Subject: theide with libclang
Posted by mirek on Sun, 26 Jun 2022 07:32:38 GMT
View Forum Message <> Reply to Message

In https://github.com/ultimatepp/ultimatepp/tree/llbm_ide branch there lives my recent development effort of integrating theide with libclang.

Now I can proudly announce that I have reached the first milestone: libclang autocomplete is now integrated well enough for me to actually start using this version of theide. Did not tested/fixed outside of windows yet though... (that said, windows seems the most complicated as we have to provide libclang.dll to the release).

Anyway, I would appreciate some more testers for this. In order to compile/use this version in windows, you need libclang.dll from official llvm release and you will also need to add include/lib paths to build method (in future, this will be coming with theide).

Mirek

Subject: Re: theide with libclang, first milestone finished
Posted by Oblivion on Sun, 26 Jun 2022 10:53:49 GMT
View Forum Message <> Reply to Message

Hello Mirek,

I've succesfully compiled the branch on Linux, however, it doesn't seem to work at all. (No assist/code completion, etc...)

(Linux 5.18, GNOME 42.1 (Wayland), LLVM/CLANG 13.0.1 (x64))

Keep in mind that this was just a quick test, Later I'll do some more...

Best regards,
Oblivion

Subject: Re: theide with libclang, first milestone finished
Posted by mirek on Sun, 26 Jun 2022 12:18:55 GMT
View Forum Message <> Reply to Message

Oblivion wrote on Sun, 26 June 2022 12:53Hello Mirek,

I've succesfully compiled the branch on Linux, however, it doesn't seem to work at all. (No assist/code completion, etc...)

(Linux 5.18, GNOME 42.1 (Wayland), LLVM/CLANG 13.0.1 (x64))

Keep in mind that this was just a quick test, Later I'll do some more...

Best regards,
Oblivion

Probably RedefineMacros() function problems - in linux, I believe it should not be there at all.

Subject: Re: theide with libclang, first milestone finished
Posted by Klugier on Sun, 26 Jun 2022 14:37:24 GMT
View Forum Message <> Reply to Message

Hello Mirek,

Good news! I will test when it will be compatible with Linux.

Klugier

Subject: Re: theide with libclang, first milestone finished
Posted by mirek on Sun, 26 Jun 2022 16:07:43 GMT
View Forum Message <> Reply to Message

OK, good news and bad news (for now):

The problem is caused by CXTranslationUnit_PrecompiledPreamble flag for clang parser. For some reason, when I specify this in Linux, libclang ignores #includes.

Without this flag, it now works. Downside is that it is much slower than in Win32 where CXTranslationUnit_PrecompiledPreamble works...

Now I am pretty sure this must be some problem of my code, as you are actually supposed to use this and I bet other ides are using this flag. But so far I was not able to crack it...

Anyway, you should now be able to test it, hopefully, for initial impressions.

Subject: Re: theide with libclang, first milestone finished
Posted by Tom1 on Mon, 27 Jun 2022 08:58:12 GMT
View Forum Message <> Reply to Message

Hi,

I must be missing something here... Compilation fails with:
C:\upp-git\upp.src\uppsrc\ide/clang/clang.h (5): fatal error C1083: Cannot open include file: 'clang-c/Index.h': No such file or directory

Best regards,

Tom

EDIT: This is on Windows using MSBT22x64 and the same happens with CLANGx64.

Subject: Re: theide with libclang, first milestone finished
Posted by mirek on Mon, 27 Jun 2022 09:16:50 GMT
View Forum Message <> Reply to Message

Tom1 wrote on Mon, 27 June 2022 10:58Hi,

I must be missing something here... Compilation fails with:
C:\upp-git\upp.src\uppsrc\ide/clang/clang.h (5): fatal error C1083: Cannot open include file: 'clang-c/Index.h': No such file or directory

Best regards,

Tom

EDIT: This is on Windows using MSBT22x64 and the same happens with CLANGx64.

It is using LLVM, so you need to have LLVM installed and paths properly set (in build method), libclang.dll has to be in the same dir as the new theide.exe.

Alternatively, you can wait before it becomes part of nightly (in the beginning, I plan to just add .dll and part of llvm sdk so that you can build llbm_ide out of box).

Subject: Re: theide with libclang, first milestone finished
Posted by Tom1 on Mon, 27 Jun 2022 09:25:07 GMT
View Forum Message <> Reply to Message

OK, I see. Maybe I'll wait a while, until it becomes part of nightly with those dependencies and settings included...

Thanks and best regards,

Tom

Subject: Re: theide with libclang, first milestone finished
Posted by mirek on Mon, 27 Jun 2022 12:35:35 GMT
View Forum Message <> Reply to Message

Tom1 wrote on Mon, 27 June 2022 11:25OK, I see. Maybe I'll wait a while, until it becomes part of nightly with those dependencies and settings included...

Thanks and best regards,

Tom

That might take a while, I do not think I will merge before everything else is changed to use libclang (like navigator, Alt+J etc...)

Mirek

Subject: Re: theide with libclang, first milestone finished
Posted by Novo on Sun, 03 Jul 2022 03:28:14 GMT
View Forum Message <> Reply to Message

Linux:
ultimatepp/uppsrc/ide/clang/clang.cpp:105:2: error: expected expression
DDUMP(cmdline);
^

Subject: Re: theide with libclang, first milestone finished
Posted by Novo on Sun, 03 Jul 2022 14:19:52 GMT
View Forum Message <> Reply to Message

I checked a clang-enabled version of TheIDE in Linux (debug version).
It is using 100% of CPU all the time. Initially, I thought that this is Clang indexing UPP files, but after waiting for 50 minutes I figured out that this must be a bug with TheIDE.

Subject: Re: theide with libclang, first milestone finished
Posted by mirek on Sun, 03 Jul 2022 22:08:08 GMT
View Forum Message <> Reply to Message

Novo wrote on Sun, 03 July 2022 16:19I checked a clang-enabled version of TheIDE in Linux (debug version).
It is using 100% of CPU all the time. Initially, I thought that this is Clang indexing UPP files, but after waiting for 50 minutes I figured out that this must be a bug with TheIDE.

Yeah, that is temporary issue, it is fixed in latest version (hopefully).

Mirek

Subject: Re: theide with libclang, first milestone finished
Posted by mirek on Thu, 07 Jul 2022 11:23:12 GMT
View Forum Message <> Reply to Message

So after developing this in Windows for a while (the hardest part now is to make clang obtained code references compatible with existing topic++ documentation) I have returned to try to fix linux issue with preamble and to my surprise it worked immediately, so probably something I have accidentally fixed in Windows.

Can please somebody retest in Linux? I believe autocomplete works now great (I mean, fast enough).

BTW, one hard job I would be happy to delegate is changing installation scripts to accound for libclang - or maybe even lack of it.

Mirek

Subject: Re: theide with libclang
Posted by mirek on Thu, 07 Jul 2022 13:32:15 GMT
View Forum Message <> Reply to Message

Second milestone finished: File annotations and Topic++ (I mean, those documentation boxes in the left bar) are now clang based.

Moving on to third milestone: Alt+C, hopefully this one will be easier...

Subject: Re: theide with libclang, first milestone finished
Posted by Novo on Thu, 07 Jul 2022 14:33:45 GMT
View Forum Message <> Reply to Message

mirek wrote on Thu, 07 July 2022 07:23Can please somebody retest in Linux? I believe autocomplete works now great (I mean, fast enough).

"make -f umkMakefile" - works.
Regular "make" (build TheIDE) didn't work (problem with linking).

Problem with CPU is fixed.
Autocomplete doesn't work for me. For example:
WStringBuffer txt;
txt.

Dot won't be autocompled. Ctrl+Space doesn't help either. I also tried autocomplete with TAB.
To be fair, I never used autocomplete with TheIDE before, but I use it with VIM all the time.

Subject: Re: theide with libclang, first milestone finished
Posted by mirek on Thu, 07 Jul 2022 15:01:25 GMT
View Forum Message <> Reply to Message

Novo wrote on Thu, 07 July 2022 16:33mirek wrote on Thu, 07 July 2022 07:23Can please somebody retest in Linux? I believe autocomplete works now great (I mean, fast enough).

"make -f umkMakefile" - works.
Regular "make" (build TheIDE) didn't work (problem with linking).

Problem with CPU is fixed.
Autocomplete doesn't work for me. For example:
WStringBuffer txt;
txt.

Dot won't be autocompled. Ctrl+Space doesn't help either. I also tried autocomplete with TAB.
To be fair, I never used autocomplete with TheIDE before, but I use it with VIM all the time.

That is weird. Looks like the whole thing is somewhat fragile in Linux

Are you testing latest llbm_ide branch, right? (make reference is confusing me a bit)

Mirek

Subject: Re: theide with libclang, first milestone finished
Posted by mirek on Thu, 07 Jul 2022 15:02:14 GMT
View Forum Message <> Reply to Message

Can you test removing "CXTranslationUnit_PrecompiledPreamble" from sources (you will easily find where)?

Subject: Re: theide with libclang, first milestone finished
Posted by Novo on Thu, 07 Jul 2022 16:00:56 GMT
View Forum Message <> Reply to Message

mirek wrote on Thu, 07 July 2022 11:01
Are you testing latest llbm_ide branch, right? (make reference is confusing me a bit)

Yes, I'm using the llbm_ide branch. Last time I tested against rev. 14dad3e.
The only possible problem I can imagine is that I copied .config (all of it) from the previous version of TheIDE.

Subject: Re: theide with libclang, first milestone finished
Posted by Novo on Thu, 07 Jul 2022 16:41:15 GMT
View Forum Message <> Reply to Message

mirek wrote on Thu, 07 July 2022 11:02Can you test removing "CXTranslationUnit_PrecompiledPreamble" from sources (you will easily find where)?
I checked that. Removing "CXTranslationUnit_PrecompiledPreamble" doesn't make any difference.
Ctrl+Space uses string on the left as a prefix for search.
Dot doesn't trigger method autocomplition.

Interestingly, umk doesn't check dependencies anymore. I need to "rebuild all" each time.

Subject: Re: theide with libclang, first milestone finished
Posted by mirek on Thu, 07 Jul 2022 16:49:11 GMT
View Forum Message <> Reply to Message

So Ctrl+Space does show something?

You can also try something selfcontained (no headers) like

```
struct Foo { int x; };

void foo() { Foo(). };
```

Mirek

Subject: Re: theide with libclang, first milestone finished
Posted by Novo on Thu, 07 Jul 2022 16:55:06 GMT
View Forum Message <> Reply to Message

mirek wrote on Thu, 07 July 2022 12:49So Ctrl+Space does show something?

You can also try something selfcontained (no headers) like

```
struct Foo { int x; };

void foo() { Foo(). };
```

Mirek
With or without CXTranslationUnit_PrecompiledPreamble?

Subject: Re: theide with libclang, first milestone finished
Posted by Novo on Thu, 07 Jul 2022 16:58:53 GMT
View Forum Message <> Reply to Message

Novo wrote on Thu, 07 July 2022 12:41
Interestingly, umk doesn't check dependencies anymore. I need to "rebuild all" each time.
This happens after "git restore". Maybe, git is getting time of last change and setting it on a file ...
It is hard to tell what is going on here.

Subject: Re: theide with libclang, first milestone finished
Posted by Novo on Thu, 07 Jul 2022 17:05:37 GMT
View Forum Message <> Reply to Message

mirek wrote on Thu, 07 July 2022 12:49So Ctrl+Space does show something?

You can also try something selfcontained (no headers) like

```
struct Foo { int x; };

void foo() { Foo(). };
```

Mirek
In this case it works (with CXTranslationUnit_PrecompiledPreamble and no headers at all).

Subject: Re: theide with libclang, first milestone finished
Posted by mirek on Thu, 07 Jul 2022 17:10:24 GMT
View Forum Message <> Reply to Message

Novo wrote on Thu, 07 July 2022 19:05mirek wrote on Thu, 07 July 2022 12:49So Ctrl+Space does show something?

You can also try something selfcontained (no headers) like

```
struct Foo { int x; };

void foo() { Foo(). };
```

Mirek
In this case it works (with CXTranslationUnit_PrecompiledPreamble and no headers at all).

Now probably stupid question, I am just asking because I was there before: Are you sure your test code is syntactically correct? Does it compile? Is not it e.g. in #if 0 block?

Also: What happens if you actually add some header? Does it break?

Mirek

Subject: Re: theide with libclang, first milestone finished
Posted by Novo on Thu, 07 Jul 2022 17:39:09 GMT
View Forum Message <> Reply to Message

mirek wrote on Thu, 07 July 2022 13:10
Now probably stupid question, I am just asking because I was there before: Are you sure your test code is syntactically correct? Does it compile? Is not it e.g. in #if 0 block?

I tested with this app.
It is a real app. And autocomplition doesn't work at any place.

I checked with another app. Dot works there ...

Another problem: Ctrl+Space ignores locally defined symbols.

Subject: Re: theide with libclang, first milestone finished
Posted by Novo on Thu, 07 Jul 2022 17:45:23 GMT
View Forum Message <> Reply to Message

mirek wrote on Thu, 07 July 2022 13:10
Also: What happens if you actually add some header? Does it break?

Adding "#include <iostream>" doesn't break it.

Subject: Re: theide with libclang, first milestone finished
Posted by Novo on Thu, 07 Jul 2022 18:15:47 GMT
View Forum Message <> Reply to Message

I'm testing TheIDE on Void Linux (rolling distro), which is using clang version 12.0.1 at this time.

Subject: Re: theide with libclang, first milestone finished
Posted by mirek on Thu, 07 Jul 2022 18:44:02 GMT
View Forum Message <> Reply to Message

Novo wrote on Thu, 07 July 2022 20:15I'm testing TheIDE on Void Linux (rolling distro), which is using clang version 12.0.1 at this time.

Well, in Windows I am at clang 14, in Linux clang 10 so that is unlikely the issue.

That said, you still have not told me what Ctrl+Space does. Obviously it shows something based on your posts.

Does it show U++ functions e.g.?

Or, opposite, does autocomplete work with std::string? (with proper includes, of course) (in that case there would be a problem with include paths)

Mirek

Subject: Re: theide with libclang, first milestone finished
Posted by Novo on Thu, 07 Jul 2022 19:35:01 GMT
View Forum Message <> Reply to Message

mirek wrote on Thu, 07 July 2022 14:44
That said, you still have not told me what Ctrl+Space does. Obviously it shows something based on your posts.

Does it show U++ functions e.g.?

App, where autocompletion works:

```
bool amap::AnalyzeMap(String fn) {
 String map = LoadFile(fn);
 char nm[1024];
 if (IsNull(map))
  return false;

 MapFileParser::MSVC MSVC(map);
 MapFileParser::CodeWarriorW CWW(map);
 MapFileParser::CodeWarrior CW(map);
 MapFileParser::PS3 PS3(map);
 MapFileParser::GCC3 GCC3(map, mcr);

 int addr;
 int sect_len;
 int prev_size = 0;
 uint64 pref_addr;
 int size;
 int modulen;
 int filen;
 String sect_name;
```

```
String sect_class;
String funct_name;
String module_name;
String file_name;
String app_name;
String time_stamp;
One<MFile> cur_mfile(new MFile);

if (map.StartsWith(elf::ElfMagic, 4)) {
 // Elf file.
 is32bit = map[4] == elf::ELFCLASS32;
 isLittleEndian = map[5] == elf::ELFDATA2LSB;
 const uint8 osABI = map[6];
 const uint8 abiVersion = map[7];
 const elf::Elf32_Ehdr* hdr = reinterpret_cast<const elf::Elf32_Ehdr*>(map.Begin());
```

If I enter "file_" and press Ctrl+Space I get "file_handle". file_name, which is defined locally, is ignored.
I do not know what Ctrl+Space is supposed to do, but it definitely not doing anything useful to me.

Subject: Re: theide with libclang, first milestone finished
Posted by Novo on Thu, 07 Jul 2022 19:42:06 GMT
View Forum Message <> Reply to Message

mirek wrote on Thu, 07 July 2022 14:44
Or, opposite, does autocomplete work with std::string? (with proper includes, of course) (in that case there would be a problem with include paths)

std::string works. Below is my test case.

```
#include <string>

struct Foo { int x; };

void foo() { Foo(); };

int main(int argc, const char *argv[])
{
 std::string str;
 str.
 return 0;
}
```

Subject: Re: theide with libclang, first milestone finished
Posted by Novo on Thu, 07 Jul 2022 20:04:43 GMT

View Forum Message <> Reply to Message

mirek wrote on Thu, 07 July 2022 14:44
That said, you still have not told me what Ctrl+Space does. Obviously it shows something based on your posts.

Does it show U++ functions e.g.?

In an app, where "dot" doesn't work Ctrl+Space shows functions from the top level (U++ functions). That works.

Subject: Re: theide with libclang, first milestone finished
Posted by Novo on Fri, 08 Jul 2022 22:32:13 GMT
View Forum Message <> Reply to Message

Novo wrote on Thu, 07 July 2022 13:39
I tested with this app.
It is a real app. And autocompletion doesn't work at any place.

Actually, it works for locally defined symbols.
TheIDE is unable to resolve and autocomplete members of a class.
The app is just one cpp file less that 2K lines of code.

I believe this is a problem with libclang because I'm getting the same behavior with VIM and ccls Language Server.

I believe you need to use LibTooling instead of LibClang with TheIDE.

Subject: Re: theide with libclang, first milestone finished
Posted by mirek on Sat, 09 Jul 2022 11:11:48 GMT
View Forum Message <> Reply to Message

Novo wrote on Sat, 09 July 2022 00:32Novo wrote on Thu, 07 July 2022 13:39
I tested with this app.
It is a real app. And autocompletion doesn't work at any place.

Actually, it works for locally defined symbols.
TheIDE is unable to resolve and autocomplete members of a class.
The app is just one cpp file less that 2K lines of code.

I believe this is a problem with libclang because I'm getting the same behavior with VIM and ccls Language Server.

I believe you need to use LibTooling instead of LibClang with TheIDE.

After playing with it for a while I think it is that preamble issue again. I have returned #ifdef to

HEAD, after that your code snippet started showing correct autocomplete.

Now it is definitely a temporary solution as it is quite slow that way, hopefully we will find a fix soon.

Mirek

Subject: Re: theide with libclang, first milestone finished
Posted by Novo on Sat, 09 Jul 2022 17:44:48 GMT
View Forum Message <> Reply to Message

mirek wrote on Sat, 09 July 2022 07:11
Now it is definitely a temporary solution as it is quite slow that way, hopefully we will find a fix soon.

I tested latest source code.
I do not see any changes.
Basically, source code where autocompletion doesn't work:

```
bool App::Parse(const String& file_name) {
 using namespace conllu;

 FileMapping s;
 if (!s.Open(file_name))
  return false;
 if (!s.Map(0, static_cast<dword>(s.GetFileSize())))
  return false;

 arr.Clear();
 sent.Clear();
 conlluTree.Clear();

 sentv.Clear();
 Parser p(reinterpret_cast<const char*>(s.Begin()), reinterpret_cast<const char*>(s.End()), sentv);

 while (p.Sentence()) ;

 for (const drtree::Sentence& v: sentv)
  sent.Add(AsString(v.GetWordV().GetCount()), v.GetText());

 return true;
}
```

In case of "arr.", "sent.", "conlluTree.", e.t.c. autocompletion doesn't work.
In case of "s." and "p." it does.

The only reason LibClang is needed is "name resolution". It is much harder to develop a name resolution algorithm than to make a C++ parser itself. If name resolution doesn't work for some

reason, then there is no reason to use LibClang. There are better solutions for parsing.

Subject: Re: theide with libclang, first milestone finished
Posted by mirek on Sat, 09 Jul 2022 22:49:30 GMT
View Forum Message <> Reply to Message

Novo wrote on Sat, 09 July 2022 19:44mirek wrote on Sat, 09 July 2022 07:11
Now it is definitely a temporary solution as it is quite slow that way, hopefully we will find a fix soon.

I tested latest source code.
I do not see any changes.
Basically, source code where autocompletion doesn't work:

```
bool App::Parse(const String& file_name) {
 using namespace conllu;

 FileMapping s;
 if (!s.Open(file_name))
  return false;
 if (!s.Map(0, static_cast<dword>(s.GetFileSize())))
  return false;

 arr.Clear();
 sent.Clear();
 conlluTree.Clear();

 sentv.Clear();
 Parser p(reinterpret_cast<const char*>(s.Begin()), reinterpret_cast<const char*>(s.End()), sentv);

 while (p.Sentence()) ;

 for (const drtree::Sentence& v: sentv)
  sent.Add(AsString(v.GetWordV().GetCount()), v.GetText());

 return true;
}
```

In case of "arr.", "sent.", "conlluTree.", e.t.c. autocompletion doesn't work.
In case of "s." and "p." it does.

The only reason LibClang is needed is "name resolution". It is much harder to develop a name resolution algorithm than to make a C++ parser itself. If name resolution doesn't work for some reason, then there is no reason to use LibClang.

It works 100% with clang 14 in Windows, hence I think the problem is somewhere else than in

clang. It works without preamble in clang 10 for me too. Too soon to try something else at this point.

Now this is one of troubles I expected before going clang - we are now victim of external incompatibilities. But what is done is done.

Quote:There are better solutions for parsing.

And these are?

Mirek

Subject: Re: theide with libclang, first milestone finished
Posted by Novo on Sat, 09 Jul 2022 23:49:46 GMT
View Forum Message <> Reply to Message

mirek wrote on Sat, 09 July 2022 18:49
And these are?

Tree-sitter - it has been a big thing for several years.

Subject: Re: theide with libclang, first milestone finished
Posted by Novo on Sun, 10 Jul 2022 00:53:55 GMT
View Forum Message <> Reply to Message

mirek wrote on Sat, 09 July 2022 18:49
It works 100% with clang 14 in Windows, hence I think the problem is somewhere else than in clang. It works without preamble in clang 10 for me too. Too soon to try something else at this point.

IMHO, you just haven't tested it very well yet. I was just "lucky" to hit a bug with my first test. My code is not that complicated. It is mostly C++98, but I use a lot of simple templates, but nothing fancy like SFINAE.

mirek wrote on Sat, 09 July 2022 18:49
Now this is one of troubles I expected before going clang - we are now victim of external incompatibilities. But what is done is done.

I do not think this is a Windows/Linux (or version of clang) incompatibility problem. I'm getting the same behavior with VIM and ccls (which is similar to clangd), and with Kakoune and kak-lsp + ccls. I use this technology for several years and I never rely on clang autocompletion, but "goto definition" works very well for me. Even in this broken app "goto definition" of sentv work fine, but for some reason language server is unable to figure out a type of sentv. This is weird. This shouldn't be a problem.

So, I personally feel fine with using LibClang + VIM because VIM has at least half a dozen of different builtin autocompletion tools. I do not care even when LibClang-related tools crash from time to time because they are out of process tools. I just do not know which part of the system (libclang or ccls) is responsible for all these problems.
But in case of TheIDE this can be a bigger problem, IMHO.

Subject: Re: theide with libclang, first milestone finished
Posted by mirek on Sun, 10 Jul 2022 06:46:59 GMT
View Forum Message <> Reply to Message

Novo wrote on Sun, 10 July 2022 02:53mirek wrote on Sat, 09 July 2022 18:49
It works 100% with clang 14 in Windows, hence I think the problem is somewhere else than in clang. It works without preamble in clang 10 for me too. Too soon to try something else at this point.

IMHO, you just haven't tested it very well yet. I was just "lucky" to hit a bug with my first test. My code is not that complicated. It is mostly C++98, but I use a lot of simple templates, but nothing fancy like SFINAE.

In Windows, I am using autocomplete for real work (really complex C++17 rich project) for about 14 days. Works flawlessly (of course, gradually maturing some things, but I have not encountered moment when it would fail to provide correct list of methods/variables).

I could try with your code if I could download it somewhere. If I delete "unknown" stuff from short snippets you are posting, it works fine in Windows/libclang 14.

Mirek

Subject: Re: theide with libclang, first milestone finished
Posted by Klugier on Sun, 10 Jul 2022 09:06:38 GMT
View Forum Message <> Reply to Message

Hello Mirek,

Few remarks from me while testing new implementation with uppsrc/ide project on Linux.

