Subject: Looking for windows compiler possibilities
Posted by mirek on Wed, 01 Jul 2015 12:13:38 GMT
View Forum Message <> Reply to Message

I am becoming uneasy about windows releases.

It would be great to

a) move to C++11
b) provide compiler toolchain as part of release

Current 'default' way is to install Win7 SDK. That is good, but not C++11...

Later Microsoft compilers need to be installed with Visual Studio and (AFAIK, please correct me) with separate SDK. And, of course, we cannot ship them with U++. Maybe it would be possible to do some sort of automated web install?

clang-cl (Microsoft compatible) seems to almost work, but is not quite mature now.

Then there is mingw-w64. This is quite close to what we need, but mingw still is slower than msc/clang and resulting binaries are quite larger than with MSC. Another issue is that most 3rd party libraries (like DB clients or openssl) are coming with MSC compatible headers/libraries. Might be fixed by providing all imaginable libraries in the release (nowadays, I would not care if U++ installer has 300MB).

It also seems possible to use clang with mingw. It however seems that clang-mingw has problems with exception in 64bit mode.

Any thoughts or suggestions?

Subject: Re: Looking for windows compiler possibilities
Posted by cbpporter on Wed, 01 Jul 2015 12:33:58 GMT
View Forum Message <> Reply to Message

I've been investigating this myself the past few weeks and found that:

1. clang++ seems to be just a compiler. Not sure if 100% mature, but probably good enough as a compiler. It still needs an SDK. I managed to compile C++ with it, but I had to use the MS SDK and point clang to the headers and libs. I do not know about clang-cl, maybe that one is a full SDK?
2. I'm still wrestling with mingw-w64 and getting it to install. I managed to get it though cygwin, but that seems a bit of an overkill for me.

Subject: Re: Looking for windows compiler possibilities
Posted by mirek on Wed, 01 Jul 2015 14:17:11 GMT

View Forum Message <> Reply to Message

By clang-cl I mean Microsoft SDK compatible clang - the one you have used.

There is another win variant, which is mingw-w64 compatible.

Subject: Re: Looking for windows compiler possibilities
Posted by koldo on Thu, 02 Jul 2015 07:11:57 GMT

View Forum Message <> Reply to Message

I use MSC12-64 bits as main compiler and TDM-GCC-64 as a second option to check code compatibility with GCC and to have a "second opinion" when I get cryptic error messages from MSC.

It would be great if U++ installer could include a GCC compiler. TDM compiles all U++ sources without problems. Options:
- TDM bundled with U++: Installer could be too big. In the past I prepared a zip with all the basics of MinGW. However, as GCC goes changing this could not be efficient.
- U++ installer asks the user to install TDM and if yes, downloads and calls the installer.
- U++ installer downloads TDM files in U++ compiler folder. Possibly the best option.

Subject: Re: Looking for windows compiler possibilities
Posted by mirek on Sun, 05 Jul 2015 05:52:44 GMT

View Forum Message <> Reply to Message

Interesting overview here:

https://wiki.qt.io/MinGW-64-bit

Now I have rechecked Mingw-w64. It works fine, more or less, the problem is speed - while it takes 30s for MSC to recompile debug theide, for mingw it is twice as much (~65s). Worse, significant part is spent in linker (~15s), which has to be done even on recompiling the single file... :(

It is really mess. The only chance is clang...

Mirek

Subject: Re: Looking for windows compiler possibilities
Posted by Klugier on Sun, 05 Jul 2015 10:55:42 GMT

View Forum Message <> Reply to Message

Hello Mirek,

In my opinion speed is not big problem. The real problem on Windows platform is that we don't supply any compile. For new users it is big think to have compile out of the box. Even if they didn't link Windows SDK. They can try Core or even start coding in pure c++ with standard library. If they begin to like TheIDE they start looking around more advanced topics. So, entry barrier shouldn't be that big.

You can try clang, but even if it does not meet your expectations. You should consider bundle any compiler. More advanced users will always be able to link MSC...

