
Subject: What is the difference between learning U++ and Learning C++ through Microsoft Visual Studio?

Posted by [RaySynthesyzer](#) on Fri, 07 Jun 2019 04:44:11 GMT

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

Subject: Re: What is the difference between learning U++ and Learning C++ through Microsoft Visual Studio?

Posted by [Xemuth](#) on Fri, 07 Jun 2019 19:42:36 GMT

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

Hello RaySynthesyzer,

This question seems to dont have any sense since U++ can be used with Microsoft Visual C++ compiler because U++ is a framework.

But, I think your question is more like "Should I choose 'U++ with MinGW compiler' OR 'standard C++ with Microsoft Visual Studio compiler' to learn C++ ? And Why ?"

Well, based on my 11 Months of experiencing U++ (using it atless 2 day a week) and having started with basic knowledge of C as well as a solid foundation of OOP(Object-oriented programming). I would say :

If you have bit knowledge about programmation, I would say U++, it will allow you to reach really good result on GUI/Console Application without much effort.

Moreover, when the framework is known, you will be able to build complex apps really quickly, and most of the time they would be Windows/Linux compatible !

If you have no idea about programmation and you planned to learn it on C++ then, maybe you should go on standard C++ (no matter the compiler (MinGW/VC++))(no matter the IDE(U++ ide/Code::Block/Microsoft Visual Studio)) and start slowly by using standard C++ lib(Wiki here).

Why I recommand you to start with Standard Lib ? Because, I think that knowing the standard before using framework can only be benefit, Mostly if you are beginner.

there is also a point on performance, U++ is really more performance than the Standard on certain aspect : U++ Core vs standard C++ library But I'm not the right person to spoke about it.

But, the most important point is that Upp and thelde are f**king awesome! :d

If one of Upp Member noticed that's I had said bullshit, made me notice !

Hope It could help.

Best regard.

Subject: Re: What is the difference between learning U++ and Learning C++ through Microsoft Visual Studio?

Posted by [Novo](#) on Sat, 08 Jun 2019 01:42:05 GMT

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

Microsoft Visual Studio is an IDE (an editor + other tools) shipped with Visual C/C++ compiler, which can be downloaded separately and is free to use. (More is here, but you better install the latest version of build tools)

U++ is also an IDE, which is shipped with a different compiler (MinGW)

If you install build-tools-for-visual-studio on your machine, you can tell U++ to start using them. In order to do that launch TheIDE and go to Setup --> "Automatic build methods setup", if you want to add your MSVC compiler manually, go to Setup --> "Build methods".

Microsoft Visual Studio is a monster, huge and slow. U++ is tiny and super fast.

Subject: Re: What is the difference between learning U++ and Learning C++ through Microsoft Visual Studio?

Posted by [RaySynthesizer](#) on Sat, 08 Jun 2019 05:44:17 GMT

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

Dear Novo and Xemuth,

Thanks a lot, that gave me some insight.

I basically need to work on a mathematical modelling project, and I have been requested to develop them in U++, as it has BSD licencing and commercializing it is easier and cheaper. Also, they like that it can be used in both Linux and Windows.

However, I have my concerns. I haven't been able to find much info about U++ anywhere except this forum and Wikipedia. Many of my friends who work and develop software using C++, haven't even heard of U++.

So, will I be able to find the libraries, and support, that I would need to develop somewhat complicated mathematical algorithms if I use U++? if not will I need to develop them myself? Sounds difficult to me. Also, given the fact that I don't have any experience in C/C++ yet, I am worried that if I have doubts in programming using U++, I will be pretty much stranded.

In this project, as far as I've been told the end product needs to be a GUI type .exe file that the customer should be able to use. Based on your responses, I believe that later part of the project will be easier using U++.

Is it possible for doing something like developing the back-end / algorithm in another software, and then using U++ to make the GUI? if it possible, what are some of the popular open source softwares (popular in terms of having open source libraries) that can be used for this purpose?

Also, I am curious as to why a software that seems to be so handy isn't popular or at least well known.

I would really appreciate some advise on these matters. Thanks a lot again, your replies were really helpful.

Regards,
Ray

Subject: Re: What is the difference between learning U++ and Learning C++ through Microsoft Visual Studio?

Posted by [mr_ped](#) on Sat, 08 Jun 2019 08:24:41 GMT

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

RaySynthesizer wrote on Sat, 08 June 2019 07:44Dear Novo and Xemuth,

So, will I be able to find the libraries, and support, that I would need to develop somewhat complicated mathematical algorithms if I use U++? if not will I need to develop them myself? Sounds difficult to me. Also, given the fact that I don't have any experience in C/C++ yet, I am worried that if I have doubts in programming using U++, I will be pretty much stranded.

When writing application in "U++", you are writing in regular C++ language, i.e. theoretically you can use any other C++ library/API/tools together (but...).