It doesn't assist with member variables. For example in idewin.cpp (Ide::ConsolePaste()) with following code:

```
void Ide::ConsolePaste()
{
 String s = ReadClipboardText();
 if(!IsNull(s)) {
  s.Insert(0, '\n');
```

```
  int len = console.GetLength();
  console.Insert(len, s.ToWString());
  console.SetCursor(len + 1);
  console. // <- No help at all
 }
}
```

The other problem within the same method is that CTRL+Space work with s. It doesn't show functionality with AString, which is essential for working with String:

```
void Ide::ConsolePaste()
{
 String s = ReadClipboardText();
 if(!IsNull(s)) {
  s.Insert(0, '\n');
  int len = console.GetLength();
  console.Insert(len, s.ToWString());
  console.SetCursor(len + 1);
  s. // <- No AString method, only derived method from String
 }
}
```

In the same place when you "include <string>" from standard library, assist will do not show anything:

```
void Ide::ConsolePaste()
{
 String s = ReadClipboardText();
 if(!IsNull(s)) {
  s.Insert(0, '\n');
  int len = console.GetLength();
  console.Insert(len, s.ToWString());
  console.SetCursor(len + 1);
  std::string str2;
            str2. // <- No assist...
 }
}
```

Also, I just noticed that CTRL+Space for String is slow for example in release mode. It is not immediate and it takes some time until dialog appear on the screen. The lag is perceptible.

The code navigation works correct and it is fast (CTRL+Left Mouse Button) and for me it looks like improvement.

Klugier

Subject: Re: theide with libclang, first milestone finished
Posted by mirek on Sun, 10 Jul 2022 14:56:30 GMT
View Forum Message <> Reply to Message

So there is definitely some issue with my libclang usage that demonstrates as a problem in Linux, something to figure out, but I do not think this is time to capitulate yet.

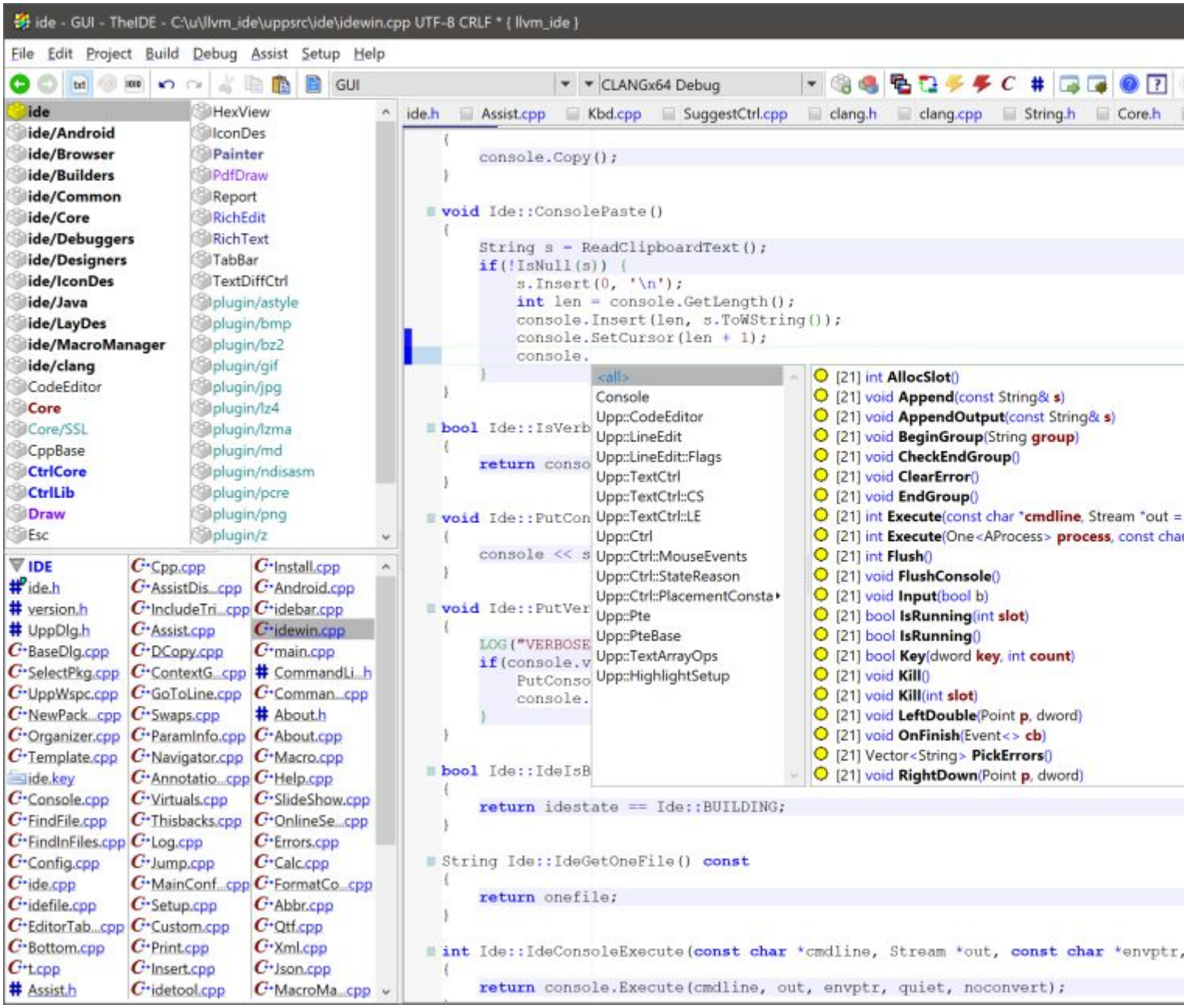
As for code navigation, that is still my old code, that is the next milestone to change that to libclang.

Things being slow are worked on as well.

Mirek

File Attachments

1) autocomplete.jpg, downloaded 2062 times



Subject: Re: theide with libclang, first milestone finished
Posted by mirek on Mon, 11 Jul 2022 16:40:24 GMT
View Forum Message <> Reply to Message

OK, so I have tried to install llvm-14 in my linux distro and linked theide with its libclang - and things started to work even with preamble, totally fine, fast, just as in Windows (except some fixing of include paths to get autocomplete of std:: stuff, but nothing serious; that part is under development anyway).

(Note - if you wonder why I am speaking about preamble so much - preamble is a flag that you give to libclang that makes it create implicit precompiled header in the first file parse. This is very

important for autocomplete because we do that parse in background thread and when user needs the autocomplete, all is ready and fast. So it is really important for comfort - using preamble, autocomplete is almost instant (~400ms). Without it can take 1-5 seconds to popup).

So my conclusion: I will only use preamble only when I detect libclang 14, I do not care about outdated distros. Autocomplete will be slow if libclang 14 is not available.

Latest ubuntu has 14, so... Maybe one last thing to try is whether libclang 13 works, if somebody has time to test.

This will be hell in the installation script (to get the most fresh libclang for given distro or even disable libclang completely), but what can we do...

I really would appreciate help with installation script BTW!

Mirek

Subject: Re: theide with libclang, first milestone finished
Posted by Klugier on Mon, 11 Jul 2022 19:57:30 GMT
View Forum Message <> Reply to Message

Hello Mirek,

libclang-14 requierments explains a lot. On Manjaro current available version is libclang-13, so maybe the issues mentioned by me in my previous message are caused by insufficient libclang library version. Manjaro is rolling release, it means that it has relatively new packages. So, the general adoption of libclang-14 in the Linux word is low.

Quote: So it is really important for comfort - using preamble, autocomplete is almost instant (~400ms). Without it can take 1-5 seconds to popup).
I can confirm that CTRL+Space take some time to appear on the screen. Is there a way to optimize it for example by pre generating auto complete or using caching? Also, I think user should be somehow inform that something is generating under the hood. For me it looks like TheIDE is unresponsive after using Ctrl+Space. Moreover, ~400ms might be too much. You need to take into consideration that it calculates that time on your CPU. With weaker CPU's this value might be bigger. So, maybe we could also optimize this using similar techniques I proposed.

Could you also tell me from where on Windows I can download libclang-14? I would like to have comparison.

Klugier

Subject: Re: theide with libclang, first milestone finished
Posted by mirek on Mon, 11 Jul 2022 20:27:10 GMT
View Forum Message <> Reply to Message

Klugier wrote on Mon, 11 July 2022 21:57Hello Mirek,

libclang-14 requierments explains a lot.

OK, some good news. I have finally figured it out and it now works fine with libclang 10 as well (commit 046f816b5a12aae4813cf4a7a83e4b6f11c13ca9)

The problem was actually in those headers, sort of as expected. In windows, I need to add compiler options for header paths even for standard headers. I left that feature in linux as well and it turns out that this is the primary cause of the problem. I guess the real issue is there is too many header paths or maybe duplicating standard paths is confusing clang. All in all, it seems to work great in linux now.

Mirek

Subject: Re: theide with libclang, first milestone finished
Posted by Klugier on Mon, 11 Jul 2022 20:49:17 GMT
View Forum Message <> Reply to Message

Hello Mirek,

I tested on the latest commit and from my perspective the performance is better, however my feelings are that Assist++ is not supper responsive and you can fell that it takes some time to appear. Could you add a release log file that will measure the Assist++ opening time? From request to appear.

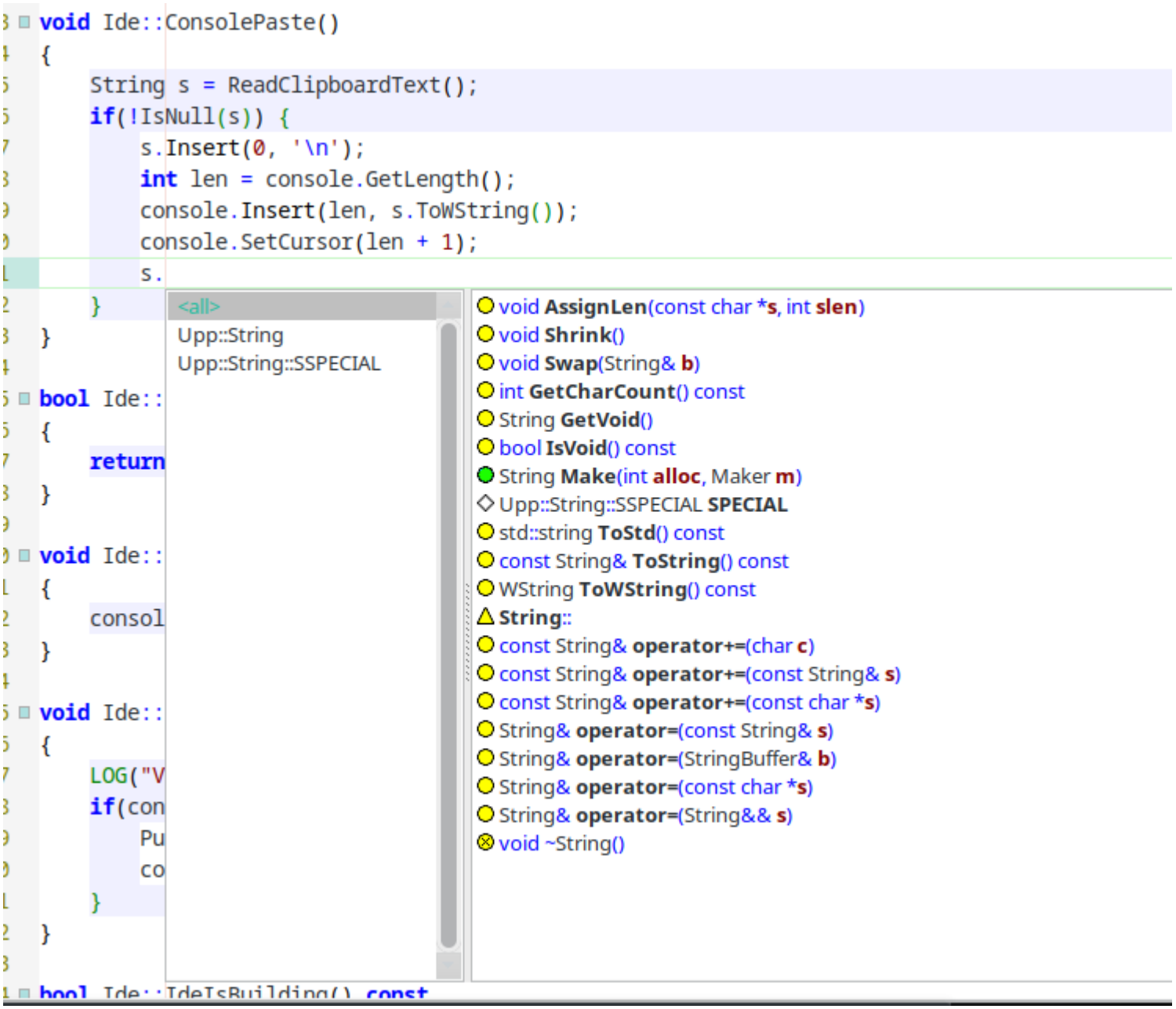
For the rest of the issues I have mentioned in my previous message, everything is the same. Here is screenshot from String without AString methods:

The console.${Assist++} doesn't open assist.

Klugier

File Attachments

1) Screenshot_20220711_224450.png, downloaded 1533 times



Subject: Re: theide with libclang, first milestone finished
Posted by Klugier on Mon, 11 Jul 2022 21:00:46 GMT
View Forum Message <> Reply to Message

Hello Mirek,

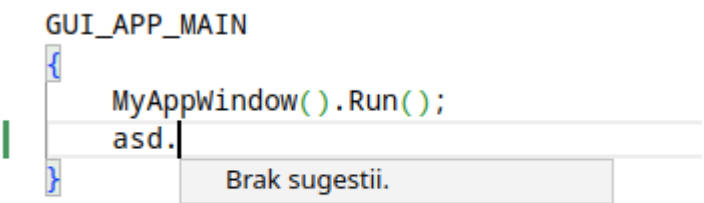
I think it would be nice to add "No suggestions" information when you explicitly hit CTRL+Space. Right now it is difficult to distinguish whenever Assist++ can not find any symbols or it is working under the hood.

Here is the feature from VSCode ("Brak sugesti." = "No sugesttions."):

Klugier

File Attachments

1) NoSuggestions.png, downloaded 1513 times



Subject: Re: theide with libclang, first milestone finished
Posted by mirek on Mon, 15 Aug 2022 08:44:33 GMT
View Forum Message <> Reply to Message

So I guess I have reached another (and sort of final) milestone - for more than week I am now working with theide with clang based navigator (that means, it is running 'indexer' in background - the code that continually maintains the database of the whole project). BTW, it takes 20 seconds to reindex theide sources on my machine - BLITZ and 16 cores in action....

Not everything is happy valley though, theide requires far more powerful HW than before. Reindexing consumes a lot of memory (around 6GB) - that is understandable considering we are running 32 C++ compilers in parallel to do it. Also autocomplete is sort of lazy in large projects as compared to my in-house parser, but that is the price to pay I guess...

There is a lot of things to fix and mature yet, my defect list has around 70 entries... Simultaneously, I will be adding nice new shiny features; new theide already got "Search for usages" that seems to be pretty useful.

I have also changed win32 nightly to contain all .dll and headers to be able to compile this llbm_ide branch directly, all you need to make sure is that new ide.exe is placed into the same directory as installed one (you can actually even use "Setup/Upgrade theide.." I think).

The remaining major issue is how to install this in linux....

Subject: Re: theide with libclang, first milestone finished
Posted by Klugier on Mon, 15 Aug 2022 16:44:42 GMT
View Forum Message <> Reply to Message

Hello Mirek,

Thanks for the update, I will play with llbm_ide branch.

Quote:Not everything is happy valley though, theide requires far more powerful HW than before. Reindexing consumes a lot of memory (around 6GB) - that is understandable considering we are running 32 C++ compilers in parallel to do it. Also autocomplete is sort of lazy in large projects as compared to my in-house parser, but that is the price to pay I guess...
As, I understand the number of C++ compiler jobs depends on HYDRA threads parameter. Could it be limited to one thread only if HYDRA threads are set to one? It would be good if TheIDE will still work on all these ARM devices that has single CPU. With only one thread running, the memory requirements should also be lower.

We have a lot of users from countries where there are difficulties to have access to modern hardware. In such cases TheIDE should work fine, even if indexing will take longer.

Scanning operation takes some time, it would be good to let user know that something is happening under the hood. We should avoid situation when user fells confusion. I think we can introduce following things:
- scanning progress is present in status bar (A lot of ide's are using this option...). Status bar is visible by default.
- CTRL+SPACE should indicate that the scanning is in progress and symbols are not ready yet. The information can be displayed in tooltip window.
- CTRL+SPACE should indicate that there are no symbols after scan (please back to my previous message). The same situation it can be displayed in tooltip window. This is exactly what I have presented in my previous message.

Klugier

Subject: Re: theide with libclang, first milestone finished
Posted by mirek on Tue, 16 Aug 2022 06:04:31 GMT
View Forum Message <> Reply to Message

Klugier wrote on Mon, 15 August 2022 18:44- CTRL+SPACE should indicate that there are no symbols after scan (please back to my previous message). The same situation it can be displayed in tooltip window. This is exactly what I have presented in my previous message.
Klugier

Well, there are always some symbols to display after CTRL+SPACE (unless you are in the completely empty file)...

Better indication of autocomplete in progress is coming..

Mirek

Subject: Re: theide with libclang
Posted by Novo on Tue, 16 Aug 2022 17:18:36 GMT
View Forum Message <> Reply to Message

Linux. Branch: llbm_ide
Command: make -f umkMakefile
In file included from ./uppsrc/CtrlCore/CtrlCore.h:49,
                 from ./uppsrc/CtrlLib/CtrlLib.h:4,
                 from ./uppsrc/RichEdit/RichEdit.h:4,
                 from ./uppsrc/ide/ide.h:4,
                 from ./uppsrc/ide/Core/Ide.cpp:1:
./uppsrc/CtrlCore/Gtk.h:12:10: fatal error: gtk/gtk.h: No such file or directory
   12 | #include <gtk/gtk.h>
      | ^~~~~~~~~~~

umk is a console app ...

Subject: Re: theide with libclang
Posted by Novo on Tue, 16 Aug 2022 17:25:21 GMT
View Forum Message <> Reply to Message

Linux. Branch: llbm_ide
Command: ./umk uppsrc ide CLANG -arbus
Problem: Fatal error Invalid memory access! (on opening a project)

Subject: Re: theide with libclang
Posted by mirek on Fri, 19 Aug 2022 11:03:03 GMT
View Forum Message <> Reply to Message

Novo wrote on Tue, 16 August 2022 19:18Linux. Branch: llbm_ide
Command: make -f umkMakefile
In file included from ./uppsrc/CtrlCore/CtrlCore.h:49,
                 from ./uppsrc/CtrlLib/CtrlLib.h:4,
                 from ./uppsrc/RichEdit/RichEdit.h:4,
                 from ./uppsrc/ide/ide.h:4,
                 from ./uppsrc/ide/Core/Ide.cpp:1:
./uppsrc/CtrlCore/Gtk.h:12:10: fatal error: gtk/gtk.h: No such file or directory
   12 | #include <gtk/gtk.h>
      | ^~~~~~~~~~~

umk is a console app ...

Fixed.

Subject: Re: theide with libclang, first milestone finished
Posted by mirek on Fri, 19 Aug 2022 11:04:58 GMT
View Forum Message <> Reply to Message

Klugier wrote on Mon, 15 August 2022 18:44
- scanning progress is present in status bar (A lot of ide's are using this option...). Status bar is visible by default.

Working on it all, however the idea is that scanning will be represented by left bar changing color (which makes sense as there are annotation that get updated).

Subject: Re: theide with libclang, first milestone finished
Posted by mirek on Fri, 19 Aug 2022 11:05:14 GMT
View Forum Message <> Reply to Message

Hopefully fixed Linux crashing in release mode.

Subject: Re: theide with libclang, first milestone finished
Posted by Novo on Sat, 20 Aug 2022 18:00:24 GMT
View Forum Message <> Reply to Message

mirek wrote on Fri, 19 August 2022 07:05Hopefully fixed Linux crashing in release mode.
Thanks. TheIde works fine on Linux.

Subject: Re: theide with libclang, first milestone finished
Posted by Novo on Sat, 20 Aug 2022 18:33:27 GMT
View Forum Message <> Reply to Message

CPU usage.

TheIde is heavily using CPU even if I change one character.
This is a problem for my ten-years-old laptop.
Is it possible to add an options, which will
1) disable libclang completely.
2) limit reindexing to a "Save" command.
3) keep libclang running all the time.

TIA.

Subject: Re: theide with libclang, first milestone finished
Posted by Tom1 on Sat, 20 Aug 2022 18:34:45 GMT

View Forum Message <> Reply to Message

Hi,

I had to move #include <X11/Xlib.h> in GtkApp.cpp in order to compile in Windows with clang:
#include <CtrlCore/CtrlCore.h>

#ifdef GUI_GTK
#include <X11/Xlib.h>
#define CATCH_ERRORS 1

Additionally, compilation fails to missing include in MSBT22x64:
C:\upp-16323\upp.src\uppsrc\ide/clang/clang.h (5): fatal error C1083: Cannot open include file: 'clang-c/Index.h': No such file or directory

Best regards,

Tom

Subject: Re: theide with libclang, first milestone finished
Posted by mirek on Sat, 20 Aug 2022 22:42:58 GMT
View Forum Message <> Reply to Message

Tom1 wrote on Sat, 20 August 2022 20:34Hi,

I had to move #include <X11/Xlib.h> in GtkApp.cpp in order to compile in Windows with clang:
#include <CtrlCore/CtrlCore.h>

#ifdef GUI_GTK
#include <X11/Xlib.h>
#define CATCH_ERRORS 1

Additionally, compilation fails to missing include in MSBT22x64:
C:\upp-16323\upp.src\uppsrc\ide/clang/clang.h (5): fatal error C1083: Cannot open include file: 'clang-c/Index.h': No such file or directory

Best regards,

Tom

Compilation of theide with MSC is not supported at the moment. Not sure it will be possible. Use

CLANG64...

Mirek

Subject: Re: theide with libclang, first milestone finished
Posted by mirek on Sat, 20 Aug 2022 22:46:22 GMT
View Forum Message <> Reply to Message

Novo wrote on Sat, 20 August 2022 20:33CPU usage.

TheIde is heavily using CPU even if I change one character.
This is a problem for my ten-years-old laptop.
Is it possible to add an options, which will
1) disable libclang completely.
2) limit reindexing to a "Save" command.
3) keep libclang running all the time.

TIA.

Disable is needed. However, reindexing is already issued only in specific moments (save being one of them). The problem is that initially it has to do it all.

Also, changing headers can trigger reindexing of a lot of files.

This is the price to pay.... (this is the reason I was reluctant to do abandon my parser....)

Mirek

Subject: Re: theide with libclang, first milestone finished
Posted by mirek on Tue, 30 Aug 2022 14:04:45 GMT
View Forum Message <> Reply to Message

Good news: I have found a bug in my code (unitialised variable for all things) that was making autocomplete slow. It now seems to be superfast after the initial parse.

Things appear to fall in place. The last major hurdle is to figure out Linux installation. The new idea to try is to use runtime .so loading...

Mirek

Subject: Re: theide with libclang, first milestone finished

Posted by Tom1 on Wed, 31 Aug 2022 08:00:43 GMT
View Forum Message <> Reply to Message

Hi Mirek,

Each time I have e.g.:
 void LeftDown(Point p, dword keyflags) override {

I get a row: "WARNING: EMIT" in my Navigator. If I remove the override, it goes away. (I do not think this is anything new, but it shows up in the latest version too.)

Second, there are some warnings (at least on MSBT):
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'

Best regards,

Tom

Subject: Re: theide with libclang, first milestone finished
Posted by mirek on Wed, 31 Aug 2022 08:13:21 GMT
View Forum Message <> Reply to Message

Tom1 wrote on Wed, 31 August 2022 10:00Hi Mirek,

Each time I have e.g.:
 void LeftDown(Point p, dword keyflags) override {

I get a row: "WARNING: EMIT" in my Navigator. If I remove the override, it goes away. (I do not think this is anything new, but it shows up in the latest version too.)

