Subject: 2022.3rc5
Posted by mirek on Fri, 23 Dec 2022 08:52:29 GMT
View Forum Message <> Reply to Message

- fixed problem with console colors in certain Linux settings

https://sourceforge.net/projects/upp/files/upp/2022.3rc5/

Subject: Re: 2022.3rc4
Posted by Tom1 on Fri, 23 Dec 2022 10:51:52 GMT
View Forum Message <> Reply to Message

Hi,

I looked into the menu item text color issue on Windows 11, and discovered that it is related to the fix #97 in ChWin32.cpp by Klugier on September 2022. While it fixed the light theme menu item text colors visibility, it makes dark theme look bad, i.e. it now uses black menu item text on dark selection bar. The text should obviously remain white here. Could you please look at this?

Best regards,

Tom

Subject: Re: 2022.3rc4
Posted by Klugier on Fri, 23 Dec 2022 12:45:01 GMT
View Forum Message <> Reply to Message

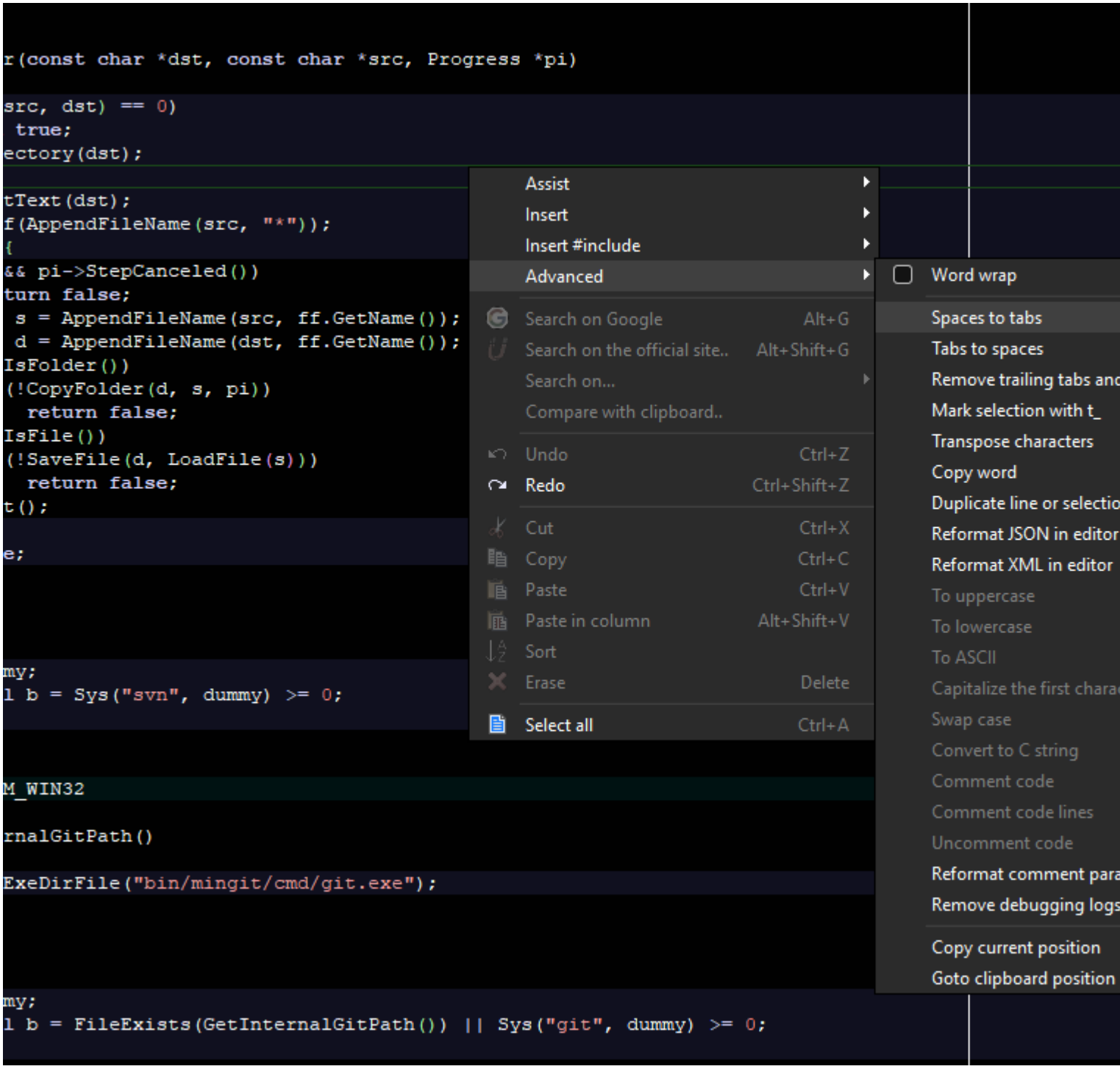
Hello Tom,

Could you share a screenshot with the issue? For me everything looks fine on the latest Windows11 version 22H2.

Also, please compare to what Windows displays in other applications such as file explore. I can say that it is exactly the same for black and light themes. Please noticed that 22H2 version of Windows11 changed the style of menu bars and made the more similar to what we have in Windows10. If the issue is reproducible on older version of Windows11 and it is fine for WIndows10 I would ignore it. 21H2 supports ends on 10th of October 2023.

Klugier

File Attachments
1) DarkThemMenuBar.png, downloaded 1424 times



Subject: Re: 2022.3rc4
Posted by Tom1 on Fri, 23 Dec 2022 13:37:01 GMT
View Forum Message <> Reply to Message

Hi Klugier,

You're right. I have W11 22H1 in the office and that shows the problem. However, W11 22H2

(here at home) works correctly, so it's all good. I do not think we need to fix previous W11 since it will eventually be updated everywhere anyway.

Thanks and best regards,

Tom