U++ is platform/GUI/core/... framework, i.e. set of many helpful utilities and libraries (for C++). But it's so much of them, that the "but" part is, that default U++ project does use even the U++ memory allocator or it's own implementation of String class, etc.. Which may cause some conflicts or difficulties when you would try to use other complex C++ library or just DLL file in windows. If you are experienced C++ programmer, you can deal with these issues (the U++ libraries can be reconfigured a lot, to avoid some conflicts, like you can switch back to standard memory allocator, etc.), but if you are new to both C++ and U++, it may be too much. Then again if you are new to C++, it may be even too much to create some GUI application quickly, C++ without framework is actually not very good fit for that. Then starting with U++ may get you productive much quicker, for the price of being a bit more locked to it, until you learn also lot of deep details of C++ and how U++ is written.

So it depends a lot, what kind of other libraries you want to use. If they are C++ libraries focused on calculations (like matrix math, CFD simulation, etc) and distributed as source code, it will very likely fit together with U++ nicely or with just some minor extra configuration (that's ideal case for U++ usage). If it's for example something like windows-GUI specific extension or DLL files, or just binary closed source libraries, it may be more tricky, and you will lose the cross-platform aspect any way.

Also while the language is regular C++, U++ provides own alternatives to almost everything in the STL (Standard Template Library), which renders many C++ tutorials less usable (as C++ is usually shown together with STL, because bare C++ is sort of difficult to showcase). But most of the U++ alternatives are really similar to STL in their principle, just the naming of API is somewhat different, etc, so I wouldn't be that much afraid of U++. Still learning also a bit of standard C++ and STL helps, to make you better recognize where you are using U++ framework.

Quote:In this project, as far as I've been told the end product needs to be a GUI type .exe file that the customer should be able to use. Based on your responses, I believe that later part of the project will be easier using U++.

Doing *some* GUI "exe" in U++ can be really a very rapid process, if you know U++ well, you can be probably as productive as in many script languages or web development. It will take a bit more effort to polish the GUI to certain design (if you have very precise idea how the GUI should look and the default look is not good enough), then again it's C++, so if you spend enough effort, you can create pretty much anything what the target platform is capable of (while things written in HTML+CSS may be faster to get some particular design, but you may hit limits of web browser functionality and not being capable to implement some functionality without extending the browser itself).

You should try to install U++ and go through the GUI examples, check their source and the result, maybe you will like it instantly, to me it feels as quite good design of framework.

Quote:Is it possible for doing something like developing the back-end / algorithm in another software, and then using U++ to make the GUI? If it possible, what are some of the popular open source softwares (popular in terms of having open source libraries) that can be used for this purpose?

Again, it's C++, so the question is not if you can (of course you can), but how difficult it is.

Quote:Also, I am curious as to why a software that seems to be so handy isn't popular or at least well known.

Marketing? Obscurity? And U++ is mostly work of 2-3 authors, although being developed and improved over decade, so while it covers many common problems of app development, it isn't as huge as for example Qt5 (which is being developed by tens of core paid developers with help of hundreds of community devs). It also means U++ doesn't take hours to build from sources and you can actually check the sources yourself (in a meaningful way, you can also check Qt5 framework sources, but it's many more source code lines and lot more complex project, so it takes more time to decipher Qt5 internals in case you are looking for some framework bug, than U++ internals).

Quote:I would really appreciate some advise on these matters. Thanks a lot again, your replies were really helpful.

If you are also new to C++, this may be more difficult, than usage of U++. C++ is sort of

demanding programming language, it takes some time and experience (usually couple of years) before you will write your first code which is reasonably correct and complex. Most of the programmers in their early years of C++ write lot of useless or even plain wrong stuff, which works mostly by accident, because they don't understand well computers and C++ is quite low level language. In this regard, if you will stick to U++ tutorials and get the feel for U++ style of coding, it may actually give you huge jumpstart into C++ programming world, avoiding lot of inferior quality tutorials where even the author of tutorial doesn't know what he is doing...

Subject: Re: What is the difference between learning U++ and Learning C++ through Microsoft Visual Studio?

Posted by [RaySynthesizer](#) on Sat, 08 Jun 2019 18:47:00 GMT

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

Dear Mr. Ped,

Thank you sincerely for putting in the effort to write out a descriptive reply, I really appreciate it.

I am a mechanical engineering student - I have been able to understand a lot of what you are saying, but not completely, probably due to my lack of experience with coding(I have only ever used MATLAB). As of now, I will be going to some programmers who have built and an entire interface using U++ in a few days and learn to use TheIDE and U++ from them, as directed by my superiors. In this light, reading your response felt relieving as I feel little more certain, and I believe I'll be putting more effort into learning U++ now, then I would have otherwise put.