I have to say I am a bit confused here...

Can you post a screenshot? (full window so that I can see the source etc.. too)

Subject: Re: theide with libclang, first milestone finished
Posted by Tom1 on Wed, 31 Aug 2022 08:50:42 GMT

View Forum Message <> Reply to Message

Hi,

Sorry. This must be caused by something strange deeper in my own code, since creating a simple test case does not seem to trigger showing it.

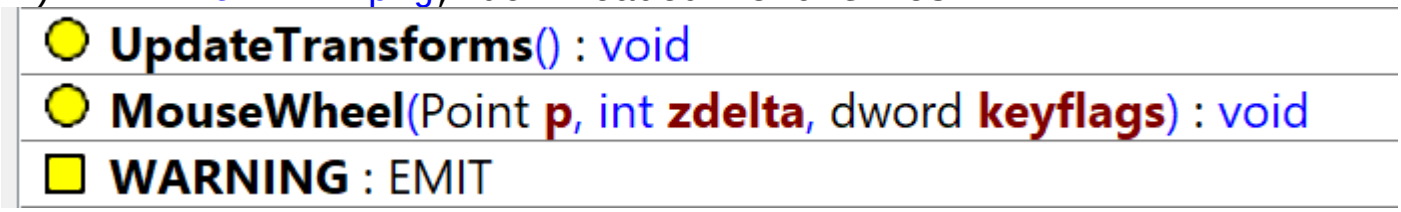
Anyway, I will stop wasting your time on this one for now. (I bet you have something better to do...

Thanks and best regards,

Tom

File Attachments

1) WARNING-EMIT.png, downloaded 1570 times



Subject: Re: theide with libclang, first milestone finished
Posted by mirek on Wed, 31 Aug 2022 09:37:58 GMT
View Forum Message <> Reply to Message

Tom1 wrote on Wed, 31 August 2022 10:50Hi,

Sorry. This must be caused by something strange deeper in my own code, since creating a simple test case does not seem to trigger showing it.

Anyway, I will stop wasting your time on this one for now. (I bet you have something better to do...

Thanks and best regards,

Tom

Ah, ok, by the screenshot I immediatelly see this non-clang version anyway (new theide has arrow instead of ":")

Mirek

Subject: Re: theide with libclang, first milestone finished
Posted by Tom1 on Wed, 31 Aug 2022 10:33:47 GMT
View Forum Message <> Reply to Message

mirek wrote on Wed, 31 August 2022 12:37Tom1 wrote on Wed, 31 August 2022 10:50Hi,

Sorry. This must be caused by something strange deeper in my own code, since creating a simple test case does not seem to trigger showing it.

Anyway, I will stop wasting your time on this one for now. (I bet you have something better to do...

Thanks and best regards,

Tom

Ah, ok, by the screenshot I immediatelly see this non-clang version anyway (new theide has arrow instead of ":")

Mirek

Thanks Mirek!

For some reason my upp.src had slipped back to master branch. Anyway, I switched back to clang and now all my problems are gone. Arrows are shown instead of colon and there are no more "WARNING: EMIT" for overrides. Thanks!

Best regards,

Tom

Subject: Re: theide with libclang, first milestone finished
Posted by Tom1 on Thu, 15 Sep 2022 06:31:11 GMT
View Forum Message <> Reply to Message

Mirek,

Would it be possible to add support for working with git branches in TheIDE menu?

Support for 'switching branch' for any git repository currently in use would be most useful, but possibly also support for creating and merging of branches.

Best regards,

Tom

Subject: Re: theide with libclang, first milestone finished
Posted by mirek on Thu, 15 Sep 2022 07:48:18 GMT
View Forum Message <> Reply to Message

Tom1 wrote on Thu, 15 September 2022 08:31Mirek,

Would it be possible to add support for working with git branches in TheIDE menu?

Support for 'switching branch' for any git repository currently in use would be most useful, but possibly also support for creating and merging of branches.

Best regards,

Tom

I would like to, I just do not quite know how to do it right. So far "Terminal at file directory" was my favorite tool in this regard...

Plus it is not quite related to Assist/clang, correct?

I plan a batch of another theide improvements after libclang is merged. Which now seems will be quite soon.

Mirek

Subject: Re: theide with libclang, first milestone finished
Posted by Tom1 on Thu, 15 Sep 2022 12:10:15 GMT
View Forum Message <> Reply to Message

mirek wrote on Thu, 15 September 2022 10:48Tom1 wrote on Thu, 15 September 2022 08:31Mirek,

Would it be possible to add support for working with git branches in TheIDE menu?

Support for 'switching branch' for any git repository currently in use would be most useful, but possibly also support for creating and merging of branches.

Best regards,

Tom

I would like to, I just do not quite know how to do it right. So far "Terminal at file directory" was my favorite tool in this regard...

Plus it is not quite related to Assist/clang, correct?

I plan a batch of another theide improvements after libclang is merged. Which now seems will be

quite soon.

Mirek

Yes, sorry. This is not the place for this discussion. Anyway, I'm looking forward to progress on this too.

Best regards,

Tom

Subject: Re: theide with libclang, first milestone finished
Posted by mirek on Fri, 16 Sep 2022 09:28:14 GMT
View Forum Message <> Reply to Message

Merged...

Subject: Re: theide with libclang, first milestone finished
Posted by Novo on Sat, 17 Sep 2022 03:54:28 GMT
View Forum Message <> Reply to Message

Merged version. Linux.
libclang: crash detected during reparsing
libclang: crash detected during parsing: {
'source_filename' : '(null)'
'command_line_args' : ['clang', '-I/home/ssg/dvlp/cpp/sergey/upp',
'-I/home/ssg/dvlp/cpp/code/upp/git/uppsrc', '-I/home/ssg/dvlp/cpp/code/upp/git/bazaar',
'-I/home/ssg/.local/tmp/dvlp/cpp/code/upp/.config/u++/ide/UppHub/SysExec',
'-I/home/ssg/.local/tmp/dvlp/cpp/code/upp/.config/u++/ide/UppHub/Updater',
'-I/usr/include/freetype2', '-I/usr/include/libpng16
', '-I/usr/include/freetype2', '-I/usr/include/libpng16
', '-I/usr/include/freetype2', '-I/usr/include/libpng16
', '-I/usr/include/gtk-3.0', '-I/usr/include/at-spi2-atk/2.0', '-I/usr/include/at-spi-2.0',
'-I/usr/include/dbus-1.0', '-I/usr/lib64/dbus-1.0/include', '-I/usr/include/gtk-3.0',
'-I/usr/include/gio-unix-2.0', '-I/usr/include/cairo', '-I/usr/include/pango-1.0', '-I/usr/include/harfbuzz',
'-I/usr/include/pango-1.0', '-I/usr/include/fribidi', '-I/usr/include/harfbuzz', '-I/usr/include/atk-1.0',
'-I/usr/include/cairo', '-I/usr/include/pixman-1', '-I/usr/include/freetype2', '-I/usr/include/libpng16',
'-I/usr/include/gdk-pixbuf-2.0', '-I/usr/include/libpng16', '-I/usr/include/libmount', '-I/usr/include/blkid',
'-I/usr/include/glib-2.0', '-I/usr/lib64/glib-2.0/include
', '-I/usr/include/gdk-pixbuf-2.0', '-I/usr/include/libpng16', '-I/usr/include/libmount',
'-I/usr/include/blkid', '-I/usr/include/glib-2.0', '-I/usr/lib64/glib-2.0/include
', '-I/usr/include/libpng16
', '-I/usr/include/c++/10.2', '-I/usr/include/c++/10.2/x86_64-unknown-linux-gnu',
'-I/usr/include/c++/10.2/backward', '-I/usr/local/include', '-I/usr/lib/clang/12.0.1/include',
'-I/usr/include', '-I/usr/include/c++/10.2', '-I/usr/include/c++/10.2/x86_64-unknown-linux-gnu',

'-I/usr/include/c++/10.2/backward', '-I/usr/local/include', '-I/usr/lib/clang/12.0.1/include',
'-I/usr/include', '-DflagGUI', '-DflagPOSIX', '-DflagLINUX', '-DCLANG',
'/home/ssg/dvlp/cpp/code/upp/git/uppsrc/Core/lheap.cpp', '-DflagDEBUG', '-DflagDEBUG_FULL',
'-DflagMAIN', '-DflagCLANG', '-xc++', '-std=c++17', '-Dlint'],
  'unsaved_files' : [],
  'options' : -2147483136,
}

Subject: Re: theide with libclang, first milestone finished
Posted by Novo on Sat, 17 Sep 2022 04:44:49 GMT
View Forum Message <> Reply to Message

Is it possible to disable libclang support during compilation?
I'm having troubles trying to cross-compile Clang. Actually, I couldn't compile it at all.
I'm trying to cross-compile TheIDE for Windows on Linux.

Subject: Re: theide with libclang, first milestone finished
Posted by mirek on Sat, 17 Sep 2022 09:04:26 GMT
View Forum Message <> Reply to Message

What is your HW configuration?

I am setting some defaults for assist config, but the lowest HW I can test with is Ryzen 2400G 8GB Linux - they seems to work fine.

Check Setup BTW, I would start with disabling automatic indexer... BTW, I might be wrong, but I think for theide, every "indexer thread" and "autocomplete file cache" is about 200MB of memory.

Mirek

Subject: Re: theide with libclang, first milestone finished
Posted by Novo on Sat, 17 Sep 2022 10:23:38 GMT
View Forum Message <> Reply to Message

mirek wrote on Sat, 17 September 2022 05:04What is your HW configuration?
Intel(R) Core(TM) i7-3720QM CPU @ 2.60GHz (4 cores / 8 threads)
16GM RAM
It is a Lenovo W530. (I also have a T530)
I like the keyboard on these models

Subject: Re: theide with libclang, first milestone finished

Posted by Novo on Sat, 17 Sep 2022 10:37:03 GMT
View Forum Message <> Reply to Message

mirek wrote on Sat, 17 September 2022 05:04
Check Setup BTW, I would start with disabling automatic indexer... BTW, I might be wrong, but I think for theide, every "indexer thread" and "autocomplete file cache" is about 200MB of memory.

Opening a project and closing it in a couple of seconds leads to "Invalid memory access" ...
EDIT: Happens after full rebuild.

Subject: Re: theide with libclang, first milestone finished
Posted by mirek on Sat, 17 Sep 2022 10:48:47 GMT
View Forum Message <> Reply to Message

Novo wrote on Sat, 17 September 2022 12:23mirek wrote on Sat, 17 September 2022 05:04What is your HW configuration?
Intel(R) Core(TM) i7-3720QM CPU @ 2.60GHz (4 cores / 8 threads)
16GM RAM
It is a Lenovo W530. (I also have a T530)
I like the keyboard on these models

That is strange, that is about the same or better than my linux machine....

Maybe your libclang is somehow deffective?

Mirek

Subject: Re: theide with libclang, first milestone finished
Posted by Novo on Sat, 17 Sep 2022 10:57:33 GMT
View Forum Message <> Reply to Message

mirek wrote on Sat, 17 September 2022 06:48
Maybe your libclang is somehow deffective?

I'm using "Void Linux". Its default clang version is 12.0.1.
Void Linux is like a stable version of Arch.
I do not have any problems with ccls on these machines ...

Subject: Re: theide with libclang, first milestone finished
Posted by Novo on Sat, 17 Sep 2022 11:01:02 GMT
View Forum Message <> Reply to Message

I'm getting weird error messages in console on exit.
/usr/include/glib-2.0/glib/gtypes.h:32:10: error: 'glibconfig.h' file not found

/usr/include/glib-2.0/glib/gversionmacros.h:375:2: error: "GLIB_VERSION_MIN_REQUIRED must be >= GLIB_VERSION_2_26"
/usr/include/glib-2.0/glib/gtypes.h:548:2: error: unknown ENDIAN type
/usr/include/glib-2.0/glib/garray.h:49:3: error: unknown type name 'guint8'
/usr/include/glib-2.0/glib/garray.h:75:36: error: unknown type name 'gsize'
...
/home/ssg/dvlp/cpp/sergey/upp/amap/amap.cpp:81:2: error: cannot initialize object parameter of type 'Upp::TopWindow' with an expression of type 'amap'
/home/ssg/dvlp/cpp/sergey/upp/amap/amap.cpp:83:2: error: no matching member function for call to 'Icon'
/home/ssg/dvlp/cpp/code/upp/git/uppsrc/CtrlCore/TopWindow.h:174:13: note: candidate function not viable: no known conversion from 'amap' to 'Upp::TopWindow' for object argument
/home/ssg/dvlp/cpp/code/upp/git/uppsrc/CtrlCore/TopWindow.h:176:13: note: candidate function not viable: requires 2 arguments, but 1 was provided
...
fatal error: too many errors emitted, stopping now [-ferror-limit=]

Subject: Re: theide with libclang
Posted by jjacksonRIAB on Sun, 18 Sep 2022 10:22:57 GMT
View Forum Message <> Reply to Message

Seemed OK at first but:

- Include file completion doesn't work (you're probably already aware of this)
- Alt+C works in .cpp files (only one direction though, from inner class scope to outer) but not .h files. It also appears to sometimes get the outer class name wrong.

Biggest issue thus far is 100% CPU and memory exhaustion. Before indexing, TheIDE uses around 50 megabytes, after first indexing it goes up to around 150, then eventually libclang breaks, autocomplete stops working for some symbols, then all - a second indexing increases memory usage until it reaches around 1.8 gig. This is with only one core enabled. With all cores, it uses all available system memory and freezes the OS... typically when compiling.

Subject: Re: theide with libclang
Posted by mirek on Sun, 18 Sep 2022 11:05:09 GMT
View Forum Message <> Reply to Message

jjacksonRIAB wrote on Sun, 18 September 2022 12:22Seemed OK at first but:

- Include file completion doesn't work (you're probably already aware of this)

I went really far to make it work (is not really simple with libclang), so that is unpleasant surprise.

Quote:
Biggest issue thus far is 100% CPU and memory exhaustion. Before indexing, TheIDE uses around 50 megabytes, after first indexing it goes up to around 150, then eventually libclang breaks, autocomplete stops working for some symbols, then all - a second indexing increases memory usage until it reaches around 1.8 gig. This is with only one core enabled. With all cores, it uses all available system memory and freezes the OS... typically when compiling.

Can you please tell me details about the machine?

I was mostly testing / using in Windows; there with 32 indexing threads the memory goes to somewhere like 6GB.

Subject: Re: theide with libclang
Posted by jjacksonRIAB on Sun, 18 Sep 2022 11:27:32 GMT
View Forum Message <> Reply to Message

Linux mint, it's an older machine with 8 cores and 16 gig of memory, libclang should be version 12 IIRC. If 6 gig is common I'll probably have to turn it off because the compiler causes a lot of memory pressure as is, oh well

---

Subject: Re: theide with libclang
Posted by Novo on Sun, 18 Sep 2022 14:57:16 GMT
View Forum Message <> Reply to Message

In my case (Lenovo ThikPad T530 8GB RAM, 2 cores (4 threads), indexer is using 4 threads) when indexing TheIde itself:

- 1.3GB when reindexing all src. Memory usage doesn't go down after reindexing.
- 300MB when opening an already indexed TheIde project.

I guess, memory allocation strategy is inefficient.

Subject: Re: theide with libclang
Posted by Novo on Sun, 18 Sep 2022 19:23:27 GMT
View Forum Message <> Reply to Message

valgrind explains everything.

File Attachments
1) memchk.zip, downloaded 304 times

Subject: Re: theide with libclang
Posted by mirek on Tue, 20 Sep 2022 11:09:38 GMT
View Forum Message <> Reply to Message

Novo wrote on Sun, 18 September 2022 21:23valgrind explains everything.

While it is definitely a good idea, only one of these errors seems libclang related and fixable and it happens on theide exit. Anyway, now fixed.

That said, everything valigrind says is worth investigation. I was able to fix many of these issues (in master now), unfortunately it looks like majority of remaining problems is cairo related (these happen when we are reading images from gtk).

Any help identifying the cause(s) is welcome.

Mirek

Subject: Re: theide with libclang
Posted by Novo on Tue, 20 Sep 2022 17:16:26 GMT
View Forum Message <> Reply to Message

mirek wrote on Tue, 20 September 2022 07:09Novo wrote on Sun, 18 September 2022 21:23valgrind explains everything.

While it is definitely a good idea, only one of these errors seems libclang related and fixable and it happens on theide exit. Anyway, now fixed.

That said, everything valigrind says is worth investigation. I was able to fix many of these issues (in master now), unfortunately it looks like majority of remaining problems is cairo related (these happen when we are reading images from gtk).

Any help identifying the cause(s) is welcome.

Mirek
Thanks a lot! TheIde doesn't crash anymore.
I personally do not see any cairo problems. I guess Void Linux has a better valgrind exceptions list.
The only problem I do see is an uninitialized value in Ide::Serialize.
Interestingly, this problem depends on when you close TheIde.
I guess this data is uninitialized during libclang indexing/reindexing.
My command line:
valgrind --track-origins=yes --log-file=memchk.txt
~/dvlp/cpp/code/upp/out/CLANG.Debug.Debug_Full.Gui.Shared/ide

File Attachments

1) memchk.02.zip, downloaded 287 times

Subject: Re: theide with libclang
Posted by jjacksonRIAB on Tue, 20 Sep 2022 17:21:00 GMT
View Forum Message <> Reply to Message

Quote:
Thanks a lot! TheIde doesn't crash anymore.

Really? I got the latest updates and it was completely broken for me. My assemblies/nests were all empty except some example code and then trying to open even one of those crashed the ide. I must be having a run of bad luck lately, I'll try to clone the whole thing again.

Subject: Re: theide with libclang
Posted by mirek on Tue, 20 Sep 2022 17:37:57 GMT
View Forum Message <> Reply to Message

Novo wrote on Tue, 20 September 2022 19:16mirek wrote on Tue, 20 September 2022 07:09Novo wrote on Sun, 18 September 2022 21:23valgrind explains everything.

While it is definitely a good idea, only one of these errors seems libclang related and fixable and it happens on theide exit. Anyway, now fixed.

That said, everything valigrind says is worth investigation. I was able to fix many of these issues (in master now), unfortunately it looks like majority of remaining problems is cairo related (these happen when we are reading images from gtk).

Any help identifying the cause(s) is welcome.

Mirek
Thanks a lot! TheIde doesn't crash anymore.
I personally do not see any cairo problems. I guess Void Linux has a better valgrind exceptions list.
The only problem I do see is an uninitialized value in Ide::Serialize.
Interestingly, this problem depends on when you close TheIde.
I guess this data is uninitialized during libclang indexing/reindexing.

I am getting that one (in Ide::Serialize) too; my guts feeling is that it is not related to libclang, just some old serialisation bug like forgotten unitialised variable. Want to hunt that down, but it is time consuming....

What about those "_dl_catch_exception " things? Does not look like U++ fault, right?

Subject: Re: theide with libclang

Posted by jjacksonRIAB on Tue, 20 Sep 2022 17:46:20 GMT
View Forum Message <> Reply to Message

Quote:
I am getting that one (in Ide::Serialize) too; my guts feeling is that it is not related to libclang

Possible but I was getting crashes on close on the previous version and it was dumping a libclang exception to the terminal because it was indexing at the time, so it was reporting an indexing exception. I don't have the text for it anymore unfortunately.

Subject: Re: theide with libclang
Posted by jjacksonRIAB on Tue, 20 Sep 2022 18:13:12 GMT
View Forum Message <> Reply to Message

OK looks like I had a bad repo. I rebuilt and I'm back in. I'm still having excessive memory usage though and another curious thing I noticed is that it reindexes the file I have open in the editor regardless of whether it's changed. If I click away to another app and back to theide it will reindex. It never frees memory, even if I switch to another package entirely.

I'm not getting the clang error in the console anymore though, so that appears to be fixed. I am, however, getting freezes if I cancel a build.

Subject: Re: theide with libclang
Posted by jjacksonRIAB on Tue, 20 Sep 2022 19:32:31 GMT
View Forum Message <> Reply to Message

OK so with address sanitizer and USEMALLOC enabled, I'm getting a bunch of reported heap overflows all on memcmp.

lib/clang Signature.cpp line 171, for example:

if(memcmp(s, " = {", 4) == 0)

what happens if s is shorter than the sequence it's being compared against? Does it overrun?

Subject: Re: theide with libclang
Posted by jjacksonRIAB on Tue, 20 Sep 2022 22:33:41 GMT
View Forum Message <> Reply to Message

Once I replaced those memcmp instructions with strnlen it shut up address sanitizer. They're probably fine but at least I was able to get further and... there was nothing worthwhile except the usual font stuff.

Found another errata to potentially test. If I go into any sufficiently complex package (like CtrlCore or CtrlLib) and hold down alt + down arrow to cycle through/open as many files as quickly as possible, libclang gets stuck in an infinite reparsing loop on one file until I click off to another file. I'm assuming this is because I'm leaving the last parsable .cpp/.h file and moving into other non-parseable file types rapidly.

Edit also if you left ctrl click a symbol with indexing turned off it will still try to index but pin a core instead.

Subject: Re: theide with libclang
Posted by Novo on Tue, 20 Sep 2022 23:33:00 GMT
View Forum Message <> Reply to Message

mirek wrote on Tue, 20 September 2022 13:37
I am getting that one (in Ide::Serialize) too; my guts feeling is that it is not related to libclang, just some old serialisation bug like forgotten unitialised variable. Want to hunt that down, but it is time consuming....

One time I ran valgring I saw ZERO problems. I didn't believe my eyes. I reran it and got this Ide::Serialize problem. This is why I think it depends on something else.

mirek wrote on Tue, 20 September 2022 13:37
What about those "_dl_catch_exception " things? Does not look like U++ fault, right?

It is hard to tell. In my latest rerun I didn't get this "_dl_catch_exception " thing.
I believe it happens when I close TheIde before it finishes loading of libclang data.
I'm closing it via unorthodox window manager called herbstluft, which is sending a message to TheIde, I believe.
So, something gets interrupted in the middle, I guess.
Theoretically, a loader should automatically unload everything correctly ...

File Attachments

1) memchk.03.zip, downloaded 298 times

Subject: Re: theide with libclang
Posted by Novo on Tue, 20 Sep 2022 23:56:10 GMT
View Forum Message <> Reply to Message

I've got a different call stack with Ide::Serialize.
Maybe it will help to locate a problem.

File Attachments
1) memchk.04.zip, downloaded 308 times

Subject: Re: theide with libclang
Posted by mirek on Wed, 21 Sep 2022 08:52:28 GMT
View Forum Message <> Reply to Message

Novo wrote on Wed, 21 September 2022 01:56I've got a different call stack with Ide::Serialize. Maybe it will help to locate a problem.

Thanks, fixed. (Unfortunately it was complete non-bug that could not have influenced anything real, just unitialized variable that is never used again).

Subject: Re: theide with libclang
Posted by jjacksonRIAB on Wed, 21 Sep 2022 20:09:15 GMT
View Forum Message <> Reply to Message

Thanks for doing this stuff Mirek. I was looking through the code trying to figure it out and learn about libclang ASTs but I simply don't know enough about TheIDE's internal operations (or libclang) to make heads or tails of it. I also swear I find you everywhere.

I came across you on Stack Overflow asking about libclang, then a couple of weeks ago I was reading some posts from Andrei Alexandrescu and I saw you in there too.

As for libclang one of the strange problems I had that shouldn't happen is that clang_parseTranslationUnit was returning a translation unit of 0 and as a consequence Dispose() was never getting called. I wasn't able to fix it because it mysteriously stopped happening and I don't know if it's anything I did. All I did was put in the hook for clang_parseTranslationUnit2 to see if it would give me a more meaningful error... but that turned out to be useless. It stopped happening even after I switched back.

