Subject: ide now using libclang
Posted by mirek on Fri, 16 Sep 2022 08:57:10 GMT
View Forum Message <> Reply to Message

After 3 months of development, theide is now using libclang for Assist. There are still some issues to fix/improve, but overall it now seems stable enough (and useful) for the community to start using/testing.

Some notes and (sometimes) temporary issues:

- in Linux, we have yet to alter the installation script to install libclang. However, as we are using runtime loading for it, when not installed it just means Assist does not work...

- Unfortunately one of drawbacks of libclang is that it does not work inside templates. I have a workaround in mind for autocomplete maybe, but for now it is a feature, not a bug.

- There is a new function, Alt-U - find all usage of current function/variables. There are more to come...

- Navigator is now usable, but still needs some work. Will work a bit differently.

- All libclang parsing is done in background threads. It is MUCH heavier on computing resources than our home-grown parser. The impact can be to a degree adjusted in Setup... Especially "indexer" - the component that needs to recompile all files of project to keep track of everything - is heavy, probably needs at least 4 CPU cores to be used reasonable. It is possible to make it "on command" in setup. That said, with 5950X, it takes 11s to reindex theide sources, which IMO is fine.

(this merge happened right now, so it will be in nightly builds tomorrow)

Subject: Re: ide now using libclang
Posted by Novo on Fri, 16 Sep 2022 11:25:03 GMT
View Forum Message <> Reply to Message

```
./umk reference CoPipe CLANG -bus
/home/buildbot/worker/l-upp/build/reference/CoPipe/CoPipe.cpp:21:7: error: no member named
'Pipe' in 'Upp::CoWork'
                co.Pipe(PROCESSLINE, [=] { SplitLine(line); });
                ~~ ^
/home/buildbot/worker/l-upp/build/reference/CoPipe/CoPipe.cpp:35:8: error: no member named
'Pipe' in 'Upp::CoWork'
                        co.Pipe(PROCESSID, [=] { ProcessId(w); });
                        ~~ ^
```

Subject: Re: ide now using libclang
Posted by mirek on Fri, 16 Sep 2022 11:47:51 GMT
View Forum Message <> Reply to Message

Novo wrote on Fri, 16 September 2022 13:25./umk reference CoPipe CLANG -bus
/home/buildbot/worker/l-upp/build/reference/CoPipe/CoPipe.cpp:21:7: error: no member named 'Pipe' in 'Upp::CoWork'
co.Pipe(PROCESSLINE, [=] { SplitLine(line); });
~~ ^
/home/buildbot/worker/l-upp/build/reference/CoPipe/CoPipe.cpp:35:8: error: no member named 'Pipe' in 'Upp::CoWork'
co.Pipe(PROCESSID, [=] { ProcessId(w); });
~~ ^

Thanks, that example needs to be deleted. Pipe was an experimental concept that failed.

Mirek

Subject: Re: ide now using libclang
Posted by Tom1 on Fri, 23 Sep 2022 10:49:07 GMT
View Forum Message <> Reply to Message

Hi Mirek,

When compiling with MSBT22x64, I get:
C:\upp-git\upp.src\uppsrc\Core\Other.h(123): warning C4267: 'argument': conversion from 'size_t' to 'int', possible loss of data
C:\upp-git\upp.src\uppsrc\Core\Other.h(143): warning C4267: 'argument': conversion from 'size_t' to 'int', possible loss of data
C:\upp-git\upp.src\uppsrc\CtrlLib/DropChoice.h(83): warning C4099: 'Upp::PopUpList::Popup': type name first seen using 'struct' now seen using 'class'
C:\upp-git\upp.src\uppsrc\CtrlLib/DropChoice.h(54): note: see declaration of 'Upp::PopUpList::Popup'
C:\upp-git\upp.src\uppsrc\CtrlCore\ImageWin32.cpp(233): warning C4267: 'argument': conversion from 'size_t' to 'int', possible loss of data
C:\upp-git\upp.src\uppsrc\CtrlCore\ImageWin32.cpp(263): warning C4267: 'argument': conversion from 'size_t' to 'int', possible loss of data
C:\upp-git\upp.src\uppsrc\CtrlCore\ImageWin32.cpp(297): warning C4267: 'return': conversion from 'size_t' to 'int', possible loss of data
C:\upp-git\upp.src\uppsrc\CtrlCore\ImageWin32.cpp(323): warning C4267: '+=': conversion from 'size_t' to 'int', possible loss of data
C:\upp-git\upp.src\uppsrc\CtrlCore\CtrlAttr.cpp(137): warning C4244: 'initializing': conversion from 'Upp::int64' to 'Upp::dword', possible loss of data
C:\upp-git\upp.src\uppsrc\CtrlCore\CtrlAttr.cpp(161): warning C4244: 'return': conversion from 'Upp::int64' to 'int', possible loss of data
C:\upp-git\upp.src\uppsrc\CtrlCore\CtrlAttr.cpp(162): warning C4244: 'return': conversion from

'Upp::int64' to 'int', possible loss of data
C:\upp-git\upp.src\uppsrc\CtrlCore\CtrlDraw.cpp(282): warning C4101: 'q': unreferenced local variable
C:\upp-git\upp.src\uppsrc\CtrlCore\Win32Clip.cpp(412): warning C4267: 'argument': conversion from 'size_t' to 'int', possible loss of data
C:\upp-git\upp.src\uppsrc\plugin\sqlite3\Sqlite3upp.cpp(644): warning C4244: 'argument': conversion from '__int64' to 'int', possible loss of data

Best regards,

Tom

Subject: Re: ide now using libclang
Posted by koldo on Sat, 24 Sep 2022 08:20:24 GMT
View Forum Message <> Reply to Message

Some of them also happen in MSVC 2017.

Subject: Re: ide now using libclang
Posted by zsolt on Thu, 29 Sep 2022 00:37:51 GMT
View Forum Message <> Reply to Message

Thanks for your work. I compiled it.
It is not so slow, quite usable on an ancient i7.

I can see two problems now:

In AddressBook example, in a method of AddressBook class, if I type a member's name and a dot, it doesn't see the member's members. It is OK on local/global variables or arguments. Is this because of that template "feature"?

An other thing is, that Help's history is buggy. It works only, if the topic has internal links with labels (e.g. TheIDE help / Build flags).
Cross references and clicking on tree items are not going into history (e.g. Used packages / Core / ArrayMap, and you click to a cross link, such as AMap or moveable). (I think, this bug is since implementing label handling in topic links.)

Edit:
My system is Ubuntu 20.4 with libclang1-10

Subject: Re: ide now using libclang
Posted by zsolt on Thu, 29 Sep 2022 01:00:55 GMT

View Forum Message <> Reply to Message

BTW, this new assist is extremely useful. It can recognize my macro generated class declarations. Cool! Thanks a lot!

Subject: Re: ide now using libclang
Posted by peterh on Thu, 29 Sep 2022 06:22:09 GMT
View Forum Message <> Reply to Message

Yes, it can also find overloaded operators with Alt-J and functionality of the "C" Button and find references with Alt-U in the designer is much improved and helpful.