Subject: Re: 2022.3rc4
Posted by lindquist on Sat, 24 Dec 2022 10:02:19 GMT
View Forum Message <> Reply to Message

Hi,
Found a problem in the release. When creating a "clip" for DoDragAndDrop, when using AppendFiles to build the clip, then program receiving the dropped file will see a list of single char paths.

The problem seems to be in the String -> WString conversion. Changing AppendFiles to use just String (and unicode=false), it works just fine.

eg.

```
// somethings broken here
 String dropFile = ConfigFile("foobar");
 VectorMap<String, ClipData> clip;
 AppendFiles(clip, { dropFile } );
 if (DoDragAndDrop(clip, img, DND_COPY) == DND_COPY) {
  // receiver sees {"f","o","o","b","a","r" } not "foobar"
 }
```

I hope this report is sufficient.

Subject: Re: 2022.3rc4
Posted by lindquist on Sat, 24 Dec 2022 10:08:22 GMT
View Forum Message <> Reply to Message

Bug with drag and drop and TabBar.

The following program, lets you drag an icon from the main window.

When doing so, as the dragging passes over hte TabBar, the program crashes:

```
#include <CtrlLib/CtrlLib.h>
#include <TabBar/TabBar.h>
```

```
using namespace Upp;

struct DragTabBug : TopWindow {
 TabBar tabs;
 DragTabBug() {
  AddFrame(tabs);
 }
 void LeftDrag(Point p, dword keyflags) override {
  Image i = CtrlImg::new_doc();
  VectorMap<String, ClipData> clip;
  Append(clip, i);
  DoDragAndDrop(clip, i);
 }
};

GUI_APP_MAIN
{
 DragTabBug win;
 win.Run();
}
```

ie. Click and drag the background, while dragging, move the cursor across the tabbar.

Subject: Re: 2022.3rc4
Posted by Oblivion on Sat, 24 Dec 2022 11:25:41 GMT
View Forum Message <> Reply to Message

Quote:
Bug with drag and drop and TabBar.

The following program, lets you drag an icon from the main window.

When doing so, as the dragging passes over hte TabBar, the program crashes:

This is due to an out-of-bounds access. (access violation)

I've made a pull request to add a check.