The only other change I made was I hooked clang_defaultReparseOptions and passed that to clang_reparseTranslationUnit. Supposedly it provides some default options that improve performance, don't really know if it did... so I flailed around uselessly for a few hours. The other behavior I had and am still having is that it's not simply parsing and then calling reparse after the initial parse, it's calling both every time. Once I got into CurrentFileThread it got too deep into TheIDE internals to understand what was going on.

Subject: Re: theide with libclang
Posted by Novo on Fri, 23 Sep 2022 03:41:41 GMT
View Forum Message <> Reply to Message

mirek wrote on Wed, 21 September 2022 04:52
Thanks, fixed.
Thanks a lot!
Could you also fix a problem below, please?
./umk reference SMTP CLANG -busr
/home/buildbot/worker/l-upp/build/uppsrc/Core/SMTP/Smtp.cpp:457:2: error: expected expression
	DDUMP(GetMessage(true));
	^
/home/buildbot/worker/l-upp/build/uppsrc/Core/Diag.h:159:26: note: expanded from macro 'DDUMP'
#define DDUMP(x)	@ // To clean logs after debugging, this produces error in release mode
                  ^
1 error generated.

Subject: Re: theide with libclang
Posted by Novo on Fri, 23 Sep 2022 03:52:28 GMT
View Forum Message <> Reply to Message

mirek wrote on Wed, 21 September 2022 04:52Novo wrote on Wed, 21 September 2022 01:56I've got a different call stack with Ide::Serialize.
Maybe it will help to locate a problem.

Thanks, fixed. (Unfortunately it was complete non-bug that could not have influenced anything real, just unitialized variable that is never used again).
I do see two problems with valgrind messages:

1) Tons of "uninitialized value" messages hide real problems.
2) Nobody will use Upp with commercial apps for a very simple reason: who knows what kind of problems can be caused by these "uninitialized values".

IMHO, Upp should be checked with all available valgrind tools and all available sanitizers. At least before release.

Subject: Re: theide with libclang
Posted by jjacksonRIAB on Fri, 23 Sep 2022 05:09:47 GMT
View Forum Message <> Reply to Message

Mirek,

Runaway memory allocations seem not to be an issue using it all day. The one thing I can report today is libclang getting stuck in a reparse loop on app exit
and refusing to let the ide close. It happens rarely but I can reliably force it to happen by attempting to exit the ide during a build.

Subject: Re: theide with libclang
Posted by mirek on Fri, 23 Sep 2022 08:48:15 GMT
View Forum Message <> Reply to Message

Novo wrote on Fri, 23 September 2022 05:52mirek wrote on Wed, 21 September 2022 04:52Novo wrote on Wed, 21 September 2022 01:56I've got a different call stack with Ide::Serialize.
Maybe it will help to locate a problem.

Thanks, fixed. (Unfortunately it was complete non-bug that could not have influenced anything real, just unitialized variable that is never used again).
I do see two problems with valgrind messages:

1) Tons of "uninitialized value" messages hide real problems.
2) Nobody will use Upp with commercial apps for a very simple reason: who knows what kind of problems can be caused by these "uninitialized values".

IMHO, Upp should be checked with all available valgrind tools and all available sanitizers. At least before release.

I do not argue with that (as you have seen, I have fixed what I could).

The comment was not meant as an apology for the bug, just a note that it cannot explain any weird behaviour with libclang.

Subject: Re: theide with libclang
Posted by Lance on Sat, 24 Sep 2022 02:41:10 GMT
View Forum Message <> Reply to Message

I don't know how much improvement we can expect in, say half an year's time. At the moment, I Ctrl+Left click on a member function in a moderately complex project, the parsing starts and goes on for ever. At least 30 minutes past and no idea how much longer it's going to take. In contrast, it used to work so smoothly.

Theoretically moving to libclang is the right thing to do, but there is a huge chance it may render theide completely useless.

Some system info
Quote:
Upp version: 16443

OS: Ubuntu 22.04.1 LTS x86_64
CPU: Intel i7-8650U (8) @ 4.200GHz
GPU: Intel UHD Graphics 620
Memory: 3757MiB / 15875MiB

PS: I tested with Examples/HomeBudget. In HomeBudget.cpp, Ctrl+left click some member

function, eg LoadCategories(...) to start parsing. Now roughly 10 minutes past, still parsing. Something might not be right.

Subject: Re: theide with libclang
Posted by peterh on Sat, 24 Sep 2022 05:26:23 GMT
View Forum Message <> Reply to Message

Fortunately I can say, on my computer it works almost immediately.
I am on Windows 10 64 bit.
The computer is a Ryzen 7, 32 GByte Memory.

I had a look into taskmanager. Total memory used was 12 GBytes.
I had some other programs open, Browser, email, Fotoviewer Faststone, Fotosoftware.
TheIde used 2 GBytes and had the biggest memory chunk in use.

I can imagine, if the computer had less memory, if the drive is a harddrive and not a SSD, and if TheIDE temporarily wants more memory, then the computer would start swapping and it could terribly slow down.
(If it had an SSD it could be faster, but would lead to unneccessary SSD wear) (I have a SSD but I hope it will not be used for swapping)

Fortunately swapping does not happen here.
It might be useful to monitor and report this machine data and properties.

Possibly reboot and do not start other programs than TheIDE and compare the behavior and memory consumption.
So far I know, libclangd can be disabled in the setup. This could be a solution as a last ressort if memory is low.

Subject: Re: theide with libclang
Posted by mirek on Sat, 24 Sep 2022 06:08:35 GMT
View Forum Message <> Reply to Message

Lance wrote on Sat, 24 September 2022 04:41I don't know how much improvement we can expect in, say half an year's time. At the moment, I Ctrl+Left click on a member function in a moderately complex project, the parsing starts and goes on for ever. At least 30 minutes past and no idea how much longer it's going to take. In contrast, it used to work so smoothly.

Theoretically moving to libclang is the right thing to do, but there is a huge chance it may render theide completely useless.

It is just first week after the merge...

Quote:Something might not be right.

The is the most likely explanation. There is no reason for it to take so long, it is probably stuck somewhere.

Is autocomplete working?

Mirek

Subject: Re: theide with libclang
Posted by mirek on Sat, 24 Sep 2022 06:44:58 GMT
View Forum Message <> Reply to Message

OK, so one week in, I guess I owe the community a bit of explanation about libclang and the way theide uses it...

So, obviously what libclang does is parsing the files and giving info about them. Additionally, it can provide autocomplete lists at specified point in the source files, but NOT HEADERS (and also not inside templates, but that is store for another time).

The important thing to know is that parsing the file with all headers is long: on my not so slow machine, it can take up to several seconds to parse the file. To remedy this, libclang has a concept of "preamble" and "reparsing". Preamble is basically automated precompiled header - if file starts with multiple headers, preamble is creating and when after the first parse the file is reparsed or autocomplete requested, preamble is used to greatly speedup the process. With preamble, autocomplete (after the first parse) is almost immediate. That said, premable creation is sort of fragile. Also, normally it gets created in system temp directory, which would be fine unless sometimes libclang does not delete it. But it can be solved be setting env variable which we do now.

So what we do now is the when you start editing some file, in the background thread the first parse of the file is started with a request to create preamble. In theide, you can know that this is happening by left bar going "sort of orange" for a couple of seconds - it does that each time file is parsed or reparsed. If you then change the file, it gets (after some time of inactivity) reparsed again. Parsing current file gets it ready for autocomplete, but also provides information about file for file annotations, current file navigation and "jumps" (Alt-J / Ctrl clicks).

Originally, there was just one parsed file, which meant each time after switching to another file it had to have that initial a couple of seconds parsing period. Later I have added a cache, there are now several files with parsed info. Downside is that each "parsing unit" consumes a lot of memory, about 300MB so with 12 entries in the cache, you can do the math....

Include files that clang is unable to process: What we do is that we find "master file" - the well defined file that includes the header and we pretend that we are parsing that one. It is not so simple: we still want to use preamble to make things fast and if we just parsed master file and hoped to catch the info include file from it, it would not work with preambles as our include file would actually be in preamble. I have spent a lot of time trying to solve this: What we actually do now is that we create a syntetic file content that actually manually does including so that it all

works (it is hard to explain in detail; in assist diagnostic mode you can see the content of the syntetic file with Assist/Current parsed file content).

So that is about all I can say about "current file" without going into gruesome details; if you want to study the code, libclang/CurrentFile.cpp and ide/AssistTrick.cpp are good starting points.

Next troublesome part of code is "Indexer" - that is something that scans the whole sources and creates a program database of, well, everything everywhere. That is needed for Navigator and Alt+J command. We are using blitz(like) process there and we are running that in multiple threads. Now there is "Scheduler" (Indexer::SchedulerThread()) thread that scans for changed files and basically prepares jobs for "Indexing" (Indexer::IndexerThread) threads. Once again, each of indexer threads can eat up to 300MB of memory, so finetuning default values will take some time. More indexer threads mean faster reindexing but perhaps too much memory...

In terms of slowdown/freezing during reindexing, well, the plan was to give indexer threads really low priority. It should be possible with

```
  clang_CXIndex_setGlobalOptions(clang.index,
CXGlobalOpt_ThreadBackgroundPriorityForIndexing);
```

but maybe this does not work as expected in Linux. In Windows, it seems to work fine.

Now for testing and troubleshooting, please note that there is "Assist diagnostics" option in Assist/Debug. When active, it will start reporting in Console what it is doing and also there are new entries at the end of assist menu, showing a dump of actual index, errors libclang reported during parsing and what was given to clang to parse as current file.

So if e.g. there is no autocomplete for include files, please check "Current file parse errors" and "Current parsed file content".

I know this is a big change (and one I was extremly hesitant to do) and not everything works smoothly at this point, but hopefully all will be finetuned in time. I guess we fixed a lot already in the first week....

Mirek

Subject: Re: theide with libclang
Posted by jjacksonRIAB on Sat, 24 Sep 2022 07:53:19 GMT
View Forum Message <> Reply to Message

Nah, Mirek, I don't feel I'm owed an explanation

Windows users seem to be OK, like you said it's linux that needs some love. In my case I had an ancient version of libclang 7.0 sitting in /usr/local/lib. Once I zapped that the excessive memory consumption disappeared so I'm pretty sure that was libclang. I updated to 15 and things are still looking good. I did find valgrind complain about some branches on uninitialized variables in theide itself:

```
==562946== Conditional jump or move depends on uninitialised value(s)
==562946==    at 0x56DB7A: Upp::CodeEditor::DirtyFrom(int) (CodeEditor.cpp:74)
==562946==    by 0x5A2DEF: Upp::TextCtrl::Insert0(int, Upp::WString const&) (Text.cpp:799)
==562946==    by 0x5A3C2E: Upp::TextCtrl::InsertU(int, Upp::WString const&, bool)
(Text.cpp:935)
==562946==    by 0x5A3FB7: Upp::TextCtrl::Insert(int, Upp::WString const&, bool) (Text.cpp:980)
==562946==    by 0x315C96: Upp::TextCtrl::Insert(int, Upp::WString const&) (TextEdit.h:218)
==562946==    by 0x5A4E6B: Upp::TextCtrl::Paste(Upp::WString const&) (Text.cpp:1153)
==562946==    by 0x3075CD: Console::Append(Upp::String const&) (Console.cpp:77)
==562946==    by 0x326EDB: Console::operator<<(Upp::String const&) (ide.h:127)
==562946==    by 0x3CA52A: Ide::PutConsole(char const*) (idewin.cpp:50)
==562946==    by 0x46B05A: PutConsole(char const*) (Ide.cpp:9)
==562946==    by 0x34EC3E: AssistEditor::SyncMaster() (IncludeTrick.cpp:18)
==562946==    by 0x3536C7: AssistEditor::NewFile(bool) (Assist.cpp:438)
==562946==
==562946== Conditional jump or move depends on uninitialised value(s)
==562946==    at 0x56D63F: Upp::CodeEditor::GetSyntax(int) (CodeEditor.cpp:29)
==562946==    by 0x56DC5B: Upp::CodeEditor::GetRefreshInfo(int) (CodeEditor.cpp:94)
==562946==    by 0x56DEFA: Upp::CodeEditor::PostInsert(int, Upp::WString const&)
(CodeEditor.cpp:111)
==562946==    by 0x5A335E: Upp::TextCtrl::Insert0(int, Upp::WString const&) (Text.cpp:851)
==562946==    by 0x5A3C2E: Upp::TextCtrl::InsertU(int, Upp::WString const&, bool)
(Text.cpp:935)
==562946==    by 0x5A3FB7: Upp::TextCtrl::Insert(int, Upp::WString const&, bool) (Text.cpp:980)
==562946==    by 0x315C96: Upp::TextCtrl::Insert(int, Upp::WString const&) (TextEdit.h:218)
==562946==    by 0x5A4E6B: Upp::TextCtrl::Paste(Upp::WString const&) (Text.cpp:1153)
==562946==    by 0x3075CD: Console::Append(Upp::String const&) (Console.cpp:77)
==562946==    by 0x326EDB: Console::operator<<(Upp::String const&) (ide.h:127)
==562946==    by 0x3CA52A: Ide::PutConsole(char const*) (idewin.cpp:50)
==562946==    by 0x46B05A: PutConsole(char const*) (Ide.cpp:9)
```

I was also able to trigger a crash of TheIDE by just holding down Ctrl+Space for a minute or so to make it continually attempt to autocomplete, put some pressure on it. Unfortunately it only happens in release mode, I cannot reproduce it in debug mode.

Subject: Re: theide with libclang
Posted by peterh on Sat, 24 Sep 2022 08:03:18 GMT
View Forum Message <> Reply to Message

Hi,

I have reported 3 GB memory in windows.

I must add, I had the IDE open for probably 2 days.
Normally I do not shut down my computer but put it to sleep at night or when I do not use it.
I have now rebooted and after opening TheIDE with the same project and having tried "Homebudget" also, it consumes only 300MB.

Subject: Re: theide with libclang
Posted by mirek on Sat, 24 Sep 2022 08:16:09 GMT
View Forum Message <> Reply to Message

jjacksonRIAB wrote on Sat, 24 September 2022 09:53Nah, Mirek, I don't feel I'm owed an explanation

It was not meant specifically for you...

I felt that general explanation of at least some inner workings will help us fix things.

In your case I remember complaining about headers. Do they work now? (If not, there are some hints in my long post where to look for troubles).

As for old libclang... I can put limit for minimal version, e.g. libclang 10 (I am quite sure that one works), what do you think?

Mirek

Subject: Re: theide with libclang
Posted by jjacksonRIAB on Sat, 24 Sep 2022 08:30:01 GMT
View Forum Message <> Reply to Message

mirek wrote on Sat, 24 September 2022 10:16
It was not meant specifically for you...

Serves me right for attempting to be conciliatory.

Quote:
In your case I remember complaining about headers. Do they work now?

I'll get latest and check. But the list of outstanding issues I have:

1. Includes don't autocomplete
2. Parser gets stuck in a reparse loop if I cycle through files in a project using alt+down and hit a file that is not code (like a .lay or .iml file)

3. Alt+C, which you are already working on

Quote:
As for old libclang... I can put limit for minimal version, e.g. libclang 10 (I am quite sure that one works), what do you think?

Sounds good.

Subject: Re: theide with libclang
Posted by jjacksonRIAB on Sat, 24 Sep 2022 08:45:48 GMT
View Forum Message <> Reply to Message

Still no joy on include autocomplete.

Subject: Re: theide with libclang
Posted by jjacksonRIAB on Sat, 24 Sep 2022 09:10:47 GMT
View Forum Message <> Reply to Message

Mirek,

The problem appears to manifest with assist_cursor in Assist.cpp line 502.

It gathers all of the potential include files appropriately but assist_cursor never matches current pos so it returns before performing the completion.

```
void AssistEditor::PopUpAssist(bool auto_insert)
{
 LTIMING("PopUpAssist");
 int pos = GetCursor();
 ReadIdBackPos(pos, false);
 if(pos != assist_cursor)
  return;
```

EDIT It's uninitialized. The only place I can find it being set is around line 463 but there's an early return

```
include_assist = false;
if(IncludeAssist())
    return;

int pos = GetCursor();
```

ReadIdBackPos(pos, false); // libclang does not work well if file is not truncated for autocomplete (?)
assist_cursor = pos;

before assist_cursor is set. If I move that block of code after the pos set it shows the files but it has strange index numbers next to the icons. <- never mind, that appears to be for debugging purposes and doesn't appear in release mode.

So the fix is:

```
int pos = GetCursor();
ReadIdBackPos(pos, false); // libclang does not work well if file is not truncated for autocomplete (?)
assist_cursor = pos;

include_assist = false;
if(IncludeAssist())
    return;
```

Subject: Re: theide with libclang
Posted by Lance on Sat, 24 Sep 2022 13:00:34 GMT
View Forum Message <> Reply to Message

mirek wrote on Sat, 24 September 2022 02:08
Is autocomplete working?

No.It is not. Is it because libclang not present in my system?

PS:
I already have clang installed. After I installed libclang-dev, key event in theide will immediately crash theide with error message
Quote:
Fatal Error!
Invalid memory access!

The libclang.so shipped with most recent(?) libclang-dev(1:14.0-55~exp2) doesn't seem to work with theide. It's most likely to be some kind of chaos in my system.

ps2: Now even after I remove'd libclang-dev, theide in upp version 16443 won't work. You don't

even need to type in anything, it will crash itself with invalid memory access in just a few seconds(when background parsing threads start). It looks like a libclang version problem. For now, I will go back to upp version 16303 on which autocomplete works fine.

Subject: Re: theide with libclang
Posted by jjacksonRIAB on Sat, 24 Sep 2022 17:55:26 GMT
View Forum Message <> Reply to Message

Lance wrote on Sat, 24 September 2022 15:00
ps2: Now even after I remove'd libclang-dev, theide in upp version 16443 won't work. You don't even need to type in anything, it will crash itself with invalid memory access in just a few seconds(when background parsing threads start). It looks like a libclang version problem. For now, I will go back to upp version 16303 on which autocomplete works fine.

I didn't try dev packages, I just uninstalled clang through package manager and ran

sudo bash -c "$(wget -O - https://apt.llvm.org/llvm.sh)"

to bring everything to clang-15, including libclang. Should work with debian/apt systems.

Subject: Re: theide with libclang
Posted by mirek on Sat, 24 Sep 2022 19:44:36 GMT
View Forum Message <> Reply to Message

jjacksonRIAB wrote on Sat, 24 September 2022 11:10before assist_cursor is set. If I move that block of code after the pos set it shows the files but it has strange index numbers next to the icons. <- never mind, that appears to be for debugging purposes and doesn't appear in release mode.

That is actually the feature of "assist diagnostics". (Those numbers are libclang defined constants for that type of object, listing it helps in case the icon is wrong etc...)

Otherwise, thanks for spotting the bug. Does include assist work now?

Mirek

Subject: Re: theide with libclang
Posted by jjacksonRIAB on Sat, 24 Sep 2022 19:45:36 GMT
View Forum Message <> Reply to Message

mirek wrote on Sat, 24 September 2022 21:44
Otherwise, thanks for spotting the bug. Does include assist work now?

Mirek

Yes, it works now.

Subject: Re: theide with libclang
Posted by jjacksonRIAB on Sat, 24 Sep 2022 21:09:56 GMT
View Forum Message <> Reply to Message

This is the problem I'm currently trying to work out but not getting anywhere yet:

https://files.catbox.moe/t4kcpd.mp4

It will get stuck in a reparse loop. Same thing also happens occasionally when I'm exiting TheIDE. Note that it happens when I move rapidly to a resource-type file. I didn't capture it in the video but it behaves normally and exits the loop if I go back to a parseable .cpp or .h file, so it's not stuck that way permanently - it is just attempting to redo the same old job repeatedly until it gets a new job. When it happens on exit it becomes permanently stuck.

Subject: Re: theide with libclang
Posted by Lance on Sun, 25 Sep 2022 00:13:48 GMT
View Forum Message <> Reply to Message

jjacksonRIAB wrote on Sat, 24 September 2022 13:55Lance wrote on Sat, 24 September 2022 15:00
ps2: Now even after I remove'd libclang-dev, theide in upp version 16443 won't work. You don't even need to type in anything, it will crash itself with invalid memory access in just a few seconds(when background parsing threads start). It looks like a libclang version problem. For now, I will go back to upp version 16303 on which autocomplete works fine.

I didn't try dev packages, I just uninstalled clang through package manager and ran

sudo bash -c "$(wget -O - https://apt.llvm.org/llvm.sh)"

to bring everything to clang-15, including libclang. Should work with debian/apt systems.

Thank you, jjacksonRIAB! Your command worked fine and I was able to upgrade my libclang. Unfortunately the problem remains with UPP version 16443: theide will crash in a few seconds after a package is open with "Invalid Memory Access" error.

Subject: Re: theide with libclang
Posted by Klugier on Sun, 25 Sep 2022 11:48:51 GMT
View Forum Message <> Reply to Message

Hello Lance,

Please remove cahce and cofnig directories:

rm -rf ${upp_dir}/.cache
rm -rf ${upp_dir}/.config

It should help.

I know that in the past we eliminated some migration issues related to new clang implementation, but it seems that there is something from the old installation that is causing crash.

Klugier

Subject: Re: theide with libclang
Posted by Lance on Sun, 25 Sep 2022 12:48:18 GMT
View Forum Message <> Reply to Message

Klugier wrote on Sun, 25 September 2022 07:48Hello Lance,

Please remove cahce and cofnig directories:

rm -rf ${upp_dir}/.cache
rm -rf ${upp_dir}/.config

It should help.

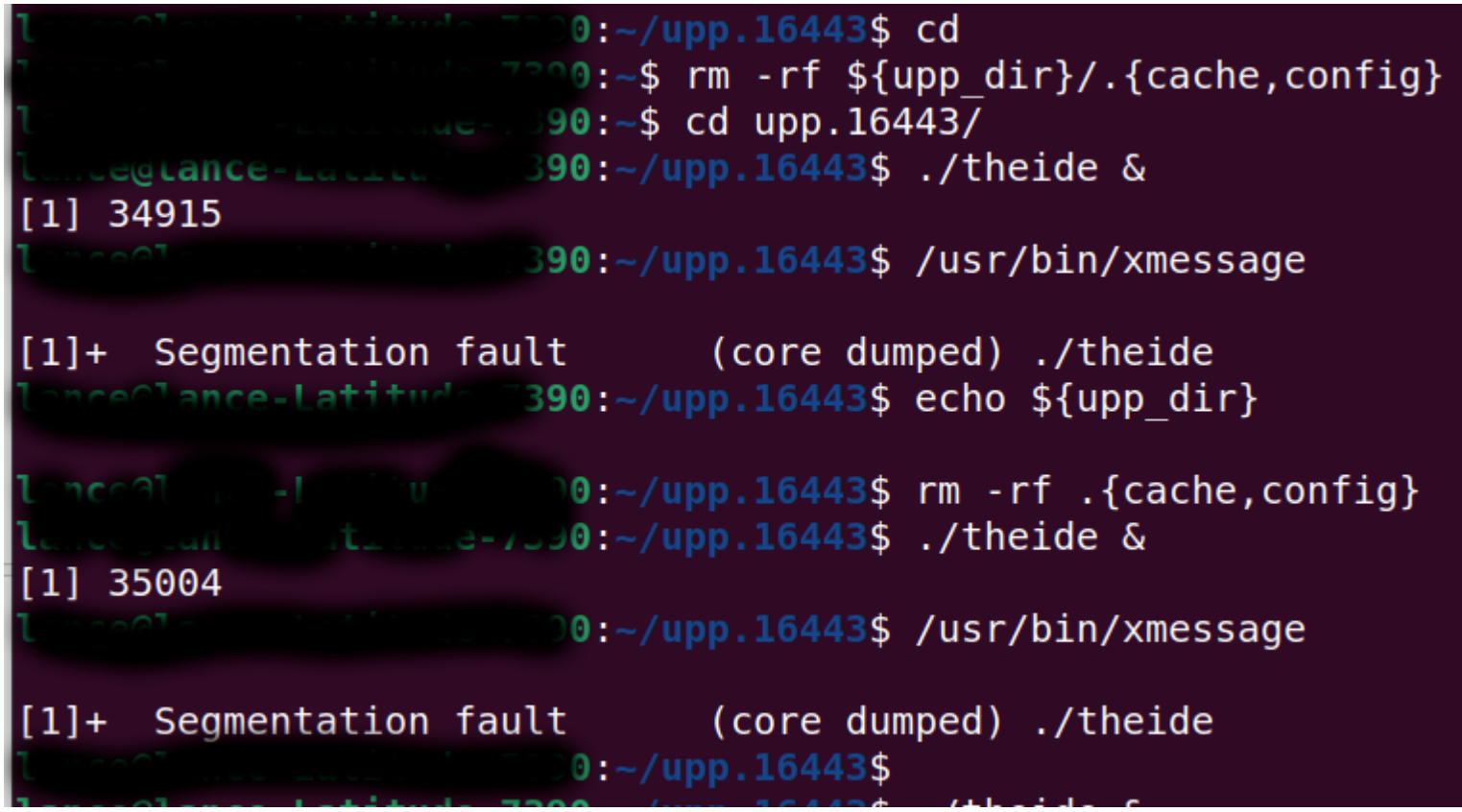
I know that in the past we eliminated some migration issues related to new clang implementation, but it seems that there is something from the old installation that is causing crash.

