
Subject: TheIDE PDB debugger now understands some U++ types

Posted by [mirek](#) on Fri, 22 Nov 2019 09:31:16 GMT

[View Forum Message](#) <> [Reply to Message](#)

So, this is for PDB debugger (one for Visual C++ compiler only) and it is limited, hardcoded and otherwise ugly implementation, plus there is some work left to do (support more types, support structured browsing), but in general I think it is already useful feature:

File Attachments

1) [Clipboard01.jpg](#), downloaded 761 times

Subject: Re: TheIDE PDB debugger now understands some U++ types

Posted by [unodgs](#) on Fri, 22 Nov 2019 14:07:10 GMT

[View Forum Message](#) <> [Reply to Message](#)

Very cool, I was missing that so many times :)

Subject: Re: TheIDE PDB debugger now understands some U++ types

Posted by [zsolt](#) on Fri, 22 Nov 2019 16:41:57 GMT

[View Forum Message](#) <> [Reply to Message](#)

Thanks a lot!

Subject: Re: TheIDE PDB debugger now understands some U++ types

Posted by [mirek](#) on Thu, 28 Nov 2019 11:25:35 GMT

[View Forum Message](#) <> [Reply to Message](#)

"recognized high-level types" are now supported in expressions:

File Attachments

1) [Clipboard02.jpg](#), downloaded 688 times

Subject: Re: TheIDE PDB debugger now understands some U++ types

Posted by [zsolt](#) on Fri, 29 Nov 2019 01:41:10 GMT

[View Forum Message](#) <> [Reply to Message](#)

How can I activate it?

I have compiled current Git version with MSC 2017 64 bit, but it seems to be working in the old

way.

Subject: Re: TheIDE PDB debugger now understands some U++ types

Posted by [mirek](#) on Fri, 29 Nov 2019 08:13:23 GMT

[View Forum Message](#) <> [Reply to Message](#)

zsolt wrote on Fri, 29 November 2019 02:41 How can I activate it?

I have compiled current Git version with MSC 2017 64 bit, but it seems to be working in the old way.

Should work out of box-> something is wrong.

Core should be trunk version for Value recognition.

Project must be compiled with MSC.

Not all U++ types are supported. Check ide/debuggers/Pretty.cpp for the list of types currently supported:

```
pretty.Add("Upp::Date", { 0, THISFN(PrettyDate) });
pretty.Add("Upp::Time", { 0, THISFN(PrettyTime) });
pretty.Add("Upp::ValueArray", { 0, THISFN(PrettyValueArray) });
pretty.Add("Upp::ValueMap", { 0, THISFN(PrettyValueMap) });
pretty.Add("Upp::Value", { 0, THISFN(PrettyValue) });
pretty.Add("Upp::String", { 0, THISFN(PrettyString) });
pretty.Add("Upp::WString", { 0, THISFN(PrettyWString) });
pretty.Add("Upp::Vector", { 1, THISFN(PrettyVector) });
pretty.Add("Upp::BiVector", { 1, THISFN(PrettyBiVector) });
pretty.Add("Upp::Array", { 1, THISFN(PrettyArray) });
pretty.Add("Upp::BiArray", { 1, THISFN(PrettyBiArray) });
pretty.Add("Upp::Index", { 1, THISFN(PrettyIndex) });
pretty.Add("Upp::VectorMap", { 2, THISFN(PrettyVectorMap) });
pretty.Add("Upp::ArrayMap", { 2, THISFN(PrettyArrayMap) });

pretty.Add("std::vector", { 1, THISFN(PrettyStdVector) });
pretty.Add("std::basic_string", { 1, THISFN(PrettyStdString) });
```

If none of those helps, we will have to dig into it... E.g. for starters, testcase would help.

Is debugged code 32 or 64?

Subject: Re: TheIDE PDB debugger now understands some U++ types

Posted by [zsolt](#) on Fri, 29 Nov 2019 12:47:02 GMT

[View Forum Message](#) <> [Reply to Message](#)

It was my fault. Works perfectly.
It's very easy to debug this way. Thank you!

Subject: Re: TheIDE PDB debugger now understands some U++ types
Posted by [mirek](#) on Fri, 29 Nov 2019 14:00:35 GMT
[View Forum Message](#) <> [Reply to Message](#)

Tree view now can be used to see any element:

File Attachments

1) [Clipboard01.jpg](#), downloaded 642 times

Subject: Re: TheIDE PDB debugger now understands some U++ types
Posted by [koldo](#) on Sat, 30 Nov 2019 15:42:21 GMT
[View Forum Message](#) <> [Reply to Message](#)

Thank you Mirek. Debugger is very improved.
How do you access to tree view?