Best regards,
Oblivion

Subject: Re: 2022.3rc4
Posted by Klugier on Sat, 24 Dec 2022 12:13:42 GMT
View Forum Message <> Reply to Message

Hello Lindquist and Mirek,

Thank you for raising this issue! I confirm that this issue is critical in the context of release. It is with us since 2022.1 when we introduced wchar that is 4 bytes length. The problem is that winapi requires 2 bytes instead of 4 bytes. The problem is with AppendFiles:

```
void AppendFiles(VectorMap<String, ClipData>& clip, const Vector<String>& files)
{
 WString wfiles;
 for(int i = 0; i < files.GetCount(); i++)
  wfiles << files[i].ToWString() << (wchar)0;
 sDROPFILES h;
 h.unicode = true;
 h.offset = sizeof(h);
 GetCursorPos(&h.pt);
 h.nc = TRUE;
 String data;
 data.Cat((byte *)&h, sizeof(h));
 data.Cat((byte *)~wfiles, 2 * (wfiles.GetCount() + 1)); // Windows wants 2 bytes unicode string
instead of 4 bytes we are providing
 clip.GetAdd("files") = ClipData(data);
}
```

Mirek, you should be able to propose the best solution here since you know the WString 4 bytes internals very well. I would opt for something like using ToSystemCharset on files to make sure that we have proper encoding.

The bug can be reproduce with reference/DropFiles example. In this case you can not drop files from the window.

The issue is only reproduce on Windows.

Klugier

Subject: Re: 2022.3rc4
Posted by mirek on Sat, 24 Dec 2022 14:23:50 GMT
View Forum Message <> Reply to Message

rc5 uploaded...

Subject: Re: 2022.3rc5
Posted by lindquist on Sun, 25 Dec 2022 12:23:05 GMT
View Forum Message <> Reply to Message

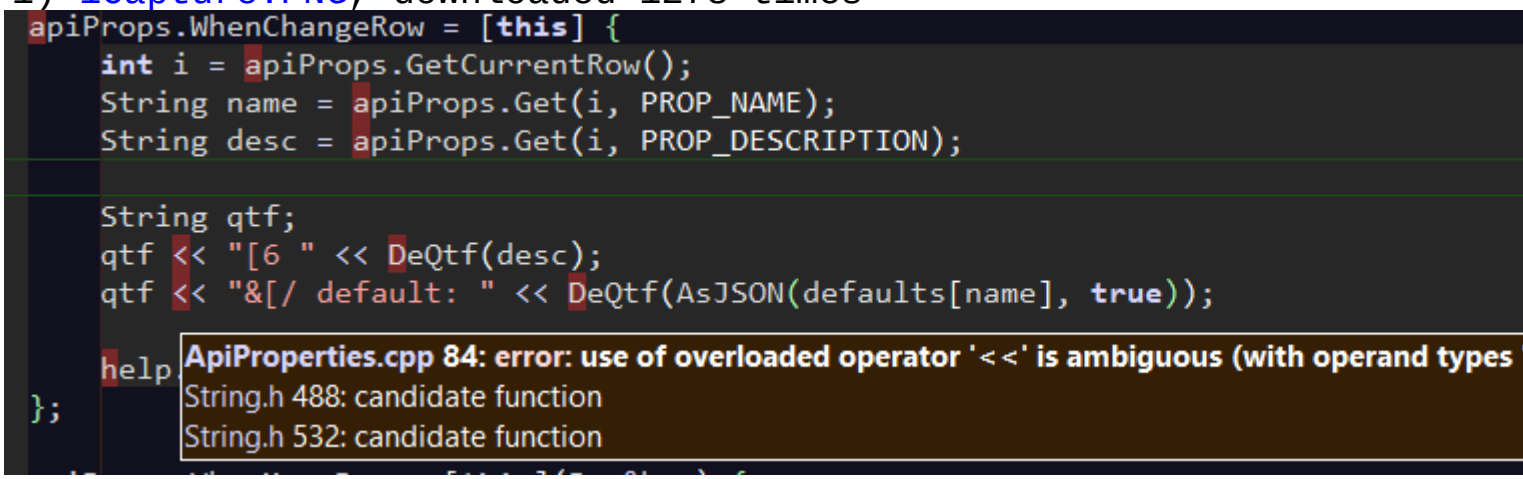
Hi,
Merry christmas

I tried the latest master and the drag and drop issue is fixed. That's awesome. Thanks!