Klugier

Thank you, Klugier. It helped but only a bit. When I opened the examples/Days package after deleting cache&config, I observed right panel starting from blank and some class/function etc are populated and then it crashed again with the same error (invalid memory access).

File Attachments

1) libclang.png, downloaded 1496 times



Subject: Re: theide with libclang
Posted by Klugier on Sun, 25 Sep 2022 13:28:14 GMT
View Forum Message <> Reply to Message

Hello Lance,

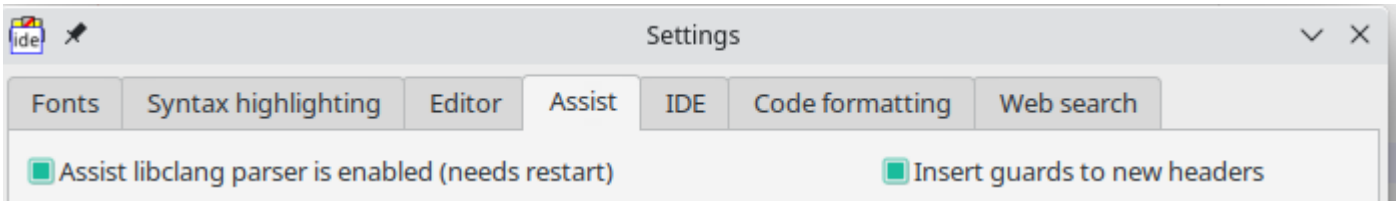
Could you open ide project and build it in debug mode. After that in TheIDE in debug mode (under debugger) open examples/Days package. Thanks to that we should be able to obtain callstack of the crash.

If you have trouble with opening uppsrc/ide package due to crash in TheIDE, disable clang parsing in Settings (Assist tab and first option from the left). Screenshot below:

Klugier

File Attachments

1) Assist.png, downloaded 1478 times



Subject: Re: theide with libclang
Posted by Lance on Sun, 25 Sep 2022 13:28:51 GMT
View Forum Message <> Reply to Message

Now I tested upp.16443 on a more recently installed machine that can be regarded as clean.

After installed and launched theide and opened Examples/Days in theide, I notice the right panel is blank and not changing. Presumably theide could not find libclang.so and no parsing were taking place.

Then I ran the jjacksonRIAB script
sudo bash -c "$(wget -O - https://apt.llvm.org/llvm.sh)"
to update/make sure the presence of libclang. After that I launched theide and examples/Days again, right panel remained blank. Then I delete ${upp_src}/.{config,cache}, launched theide & examples/Days, same blank. Now on this supposedly cleaner system, theide.16443 could no even locate libclang.so.

Subject: Re: theide with libclang
Posted by Lance on Sun, 25 Sep 2022 13:30:51 GMT
View Forum Message <> Reply to Message

Klugier wrote on Sun, 25 September 2022 09:28Hello Lance,

Could you open ide project and build it in debug mode. After that in TheIDE in debug mode (under debugger) open examples/Days package. Thanks to that we should be able to obtain callstack of the crash.

If you have trouble with opening uppsrc/ide package due to crash in TheIDE, disable clang parsing in Settings (Assist tab and first option from the left). Screenshot below:

Klugier

libclang.so obviously is successfully detected. In the generated theide, I opened examples/AddressBook, I can observe the right panel populated with parsed results, gradually,

starting from blank.

This time it took much longer for the generated theide to crash. Unfortunately nothing useful could be obtained from that/after that.

As you can see, I can either Wait(and it eventually bring me to the same screen) or Force Quit the generated debug-mode theide which end the generated debug-mode theide completely,leaving me no chance to check the call stack.

In the host theide(which creates the debug-mode theide), I have something like this...oh, I can not update more than one picture, anyways, it bears no information and should not be of any relevance anyways.

File Attachments

1) Screenshot from 2022-09-25 09-52-43.png, downloaded 1378 times



Subject: Re: theide with libclang
Posted by Klugier on Sun, 25 Sep 2022 13:47:13 GMT
View Forum Message <> Reply to Message

Hello Lance,

OK, I see where the problem is. We are dynamically loading libclang without notify the user about

the actions of this operation. It must change and in situation when we do not detect it we should show error and siabled parsing...

Here is the code in TheIDE (ide/main.cpp):

```
#ifdef PLATFORM_POSIX
void TryLoadLibClang() // Should return bool...
{
 String libdir = TrimBoth(Sys("llvm-config --libdir"));
 if(LoadLibClang(libdir + "/libclang.so"))
  return;
 if(LoadLibClang("/usr/lib/libclang.so"))
  return;
 for(int i = 20; i >= 10; i--)
  if(LoadLibClang("/usr/lib/llvm-" + AsString(i) + "/lib/libclang.so"))
   return;
}
#endif

#ifdef flagMAIN
GUI_APP_MAIN
#else
void AppMain___()
#endif
{
#ifdef PLATFORM_POSIX
 TryLoadLibClang(); // If it fails... just notrmal execution without any notification or error. However user will now if libclang is conffigured correctly?
#endif
	// Above operation fails and we are starting to use libclang, it can not end good...
```

The above code must be fixed and then you can try to run TheIDE. If you do not have correctly set libclang then you should see the error.

Edit: In general I assume that for your configuration there is a problem with finding libclang.so and lack of this library is causing crash.

Klugier

Subject: Re: theide with libclang
Posted by Lance on Sun, 25 Sep 2022 14:05:10 GMT
View Forum Message <> Reply to Message

Klugier wrote on Sun, 25 September 2022 09:47Hello Lance,

OK, I see where the problem is. We are dynamically loading libclang without notify the user about

the actions of this operation. It must change and in situation when we do not detect it we should show error and siabled parsing...

Here is the code in TheIDE (ide/main.cpp):

```
#ifdef PLATFORM_POSIX
void TryLoadLibClang() // Should return bool...
{
 String libdir = TrimBoth(Sys("llvm-config --libdir"));
 if(LoadLibClang(libdir + "/libclang.so"))
  return;
 if(LoadLibClang("/usr/lib/libclang.so"))
  return;
 for(int i = 20; i >= 10; i--)
  if(LoadLibClang("/usr/lib/llvm-" + AsString(i) + "/lib/libclang.so"))
   return;
}
#endif

#ifdef flagMAIN
GUI_APP_MAIN
#else
void AppMain___()
#endif
{
#ifdef PLATFORM_POSIX
 TryLoadLibClang(); // If it fails... just notrmal execution without any notification or error. However user will now if libclang is conffigured correctly?
#endif
	// Above operation fails and we are starting to use libclang, it can not end good...
```

The above code must be fixed and then you can try to run TheIDE. If you do not have correctly set libclang then you should see the error.

Edit: In general I assume that for your configuration there is a problem with finding libclang.so and lack of this library is causing crash.

Klugier

Hi Klugier,

I have updated my previous reply. Please let me know if it helps for me to try your revised code. Thanks!

Subject: Re: theide with libclang

Posted by mirek on Sun, 25 Sep 2022 14:57:59 GMT
View Forum Message <> Reply to Message

Klugier wrote on Sun, 25 September 2022 15:47

> // Above operation fails and we are starting to use libclang, it can not end good...

It is accounted for in the code, should continue working without assist.

That said, a prompt with information is a must here.

Mirek

Subject: Re: theide with libclang
Posted by jjacksonRIAB on Sun, 25 Sep 2022 19:50:48 GMT
View Forum Message <> Reply to Message

Lance wrote on Sun, 25 September 2022 15:30
As you can see, I can either Wait(and it eventually bring me to the same screen) or Force Quit the generated debug-mode theide which end the generated debug-mode theide completely,leaving me no chance to check the call stack.

Times like that you pretty much have no choice but to run gdb directly and get a call stack.

gdb <appname>
run

then when it crashes type bt for a backtrace
quit (or q) to quit.

I'm not an expert with gdb but it could get you further. It has a help section, a tui (if you run it with -tui) and I think they even added python scripting support a while ago.

Subject: Re: theide with libclang
Posted by mirek on Sun, 25 Sep 2022 21:23:41 GMT
View Forum Message <> Reply to Message

jjacksonRIAB wrote on Sat, 24 September 2022 11:10Mirek,

The problem appears to manifest with assist_cursor in Assist.cpp line 502.

It gathers all of the potential include files appropriately but assist_cursor never matches current pos so it returns before performing the completion.

And the whole time I thought the problem is with C++ autocomplete in header files....

Thanks, good catch.

Mirek

Subject: Re: theide with libclang
Posted by mirek on Sun, 25 Sep 2022 21:26:49 GMT
View Forum Message <> Reply to Message

jjacksonRIAB wrote on Sun, 25 September 2022 21:50Lance wrote on Sun, 25 September 2022 15:30
As you can see, I can either Wait(and it eventually bring me to the same screen) or Force Quit the generated debug-mode theide which end the generated debug-mode theide completely,leaving me no chance to check the call stack.

Times like that you pretty much have no choice but to run gdb directly and get a call stack.

gdb <appname>
run

then when it crashes type bt for a backtrace
quit (or q) to quit.

I'm not an expert with gdb but it could get you further. It has a help section, a tui (if you run it with -tui) and I think they even added python scripting support a while ago.

You can post the backtrace here...

Subject: Re: theide with libclang
Posted by Lance on Mon, 26 Sep 2022 23:48:09 GMT
View Forum Message <> Reply to Message

mirek wrote on Sun, 25 September 2022 17:26jjacksonRIAB wrote on Sun, 25 September 2022 21:50Lance wrote on Sun, 25 September 2022 15:30
As you can see, I can either Wait(and it eventually bring me to the same screen) or Force Quit the generated debug-mode theide which end the generated debug-mode theide completely,leaving me no chance to check the call stack.

[code]

Times like that you pretty much have no choice but to run gdb directly and get a call stack.

gdb <appname>
run

then when it crashes type bt for a backtrace
quit (or q) to quit.

I'm not an expert with gdb but it could get you further. It has a help section, a tui (if you run it with -tui) and I think they even added python scripting support a while ago.

You can post the backtrace here...

Reading symbols from ./ide...
(gdb) run
Starting program: /home/lance/.cache/upp.out/uppsrc/CLANG.Debug.Debug_Full.Gui.Shared/ide
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
[New Thread 0x7ffff3dc5640 (LWP 66838)]
[New Thread 0x7ffff35c4640 (LWP 66839)]
[New Thread 0x7fffebfff640 (LWP 66840)]
[New Thread 0x7fffeb7fe640 (LWP 66841)]
[New Thread 0x7fffeaffd640 (LWP 66842)]
[New Thread 0x7fffea7fc640 (LWP 66843)]
[New Thread 0x7fffe9ffb640 (LWP 66844)]
[New Thread 0x7fffe97fa640 (LWP 66845)]
[New Thread 0x7fffe8ff9640 (LWP 66846)]
[New Thread 0x7fffe3fff640 (LWP 66847)]
[New Thread 0x7fffe37fe640 (LWP 66848)]
[New Thread 0x7fffe2ffd640 (LWP 66849)]
[New Thread 0x7fffda1fd640 (LWP 66850)]
[New Thread 0x7fffd99fc640 (LWP 66851)]
[New Thread 0x7fffd91fb640 (LWP 66852)]
[New Thread 0x7fffd89fa640 (LWP 66854)]
[Thread 0x7fffd91fb640 (LWP 66852) exited]
[Thread 0x7fffd99fc640 (LWP 66851) exited]
[New Thread 0x7fffd99fc640 (LWP 66855)]
[New Thread 0x7fffd91fb640 (LWP 66856)]
[New Thread 0x7fffb77ff640 (LWP 66857)]
[New Thread 0x7fffb6ffe640 (LWP 66858)]
[New Thread 0x7fffb67fd640 (LWP 66859)]
[New Thread 0x7fffb5ffc640 (LWP 66860)]
[New Thread 0x7fffb57fb640 (LWP 66861)]
[Detaching after fork from child process 66862]

```
[Detaching after fork from child process 66863]
[Detaching after fork from child process 66864]
[Detaching after fork from child process 66865]
[Detaching after fork from child process 66866]
[Detaching after fork from child process 66867]
[Detaching after fork from child process 66868]
[Detaching after fork from child process 66869]
[Detaching after fork from child process 66870]
[Detaching after fork from child process 66871]
[Detaching after fork from child process 66872]
[Detaching after fork from child process 66873]
[Detaching after fork from child process 66874]
[Detaching after fork from child process 66875]
[New Thread 0x7fffb4ffa640 (LWP 66876)]
[New Thread 0x7fff9ffff640 (LWP 66877)]
[New Thread 0x7fff8f77e640 (LWP 66879)]
[New Thread 0x7fff8ef7d640 (LWP 66880)]
[New Thread 0x7fff8e77c640 (LWP 66881)]
[New Thread 0x7fff8df7b640 (LWP 66882)]
[New Thread 0x7fff89c7a640 (LWP 66883)]
[New Thread 0x7fff89479640 (LWP 66884)]
[Thread 0x7fff9ffff640 (LWP 66877) exited]
[Thread 0x7fffda1fd640 (LWP 66850) exited]
[Thread 0x7fff8f77e640 (LWP 66879) exited]
[Thread 0x7fff8e77c640 (LWP 66881) exited]
[Thread 0x7fff8df7b640 (LWP 66882) exited]
[Thread 0x7fff89c7a640 (LWP 66883) exited]
[Thread 0x7fff89479640 (LWP 66884) exited]
[Thread 0x7fff8ef7d640 (LWP 66880) exited]
[New Thread 0x7fff8ef7d640 (LWP 66888)]
[New Thread 0x7fff89479640 (LWP 66889)]

Thread 21 "ide" received signal SIGSEGV, Segmentation fault.
[Switching to Thread 0x7fffb6ffe640 (LWP 66858)]
0x00007fffe122271b in ?? () from /usr/lib/llvm-14/lib/libclang.so
(gdb) bt
#0  0x00007fffe122271b in  () at /usr/lib/llvm-14/lib/libclang.so
(gdb)
```

Subject: Re: theide with libclang
Posted by Klugier on Tue, 27 Sep 2022 18:35:00 GMT
View Forum Message <> Reply to Message

Hello Lance,

I merged the changes related to display error prompt when libclan has not been detected. In your case it won't help, because your instance of theide has access to this library.

In order to have more reasonable callstack, you can build the newest TheIDE in debug mode with the older one that works on previos Assist++ version. For example to do it you can use TheIDE from the latest release 2022.2.

Klugier

Subject: Re: theide with libclang
Posted by Lance on Tue, 27 Sep 2022 21:11:16 GMT
View Forum Message <> Reply to Message

Klugier wrote on Tue, 27 September 2022 14:35Hello Lance,

I merged the changes related to display error prompt when libclan has not been detected. In your case it won't help, because your instance of theide has access to this library.

In order to have more reasonable callstack, you can build the newest TheIDE in debug mode with the older one that works on previos Assist++ version. For example to do it you can use TheIDE from the latest release 2022.2.

Klugier

Hi Klugier,

Thanks for the reply. I tried your method with upp version 16303 whose Assist++ are still working fine on my computer with same result. When the generated ide failed with segmentation fault, I was faced with the same Force Quit or Wait choices.

I will try it with the upp version you suggested. By the way, the version of uppsrc I used is 16446.

Regards,
Lance

Same result with upp 2022.2 (version 16270).

Subject: Re: theide with libclang
Posted by Lance on Tue, 27 Sep 2022 21:12:30 GMT
View Forum Message <> Reply to Message

jjacksonRIAB wrote on Sun, 25 September 2022 15:50Lance wrote on Sun, 25 September 2022 15:30
As you can see, I can either Wait(and it eventually bring me to the same screen) or Force Quit the

generated debug-mode theide which end the generated debug-mode theide completely,leaving me no chance to check the call stack.

Times like that you pretty much have no choice but to run gdb directly and get a call stack.

gdb <appname>
run

then when it crashes type bt for a backtrace
quit (or q) to quit.

I'm not an expert with gdb but it could get you further. It has a help section, a tui (if you run it with -tui) and I think they even added python scripting support a while ago.

I forgot to thank you for your reply

Subject: Re: theide with libclang
Posted by Tom1 on Sat, 01 Oct 2022 16:46:07 GMT
View Forum Message <> Reply to Message

Hi,

While, everything seems quite smooth in Windows, I tried to run this in Linux Mint 21 with 16G RAM. Every time it crashes with segmentation fault a few seconds after opening the main package. I tried to run it in gdb with the following results:
Starting program: /home/tom/out/uppsrc/CLANG.Debug.Debug_Full.Gui.Shared/ide
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
[New Thread 0x7ffff1eaf640 (LWP 6714)]
[New Thread 0x7ffff16ae640 (LWP 6715)]
[New Thread 0x7fffebfff640 (LWP 6716)]
[New Thread 0x7fffeb7fe640 (LWP 6717)]
[New Thread 0x7fffeaffd640 (LWP 6718)]
[New Thread 0x7fffea7fc640 (LWP 6719)]
[New Thread 0x7fffe9ffb640 (LWP 6720)]
[New Thread 0x7fffe97fa640 (LWP 6721)]
[New Thread 0x7fffe8ff9640 (LWP 6722)]
[New Thread 0x7fffe3fff640 (LWP 6723)]
[New Thread 0x7fffe37fe640 (LWP 6724)]
[New Thread 0x7fffe2ffd640 (LWP 6725)]
[New Thread 0x7fffd91fd640 (LWP 6726)]
[New Thread 0x7fffd89fc640 (LWP 6728)]
[New Thread 0x7fffd1fff640 (LWP 6729)]
[New Thread 0x7fffd17fe640 (LWP 6730)]
[New Thread 0x7fffd0ffd640 (LWP 6731)]
[New Thread 0x7fffc3fff640 (LWP 6732)]

```
[New Thread 0x7fffc37fe640 (LWP 6733)]
[New Thread 0x7fffc2ffd640 (LWP 6734)]
[Detaching after fork from child process 6735]
[Detaching after fork from child process 6736]
[Detaching after fork from child process 6737]
[Detaching after fork from child process 6738]
[Detaching after fork from child process 6739]
[Detaching after fork from child process 6740]
[Detaching after fork from child process 6741]
[Detaching after fork from child process 6742]
[Detaching after fork from child process 6743]
[Detaching after fork from child process 6744]
[Detaching after fork from child process 6745]
[Detaching after fork from child process 6746]
[Detaching after fork from child process 6747]
[Detaching after fork from child process 6748]
[New Thread 0x7fffc27fc640 (LWP 6749)]
[New Thread 0x7fffc1ffb640 (LWP 6750)]
[Thread 0x7fffd91fd640 (LWP 6726) exited]
[Thread 0x7fffc1ffb640 (LWP 6750) exited]
[New Thread 0x7fffc1ffb640 (LWP 6751)]
[New Thread 0x7fffd91fd640 (LWP 6752)]
[New Thread 0x7fffa29ff640 (LWP 6753)]
[New Thread 0x7fffa21fe640 (LWP 6754)]
[New Thread 0x7fffa19fd640 (LWP 6755)]
[New Thread 0x7fff9dd2b640 (LWP 6756)]
[Thread 0x7fffd91fd640 (LWP 6752) exited]
[New Thread 0x7fffd91fd640 (LWP 6757)]
[Thread 0x7fff9dd2b640 (LWP 6756) exited]
[Thread 0x7fffa29ff640 (LWP 6753) exited]
[New Thread 0x7fffa29ff640 (LWP 6758)]
[Thread 0x7fffa21fe640 (LWP 6754) exited]
[Thread 0x7fffa19fd640 (LWP 6755) exited]
[Thread 0x7fffc1ffb640 (LWP 6751) exited]
[New Thread 0x7fffc1ffb640 (LWP 6759)]
[New Thread 0x7fffa19fd640 (LWP 6760)]
[New Thread 0x7fffa21fe640 (LWP 6761)]
[New Thread 0x7fff9dd2b640 (LWP 6762)]
[Thread 0x7fffd91fd640 (LWP 6757) exited]
[New Thread 0x7fffd91fd640 (LWP 6763)]
[Thread 0x7fffa29ff640 (LWP 6758) exited]
[New Thread 0x7fffa29ff640 (LWP 6764)]
[Thread 0x7fffa19fd640 (LWP 6760) exited]
[Thread 0x7fff9dd2b640 (LWP 6762) exited]
[Thread 0x7fffc1ffb640 (LWP 6759) exited]
[Thread 0x7fffa21fe640 (LWP 6761) exited]
[Thread 0x7fffd91fd640 (LWP 6763) exited]
[Thread 0x7fffa29ff640 (LWP 6764) exited]
```

```
[New Thread 0x7fffa29ff640 (LWP 6765)]
[New Thread 0x7fffd91fd640 (LWP 6766)]
[New Thread 0x7fffa21fe640 (LWP 6767)]
[New Thread 0x7fffc1ffb640 (LWP 6768)]
[New Thread 0x7fffa19fd640 (LWP 6769)]
[New Thread 0x7fff9dd2b640 (LWP 6770)]
[Thread 0x7fffa29ff640 (LWP 6765) exited]
[Thread 0x7fffa21fe640 (LWP 6767) exited]
[Thread 0x7fffd91fd640 (LWP 6766) exited]
[Thread 0x7fff9dd2b640 (LWP 6770) exited]
[Thread 0x7fffc1ffb640 (LWP 6768) exited]
[New Thread 0x7fffc1ffb640 (LWP 6771)]
[Thread 0x7fffc1ffb640 (LWP 6771) exited]
[New Thread 0x7fffc1ffb640 (LWP 6772)]
[Thread 0x7fffc1ffb640 (LWP 6772) exited]
[New Thread 0x7fffc1ffb640 (LWP 6773)]
[Thread 0x7fffa19fd640 (LWP 6769) exited]
[New Thread 0x7fffa19fd640 (LWP 6774)]
[New Thread 0x7fff9dd2b640 (LWP 6775)]
[New Thread 0x7fffd91fd640 (LWP 6776)]
[New Thread 0x7fffa21fe640 (LWP 6777)]
[Thread 0x7fffa21fe640 (LWP 6777) exited]
[New Thread 0x7fffa21fe640 (LWP 6778)]
[Thread 0x7fffa21fe640 (LWP 6778) exited]
[New Thread 0x7fffa21fe640 (LWP 6779)]
[Thread 0x7fffa21fe640 (LWP 6779) exited]
[New Thread 0x7fffa21fe640 (LWP 6780)]
[Thread 0x7fffa21fe640 (LWP 6780) exited]
[New Thread 0x7fffa21fe640 (LWP 6781)]
[Thread 0x7fffa21fe640 (LWP 6781) exited]
[New Thread 0x7fffa21fe640 (LWP 6782)]
[Thread 0x7fffa21fe640 (LWP 6782) exited]
[New Thread 0x7fffa21fe640 (LWP 6783)]
[Thread 0x7fffa21fe640 (LWP 6783) exited]
[New Thread 0x7fffa21fe640 (LWP 6784)]
[Thread 0x7fffa21fe640 (LWP 6784) exited]
[New Thread 0x7fffa21fe640 (LWP 6785)]
[Thread 0x7fffa21fe640 (LWP 6785) exited]
[New Thread 0x7fffa21fe640 (LWP 6786)]
[Thread 0x7fffa21fe640 (LWP 6786) exited]
[New Thread 0x7fffa21fe640 (LWP 6787)]
[Thread 0x7fffa21fe640 (LWP 6787) exited]
[New Thread 0x7fffa21fe640 (LWP 6788)]
[Thread 0x7fffa21fe640 (LWP 6788) exited]
[New Thread 0x7fffa21fe640 (LWP 6789)]
[Thread 0x7fffa21fe640 (LWP 6789) exited]
[New Thread 0x7fffa21fe640 (LWP 6790)]
[Thread 0x7fffa21fe640 (LWP 6790) exited]
```