Subject: Re: TheIDE PDB debugger now understands some U++ types
Posted by [mirek](#) on Sun, 01 Dec 2019 15:07:16 GMT
[View Forum Message](#) <> [Reply to Message](#)

koldo wrote on Sat, 30 November 2019 16:42 Thank you Mirek. Debugger is very improved.
How do you access to tree view?

Click the line with the value.

Subject: Re: TheIDE PDB debugger now understands some U++ types
Posted by [koldo](#) on Sun, 01 Dec 2019 20:54:52 GMT
[View Forum Message](#) <> [Reply to Message](#)

Very smart! :lol:

Let me show to the Forum the full picture.
The trick was to insist some times on moving the right vertical bar to the left.

File Attachments

1) [Sin título.png](#) , downloaded 524 times

Subject: Re: TheIDE PDB debugger now understands some U++ types

Posted by [mirek](#) on Tue, 03 Dec 2019 10:10:11 GMT

[View Forum Message](#) <> [Reply to Message](#)

Could not resist:

File Attachments

1) [Clipboard01.jpg](#) , downloaded 561 times

Subject: Re: TheIDE PDB debugger now understands some U++ types

Posted by [koldo](#) on Fri, 06 Dec 2019 09:21:26 GMT

[View Forum Message](#) <> [Reply to Message](#)

Really useful.

Could you add a copy to clipboard option to get the variable text?

Subject: Re: TheIDE PDB debugger now understands some U++ types

Posted by [Klugier](#) on Sat, 07 Dec 2019 22:12:39 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hello koldo,

Some time ago I implemented feature for GDB that copy the text to the clipboard from the tree. Optimally, we should have the same functionality for PDB with the shared code to reduced code maintenance. I also planned to add window that will open when you want to examine full string value inside TheIDE. This functionality should also be shared between debuggers. This is optimal implementation to not favor any of debugger. Especially that PDB is not available on Linux.

I think above improvements are great, but they are not present with the GDB back-end. Mirek, do you think we can port this functionality without much effort (by extracting common code?)?

Sincerely,
Klugier

Subject: Re: TheIDE PDB debugger now understands some U++ types

Posted by [mirek](#) on Sun, 08 Dec 2019 07:40:07 GMT

[View Forum Message](#) <> [Reply to Message](#)

Klugier wrote on Sat, 07 December 2019 23:12

I think above improvements are great, but they are not present with the GDB back-end. Mirek, do you think we can port this functionality without much effort (by extracting common code?)?

Unfortunately not. That is one of reasons I have resisted these features for so long.

GDB and PDB are completely different beasts. GDB is just issuing text commands to gdb binary via text pipe and interpreting results.

PDB code is really a full featured debugger. It is using Win32 debugging API directly and symbol server to really understand the layout of objects, directly reads debuggee memory etc... That are actually requirements to make this kind of functionality possible.

This definitely possible to binaries produced by GCC toolchain too. But it is a LOT of work, probably would involve thorough study of GDB sources and about a year to complete.

Mirek

Subject: Re: TheIDE PDB debugger now understands some U++ types

Posted by [Novo](#) on Sun, 08 Dec 2019 15:27:44 GMT

[View Forum Message](#) <> [Reply to Message](#)

mirek wrote on Sun, 08 December 2019 02:40

This definitely possible to binaries produced by GCC toolchain too. But it is a LOT of work, probably would involve thorough study of GDB sources and about a year to complete.

IMHO, it is easier to use LLDB, which is available as a lib.

"The LLDB debugger APIs are exposed as a C++ object oriented interface in a shared library. The lldb command line tool links to, and uses this public API."

Subject: Re: TheIDE PDB debugger now understands some U++ types

Posted by [mirek](#) on Sat, 14 Dec 2019 14:20:18 GMT

[View Forum Message](#) <> [Reply to Message](#)

koldo wrote on Fri, 06 December 2019 10:21 Really useful.

Could you add a copy to clipboard option to get the variable text?

Done.

Subject: Re: TheIDE PDB debugger now understands some U++ types

Posted by [koldo](#) on Sat, 14 Dec 2019 15:33:09 GMT

[View Forum Message](#) <> [Reply to Message](#)

Cool! It runs, thank you Mirek.

The last features you are adding to debugger are really useful, and they have helped me to detect errors.