Supposedly this project will take a while... so I'll be counting on you, others like Nova and Xemuth and this forum to guide me if I get struck somewhere along. Thanks a lot again.

Warm Regards,

Ray

Subject: Re: What is the difference between learning U++ and Learning C++ through Microsoft Visual Studio?

Posted by [koldo](#) on Sat, 08 Jun 2019 19:11:51 GMT

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

Hi Ray

I fully agree with my U++ colleagues. In addition if you are interested in handling MATLAB files I am developing a simple lib to handling .mat files.

I propose you to begin with a very simple, but useful project. The simpler is your project, the easier will be for us to support you.

Have it clear, if you show us simple cases and do the right questions, in U++ you don't feel alone.

Subject: Re: What is the difference between learning U++ and Learning C++ through Microsoft Visual Studio?

Posted by [peterh](#) on Sun, 09 Jun 2019 08:14:19 GMT

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

It is -from my beginners experience- useful to use both, U++ and Visual Studio.

U++ is full of examples and probably you first open one of the bazaar, tutorial or reference examples, and build it and try it.

This is easy to do in TheIDE.

If Visual C++ is installed, Upp can use the VC++ compiler to build the example projects.

You can use TheIDE to browse and read the source code.

You can also use the debugger in U++ to learn about the source code.

If the program was compiled with VC++, then you can also use VS to debug, because VS can read the debug information.

Using TheIDE has these advantages:

You have a GUI designer.

A lot of documentation is inbuilt to the sourcecode of U++ and you can only access this documentation in TheIDE.

Using VS has these Advantages:

U++ uses generic programming and overloaded operators a lot.

This to an extend, that even a skilled C++ Programmer cannot read an U++ program, because he doesn't know the meaning of these overloaded operators. He has first to learn the concepts of U++, before he can read the program. Now, if he cannot read it, he cannot write it. ;)

For this reason it is good to start with the examples.

If you use VS 17 or 19, then you can open the main.cpp of an U++ project. VS will analyze and crossreference the source code (be patient, it happens in background and takes some time) and if you point to an unknown operator VS displays nicely the type of the operator and can show its definition. TheIDE (the IDE of U++) cannot do this so fast and easy.

Unfortunately VS cannot build a U++ project, because it cannot interpret the makefile and so you probably use both, U++ TheIDE and VS.

This said, even learning C needs a lot of time, and before you learn C++ you should learn C. This is a relation like algebra and differential calculus, C is not different from C++, C++ is a superset of C.

To learn C, Visual Studio is probably better, because there is a lot of inbuilt help and documentation and countless (free) books and online communities.

So you can use TheIDE and VS both at the same time on the same file.

All the best,

Peter

Subject: Re: What is the difference between learning U++ and Learning C++ through Microsoft Visual Studio?

Posted by [peterh](#) on Sun, 09 Jun 2019 10:37:24 GMT

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

Just an Example, why, even with U++, VS will be useful for a beginner:

We can see here (in Visual Studio) what edrd <<= ~ed2 means:

<<= is an overloaded operator that means SetData(). (in this context)

~ is an overloaded operator that means GetData()

Without this knowledge the program is not readable.

The context is not visible in the source, but VS makes it visible.

Therefore, using VS together with U++ is a big help.

BTW, in Free Pascal this is written as "edrd.Caption := ed2.Caption" which is pretty readable, even if you dont know Pascal. :roll: Unfortunately C++ has no Property members. :twisted:

All the best,

Peter

File Attachments

1) [VSexample.png](#), downloaded 931 times

Subject: Re: What is the difference between learning U++ and Learning C++ through Microsoft Visual Studio?

Posted by [Novo](#) on Sun, 09 Jun 2019 14:14:40 GMT

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

peterh wrote on Sun, 09 June 2019 06:37 Unfortunately C++ has no Property members. :twisted:

This statement is not entirely correct :roll:

C++ doesn't have properties built into the language itself, but properties can be implemented in just a few lines of code. I did that by accident when working on a "restricted value" framework last weekend.

Property is a value wrapper, which doesn't store an actual value ... :roll:

C++ has ZERO builtin functionality. Even STL is a replaceable library ...

Subject: Re: What is the difference between learning U++ and Learning C++ through Microsoft Visual Studio?

Posted by [Novo](#) on Sun, 09 Jun 2019 14:21:29 GMT

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

peterh wrote on Sun, 09 June 2019 04:14

If you use VS 17 or 19, then you can open the main.cpp of an U++ project. VS will analyze and crossreference the source code (be patient, it happens in background and takes some time) and if you point to an unknown operator VS displays nicely the type of the operator and can show its definition. TheIDE (the IDE of U++) cannot do this so fast and easy.

TheIDE can do that. It is doing that each time you are opening up a project. And it is doing this much faster. Useful short-cuts:

Alt-J - Context go to ...