[New Thread 0x7fffa21fe640 (LWP 6791)]
[Thread 0x7fffa21fe640 (LWP 6791) exited]
[New Thread 0x7fffa21fe640 (LWP 6792)]
[New Thread 0x7fffa29ff640 (LWP 6793)]
[Thread 0x7fffd91fd640 (LWP 6776) exited]
[Thread 0x7fff9dd2b640 (LWP 6775) exited]
[Thread 0x7fffa19fd640 (LWP 6774) exited]
[Thread 0x7fffa21fe640 (LWP 6792) exited]
[New Thread 0x7fffa21fe640 (LWP 6794)]
[New Thread 0x7fffa19fd640 (LWP 6795)]
[New Thread 0x7fff9dd2b640 (LWP 6796)]
[Thread 0x7fffa29ff640 (LWP 6793) exited]
[New Thread 0x7fffa29ff640 (LWP 6797)]
[Thread 0x7fffa29ff640 (LWP 6797) exited]
[New Thread 0x7fffa29ff640 (LWP 6798)]
[Thread 0x7fffa29ff640 (LWP 6798) exited]
[New Thread 0x7fffa29ff640 (LWP 6799)]
[Thread 0x7fffa29ff640 (LWP 6799) exited]
[New Thread 0x7fffa29ff640 (LWP 6800)]
[Thread 0x7fffa29ff640 (LWP 6800) exited]
[New Thread 0x7fffa29ff640 (LWP 6801)]
[Thread 0x7fffa29ff640 (LWP 6801) exited]
[New Thread 0x7fffa29ff640 (LWP 6802)]
[Thread 0x7fffa29ff640 (LWP 6802) exited]
[New Thread 0x7fffa29ff640 (LWP 6803)]
[Thread 0x7fffa29ff640 (LWP 6803) exited]
[New Thread 0x7fffa29ff640 (LWP 6804)]
[Thread 0x7fffa29ff640 (LWP 6804) exited]
[New Thread 0x7fffa29ff640 (LWP 6805)]
[Thread 0x7fffa29ff640 (LWP 6805) exited]
[New Thread 0x7fffa29ff640 (LWP 6806)]
[Thread 0x7fffa29ff640 (LWP 6806) exited]
[New Thread 0x7fffa29ff640 (LWP 6807)]
[Thread 0x7fffa29ff640 (LWP 6807) exited]
[New Thread 0x7fffa29ff640 (LWP 6808)]
[Thread 0x7fffa29ff640 (LWP 6808) exited]
[New Thread 0x7fffa29ff640 (LWP 6809)]
[Thread 0x7fffa29ff640 (LWP 6809) exited]
[New Thread 0x7fffa29ff640 (LWP 6810)]
[Thread 0x7fffa29ff640 (LWP 6810) exited]
[New Thread 0x7fffa29ff640 (LWP 6811)]
[Thread 0x7fffa29ff640 (LWP 6811) exited]
[New Thread 0x7fffa29ff640 (LWP 6812)]
[Thread 0x7fffa29ff640 (LWP 6812) exited]
[New Thread 0x7fffa29ff640 (LWP 6813)]
[Thread 0x7fffa29ff640 (LWP 6813) exited]
[New Thread 0x7fffa29ff640 (LWP 6814)]
[Thread 0x7fffa29ff640 (LWP 6814) exited]

```
[New Thread 0x7fffa29ff640 (LWP 6815)]
[Thread 0x7fffa29ff640 (LWP 6815) exited]
[New Thread 0x7fffa29ff640 (LWP 6816)]
[Thread 0x7fffa29ff640 (LWP 6816) exited]
[New Thread 0x7fffa29ff640 (LWP 6817)]
[Thread 0x7fffa29ff640 (LWP 6817) exited]
[New Thread 0x7fffa29ff640 (LWP 6818)]
[Thread 0x7fffa29ff640 (LWP 6818) exited]
[New Thread 0x7fffa29ff640 (LWP 6819)]
[Thread 0x7fffa29ff640 (LWP 6819) exited]
[New Thread 0x7fffa29ff640 (LWP 6820)]
[Thread 0x7fffa29ff640 (LWP 6820) exited]
[New Thread 0x7fffa29ff640 (LWP 6821)]
[Thread 0x7fffa29ff640 (LWP 6821) exited]
[New Thread 0x7fffa29ff640 (LWP 6822)]
[Thread 0x7fffc1ffb640 (LWP 6773) exited]
[Thread 0x7fffa29ff640 (LWP 6822) exited]
[New Thread 0x7fffa29ff640 (LWP 6823)]
[Thread 0x7fffa29ff640 (LWP 6823) exited]
[New Thread 0x7fffa29ff640 (LWP 6824)]
[Thread 0x7fffa29ff640 (LWP 6824) exited]
[New Thread 0x7fffa29ff640 (LWP 6825)]
[Thread 0x7fffa29ff640 (LWP 6825) exited]
[New Thread 0x7fffa29ff640 (LWP 6826)]
[Thread 0x7fffa29ff640 (LWP 6826) exited]
[New Thread 0x7fffa29ff640 (LWP 6827)]
[Thread 0x7fffa29ff640 (LWP 6827) exited]
[New Thread 0x7fffa29ff640 (LWP 6828)]
[New Thread 0x7fffc1ffb640 (LWP 6829)]
[Thread 0x7fffa29ff640 (LWP 6828) exited]
[New Thread 0x7fffa29ff640 (LWP 6830)]
[Thread 0x7fffc1ffb640 (LWP 6829) exited]
[New Thread 0x7fffc1ffb640 (LWP 6831)]
[Thread 0x7fffa29ff640 (LWP 6830) exited]
[New Thread 0x7fffa29ff640 (LWP 6832)]
[Thread 0x7fffc1ffb640 (LWP 6831) exited]
[New Thread 0x7fffc1ffb640 (LWP 6833)]
[Thread 0x7fffa29ff640 (LWP 6832) exited]
[New Thread 0x7fffa29ff640 (LWP 6834)]
[Thread 0x7fffc1ffb640 (LWP 6833) exited]
[New Thread 0x7fffc1ffb640 (LWP 6835)]
[Thread 0x7fffa29ff640 (LWP 6834) exited]
[New Thread 0x7fffa29ff640 (LWP 6836)]
[Thread 0x7fffc1ffb640 (LWP 6835) exited]
[New Thread 0x7fffc1ffb640 (LWP 6837)]
[Thread 0x7fffa29ff640 (LWP 6836) exited]
[Thread 0x7fffc1ffb640 (LWP 6837) exited]
[New Thread 0x7fffc1ffb640 (LWP 6838)]
```

```
[New Thread 0x7fffa29ff640 (LWP 6839)]
[Thread 0x7fffc1ffb640 (LWP 6838) exited]
[Thread 0x7fffa29ff640 (LWP 6839) exited]
[New Thread 0x7fffa29ff640 (LWP 6840)]
[New Thread 0x7fffc1ffb640 (LWP 6841)]
[Thread 0x7fffa29ff640 (LWP 6840) exited]
[Thread 0x7fffc1ffb640 (LWP 6841) exited]
[New Thread 0x7fffc1ffb640 (LWP 6842)]
[New Thread 0x7fffa29ff640 (LWP 6843)]
[Thread 0x7fffa29ff640 (LWP 6843) exited]
[Thread 0x7fffc1ffb640 (LWP 6842) exited]
[New Thread 0x7fffc1ffb640 (LWP 6844)]
[New Thread 0x7fffa29ff640 (LWP 6845)]
[Thread 0x7fffc1ffb640 (LWP 6844) exited]
[Thread 0x7fffa29ff640 (LWP 6845) exited]
[New Thread 0x7fffa29ff640 (LWP 6846)]
[New Thread 0x7fffc1ffb640 (LWP 6847)]
[Thread 0x7fffa29ff640 (LWP 6846) exited]
[Thread 0x7fffc1ffb640 (LWP 6847) exited]
[New Thread 0x7fffc1ffb640 (LWP 6848)]
[New Thread 0x7fffa29ff640 (LWP 6849)]
[Thread 0x7fffa29ff640 (LWP 6849) exited]
[Thread 0x7fffc1ffb640 (LWP 6848) exited]
[New Thread 0x7fffa29ff640 (LWP 6850)]
[New Thread 0x7fffc1ffb640 (LWP 6851)]
[Thread 0x7fffa29ff640 (LWP 6850) exited]
[New Thread 0x7fffa29ff640 (LWP 6852)]
[Thread 0x7fffc1ffb640 (LWP 6851) exited]
[New Thread 0x7fffc1ffb640 (LWP 6853)]
[Thread 0x7fffa29ff640 (LWP 6852) exited]
[New Thread 0x7fffa29ff640 (LWP 6854)]
[Thread 0x7fffc1ffb640 (LWP 6853) exited]
[New Thread 0x7fffc1ffb640 (LWP 6855)]
[Thread 0x7fffa29ff640 (LWP 6854) exited]
[New Thread 0x7fffa29ff640 (LWP 6856)]
[Thread 0x7fffc1ffb640 (LWP 6855) exited]
[New Thread 0x7fffc1ffb640 (LWP 6857)]

Thread 19 "ide" received signal SIGSEGV, Segmentation fault.
[Switching to Thread 0x7fffc3fff640 (LWP 6732)]
0x00007fffe122271b in ?? () from /usr/lib/llvm-14/lib/libclang.so
(gdb)
```

The above does not say much to me.
Anyway, uninstalling libclang allows me to run ide, so I can compile new versions for testing...

Best regards,

Tom

Subject: Re: theide with libclang
Posted by Klugier on Sat, 01 Oct 2022 20:49:24 GMT
View Forum Message <> Reply to Message

Hello Tom

You need to have TheIDE in debug mode in order to have reasonable callstack, release mode doesn't posses debug symbols. There are multiple ways to do, you can build newer ide in debug mode with the older one.

Klugier

Subject: Re: theide with libclang
Posted by Lance on Sun, 02 Oct 2022 00:19:39 GMT
View Forum Message <> Reply to Message

I did a clean installation of Ubuntu 22.04 LTS. After update and upgrade, first thing I did was download and ./Install UPP version 16450.

Turns out

```
bool LoadLibClangAutomatically()
{
 String libdir = TrimBoth(Sys("llvm-config --libdir"));
 if(LoadLibClang(libdir + "/libclang.so")) {
  return true;
 }
 if(LoadLibClang("/usr/lib/libclang.so")) {
  return true;
 }
 for(int i = 20; i >= 10; i--) {
  if(LoadLibClang("/usr/lib/llvm-" + AsString(i) + "/lib/libclang.so")) {
   return true;
  }
 }

 return false;
}
```

failed even though a libclang is present in the system.

After I made the following change, libclang can be successfully located. It seems some expected symlinks were not present in the system.

```
bool LoadLibClangAutomatically()
```

```
{
 String libdir = TrimBoth(Sys("llvm-config-14 --libdir"));
 if(LoadLibClang(libdir + "/libclang.so.1")) {
  return true;
 }
 if(LoadLibClang("/usr/lib/libclang.so")) {
  return true;
 }
 for(int i = 20; i >= 10; i--) {
  if(LoadLibClang("/usr/lib/llvm-" + AsString(i) + "/lib/libclang.so")) {
   return true;
  }
 }

 return false;
}
```

After that, Assist++ is up and running, parsing the sources and populating the right panel with class/function/etc info. I was even successful to invoke code completion with object name following by a dot(.). It went much further than it used to. Unfortunately segmentation fault follows. Yes, it's built in debug mode. No, gdb, run & bt would not produce anything useful, essentially very similar to what Tom and I have posted.

Subject: Re: theide with libclang
Posted by mirek on Sun, 02 Oct 2022 09:57:21 GMT
View Forum Message <> Reply to Message

Lance wrote on Sun, 02 October 2022 02:19I did a clean installation of Ubuntu 22.04 LTS. After update and upgrade, first thing I did was download and ./Install UPP version 16450.

After that, Assist++ is up and running, parsing the sources and populating the right panel with class/function/etc info. I was even successful to invoke code completion with object name following by a dot(.). It went much further than it used to. Unfortunately segmentation fault follows. Yes, it's built in debug mode. No, gdb, run & bt would not produce anything useful, essentially very similar to what Tom and I have posted.

One thing worth testing: try with USEMALLOC...

(I will try to look into this mess ASAP).

Subject: Re: theide with libclang

Posted by Lance on Sun, 02 Oct 2022 13:27:13 GMT
View Forum Message <> Reply to Message

mirek wrote on Sun, 02 October 2022 05:57Lance wrote on Sun, 02 October 2022 02:19I did a clean installation of Ubuntu 22.04 LTS. After update and upgrade, first thing I did was download and ./Install UPP version 16450.

After that, Assist++ is up and running, parsing the sources and populating the right panel with class/function/etc info. I was even successful to invoke code completion with object name following by a dot(.). It went much further than it used to. Unfortunately segmentation fault follows. Yes, it's built in debug mode. No, gdb, run & bt would not produce anything useful, essentially very similar to what Tom and I have posted.

One thing worth testing: try with USEMALLOC...

(I will try to look into this mess ASAP).

I wasn't able to notice anything different with USEMALLOC defined.

Speed-wise, new Assist++ is reasonably OK. On my old laptop, in debug mode, I can notice a delay of a few seconds (code completion), but on new hardware and release mode, it should be quite smooth.

Subject: Re: theide with libclang
Posted by mirek on Sun, 02 Oct 2022 14:17:46 GMT
View Forum Message <> Reply to Message

Lance wrote on Sun, 02 October 2022 15:27mirek wrote on Sun, 02 October 2022 05:57Lance wrote on Sun, 02 October 2022 02:19I did a clean installation of Ubuntu 22.04 LTS. After update and upgrade, first thing I did was download and ./Install UPP version 16450.

After that, Assist++ is up and running, parsing the sources and populating the right panel with class/function/etc info. I was even successful to invoke code completion with object name following by a dot(.). It went much further than it used to. Unfortunately segmentation fault follows. Yes, it's built in debug mode. No, gdb, run & bt would not produce anything useful, essentially very similar to what Tom and I have posted.

One thing worth testing: try with USEMALLOC...

(I will try to look into this mess ASAP).

I wasn't able to notice anything different with USEMALLOC defined.

Excellent, good info, thanks.

Subject: Re: theide with libclang
Posted by Tom1 on Sun, 02 Oct 2022 17:03:16 GMT
View Forum Message <> Reply to Message

Hi,

The same here, no help from USEMALLOC on Linux Mint 21.

As I just updated from GIT, I received also compilation issues:

```
----- ide/clang ( GUI CLANG SHARED POSIX LINUX ) (22 / 40)
libclang.cpp
clang.cpp
/home/tom/upp.src/uppsrc/ide/clang/clang.cpp:181:3: error: unknown type name
'CXDiagnosticSet'; did you mean 'CXDiagnostic'?
        CXDiagnosticSet set = clang_getChildDiagnostics(diagnostic);
        ^~~~~~~~~~~~~~~
        CXDiagnostic
/home/tom/upp.src/uppsrc/ide/clang/libclang.h:1919:15: note: 'CXDiagnostic' declared here
typedef void *CXDiagnostic;
          ^
/home/tom/upp.src/uppsrc/ide/clang/clang.cpp:181:25: error: use of undeclared identifier
'clang_getChildDiagnostics'; did you mean 'clang_getNumDiagnostics'?
        CXDiagnosticSet set = clang_getChildDiagnostics(diagnostic);
                              ^~~~~~~~~~~~~~~~~~~~~~~~~
                              clang_getNumDiagnostics
/home/tom/upp.src/uppsrc/ide/clang/libclang.h:1992:10: note: 'clang_getNumDiagnostics' declared
here
unsigned clang_getNumDiagnostics(CXTranslationUnit Unit);
       ^
/home/tom/upp.src/uppsrc/ide/clang/clang.cpp:181:51: error: cannot initialize a parameter of type
'CXTranslationUnit' (aka 'CXTranslationUnitImpl *') with an lvalue of type 'CXDiagnostic' (aka 'void
*')
        CXDiagnosticSet set = clang_getChildDiagnostics(diagnostic);
                                                        ^~~~~~~~~~
/home/tom/upp.src/uppsrc/ide/clang/libclang.h:1992:52: note: passing argument to parameter
'Unit' here
unsigned clang_getNumDiagnostics(CXTranslationUnit Unit);
                                 ^
/home/tom/upp.src/uppsrc/ide/clang/clang.cpp:182:11: error: use of undeclared identifier
'clang_getNumDiagnosticsInSet'; did you mean 'clang_getNumDiagnostics'?
        int n = clang_getNumDiagnosticsInSet(set);
            ^~~~~~~~~~~~~~~~~~~~~~~~~~~~
            clang_getNumDiagnostics
```

```
/home/tom/upp.src/uppsrc/ide/clang/libclang.h:1992:10: note: 'clang_getNumDiagnostics' declared
here
unsigned clang_getNumDiagnostics(CXTranslationUnit Unit);
         ^
/home/tom/upp.src/uppsrc/ide/clang/clang.cpp:182:40: error: cannot initialize a parameter of type
'CXTranslationUnit' (aka 'CXTranslationUnitImpl *') with an lvalue of type 'CXDiagnostic' (aka 'void
*')
                int n = clang_getNumDiagnosticsInSet(set);
                                                     ^~~
/home/tom/upp.src/uppsrc/ide/clang/libclang.h:1992:52: note: passing argument to parameter
'Unit' here
unsigned clang_getNumDiagnostics(CXTranslationUnit Unit);
                                                   ^
/home/tom/upp.src/uppsrc/ide/clang/clang.cpp:184:21: error: use of undeclared identifier
'clang_getDiagnosticInSet'; did you mean 'clang_getDiagnostic'?
                        CXDiagnostic d = clang_getDiagnosticInSet(set, i);
                                         ^~~~~~~~~~~~~~~~~~~~~~~~
                                         clang_getDiagnostic
/home/tom/upp.src/uppsrc/ide/clang/libclang.h:1993:14: note: 'clang_getDiagnostic' declared here
CXDiagnostic clang_getDiagnostic(CXTranslationUnit Unit, unsigned Index);
             ^
/home/tom/upp.src/uppsrc/ide/clang/clang.cpp:184:46: error: cannot initialize a parameter of type
'CXTranslationUnit' (aka 'CXTranslationUnitImpl *') with an lvalue of type 'CXDiagnostic' (aka 'void
*')
                        CXDiagnostic d = clang_getDiagnosticInSet(set, i);
                                                                  ^~~
/home/tom/upp.src/uppsrc/ide/clang/libclang.h:1993:52: note: passing argument to parameter
'Unit' here
CXDiagnostic clang_getDiagnostic(CXTranslationUnit Unit, unsigned Index);
                                                   ^
7 errors generated.
/home/tom/upp.src/uppsrc/ide/clang/libclang.cpp:242:1: error: unknown type name
'CXDiagnosticSet'; did you mean 'CXDiagnostic'?
CXDiagnosticSet clang_getChildDiagnostics(CXDiagnostic d)
^~~~~~~~~~~~~~
CXDiagnostic
/home/tom/upp.src/uppsrc/ide/clang/libclang.h:1919:15: note: 'CXDiagnostic' declared here
typedef void *CXDiagnostic;
             ^
/home/tom/upp.src/uppsrc/ide/clang/libclang.cpp:244:20: error: no member named
'clang_getChildDiagnostics' in 'T_LibClang'; did you mean 'clang_disposeDiagnostic'?
        return LibClang().clang_getChildDiagnostics(d);
                          ^~~~~~~~~~~~~~~~~~~~~~~~~
                          clang_disposeDiagnostic
/home/tom/upp.src/uppsrc/ide/clang/clang.dli:50:10: note: 'clang_disposeDiagnostic' declared
here
FN(void, clang_disposeDiagnostic, (CXDiagnostic Diagnostic))
         ^
```

```
/home/tom/upp.src/uppsrc/ide/clang/libclang.cpp:244:9: error: cannot initialize return object of type
'CXDiagnostic' (aka 'void *') with an rvalue of type 'void'
        return LibClang().clang_getChildDiagnostics(d);
               ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
/home/tom/upp.src/uppsrc/ide/clang/libclang.cpp:247:39: error: unknown type name
'CXDiagnosticSet'; did you mean 'CXDiagnostic'?
unsigned clang_getNumDiagnosticsInSet(CXDiagnosticSet diags)
                                      ^~~~~~~~~~~~~~~
                                      CXDiagnostic
/home/tom/upp.src/uppsrc/ide/clang/libclang.h:1919:15: note: 'CXDiagnostic' declared here
typedef void *CXDiagnostic;
              ^
/home/tom/upp.src/uppsrc/ide/clang/libclang.cpp:249:20: error: no member named
'clang_getNumDiagnosticsInSet' in 'T_LibClang'
        return LibClang().clang_getNumDiagnosticsInSet(diags);
               ~~~~~~~~~~ ^
/home/tom/upp.src/uppsrc/ide/clang/libclang.cpp:252:39: error: unknown type name
'CXDiagnosticSet'; did you mean 'CXDiagnostic'?
CXDiagnostic clang_getDiagnosticInSet(CXDiagnosticSet diags, unsigned index)
                                      ^~~~~~~~~~~~~~~
                                      CXDiagnostic
/home/tom/upp.src/uppsrc/ide/clang/libclang.h:1919:15: note: 'CXDiagnostic' declared here
typedef void *CXDiagnostic;
              ^
/home/tom/upp.src/uppsrc/ide/clang/libclang.cpp:254:20: error: no member named
'clang_getDiagnosticInSet' in 'T_LibClang'
        return LibClang().clang_getDiagnosticInSet(diags, index);
               ~~~~~~~~~~ ^
7 errors generated.
ide/clang: 2 file(s) built in (0:03.45), 1726 msecs / file

There were errors. (0:03.48)
```

Best regards,

Tom

Subject: Re: theide with libclang
Posted by mirek on Sun, 02 Oct 2022 20:07:54 GMT
View Forum Message <> Reply to Message

I have made some additions trying to ease (and help resolve) current problems:

- there is now --noclang commandline option that simply switches libclang loading off. should help with those crashes without uninstalling libclang
- the path of libclang loaded is written to theide log
- there are also improved message boxes informing about missing libclang (or that it is now

detected if it was missing)

- most importantly:

One of likely culprits here is dynamic loading. I would like you to test with linked libclang. In order to make it easier, I have added another theide mainconfig with new "LCLANG" that forces the "normal" linking of libclang. Downside is that you probably will have to manually alter build method with include and library paths (on mine, it is /usr/lib/llvm-10/include and /usr/lib/llvm-10/lib). Can you (Tom1 and Lance) test this for me?

Subject: Re: theide with libclang
Posted by Lance on Sun, 02 Oct 2022 22:58:01 GMT
View Forum Message <> Reply to Message

To Mirek:

Same result. Crashed with sigmentation fault. Initial parsing (with a dialog box prompting or something similar) didn't show up. One out of two run, I noticed the right panel populated with class/function info. Blank in the other: it crashed before anything could come out.

The code editor didn't respond to key stoke/mouse click. I intended to enter something in the code editor to see if code completion would work, but wasn't able to activate the code editor. Maybe it's working too hard at the background doing parsing.

It behaved somewhat differently from the previous version, in a sense, worse. But that should not matter once we figure out the reason for the segmentation fault.

gdb/run/bt produced similar information.

I was able to do some screen shots of supposed calling stack, hopefully it means something to you.

BTW: the generated ide is showing version 16466.