However, as I moved to the ultimatepp master, assist++ has gotten a lot worse. I get a lot of errors indicated in the editor, that are not a problem when compiling.
ie. the code compiles and runs fine, but theide editor insists that it's full of errors, like:

File Attachments

1) 1Capture.PNG, downloaded 1278 times



Subject: Re: 2022.3rc5
Posted by lindquist on Sun, 25 Dec 2022 15:28:25 GMT
View Forum Message <> Reply to Message

I had a chat with Klugier on Discord, and after copying the "bin" dir into my git master checkout, it works fine!

Subject: Re: 2022.3rc5
Posted by Klugier on Mon, 26 Dec 2022 21:44:39 GMT
View Forum Message <> Reply to Message

Hello Mirek,

You added crash handling once TheIDE crashed. Recently, I had a crash on Linux and obtained

this prompt on startup:

TheIDE has crashed the last time it was run. As the possible cause is libclang incompatibility, we are disabling Assist features for now.
You can have them reenabled in Setup / Settings..

I think it should be yes/no option. Most of the time I do not want to disable libclang and there is no 100% that the crash was caused by libclang. Going into setting and switch assist back is from my point of view too much in the additional work I need to do. Please consider changing this behavior before release...

The sad news is that the crash is still there, however the rerpoduction scenario is not so easy. I switched main package several time.

Klugier

Subject: Re: 2022.3rc5
Posted by Oblivion on Mon, 26 Dec 2022 22:49:53 GMT
View Forum Message <> Reply to Message

Hi,

For some reason, I can't get clangd/TheIDE work on Linux (archlinux, up-to-date, clang 14.0.6); No assist,no autocomplete, and the navigator is empty. Log says:

01:20:55:490 INFO LoadLibClang0(): libclang path: "/usr/lib/libclang.so"

And I always get the below error message, which is not very helpful...

libclang additional compiler options do not seem do be correct. Assist might not work.

I have no idea what those additional compiler options should be.

Any suggestions?

Best regards,
Oblivion

Subject: Re: 2022.3rc5
Posted by mirek on Tue, 27 Dec 2022 08:49:20 GMT

View Forum Message <> Reply to Message

Oblivion wrote on Mon, 26 December 2022 23:49Hi,

For some reason, I can't get clangd/TheIDE work on Linux (archlinux, up-to-date, clang 14.0.6); No assist,no autocomplete, and the navigator is empty. Log says:

01:20:55:490 INFO LoadLibClang0(): libclang path: "/usr/lib/libclang.so"

And I always get the below error message, which is not very helpful...

libclang additional compiler options do not seem do be correct. Assist might not work.

I have no idea what those additional compiler options should be.

Any suggestions?

Best regards,
Oblivion

These are in settings/assist. The original reason is to allow activating c++20. The semantics is basically the same as compiler options in build method, however as Assist is not really connected to build method (well, it fetches include paths for the most likely clang build method, but that might not be one with c++20), it needs separate set.

Mirek

Subject: Re: 2022.3rc5
Posted by mirek on Tue, 27 Dec 2022 08:51:12 GMT
View Forum Message <> Reply to Message

Anyway, that error means clang_parseTranslationUnit, which usually is caused by invalid commandline. If you have time, please try to investigate in Clang::Parse ide/clang/clang.cpp:77

Subject: Re: 2022.3rc5
Posted by mirek on Tue, 27 Dec 2022 11:23:39 GMT
View Forum Message <> Reply to Message

Klugier wrote on Mon, 26 December 2022 22:44
The sad news is that the crash is still there, however the rerpoduction scenario is not so easy. I switched main package several time.

Klugier

That is a clue (and also explains why I am not getting these, I rarely switch, prefer to start a new theide at all times).

Maybe when switching while current file parsing is running? Does it crash soon after switching?

Subject: Re: 2022.3rc5
Posted by Klugier on Tue, 27 Dec 2022 12:48:19 GMT
View Forum Message <> Reply to Message

Hello Mirek,

I see one problem - while switching package in single TheIDE instance mode the build continues. It means that for example you select ide package and start the build. In the meanwhile you decided to hit CTRL+M and switch package to other one. The build for previously selected package continues. This is very risky behavior and can lead to unexpected behaviors like crashes and faulty debug session.