Alt-I - Go to definition/declaration

You will find a lot of interesting stuff if you click on the "Assist" menu :roll:

Subject: Re: What is the difference between learning U++ and Learning C++ through Microsoft Visual Studio?

Posted by [peterh](#) on Mon, 10 Jun 2019 08:27:31 GMT

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

Novo wrote on Sun, 09 June 2019 16:21peterh wrote on Sun, 09 June 2019 04:14

If you use VS 17 or 19, then you can open the main.cpp of an U++ project. VS will analyze and crossreference the source code (be patient, it happens in background and takes some time) and if you point to an unknown operator VS displays nicely the type of the operator and can show its definition. TheIDE (the IDE of U++) cannot do this so fast and easy.

TheIDE can do that. It is doing that each time you are opening up a project. And it is doing this much faster. Useful short-cuts:

Alt-J - Context go to ...

Alt-I - Go to definition/declaration

You will find a lot of interesting stuff if you click on the "Assist" menu :roll:

I know this.

TheIDE doesn't do that for overloaded operators.

In VS, you point the mouse to an operator, variable or function and a popup window gives a short description for its kind and precise type.

Then you rightclick and you get the source embedded as shown above or goto the definition.

And UPP uses overloaded operators a lot. If you are a newbie and use these operators wrong, then you get a more or less cryptic error message at compile time, nothing more.

Possibly the compiler found an error in a template that is deeply embedded in some other templates and it is hard to figure this out.

In Freepascal Lazarus IDE, if you point to a property, variable or procedure you get an informative popup, this shows type and usage, eventually a short explanation and the choice to open the definition or declaration.

I still think Upp is great work, but to become more accepted, it should create a CMake project that can be used by mainstream IDEs.

"TheIDE" is still needed and it is great to design and create the project.

I am currently working and learning with TheIDE, tracking down a compile error.

I have used Alt-J a lot in the past.

If I press Alt-I or Alt-J on B::WhenAction then nothing happens.

Apparently this doesn't work in templates.

It works in VS.

Edit: Sorry this was an error, it doesn't work in VS too.

I have to replace B by EditString, then it works in VS, but still doesn't work in TheIDE :d

I would not be too surprised, if it works in VS if I could build the project in VS.... but I'm a beginner with UPP and VS, so I don't know.

All the best,

Peter

File Attachments

1) [TheIDE.png](#), downloaded 753 times

Subject: Re: What is the difference between learning U++ and Learning C++ through Microsoft Visual Studio?

Posted by [Novo](#) on Mon, 10 Jun 2019 20:40:38 GMT

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

peterh wrote on Mon, 10 June 2019 04:27

In VS, you point the mouse to an operator, variable or function and a popup window gives a short description for its kind and precise type.

VS is using its C++ parser for that, which is not available as a standalone product, AFAIK. The only parser, which TheIDE could use is Clang, but in this case your code should be developed for Clang (you should be able to compile your project with Clang).

What TheIDE is able to do is a compromise. Code navigation isn't perfect, but it works in many cases.

Another "nice" feature of VS is that it often takes hours to open up a relatively big project.

peterh wrote on Mon, 10 June 2019 04:27

Then you rightclick and you get the source embedded as shown above or goto the definition. And UPP uses overloaded operators a lot. If you are a newbie and use these operators wrong,

then you get a more or less cryptic error message at compile time, nothing more. Possibly the compiler found an error in a template that is deeply embedded in some other templates and it is hard to figure this out.

IMHO, U++ is very simple, much simpler than anything else in the C++ world. You just need to learn its basic concepts to stop making mistakes. If something doesn't compile it is often enough to just take a look at a class/function declaration. This is just a workflow you need to learn.

Many things are done differently in U++ comparing to other frameworks. In almost all cases U++'s way is better.

peterh wrote on Mon, 10 June 2019 04:27

In Freepascal Lazarus IDE, if you point to a property, variable or procedure you get an informative popup, this shows type and usage, eventually a short explanation and the choice to open the definition or declaration.

Freepascal/Pascal's syntax is very simple. Parser for this language can be implemented in a couple of hours/days depending on your experience. The same story is with Java and .NET.

C/C++ is a completely different story.

peterh wrote on Mon, 10 June 2019 04:27

I still think Upp is great work, but to become more accepted, it should create a CMake project that can be used by mainstream IDEs.

I would not be too surprised, if it works in VS if I could build the project in VS.... but Im a beginner with UPP and VS, so I dont know.

Fixing compilation errors and navigating code (especially U++ code) is a trivial task. As soon as you get more familiar with C++ this won't bother you at all.

U++ is something much bigger than just compilation and TheIDE. U++ is about software design, I'd say it is about software philosophy. It is about a different way of thinking. IMHO, this is why it is not that widely accepted. :roll:
