on Tue, 15 November 2022 18:00mirek wrote on Tue, 15 November 2022 17:19coolman wrote on Tue, 15 November 2022 16:53Hi,

I tried new U++ assist and I found a problem with external headers. When Alt+J is executed on a function from an external header, the header is opened for the first time correctly. But then Alt+J doesn't work on this function or other functions from this external header file anymore.

BR, Radek

Well, I am doing that all the time with libclang.h (I mean, during development of theide, I am often jumping into libclang.h, which is from this point of view 'external header').

So maybe can you be more specific?

Also, when I read "first time" - to me often happens that I edit the file, which introduces some compile errors, which is causing alt-j not working any more. Probably not your situation, but worth mentioning.

OK,

E.g. at this code: Alt+J on X509_time_adj() open the x509.h header file at function in this header file. I switched back to the source, moved to the function X509_time_adj() (at different position using mouse) and Alt+J did not work. When I moved out of the U++ window and back in, Alt+J worked again until I did the previous steps.

ASN1_TIME *ans1Time = X509_time_adj(X509_getm_notBefore(crt.Get()), static_cast<long>(0),

&time);

Edit:
I found that the symbols in the external file disappear after a while and then Alt+J doesn't work.
And OK in the upper left corner is still visible.

Hi, problem with Alt+J is fixed by commit 588529a7524b3d7e432438d53995edd8d2c783ff

---