Sincerely,
Klugier

Subject: Re: Looking for windows compiler possibilities
Posted by cbpporter on Sun, 05 Jul 2015 14:36:29 GMT
View Forum Message <> Reply to Message

Clang, but don't we need an SDK?

Subject: Re: Looking for windows compiler possibilities
Posted by mirek on Sun, 05 Jul 2015 18:43:04 GMT
View Forum Message <> Reply to Message

cbpporter wrote on Sun, 05 July 2015 16:36Clang, but don't we need an SDK?

mingw-w64 should be usable as SDK for clang.

It is sort of complicated and I am still missing details. That is why I have started this thread.

Anyway, it seems like clang can be compiled to be compatible with MSC SDK, or can be compiled to be compatible with mingw.

Now to make things even more fun, there are about 10 flavors of mingw. You can have 64/32 bit. You can have SjLj/Dwarf2/SEH exceptions (but to make it more fun, dwarf only with 32 bit and SEH only with 64 bit... :). Then you can have Win32 threads (and lack C++11 threading facilities), or POSIX threads, which are slow... :)

And you have to match the clang with all this mess somehow.. :)

Now I will probably concentrate on this in coming weeks. But in the same time, it would be great if somebody else tried too...

Mirek

Subject: Re: Looking for windows compiler possibilities
Posted by koldo on Mon, 06 Jul 2015 06:51:17 GMT
View Forum Message <> Reply to Message

For the sake of simplicity I would use the simplest to install and most robust compiler in U++ Windows installer.

I think that first time user needs simplicity and to see that all examples compile and work just out of the box, although it would require more time to compile than other options. In the same installer or in the download page I would put two options:
- The first one: install and run (preferred)
- The second one, for advanced users, would go to the place where compiler directories are searched and selected.

Subject: Re: Looking for windows compiler possibilities
Posted by mirek on Mon, 06 Jul 2015 08:44:28 GMT
View Forum Message <> Reply to Message

koldo wrote on Mon, 06 July 2015 08:51For the sake of simplicity I would use the simplest to install and most robust compiler in U++ Windows installer.

I think that first time user needs simplicity and to see that all examples compile and work just out of the box, although it would require more time to compile than other options. In the same installer or in the download page I would put two options:
- The first one: install and run (preferred)
- The second one, for advanced users, would go to the place where compiler directories are searched and selected.

That would probably solve first part of problem (newbie), but not exactly second one (we would like C++11).

With clang, there is a chance to solve both issues IMO.

Mirek

Subject: Re: Looking for windows compiler possibilities
Posted by cbpporter on Mon, 06 Jul 2015 08:57:32 GMT
View Forum Message <> Reply to Message

I'll try to help.

Currently, all I managed to do is compile same basic code with clang and MSC SDK. Was reasonably fast, probably a bit slower than MSC.

Could you provide the exact link to MINGW-w64 that you are using so I can install that and use it as an SDK?

And does it support 32bit?

Edit: typo.

Subject: Re: Looking for windows compiler possibilities
Posted by mirek on Mon, 06 Jul 2015 09:35:21 GMT
View Forum Message <> Reply to Message

cbpporter wrote on Mon, 06 July 2015 10:57I'll try to help.

Currently, all I managed to do is compile same basic code with clang and MSC SDK. Was reasonably fast, probably a bit slower than MSC.

Could you provide the exact link to MINGW-w64 that you are using so I can install that and use it as an SDK?

And does it support 32bit?

I wish I knew...

From what I have gathered so far, I was unable to find recent version of clang prebuilt. There is only a couple of tutorials to build it....

AFAIK clang is 'mated' with mingw distribution during build process. Not all mingws are supported though...

Mirek

Subject: Re: Looking for windows compiler possibilities
Posted by cbpporter on Mon, 06 Jul 2015 09:41:22 GMT
View Forum Message <> Reply to Message

The one that worked for me for that single test was from: http://llvm.org/releases/download.html

As for the mingw-w64 I got, it was from cygwin which is far from ideal.