Update: in all following run(5-6), right panel were populated. And code editor could be actived. Code complete performed with similar response speed as previous tested version. So, no, it's not worth in this regard.

File Attachments
1) clang.png, downloaded 1391 times

/lib/x86_64-linux-gnu/libclang-14.so.13
/lib/x86_64-linux-gnu/libclang-14.so.13
/lib/x86_64-linux-gnu/libclang-14.so.13

Subject: Re: theide with libclang
Posted by Tom1 on Mon, 03 Oct 2022 07:34:32 GMT
View Forum Message <> Reply to Message

Hi Mirek,

No change. It still crashes as before:
* /home/tom/out/uppsrc/CLANG.Blitz.Gui.Lclang.Shared/ide 03.10.2022 09:47:54, user: tom

09:47:54:552 INFO GuiMainFn_(): Version: 16466 (64 bit) (wchar32) (CLANG) (C++14) (Gtk)
Compiled: 10/03/2022 09:47:14
****************** PANIC: Invalid memory access!

Anyway, it's nice to have the --noclang switch for testing.

BTW: I have also clang-14. Could this make any difference compared to your clang-10?

Best regards,

Tom

Subject: Re: theide with libclang
Posted by Tom1 on Mon, 03 Oct 2022 08:05:16 GMT
View Forum Message <> Reply to Message

Mirek,

I installed and had success with libclang-11! It does not crash with LCLANG flag.

* /home/tom/out/uppsrc/CLANG.Blitz.Gui.Lclang.Shared/ide 03.10.2022 11:02:44, user: tom

11:02:44:268 INFO GuiMainFn_(): Version: 16466 (64 bit) (wchar32) (CLANG) (C++14) (Gtk)
Compiled: 10/03/2022 10:55:02

Without the flag it goes with libclang-14 and crashes:
* /home/tom/out/uppsrc/CLANG.Blitz.Gui.Shared/ide 03.10.2022 10:59:52, user: tom

10:59:52:892 INFO libclang path:/usr/lib/llvm-14/lib/libclang.so
10:59:52:893 INFO GuiMainFn_(): Version: 16466 (64 bit) (wchar32) (CLANG) (C++14) (Gtk)
Compiled: 10/03/2022 10:59:10
****************** PANIC: Invalid memory access!

Best regards,

Tom

Subject: Re: theide with libclang
Posted by mirek on Mon, 03 Oct 2022 08:34:29 GMT
View Forum Message <> Reply to Message

Tom1 wrote on Mon, 03 October 2022 10:05Mirek,

I installed and had success with libclang-11! It does not crash with LCLANG flag.

* /home/tom/out/uppsrc/CLANG.Blitz.Gui.Lclang.Shared/ide 03.10.2022 11:02:44, user: tom

11:02:44:268 INFO GuiMainFn_(): Version: 16466 (64 bit) (wchar32) (CLANG) (C++14) (Gtk)
Compiled: 10/03/2022 10:55:02

Without the flag it goes with libclang-14 and crashes:
* /home/tom/out/uppsrc/CLANG.Blitz.Gui.Shared/ide 03.10.2022 10:59:52, user: tom

10:59:52:892 INFO libclang path:/usr/lib/llvm-14/lib/libclang.so
10:59:52:893 INFO GuiMainFn_(): Version: 16466 (64 bit) (wchar32) (CLANG) (C++14) (Gtk)
Compiled: 10/03/2022 10:59:10
********** PANIC: Invalid memory access!

Best regards,

Tom

Can you try to LCLANG with libclang-14? (by changing build method paths)

Anyway, if this is holds true, it would be both great and sort of bad - there will be a lot of work fixing install process for clang, even maybe ide/Builders. Hopefully the community will help with that...

Mirek

Subject: Re: theide with libclang
Posted by mirek on Mon, 03 Oct 2022 08:37:12 GMT
View Forum Message <> Reply to Message

Lance wrote on Mon, 03 October 2022 00:58
I was able to do some screen shots of supposed calling stack, hopefully it means something to you.

You do not need to do it this way (if you are debugging with theide): There is a menu item in Debug that copies backtrace of all threads to clipboard.

Mirek

Subject: Re: theide with libclang
Posted by mirek on Mon, 03 Oct 2022 08:48:03 GMT
View Forum Message <> Reply to Message

[quote title=Tom1 wrote on Mon, 03 October 2022 10:05]Mirek,

I installed and had success with libclang-11! It does not crash with LCLANG flag.

* /home/tom/out/uppsrc/CLANG.Blitz.Gui.Lclang.Shared/ide 03.10.2022 11:02:44, user: tom

11:02:44:268 INFO GuiMainFn_(): Version: 16466 (64 bit) (wchar32) (CLANG) (C++14) (Gtk)
Compiled: 10/03/2022 10:55:02

What about dynamically loaded libclang-11?

Mirek

Subject: Re: theide with libclang
Posted by Tom1 on Mon, 03 Oct 2022 09:30:41 GMT
View Forum Message <> Reply to Message

mirek wrote on Mon, 03 October 2022 11:34Tom1 wrote on Mon, 03 October 2022 10:05Mirek,

I installed and had success with libclang-11! It does not crash with LCLANG flag.

* /home/tom/out/uppsrc/CLANG.Blitz.Gui.Lclang.Shared/ide 03.10.2022 11:02:44, user: tom

11:02:44:268 INFO GuiMainFn_(): Version: 16466 (64 bit) (wchar32) (CLANG) (C++14) (Gtk)
Compiled: 10/03/2022 10:55:02

Without the flag it goes with libclang-14 and crashes:
* /home/tom/out/uppsrc/CLANG.Blitz.Gui.Shared/ide 03.10.2022 10:59:52, user: tom

10:59:52:892 INFO libclang path:/usr/lib/llvm-14/lib/libclang.so
10:59:52:893 INFO GuiMainFn_(): Version: 16466 (64 bit) (wchar32) (CLANG) (C++14) (Gtk)
Compiled: 10/03/2022 10:59:10
********** PANIC: Invalid memory access!

Best regards,

Tom

Can you try to LCLANG with libclang-14? (by changing build method paths)

Anyway, if this is holds true, it would be both great and sort of bad - there will be a lot of work fixing install process for clang, even maybe ide/Builders. Hopefully the community will help with that...

Mirek

Yes, this was my first test today and it failed. That's why I decided to try libclang-11 in the first place.

BR,

Tom

Subject: Re: theide with libclang
Posted by Tom1 on Mon, 03 Oct 2022 09:35:48 GMT
View Forum Message <> Reply to Message

[quote title=mirek wrote on Mon, 03 October 2022 11:48]Tom1 wrote on Mon, 03 October 2022 10:05Mirek,

I installed and had success with libclang-11! It does not crash with LCLANG flag.

* /home/tom/out/uppsrc/CLANG.Blitz.Gui.Lclang.Shared/ide 03.10.2022 11:02:44, user: tom

11:02:44:268 INFO GuiMainFn_(): Version: 16466 (64 bit) (wchar32) (CLANG) (C++14) (Gtk) Compiled: 10/03/2022 10:55:02

What about dynamically loaded libclang-11?

Mirek
Yes, libclang-11 works without crash when loading dynamically as well:
* /home/tom/out/uppsrc/CLANG.Blitz.Gui.Shared/ide 03.10.2022 12:33:52, user: tom

12:33:52:324 INFO libclang path:/usr/lib/llvm-11/lib/libclang.so
12:33:52:324 INFO GuiMainFn_(): Version: 16466 (64 bit) (wchar32) (CLANG) (C++14) (Gtk) Compiled: 10/03/2022 10:59:10

BR,

Tom

Subject: Re: theide with libclang
Posted by Tom1 on Mon, 03 Oct 2022 09:45:43 GMT
View Forum Message <> Reply to Message

Mirek,

Further on: libclang-11, libclang-12 and libclang-13 do not crash when loading dynamically. It's just libclang-14 that crashes here.

BR,

Tom

Subject: Re: theide with libclang
Posted by mirek on Mon, 03 Oct 2022 11:38:12 GMT
View Forum Message <> Reply to Message

Tom1 wrote on Mon, 03 October 2022 11:45Mirek,

Further on: libclang-11, libclang-12 and libclang-13 do not crash when loading dynamically. It's just libclang-14 that crashes here.

BR,

Tom

Thank you, very helpful. So I guess we can rule out dynamic loading and it now appears to be 14 specific.

I guess there is some bug in using the API.

Mirek

Subject: Re: theide with libclang
Posted by Lance on Mon, 03 Oct 2022 13:51:25 GMT
View Forum Message <> Reply to Message

mirek wrote on Mon, 03 October 2022 04:37Lance wrote on Mon, 03 October 2022 00:58
I was able to do some screen shots of supposed calling stack, hopefully it means something to you.

You do not need to do it this way (if you are debugging with theide): There is a menu item in Debug that copies backtrace of all threads to clipboard.

Mirek

I see. Thanks.

Thread: 1

futex_wait (private=0, expected=2, futex_word=0x555556869ca8 <Upp::sHeapLock2+8>) at
../sysdeps/nptl/futex-internal.h:146
./nptl/lowlevellock.c:49

Thread: 2

../sysdeps/unix/sysv/linux/poll.c:29

Thread: 3

../sysdeps/unix/sysv/linux/poll.c:29

Thread: 4

./nptl/futex-internal.c:57
./nptl/futex-internal.c:87
./nptl/futex-internal.c:139

Thread: 5

./nptl/futex-internal.c:57
./nptl/futex-internal.c:87
./nptl/futex-internal.c:139

Thread: 6

./nptl/futex-internal.c:57
./nptl/futex-internal.c:87
./nptl/futex-internal.c:139

Thread: 7

./nptl/futex-internal.c:57
./nptl/futex-internal.c:87

./nptl/futex-internal.c:139

Thread: 8

./nptl/futex-internal.c:57
./nptl/futex-internal.c:87
./nptl/futex-internal.c:139

Thread: 9

./nptl/futex-internal.c:57
./nptl/futex-internal.c:87
./nptl/futex-internal.c:139

Thread: 10

./nptl/futex-internal.c:57
./nptl/futex-internal.c:87
./nptl/futex-internal.c:139

Thread: 11

./nptl/futex-internal.c:57
./nptl/futex-internal.c:87
./nptl/futex-internal.c:139

Thread: 12

./nptl/futex-internal.c:57
./nptl/futex-internal.c:87
./nptl/futex-internal.c:139

Thread: 13

./nptl/futex-internal.c:57
./nptl/futex-internal.c:87
./nptl/futex-internal.c:139

Thread: 15

syscall () at ../sysdeps/unix/sysv/linux/x86_64/syscall.S:38

Thread: 17

../sysdeps/unix/sysv/linux/poll.c:29

Thread: 18

./nptl/futex-internal.c:57
./nptl/futex-internal.c:87
./nptl/futex-internal.c:139

Thread: 19

./nptl/futex-internal.c:57
./nptl/futex-internal.c:87
./nptl/futex-internal.c:139

Thread: 20

/lib/x86_64-linux-gnu/libclang-14.so.13

Thread: 21

futex_wait (private=0, expected=2, futex_word=0x555556869ca8 <Upp::sHeapLock2+8>) at
../sysdeps/nptl/futex-internal.h:146
./nptl/lowlevellock.c:49

Thread: 22

Upp::BlkHeap<Upp::HugeHeapDetail, 4096>::DbgFreeFill (p=<error reading variable: Cannot
access memory at address 0x7fffc5fed1b8>, size=<error reading variable: Cannot access memory
at address 0x7fffc5fed1b0>) at /home/lance/upp.src/uppsrc/Core/HeapImp.h:93

Thread: 23

./nptl/futex-internal.c:57
./nptl/futex-internal.c:87
./nptl/futex-internal.c:139

Thread: 24

./nptl/futex-internal.c:57
./nptl/futex-internal.c:87
./nptl/futex-internal.c:139

Thread: 25

futex_wait (private=0, expected=2, futex_word=0x5555568d01d8 <Upp::sGLock+8>) at
../sysdeps/nptl/futex-internal.h:146
./nptl/lowlevellock.c:49

Thread: 29

futex_wait (private=0, expected=2, futex_word=0x555556869ca8 <Upp::sHeapLock2+8>) at
../sysdeps/nptl/futex-internal.h:146
./nptl/lowlevellock.c:49

Thread: 31

futex_wait (private=0, expected=2, futex_word=0x555556869ca8 <Upp::sHeapLock2+8>) at
../sysdeps/nptl/futex-internal.h:146
./nptl/lowlevellock.c:49

Thread: 32

futex_wait (private=0, expected=2, futex_word=0x555556869ca8 <Upp::sHeapLock2+8>) at
../sysdeps/nptl/futex-internal.h:146
./nptl/lowlevellock.c:49

Subject: Re: theide with libclang
Posted by Klugier on Mon, 03 Oct 2022 21:20:42 GMT
View Forum Message <> Reply to Message

Hello,

I just installed Ubuntu 22.04.01 (LTS) and I managed to reproduce the problem. After opening project, the crash happen after 4-5 seconds of running indexer. Thanks to --noclang option I managed to build TheIDE under debug mode. Here is the screenshot from the crash and callstack:

It seems that it crashes when calling clang_getCursorPrettyPrinted in ClangCursorInfo::RawId() method:

```
force_inline
String ClangCursorInfo::RawId()
{
 if(!hasraw_id) {
  raw_id = FetchString(clang_getCursorPrettyPrinted(cursor, pp_id)); // Last call in TheIDE code base before crash...
  hasraw_id = true;
 }
 return raw_id;
}
```

I have 14.0.0-1ubuntu1, on Manajaro I have 14.0.6 and there is no crash.

Once change the line to:

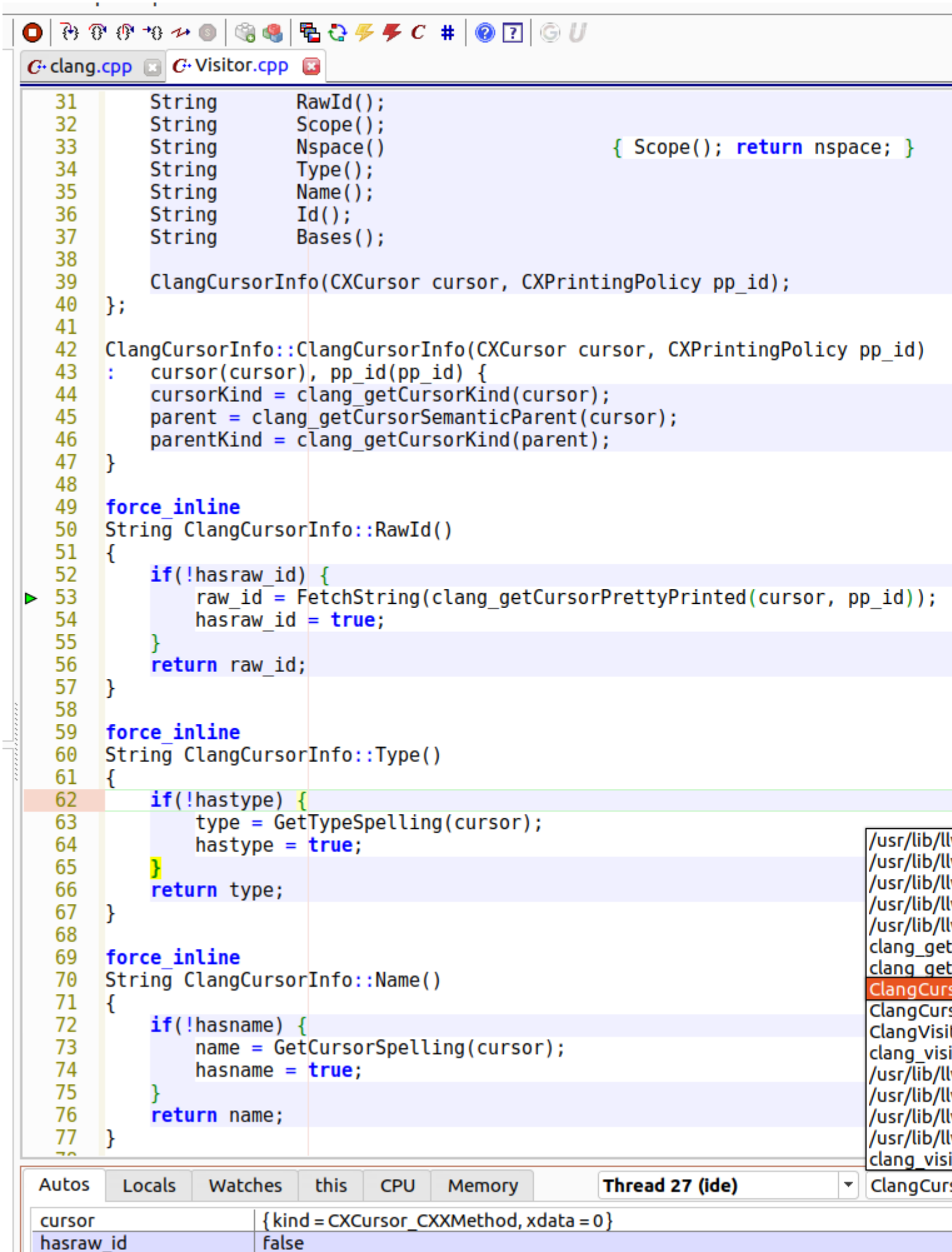
```
raw_id = FetchString(clang_getCursorPrettyPrinted(cursor, nullptr));
```

Fixed the problem, however I am not sure what is this code doing, so there is a chance that it will break something...

Klugier

File Attachments

1) Screenshot from 2022-10-03 23-15-25.png, downloaded 1583 times


clang.cpp
Visitor.cpp
31 String RawId();
32 String Scope();
33 String Nspace() { Scope(); return nspace; }
34 String Type();
35 String Name();
36 String Id();
37 String Bases();
38
39 ClangCursorInfo(CXCursor cursor, CXPrintingPolicy pp_id);
40 };
41
42 ClangCursorInfo::ClangCursorInfo(CXCursor cursor, CXPrintingPolicy pp_id)
43 : cursor(cursor), pp_id(pp_id) {
44 cursorKind = clang_getCursorKind(cursor);
45 parent = clang_getCursorSemanticParent(cursor);
46 parentKind = clang_getCursorKind(parent);
47 }
48
49 force_inline
50 String ClangCursorInfo::RawId()
51 {
52 if(!hasraw_id) {
53 raw_id = FetchString(clang_getCursorPrettyPrinted(cursor, pp_id));
54 hasraw_id = true;
55 }
56 return raw_id;
57 }
58
59 force_inline
60 String ClangCursorInfo::Type()
61 {
62 if(!hastype) {
63 type = GetTypeSpelling(cursor);
64 hastype = true;
65 }
66 return type;
67 }
68
69 force_inline
70 String ClangCursorInfo::Name()
71 {
72 if(!hasname) {
73 name = GetCursorSpelling(cursor);
74 hasname = true;
75 }
76 return name;
77 }
/usr/lib/ll
/usr/lib/ll
/usr/lib/ll
/usr/lib/ll
/usr/lib/ll
clang_get
clang_get
ClangCurs
ClangCurs
ClangVisit
clang_visi
/usr/lib/ll
/usr/lib/ll
/usr/lib/ll
/usr/lib/ll
clang_visi
Autos
Locals
Watches
this
CPU
Memory
Thread 27 (ide)
ClangCurs
cursor
{kind = CXCursor_CXXMethod, xdata = 0}
hasraw_id
false

Subject: Re: theide with libclang
Posted by mirek on Sat, 08 Oct 2022 17:47:27 GMT
View Forum Message <> Reply to Message

Klugier wrote on Mon, 03 October 2022 23:20Hello,

I just installed Ubuntu 22.04.01 (LTS) and I managed to reproduce the problem. After opening project, the crash happen after 4-5 seconds of running indexer. Thanks to --noclang option I managed to build TheIDE under debug mode. Here is the screenshot from the crash and callstack:

I now strongly believe this is libclang bug. When I comment out ide/clang/Visitor.cpp:378

// CXPrintingPolicy_FullyQualifiedName,

, there seems to be no more crashes.

And it always crashes on the same node (standard library to_pointer method).

Can you (I mean everybody with crashing libclang) check above (while I am thinking about how to fix this).

Mirek

Subject: Re: theide with libclang
Posted by Lance on Sat, 08 Oct 2022 19:29:14 GMT
View Forum Message <> Reply to Message

Mirek:

No more crashes after the line is commented out. I have tested with both llvm-14 and llvm-15.

For some reason, neither llvm-14 nor llvm-15 generate some symlinks that should be in exist for ide to properly load libclang dynamically.

I have
/usr/bin/llvm-config-15
and
/usr/bin/llvm-config-14
but not
/usr/bin/llvm-config

Also, I believe I also need to manually create

sudo ln -s /usr/lib/llvm-15/lib/libclang.so.1 /usr/lib/llvm-15/lib/libclang.so

As we cannot control how llvm installation creates symlinks, for smoother user experience, should we alter the detecting logic to accommodate this situation?

Regards,
Lance

BTW, the speed when built in release mode is quite good.

Subject: Re: theide with libclang
Posted by Tom1 on Sat, 08 Oct 2022 19:54:48 GMT
View Forum Message <> Reply to Message

Hi Mirek,

Same here: No more crashes after commenting out.

BR,

Tom

Update: And by the way: I'm really seriously missing the auto completion for macros...

Subject: Re: theide with libclang
Posted by mirek on Sat, 08 Oct 2022 21:04:15 GMT
View Forum Message <> Reply to Message

Tom1 wrote on Sat, 08 October 2022 21:54Hi Mirek,

Same here: No more crashes after commenting out.

BR,

Tom

Update: And by the way: I'm really seriously missing the auto completion for macros...

It is in todo list for now.

Mirek

Subject: Re: theide with libclang
Posted by mirek on Sun, 09 Oct 2022 18:44:50 GMT
View Forum Message <> Reply to Message

Workaround is in master.

I have also tried to

- do something with Linux performance - hopefully indexer threads now should have lower priority, maybe
- changed default Assist settings - lower the number of threads etc... (but if you have already started theide, settings are already in config)

Let is hope this will move us past linux issues....

Subject: Re: theide with libclang
Posted by Tom1 on Sun, 09 Oct 2022 19:19:12 GMT
View Forum Message <> Reply to Message

Hi Mirek,

Seems to run fine now in Linux Mint 21 here.

However, should this workaround be POSIX only, or what are the consequences in Windows?:
#define UBUNTU2204_WORKAROUND // there appears to be a bug in Ubuntu 22.04.1LTS libclang 14
Best regards,

Tom

Subject: Re: theide with libclang
Posted by mirek on Sun, 09 Oct 2022 21:18:30 GMT
View Forum Message <> Reply to Message

Tom1 wrote on Sun, 09 October 2022 21:19Hi Mirek,

Seems to run fine now in Linux Mint 21 here.

However, should this workaround be POSIX only, or what are the consequences in Windows?:
#define UBUNTU2204_WORKAROUND // there appears to be a bug in Ubuntu 22.04.1LTS libclang 14
Best regards,

Tom

Workaround basically implements what is crashing in another way. I will be better to keep the

code the same for both platforms.

I could probably do without #define, it more or less for documentation purposes...