The possible fix Ide/idebar.cpp):

```
menu.Add(AK_SETMAIN, IdeImg::MainPackage(), THISBACK(NewMainPackage))
   .Enable(!IdeIsDebugLock() && !IdeIsBuilding())
   .Help("Select global configuration (var), select / add main project package");
```

Problems with above implementation:
- it blocks spawning new main package in non single window mode which is not the best approch
- i think it would be better to stop build while changing the main package in single window instance. The same is true for indexing. It should be canceled before switching to new package.

Please do not fix this issue by making "Open main package"creates the new theide instead of just changing main package" the default option. For basic users current behavior is more friendly. However, we need to make it more isolated.

What about relaxing crash dialog to (yes/no) instead of simple always turn off. As I previously said non all crashes all caused by assist and turning it of in such cases doesn't make sens and do not fix the problem. Message that should be migrated from PromptOK to PromptYesNo:

TheIDE has crashed the last time it was run. As the possible cause is libclang incompatibility, we are disabling Assist features for now.
You can have them reenabled in Setup / Settings..

Klugier

Subject: Re: 2022.3rc5
Posted by Oblivion on Sun, 05 Feb 2023 15:36:42 GMT
View Forum Message <> Reply to Message

Hello Mirek,

Quote:
Anyway, that error means clang_parseTranslationUnit, which usually is caused by invalid commandline. If you have time, please try to investigate in Clang::Parse ide/clang/clang.cpp:77

I am still baffled by the problem. Couldn't make the new assist run on my Linux (ArchLinux) machines.
It works fine on Windows and is very cool.

It appears that clang.cpp:138, clang_parseTranslationUnit() function fails all the time, as I constantly see "failed commandline" message in the console. Logs don't show any error though...

So, I've dumped the values of main variables of the the Clang::Parse method (txt file is attached).

Where should I look at or what do you see wrong in these dumps?

Best regards,
Oblivion

File Attachments
1) dump.txt, downloaded 292 times

Subject: Re: 2022.3rc5
Posted by mirek on Sun, 05 Feb 2023 16:00:23 GMT
View Forum Message <> Reply to Message

Uhm all seems fine.

I would perhaps start by removing argv items until *maybe* it returns something else than NULL...

Subject: Re: 2022.3rc5
Posted by Oblivion on Mon, 06 Feb 2023 22:34:00 GMT
View Forum Message <> Reply to Message

Hello Mirek,

It didn't work but I've got a more detailed error -instead of plain nullptr- by using clang_parseTranslationUnit2() function, which returns an enum, defined in CxErrorCode.h):

```
/**
 * An AST deserialization error has occurred.
 */
CXError_ASTReadError = 4
```

I'll dig deeper, but if this gives you any hints, please let me know.

Best regards,
Oblivion

Subject: Re: 2022.3rc5
Posted by mirek on Mon, 06 Feb 2023 23:16:28 GMT
View Forum Message <> Reply to Message

Oblivion wrote on Mon, 06 February 2023 23:34Hello Mirek,

It didn't work but I've got a more detailed error -instead of plain nullptr- by using clang_parseTranslationUnit2() function, which returns an enum, defined in CxErrorCode.h):

```
/**
 * An AST deserialization error has occurred.
 */
CXError_ASTReadError = 4
```

I'll dig deeper, but if this gives you any hints, please let me know.

Best regards,
Oblivion

https://groups.google.com/g/llvm-dev/c/Jh0bUJz7PHw?pli=1

Based on this, maybe you could try to pass the filename instead of "nullptr" in the call and remove it from commandline? Not the same error, but could be related.

Subject: Re: 2022.3rc5
Posted by Oblivion on Wed, 08 Feb 2023 16:55:24 GMT
View Forum Message <> Reply to Message