BTW, how about updating the installer to check for prerequisites? And add support for downloading and running the SDK installer or any of the VS Express editions (in addition to detecting the installed ones).

It is a lot of work, I know, but if mingw won't work out...

Subject: Re: Looking for windows compiler possibilities
Posted by mirek on Mon, 06 Jul 2015 11:09:19 GMT
View Forum Message <> Reply to Message

Well, but AFAIK that one particular clang is not compatible with mingw SDK...

Anyway, the idea about prerequisites is now bad - I just did not knew it is possible in Windows... :) (is it possible, right?)

Subject: Re: Looking for windows compiler possibilities
Posted by mirek on Mon, 06 Jul 2015 11:10:55 GMT
View Forum Message <> Reply to Message

cbpporter wrote on Mon, 06 July 2015 11:41The one that worked for me for that single test was from: http://llvm.org/releases/download.html

BTW, funny trouble about this clang is that U++ considers it C++11 compatible (rightfully), but C++11 library is missing, so U++ sources cannot be compiled without tweaks...

Mirek

Subject: Re: Looking for windows compiler possibilities
Posted by cbpporter on Mon, 06 Jul 2015 11:18:31 GMT
View Forum Message <> Reply to Message

I compiled legitimate C++ code but did not use C++ headers, so I can confirm that C++ works fine (I did have to tweak my code since clang is far more restrictive than MSC) and the windows .lib from MS SDK are compatible with clang.

Subject: Re: Looking for windows compiler possibilities
Posted by cbpporter on Mon, 06 Jul 2015 11:26:28 GMT
View Forum Message <> Reply to Message

MINGW version are so confusing.

http://sourceforge.net/projects/mingw-w64/

Is this the fork that is semi-reverse engineered of the old abandoned MINGW? The one we are supposed to be using? Anyway, it fails to install for me. The installer might not support proxy.

Subject: Re: Looking for windows compiler possibilities
Posted by cbpporter on Tue, 07 Jul 2015 08:16:15 GMT
View Forum Message <> Reply to Message

mirek wrote on Mon, 06 July 2015 14:09Well, but AFAIK that one particular clang is not compatible with mingw SDK...

Anyway, the idea about prerequisites is now bad - I just did not knew it is possible in Windows... :) (is it possible, right?)

Sure, it is possible. Like I said, it won't be easy.

How about if we try it in steps. How about if the build methods detector detects clang? Then we add a check box with "override compiler" to the right of the build methods dropdown? So we can build a package with MSC10 or MSC10 with Clang.

And how about we leave that check box on on as default for a couple of weeks and see if Clang with MS SDK is up to par? Production ready?

Subject: Re: Looking for windows compiler possibilities
Posted by Sgifan on Tue, 07 Jul 2015 17:01:31 GMT
View Forum Message <> Reply to Message

Why not propose (and adapt The IDE to) using VS2013 community edition

It would be perfect in most situation (legalize wise) and it is C++11.

I know that currently the IDE does not work with it, at least that's what I did experience

Subject: Re: Looking for windows compiler possibilities
Posted by mirek on Wed, 08 Jul 2015 07:24:00 GMT
View Forum Message <> Reply to Message

Sgifan wrote on Tue, 07 July 2015 19:01Why not propose (and adapt The IDE to) using VS2013 community edition

It would be perfect in most situation (legalize wise) and it is C++11.

Not "enough" C++11 unfortunately AFAIK. Things required for pick/clone to work are not there. Anyway, perhaps it is worth workaround.

Quote:
I know that currently the IDE does not work with it, at least that's what I did experience

Autodetect might not work, but I believe that after fixing paths, it should work just fine.

Generally, I agree that we should support this out of box. But it would still be nice to have "complete download".