Mirek

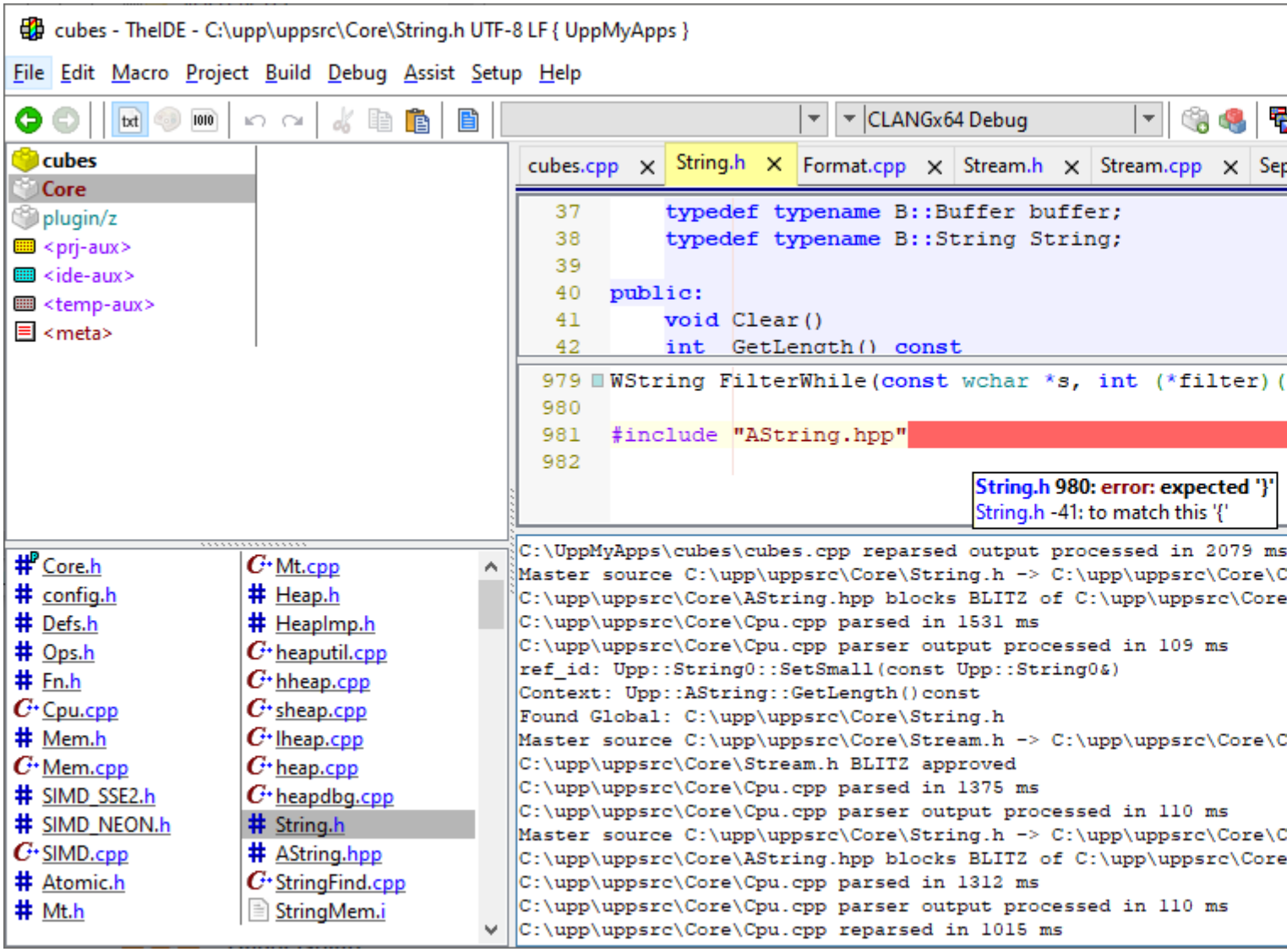
Subject: Re: theide with libclang
Posted by peterh on Sat, 15 Oct 2022 09:33:48 GMT
View Forum Message <> Reply to Message

libclangd seems to have a problem with header files.
It expects a "}" at the end of the header file and it thinks the opening bracket is at a negative line number.
I see this with other header files too.
Upp Version 16497 on Windows 10 64 bit.

In "CurrentContext.cpp" there is no matching bracket for
namespace Upp {

File Attachments

1) ide.png, downloaded 1446 times



Subject: Re: theide with libclang
Posted by mirek on Sun, 16 Oct 2022 11:28:16 GMT
View Forum Message <> Reply to Message

peterh wrote on Sat, 15 October 2022 11:33libclangd seems to have a problem with header files. It expects a "}" at the end of the header file and it thinks the opening bracket is at a negative line number.
I see this with other header files too.
Upp Version 16497 on Windows 10 64 bit.

In "CurrentContext.cpp" there is no matching bracket for
namespace Upp {

Thanks. I am aware of that, this is one of bugs to address. Work in progress.

Subject: Re: theide with libclang
Posted by Lance on Wed, 26 Oct 2022 14:00:00 GMT
View Forum Message <> Reply to Message

Not sure if this problem is unique to me: on Ubuntu 22.10+UPP.16512 (and some other recent versions),
"Find..." function cannot be used. Ctrl+F, type in string to search, won't work.

Another problem. Add new help topic will crash TheIDE. For a test,
* open "examples/AddressBook", navigate to "AddressBook.cpp" if not already in it.
* In the class "AddressBook", click any member variable or function to add cocumentation [a cyan square to the left of the function/variable line];
* in the popup menu, choose "New reference topic"
* in the "New Topic" dialog, in the Template field, change it to "Reference-en_us (in uppsrc)"
* Click "OK" button

The ide crash with invalid memory access.

I am sorry. I know Mirek has been tirelessly working on integrating libclang to theide. Not criticizing, not meaning to distract, just trying to bring awareness.

Subject: Re: theide with libclang
Posted by mirek on Wed, 26 Oct 2022 20:07:38 GMT
View Forum Message <> Reply to Message

Lance wrote on Wed, 26 October 2022 16:00Not sure if this problem is unique to me: on Ubuntu 22.10+UPP.16512 (and some other recent versions),
"Find..." function cannot be used. Ctrl+F, type in string to search, won't work.

Another problem. Add new help topic will crash TheIDE. For a test,
* open "examples/AddressBook", navigate to "AddressBook.cpp" if not already in it.
* In the class "AddressBook", click any member variable or function to add cocumentation [a cyan square to the left of the function/variable line];
* in the popup menu, choose "New reference topic"
* in the "New Topic" dialog, in the Template field, change it to "Reference-en_us (in uppsrc)"
* Click "OK" button

The ide crash with invalid memory access.

I am sorry. I know Mirek has been tirelessly working on integrating libclang to theide. Not criticizing, not meaning to distract, just trying to bring awareness.

Bug reports are always welcome!

Subject: Re: theide with libclang
Posted by mirek on Thu, 27 Oct 2022 10:21:57 GMT
View Forum Message <> Reply to Message

Lance wrote on Wed, 26 October 2022 16:00
Another problem. Add new help topic will crash TheIDE. For a test,
* open "examples/AddressBook", navigate to "AddressBook.cpp" if not already in it.
* In the class "AddressBook", click any member variable or function to add cocumentation [a cyan square to the left of the function/variable line];
* in the popup menu, choose "New reference topic"
* in the "New Topic" dialog, in the Template field, change it to "Reference-en_us (in uppsrc)"
* Click "OK" button

This part should be now fixed in master.

Subject: Re: theide with libclang
Posted by mirek on Thu, 27 Oct 2022 10:26:51 GMT
View Forum Message <> Reply to Message

Lance wrote on Wed, 26 October 2022 16:00Not sure if this problem is unique to me: on Ubuntu 22.10+UPP.16512 (and some other recent versions),
"Find..." function cannot be used. Ctrl+F, type in string to search, won't work.

This one seems to work for me.

Any chance you are trying to search in very long file? In that case, incremental search is disabled (and you need to push "Find next" button).

If that is not the explanation:

Do you see bottom bar with text?

Can you type text into into the field?

Have you tried pressing next/previous buttons?

What about Find in files?

Mirek

Subject: Re: theide with libclang
Posted by Lance on Tue, 01 Nov 2022 22:32:09 GMT
View Forum Message <> Reply to Message

Thank you, Mirek! Both are fixed in UPP version 16523.

Another bug I found a few weeks (maybe months) ago. I tested with version 16523 and it's still there.

To reproduce the experience, move away or rename libclang.so/dll etc so that it cannot be detected when theide starts. Or, change
bool TryLoadLibClang()
 at line 120 in <ide/main.cpp> to fail the LibClang loading:

```
bool TryLoadLibClang()
{
	// add the following line to
 return false;  // make sure detection fails.


 String libdir = TrimBoth(Sys("llvm-config --libdir"));
 int q = FindIndex(CommandLine(), "--clangdir");
 if(q >= 0 && q + 1 < CommandLine().GetCount()) {
  libdir = CommandLine()[q + 1];
  CommandLineRemove(q, 2);
 }


 if(LoadLibClang(libdir))
  return true;
 if(LoadLibClang("/usr/lib"))
  return true;
 for(int i = 200; i >= 10; i--)
  if(LoadLibClang("/usr/lib/llvm-" + AsString(i) + "/lib"))
   return true;
 return false;
}
```

F5 to compile and run the modified theide, in it, open button.cpp from <examples/Button>, left click on, for example the word 'TopWindow', the assist++ will start a dialog that will never end unless cancelled.

I am not sure if there are other occassions like this when assist++ will be started by some other key combinations or mouse events, even though theide knows and even prompted the end users of "libclang detection failure". This is no longer bothering me, but could confuse a person who starts to try UPP.

Subject: Re: theide with libclang
Posted by Lance on Tue, 01 Nov 2022 23:15:47 GMT
View Forum Message <> Reply to Message

Another more like a feature request:

Take for example, the following code:

```
class Person
{
 int age_;
public:
 int get_age()const { return age_; }
 void set_age(int age){ age_ = age; }
 __declspec(property(get=get_age, put = set_age)) int age;
};
```

theide/Assist++ will put a red rect at the beginning of __declspec, with error message ...'__declspec' attributes are not enabled; use '-fdeclspec'..., even though the flag has been turned on. This is kind of a fictitious example. The real issue here is, the flags I set in "Setup > Build Methods" for CLANG, under common c++ options

-std=c++20 -Wno-logical-op-parentheses -fdeclspec

is not respected by assist++.

So if someone wants to try some c++20 new language features, assist++ will stick to builtin option of '-std=c++14' and keep bugging him.

Subject: Re: theide with libclang
Posted by Lance on Wed, 02 Nov 2022 00:58:45 GMT
View Forum Message <> Reply to Message

It might be something messed up in my system.

Quote:
Any chance you are trying to search in very long file? In that case, incremental search is disabled (and you need to push "Find next" button).

No.

Quote:
Do you see bottom bar with text?

Yes.

Quote:
Can you type text into into the field?

Yes.

Quote:
Have you tried pressing next/previous buttons?

Yes, nothing happen. O occurrence, or something like that.

Quote:
What about Find in files?

This has always worked fine.

By the way, if I don't go through the menu, but use Ctrl+F instead, I receive a ding, but nothing happen afterwards, not even the text field for entering text to be searched.

I have overwritten the previous compiled version of ide. Above replies are according to my memory. I don't think they matter any more since that problem might be unique to me.

Subject: Re: theide with libclang
Posted by mirek on Wed, 02 Nov 2022 09:21:42 GMT
View Forum Message <> Reply to Message

Lance wrote on Wed, 02 November 2022 00:15Another more like a feature request:

Take for example, the following code:

```
class Person
{
 int age_;
public:
 int get_age()const { return age_; }
```

```
 void set_age(int age){ age_ = age; }
 __declspec(property(get=get_age, put = set_age)) int age;
};
```

theide/Assist++ will put a red rect at the beginning of __declspec, with error message ...'__declspec' attributes are not enabled; use '-fdeclspec'..., even though the flag has been turned on. This is kind of a fictitious example. The real issue here is, the flags I set in "Setup > Build Methods" for CLANG, under common c++ options

-std=c++20 -Wno-logical-op-parentheses -fdeclspec

is not respected by assist++.

So if someone wants to try some c++20 new language features, assist++ will stick to builtin option of '-std=c++14' and keep bugging him.

Well, this will be a bit complicated: We are not sure that libclang version is able to support -std=c++20.... Any ideas?

(In general, libclang is not the same thing as clang, so we cannot really use options from build method... Adding " -Wno-logical-op-parentheses -fdeclspec" as fixed options makes sense though.)

Mirek

Subject: Re: theide with libclang
Posted by mirek on Wed, 02 Nov 2022 14:58:40 GMT
View Forum Message <> Reply to Message

Lance wrote on Wed, 02 November 2022 01:58It might be something messed up in my system.

Quote:
Any chance you are trying to search in very long file? In that case, incremental search is disabled (and you need to push "Find next" button).

No.

Quote:
Do you see bottom bar with text?

Yes.

Quote:
Can you type text into into the field?

Yes.

Quote:
Have you tried pressing next/previous buttons?

Yes, nothing happen. O occurrence, or something like that.

Quote:
What about Find in files?

This has always worked fine.

By the way, if I don't go through the menu, but use Ctrl+F instead, I receive a ding, but nothing happen afterwards, not even the text field for entering text to be searched.

I have overwritten the previous compiled version of ide. Above replies are according to my memory. I don't think they matter any more since that problem might be unique to me.

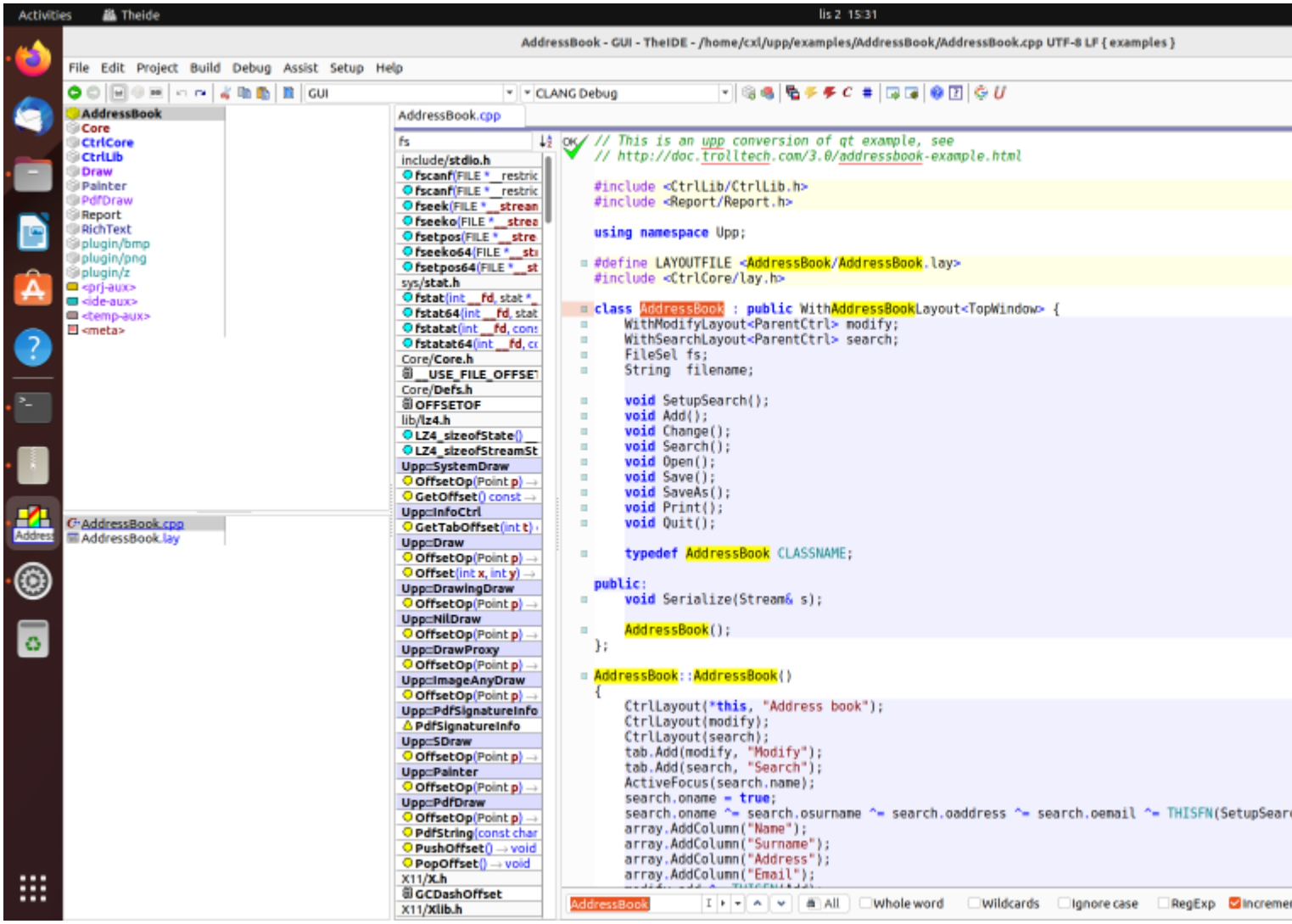
I have installed 22.10 in Virtual Box and tested:

all seems fine

(On the positively negative note, theide crashed the first time I have run it. I have noticed this before in Windows, so now it is officially a new phantom bug to fix...)

File Attachments

1) VirtualBox_Ubuntu 22.10_02_11_2022_15_31_02.png, downloaded 1364 times



Subject: Re: theide with libclang
Posted by mirek on Wed, 02 Nov 2022 18:13:15 GMT
View Forum Message <> Reply to Message

Lance wrote on Wed, 02 November 2022 00:15Another more like a feature request:

Take for example, the following code:

```
class Person
{
 int age_;
public:
 int get_age()const { return age_; }
 void set_age(int age){ age_ = age; }
 __declspec(property(get=get_age, put = set_age)) int age;
};
```

theide/Assist++ will put a red rect at the beginning of __declspec, with error message ...'__declspec' attributes are not enabled; use '-fdeclspec'..., even though the flag has been turned on. This is kind of a fictitious example. The real issue here is, the flags I set in "Setup > Build Methods" for CLANG, under common c++ options

-std=c++20 -Wno-logical-op-parentheses -fdeclspec

is not respected by assist++.

So if someone wants to try some c++20 new language features, assist++ will stick to builtin option of '-std=c++14' and keep bugging him.

Hopefully solved by adding a field for additional libclang options to Assist settings (so that you can specify -std=c++20 there...)

Subject: Re: theide with libclang
Posted by Lance on Wed, 02 Nov 2022 21:52:56 GMT
View Forum Message <> Reply to Message

Thank you, Mirek. I will test it when github/ultimatepp is updated.

BTW, "Find" now works fine in UPP.16523 like I have reported yesterday.

And yes, I also had a mysterious theide-crash-on first-run-then-works-fine-afterwords experience. I accidently updated a upp.16470 src and built theide from there, on first run, it crashed with invalid-memory-access. Strange.

Subject: Re: theide with libclang
Posted by Lance on Thu, 03 Nov 2022 19:06:28 GMT
View Forum Message <> Reply to Message

additional c++ options for libclang works like a charm in most recent version 16528.

The setting is through:
Main Menu: Setup -> Settings ...
in Assist tab
Second last item "libclang additional options"

After I set it to
-Wno-logical-op-parentheses -std=c++20 -fdeclspecthe following sample code are properly understood by Assist++

```
class Person
{
 int age_;
public:
 int get_age()const { return age_; }
 void set_age(int age){ age_ = age; }
 __declspec(property(get=get_age, put = set_age)) int age;
};

template <class T>
concept A = requires(T t) {
 ++t;
};
```

Wonderful! Imaging how much work it will involve if libclang is not used and we have to update the homegrown c++ parser to provide intellisense to the ever-updating c++ language.

I remember a while ago I dug into the parser while trying to add support to attribute: it is so complicated and involving that I had to give up.

It's great to see that libclang-powered assist++ becomes quite usable in rapid pace.

Subject: Re: theide with libclang
Posted by mirek on Mon, 21 Nov 2022 15:46:38 GMT
View Forum Message <> Reply to Message

In recent commit, I have tried to mitigate somewhat the performance issue when you are building with Indexer doing its thing. I cannot suspend threads in Linux unfortunately, so I have added sort of "blocking sleep" into indexer. Far from ideal as this only can take place when libclang is not compiling (means after starting the build, it might take some seconds to stop indexer), but better than nothing....

Subject: Re: theide with libclang
Posted by zsolt on Tue, 22 Nov 2022 07:28:21 GMT
View Forum Message <> Reply to Message

Thanks for your work.

A problem: Alt + C is not working now. It starts an endless "Parsing" and doesn't copy anything to the clipboard.

Subject: Re: theide with libclang
Posted by mirek on Tue, 22 Nov 2022 09:28:39 GMT
View Forum Message <> Reply to Message

zsolt wrote on Tue, 22 November 2022 08:28Thanks for your work.

A problem: Alt + C is not working now. It starts an endless "Parsing" and doesn't copy anything to the clipboard.

Works for me. Endless "Parsing" would mean that current file parser never stops.

Does it do that always? If yes, can you post more info so that I can reproduce?

Does autocomplete work? (That is basically the same parsing thread).

Mirek

Subject: Re: theide with libclang
Posted by zsolt on Tue, 22 Nov 2022 10:21:00 GMT
View Forum Message <> Reply to Message

mirek wrote on Tue, 22 November 2022 10:28

Does it do that always? If yes, can you post more info so that I can reproduce?

Does autocomplete work? (That is basically the same parsing thread).

Mirek

Interesting. I can not reproduce it anymore. I restarted the IDE, maybe reindexing helped.

Subject: Re: theide with libclang
Posted by mirek on Tue, 22 Nov 2022 10:25:11 GMT
View Forum Message <> Reply to Message

zsolt wrote on Tue, 22 November 2022 11:21mirek wrote on Tue, 22 November 2022 10:28

Does it do that always? If yes, can you post more info so that I can reproduce?

Does autocomplete work? (That is basically the same parsing thread).

Mirek

Interesting. I can not reproduce it anymore. I restarted the IDE, maybe reindexing helped.

It is not related to reindexing. Anyway, I have noticed that in rare cases, current file parse gets

stuck for some reason - I have it in internal todo list. Usually clicking other file and then back helps.

Subject: Re: theide with libclang
Posted by zsolt on Tue, 22 Nov 2022 10:40:58 GMT
View Forum Message <> Reply to Message

Thanks.

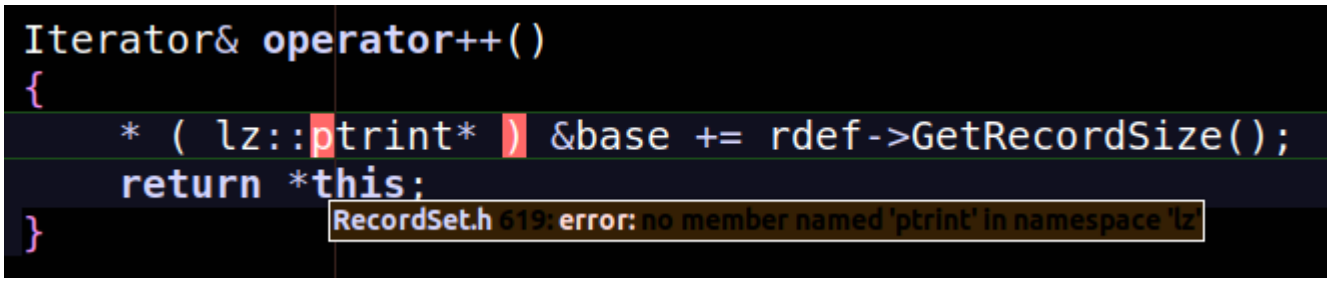
Subject: Re: theide with libclang
Posted by Lance on Tue, 22 Nov 2022 23:44:21 GMT
View Forum Message <> Reply to Message

Hi Mirek:

In dark theme, the error texts is not legible.

BR,
Lance

File Attachments
1) Screenshot from 2022-11-22 18-40-56.png, downloaded 1373 times



Subject: Re: theide with libclang
Posted by mirek on Wed, 23 Nov 2022 09:46:36 GMT
View Forum Message <> Reply to Message

Lance wrote on Wed, 23 November 2022 00:44Hi Mirek:

In dark theme, the error texts is not legible.

BR,

Lance

Hopefully fixed, thanks for reporting.

Subject: Re: theide with libclang
Posted by Lance on Thu, 24 Nov 2022 02:12:38 GMT
View Forum Message <> Reply to Message

mirek wrote on Wed, 23 November 2022 04:46Lance wrote on Wed, 23 November 2022 00:44Hi Mirek:

In dark theme, the error texts is not legible.

BR,
Lance

Hopefully fixed, thanks for reporting.

Thanks for the quick response. I will check github tomorrow. Version 16566 as of now still not fixed.

PS: Just compiled ver 16575. The problem is gone. Thanks!