Hello Mirek,

I've stripped the code, and wrote the below test:

```
struct Clang {

 CXIndex index = nullptr;
 CXTranslationUnit tu = nullptr;

 Clang()
 {
  MemoryIgnoreLeaksBlock __;
  index = clang_createIndex(0, 0);
 }

 void Parse()
 {
  Vector<const char*> args = {
   "-std=c++14",
   "-xc++",
   "-I/usr/include/llvm",
   "-I/usr/include/c++",
   "-I/usr/include/c++/12.2.1",
   "-I/usr/include/c++/12.2.1/x86_64-pc-linux-gnu",
   "-I/usr/include/c++/12.2.1/backward",
   "-I/usr/lib/clang/15.0.7/include",
   "-I/usr/include"
   "-I/usr/local/include",
  };

  tu = clang_parseTranslationUnit(
      index,
      "/home/user/test.cpp",
                    args,
                    args.GetCount(),
                    nullptr,
                    0,
                    CXTranslationUnit_None
               );
 }

 ~Clang()
 {
  MemoryIgnoreLeaksBlock __;
  if(tu) clang_disposeTranslationUnit(tu);
  clang_disposeIndex(index);
 }

CONSOLE_APP_MAIN
```

```
{
 StdLogSetup(LOG_COUT);
 Clang().Parse();
}
```

This works. clang_parseTranslationUnit() returns a translation unit handle every single time. (And the handle can be successfully used to traverse the AST.

However, the same code applied to TheIDE's Clang::Parse() method, with hard coded paths as is above,
clang_parseTranslationUnit() still fails to return a translation unit every single time.

I've tried both dynamic loading and static linking (LCLANG), tested both the above code & TheIDe on Linux 6.1.9/Clang 14 & 15

At this point I am almost sure that this is not a user-side problem, because I have installed vanilla ArchLinux on both a real hardware and on a VM (on windows machine), yet I get the same failure... Could this be a process env issue? Any ideas?

Best regards,
Oblivion

Subject: Re: 2022.3rc5
Posted by mirek on Wed, 08 Feb 2023 19:01:08 GMT
View Forum Message <> Reply to Message

Oblivion wrote on Wed, 08 February 2023 17:55Hello Mirek,
However, the same code applied to TheIDE's Clang::Parse() method, with hard coded paths as is above,
clang_parseTranslationUnit() still fails to return a translation unit every single time.

Just idea: Try to put it at the start of theide's GUI_APP_MAIN. If it works there, it must be something between that point and Clang::Parse. If not, it some linking or global constructor problem (or something like that).

Mirek

Subject: Re: 2022.3rc5
Posted by Oblivion on Wed, 08 Feb 2023 19:33:28 GMT
View Forum Message <> Reply to Message

Hello Mirek,

You're right,
I've just got an interesting result:

CONSOLE_APP_MAIN = clang_parseTranslationUnit() = OK
GUI_APP_MAIN = clang_parseTranslationUnit() = fails

Best regards,
Oblivion

Subject: Re: 2022.3rc5
Posted by Oblivion on Wed, 08 Feb 2023 19:50:17 GMT
View Forum Message <> Reply to Message

I think we're getting somewhere:

It appears to be a GTK::GUI_APP_MAIN related problem, as clangd snippet seems to work fine with NOGTK flag.

edit: Yes, I can confirm this.TheIde (NOGTK) with clangd runs just fine!

Best regards,
Oblivion

Subject: Re: 2022.3rc5
Posted by Oblivion on Thu, 09 Feb 2023 16:41:53 GMT
View Forum Message <> Reply to Message

HEllo Mirek,

The culprit seems to be gtk_init() function in InitGtkApp() function, line: 69.
If I call the clangd before that point, the parser works.

Best regards,
Oblivion

Subject: Re: 2022.3rc5
Posted by mirek on Thu, 09 Feb 2023 18:16:15 GMT
View Forum Message <> Reply to Message

Oblivion wrote on Thu, 09 February 2023 17:41HEllo Mirek,

The culprit seems to be gtk_init() function in InitGtkApp() function, line: 69.
If I call the clangd before that point, the parser works.

Best regards,
Oblivion

Cool but we definitely need to call it, right?

Maybe it alters argv/environment?

Or maybe something with threads?

Subject: Re: 2022.3rc5
Posted by Silvan on Thu, 09 Feb 2023 20:05:49 GMT
View Forum Message <> Reply to Message

maybe it is not related but I found a problem in the parser:
https://www.ultimatepp.org/forums/index.php?t=msg&th=120 25&start=0&
In short: parser give error in theide but build normally.
It depend on the machine theide is running, just after unpacking.

Bye

Subject: Re: 2022.3rc5
Posted by Oblivion on Thu, 09 Feb 2023 22:20:51 GMT
View Forum Message <> Reply to Message

I found the problem. It is env/LC_CTYPE variable.