Mirek

Subject: Re: Looking for windows compiler possibilities
Posted by cbpporter on Fri, 10 Jul 2015 08:20:46 GMT
View Forum Message <> Reply to Message

[quote]mirek wrote on Wed, 08 July 2015 10:24Sgifan wrote on Tue, 07 July 2015 19:01Why not propose (and adapt The IDE to) using VS2013 community edition

It would be perfect in most situation (legalize wise) and it is C++11.

Not "enough" C++11 unfortunately AFAIK. Things required for pick/clone to work are not there. Anyway, perhaps it is worth workaround.

Isn't && support good enough in the latest free VS?

BTW, are there plans to implement full r-value for String and other classes, like in the case of s + func1() + func2(), or would String not benefit from this since it does RC and other stuff?

Quote:
Quote:
I know that currently the IDE does not work with it, at least that's what I did experience

Autodetect might not work, but I believe that after fixing paths, it should work just fine.

Generally, I agree that we should support this out of box. But it would still be nice to have "complete download".

Mirek
I can confirm that autodetect does not work. Since I installed this version I need to not use the highest version VS available.

Subject: Re: Looking for windows compiler possibilities
Posted by mirek on Fri, 10 Jul 2015 11:11:00 GMT
View Forum Message <> Reply to Message

[quote title=cbpporter wrote on Fri, 10 July 2015 10:20]Quote:mirek wrote on Wed, 08 July 2015 10:24Sgifan wrote on Tue, 07 July 2015 19:01Why not propose (and adapt The IDE to) using VS2013 community edition

It would be perfect in most situation (legalize wise) and it is C++11.

Not "enough" C++11 unfortunately AFAIK. Things required for pick/clone to work are not there. Anyway, perhaps it is worth workaround.

Isn't && support good enough in the latest free VS?

&& is not enough. We need 'default' for constructors to work correctly.

See http://www.ultimatepp.org/srcdoc$Core$pick_$en-us.html, Composition.

Quote:
BTW, are there plans to implement full r-value for String and other classes, like in the case of s + func1() + func2(), or would String not benefit from this since it does RC and other stuff?

I am considering it, but performance gains will be negligible. In fact, such things only make sense if you do not have to maintain C++03 compatibility (which is the point of this thread).

Quote:
I can confirm that autodetect does not work. Since I installed this version I need to not use the highest version VS available.

Maybe you can fix this and post the patch. Should be easy...

Mirek

Subject: Re: Looking for windows compiler possibilities
Posted by cbpporter on Fri, 10 Jul 2015 13:26:48 GMT
View Forum Message <> Reply to Message

mirek wrote on Fri, 10 July 2015 14:11

Maybe you can fix this and post the patch. Should be easy...

Mirek

Yes, I haven't truly contributed a lot of patches lately :).

Anyway, the problem is that Microsoft loves to change the directory layout around for no good reason. Simply checking that msc.sdk + "\\lib" exists on disk won't help.

Here is a potential way that fixed bugs for me:

```
void TestLib() {
 if (create) {
  if (FindFile(sdk + "\\lib\\*.lib"))
   sdklib = sdk + "\\lib";
  if (FindFile(sdk + "\\lib\\x86\\*.lib"))
   sdklib = sdk + "\\lib\\x86";
  else if (FindFile(sdk + "\\lib\\win8\\um\\x86\\*.lib"))
   sdklib = sdk + "\\lib\\win8\\um\\x86";
  else if (FindFile(sdk + "\\lib\\winv6.3\\um\\x86"))
   sdklib = sdk + "\\lib\\winv6.3\\um\\x86";
 }
 if (create64) {
  if (FindFile(sdk + "\\lib\\*.lib"))
   sdklib64 = sdk + "\\lib";
  else if (FindFile(sdk + "\\lib\\x64\\*.lib"))
   sdklib64 = sdk + "\\lib\\x64";
  else if (FindFile(sdk + "\\lib\\win8\\um\\x64\\*.lib"))
   sdklib64 = sdk + "\\lib\\win8\\um\\x64";
  else if (FindFile(sdk + "\\lib\\winv6.3\\um\\x64"))
   sdklib64 = sdk + "\\lib\\winv6.3\\um\\x64";
 }
}
```

I had to add the sdklib fields to actually determine where the lib files are. I have a ton of VS version installed, and all use different paths...

