Subject: How to compile with MSVC... is it possible?
Posted by novak on Mon, 04 Apr 2022 18:31:13 GMT
View Forum Message <> Reply to Message

I ran the MSVC command prompt and copied the path, include and lib directories from 'set', and added to 'Build Methods' in TheIDE. Also added the release compile flags /O2, /MT, /GR- and /D_HAS_EXCEPTIONS=0 and start my attempt to compile the UWord sample.
I quickly figured out U++ needs RTTI and it also depends on exceptions, so I removed the last two compile options. I wonder if U++ really needs exceptions or this can be removed?
But there are many compile errors. For example:

core/Defs.h line 214:
```
template <typename T>
auto pick(T&& x) noexcept -> decltype(std::move(x)) { return std::move(x); }
```
- eror C3646: 'noexcept' : unknown override specifier
- error C2143: syntax error : missing ';' before '->'
... and a few more

core/Fn.h line 13:
```
template <class T>
constexpr const T& min(const T& a, const T& b)
{
 return a < b ? a : b;
}
```
- error C2143: syntax error : missing ';' before 'const'
- error C4430: missing type specifier - int assumed. Note: C++ does not support default-int
... and a few more

I use VS2013. Is it too old? I'm using wxWidgets and I'm looking around a bit.

Subject: Re: How to compile with MSVC... is it possible?
Posted by Klugier on Tue, 05 Apr 2022 08:54:39 GMT
View Forum Message <> Reply to Message

Hello,

I think it is too old. 2013 is probably do not support c++14 and we requiere that standard. Could you try installing latest Visual Studio 2022 (Community Edition) and try again?

Klugier

Subject: Re: How to compile with MSVC... is it possible?
Posted by novak on Tue, 05 Apr 2022 18:53:19 GMT
View Forum Message <> Reply to Message

That's a bit of a hassle as Visual Studio is registerware and it, since some time, won't load the login page that uses the built-in Internet Explorer. They probably do that on purpose to get you to update your OS.
So I have to install this on a VM with windows 10 and meticuously copy over the needed files to make it portable and usable.

But I'm curious, what is the executable size for an average Ultimate++ program like the UWord sample? And without exceptions? Or are they really needed?

Subject: Re: How to compile with MSVC... is it possible?
Posted by Klugier on Sun, 10 Apr 2022 17:52:02 GMT
View Forum Message <> Reply to Message

Hello,

In order to use U++ you do not need to launch Visual Studio. Installation with C++ components is sufficient. After that you should launch TheIDE and use auto-setup for MSVC detection. Also, we bundle Clang compiler with standard Windows distribution, so possessing Visual Studio is optional.

What is your current operating system and what is the reason why you do not want to update your OS?

Klugier

Subject: Re: How to compile with MSVC... is it possible?
Posted by novak on Sun, 10 Apr 2022 21:00:13 GMT
View Forum Message <> Reply to Message

You need to run Visual Studio at least once, and register it, and let it set up things before the compiler will work. Also it's better to do this installation on another computer and extract it from there to keep things clean because it has it's tentacles everywhere on the disk and in the registry. It's not too much of a problem but I dont feel like doing this now and I'm happy for the moment with MSVC 13 and wxWidgets and Clion on Windows 7.

I like small filesizes, a minimal wxWidgets program is, after removing the obvious bloat (like exceptions), 1.5 MB. With PureBasic, the same program is 50kb. So there must be a lot of things in between there that wxWidgets has that I probably don't need. And in any case with Clang there is a 4 MB "entry fee" just for trying.

But after they start waving this Ukrainian flag on the website (this business is just a continuance of the Covid cult) together with some dumb and ignorant commentary, I'm not too proud to be using wxWidgets and I wonder if there's anything better, but I haven't found anything yet... the other

options, including U++, all have too many downsides, and I'm probably not capable of writing my own "OS abstraction library".