My locale is set to tr_TR.UTF-8

For now, clangd doesn't seem to like it.
Changing LC_CTYPE to en_US did the trick for me, on all ArchLinux installations...

I am not entirely familiar with clangd's behaviour (yet) but if it requires LC_CTYPE to be ASCII en_xx values, maybe TheIDE should ensure that it is.

Best regards,
Oblivion

Subject: Re: 2022.3rc5
Posted by mirek on Fri, 10 Feb 2023 07:49:29 GMT
View Forum Message <> Reply to Message

Oblivion wrote on Thu, 09 February 2023 23:20
I found the problem. It is env/LC_CTYPE variable.

My locale is set to tr_TR.UTF-8
For now, clangd doesn't seem to like it.
Changing LC_CTYPE to en_US did the trick for me, on all ArchLinux installations...

I am not entirely familiar with clangd's behaviour (yet) but if it requires LC_CTYPE to be ASCII en_xx values, maybe TheIDE should ensure that it is.

Best regards,
Oblivion

Tried something like putting something like this at the start of theide GUI_APP_MAIN?

GUI_APP_MAIN {
#ifdef PLATFORM_POSIX
SetEnv("LC_TYPE", "en_US.UTF8")
#endif

Subject: Re: 2022.3rc5
Posted by Oblivion on Fri, 10 Feb 2023 20:45:40 GMT
View Forum Message <> Reply to Message

Quote:Tried something like putting something like this at the start of theide GUI_APP_MAIN?

GUI_APP_MAIN {
#ifdef PLATFORM_POSIX

SetEnv("LC_TYPE", "en_US.UTF8")
#endif

I've already tried this, but it doesn't work. Also, after thinking about it, I am not sure this is a good idea. LC_CTYPE variable should be set by OS user, not by TheIDE. (Unless there is a clang specific config option to set). In the meantime I'll use the desktop file to set it for TheIDE...

Best regards,
Oblivion

Subject: Re: 2022.3rc5
Posted by mirek on Fri, 10 Feb 2023 22:44:42 GMT
View Forum Message <> Reply to Message

Oblivion wrote on Fri, 10 February 2023 21:45Quote:Tried something like putting something like this at the start of theide GUI_APP_MAIN?

GUI_APP_MAIN {
#ifdef PLATFORM_POSIX
SetEnv("LC_TYPE", "en_US.UTF8")
#endif

I've already tried this, but it doesn't work. Also, after thinking about it, I am not sure this is a good idea. LC_CTYPE variable should be set by OS user, not by TheIDE. (Unless there is a clang specific config option to set). In the meantime I'll use the desktop file to set it for TheIDE...

Best regards,
Oblivion

U++ does not care about LC_TYPE, so does not TheIDE. If anything, TheIDE needs it en_us utf8...

Global constructor then (INITBLOCK) then?

Mirek

Subject: Re: 2022.3rc5
Posted by Oblivion on Sat, 11 Feb 2023 06:48:05 GMT
View Forum Message <> Reply to Message

Quote:Global constructor then (INITBLOCK) then?

Yes,

```
INITBLOCK {
 SetEnv("LC_CTYPE", "en_US.UTF-8");
}
```

This works.

Best regards,
Oblivion

Subject: Re: 2022.3rc5
Posted by Silvan on Mon, 13 Feb 2023 10:21:59 GMT
View Forum Message <> Reply to Message

Hi Oblivion,
I experienced a similar problem with the parser under Windows 10,
I tried to find out wicht enviromental variable could affect the behaviour of the ide, but with no success.
Have you an idea of the correspondent under windows of LC_CTYPE?
Thank you for any help,
Silvan

Subject: Re: 2022.3rc5
Posted by mirek on Sun, 19 Feb 2023 09:06:35 GMT
View Forum Message <> Reply to Message

Oblivion wrote on Sat, 11 February 2023 07:48Quote:Global constructor then (INITBLOCK) then?

Yes,

```
INITBLOCK {
 SetEnv("LC_CTYPE", "en_US.UTF-8");
}
```

This works.

Best regards,
Oblivion

It is now in master then...

Mirek