I'm downloading VS 2015 which for some stupid reason is MSC14 to test that too...

Subject: Re: Looking for windows compiler possibilities
Posted by cbpporter on Fri, 10 Jul 2015 14:25:36 GMT
View Forum Message <> Reply to Message

And MSC14 changes things around again:
http://blogs.msdn.com/b/vcblog/archive/2015/03/03/introducin g-the-universal-crt.aspx

Subject: Re: Looking for windows compiler possibilities
Posted by mirek on Sat, 11 Jul 2015 08:41:16 GMT
View Forum Message <> Reply to Message

cbpporter wrote on Fri, 10 July 2015 15:26mirek wrote on Fri, 10 July 2015 14:11

Maybe you can fix this and post the patch. Should be easy...

Mirek

Yes, I haven't truly contributed a lot of patches lately :).

Anyway, the problem is that Microsoft loves to change the directory layout around for no good reason. Simply checking that msc.sdk + "\\lib" exists on disk won't help.

Here is a potential way that fixed bugs for me:

```
void TestLib() {
  if (create) {
   if (FindFile(sdk + "\\lib\\*.lib"))
    sdklib = sdk + "\\lib";
   if (FindFile(sdk + "\\lib\\x86\\*.lib"))
    sdklib = sdk + "\\lib\\x86";
   else if (FindFile(sdk + "\\lib\\win8\\um\\x86\\*.lib"))
    sdklib = sdk + "\\lib\\win8\\um\\x86";
   else if (FindFile(sdk + "\\lib\\winv6.3\\um\\x86"))
    sdklib = sdk + "\\lib\\winv6.3\\um\\x86";
  }
  if (create64) {
   if (FindFile(sdk + "\\lib\\*.lib"))
    sdklib64 = sdk + "\\lib";
   else if (FindFile(sdk + "\\lib\\x64\\*.lib"))
    sdklib64 = sdk + "\\lib\\x64";
   else if (FindFile(sdk + "\\lib\\win8\\um\\x64\\*.lib"))
    sdklib64 = sdk + "\\lib\\win8\\um\\x64";
   else if (FindFile(sdk + "\\lib\\winv6.3\\um\\x64"))
    sdklib64 = sdk + "\\lib\\winv6.3\\um\\x64";
  }
 }
```

Have you tested inside theide code? Can you provide a tested patch?

Mirek

Subject: Re: Looking for windows compiler possibilities
Posted by cbpporter on Wed, 15 Jul 2015 10:45:19 GMT
View Forum Message <> Reply to Message

Not yet, since I am running into the problem of TheIDE nightly not detecting the SDK for MSC11 and 12, but when building with 10, the "ide" I'm running fromwithin TheIDE detects the sdk just fine.

Subject: Re: Looking for windows compiler possibilities
Posted by cbpporter on Thu, 30 Jul 2015 11:22:45 GMT
View Forum Message <> Reply to Message

Well I manged to fully mess up my VS and SDKs and nothing is working any more. Not even VS can find the SDK. I need to fully clean my system.

Hopefully a Windows reinstall won't be needed :).

Subject: Re: Looking for windows compiler possibilities
Posted by cbpporter on Mon, 03 Aug 2015 08:52:44 GMT
View Forum Message <> Reply to Message

cbpporter wrote on Thu, 30 July 2015 14:22Well I manged to fully mess up my VS and SDKs and nothing is working any more. Not even VS can find the SDK. I need to fully clean my system.

Hopefully a Windows reinstall won't be needed :).
Nope, needed to reinstall Windows.

Even after cleanup, VS refused to install correctly so I had to wipe the OS.

It is a really bad idea to install every VS.

Next time I'm doing it in a VM.