Subject: Ultimate++ vs Wxwidgets for desktop linux/windows/mac app
Posted by germandiago on Wed, 23 Jun 2021 13:18:17 GMT
View Forum Message <> Reply to Message

Hello everyone,

I know this project for a long time. I always thought of it as a good project, but not sure why it does not get used more, compared to, let us say, wxWidgets.

So here I have some questions:

1. For a Mac/Windows/Linux desktop app, does Ultimate++ fit well? I just want to use the GUI part, mainly.
2. In case I need to add other external dependencies, an sqlite orm or sqlpp11 for example and others, and since I use Meson for projects, will the package system get in the way for adding/removing dependencies? I want to have meson invocation as my top-level command, use theIDE for layout creation and have everything else run from Meson. I will not use TheIDE for anything else, since I use CLion mostly.
3. Can I streamline my layouts from the GUI into my build system or I have to stick in some way to TheIDE?
4. How well tested is Ultimate++ in Windows/Mac/Linux?
5. Is it possible to show/embed OpenGL and the like?
6. What are your perceived advantages/disadvantages compared to WxWidgets, limited to desktop app development for GUI only? I do
   not care about any other thing such as sockets or db access, since I will be using other projects for that.

Thanks.

Subject: Re: Ultimate++ vs Wxwidgets for desktop linux/windows/mac app
Posted by mirek on Thu, 24 Jun 2021 07:57:32 GMT
View Forum Message <> Reply to Message

germandiago wrote on Wed, 23 June 2021 15:18Hello everyone,

I know this project for a long time. I always thought of it as a good project, but not sure why it does not get used more, compared to, let us say, wxWidgets.

So here I have some questions:

1. For a Mac/Windows/Linux desktop app, does Ultimate++ fit well? I just want to use the GUI part, mainly.

You will probably be able to tell the difference from native apps if looking very carefully. You could probably do a tiny bit better with wxWidgets (which is native GUI, meaning it is using host platform widgets), but not with Qt (which is emulated, meaning it renders widgets itself, but U++ is about as good).

What U++ is to detect target platform look&feel as accurately as possible (color, shape of buttons etc...) and then use "Chameleon" system to adjust its look and feel.

Quote:
2. In case I need to add other external dependencies, an sqlite orm or sqlpp11 for example and others, and since I use Meson for projects, will the package system get in the way for adding/removing dependencies? I want to have meson invocation as my top-level command, use theIDE for layout creation and have everything else run from Meson. I will not use TheIDE for anything else, since I use CLion mostly.

With TheIDE, should not be that much of trouble.

Then if using external editor, you might consider using "umk" for building (means it will still be using U++ modular system).

You can also build U++ GUI as library - it is definitely doable and tested, but we never went beyond that (in the end, everybody ends using theide or umk and we really have limited resources). You can check the corresponding discussion here:
https://www.ultimatepp.org/forums/index.php?t=msg&th=112 72&start=0&

BTW, theide can be used as editor of layout files - if you specify the file on commandline, it will simply be open for editing (e.g. "theide mydialogs.lay")

Quote:
3. Can I streamline my layouts from the GUI into my build system or I have to stick in some way to TheIDE?

Layouts (and images) are in fact compiled with C++ compiler, so no problem there.

Quote:
4. How well tested is Ultimate++ in Windows/Mac/Linux?

Win/Linux production (in sense there are applications that sell for money).

Mac is less used and probably less matured. Definitely works, but I am not aware about any production apps there.

Quote:
5. Is it possible to show/embed OpenGL and the like?

Not 100% sure what you mean by that, but there is "GLCtrl" widget, perhaps this answers the question.

Quote:
6. What are your perceived advantages/disadvantages compared to WxWidgets, limited to desktop app development for GUI only? I do
  not care about any other thing such as sockets or db access, since I will be using other projects for that.

About the same as C++ vs C... :)

U++ is battle tested for really huge complex applications (think about application with 1000 dialogs as one of metrics). From maintainance standpoint, it is always easier to maintain smaller codebase.

Mirek

Mirek

Subject: Re: Ultimate++ vs Wxwidgets for desktop linux/windows/mac app
Posted by germandiago on Thu, 24 Jun 2021 14:56:35 GMT
View Forum Message <> Reply to Message

Hello Mirek,

Thanks for your prompt reply. I gave it a small try yesterday to U++ + TheIDE and the feelings are quite good so far. It looks like it is working well and it is battle tested.

It is true that it is not native rendering, but for my purposes it will not make a big difference (it is mainly for internal tooling).

1. What would be the effort involved (maybe man-hours?) in, let us say, have Meson wraps for the GUI part of U++?
https://mesonbuild.com/Wrap-dependency-system-manual.html

If you are not familiar with Meson wraps, these are basically overlays (patched builds) + meson.build files written to be able to consume libraries as subprojects in a standard way, think of something like Conan. I would be interested in having something that can make use of Meson standard workflow and be able to consume U++. Conan would also be ok if there is more interest, since Conan can be used from both CMake and Meson AFAIK.

Why this workflow? Basically because I already use emacs + lsp and CLion for code completion. I would still use TheIDe as the layout designer. I am really impressed that it worked so well. U++ looks like polished and the API nicer to use than wxWidgets, since everything ends up embedded in normal C++, even layouts.

As something out of my reach now, I think that if you have few resources, and unless you are under some kind of restriction, it would be nice to make U++ consumable by CMake + Meson. The library is all worth. The missing part would be a layout designer out of TheIDE. Nowadays IDEs have a really good completion engine via compile_commands.json and/or CMake but it seems that U++ is locked down under theIDE. This would free resources to focus on:

1. integrating with Conan to be able to consume from CMake/Meson --> this could bring new users
2. free from TheIde maintainance, unless it has reasons to exist still. Nowadays there are really powerful IDEs (the layout designer should be kept somewhere, as a plugin for other software or from theIde anyway).

I would be interested in porting U++, in a modular way to be consumed by Meson (Conan is ok, so Meson + CMake is viable). Thank you.

Subject: Re: Ultimate++ vs Wxwidgets for desktop linux/windows/mac app
Posted by mirek on Fri, 25 Jun 2021 05:24:12 GMT

View Forum Message <> Reply to Message

germandiago wrote on Thu, 24 June 2021 16:56Hello Mirek,

Thanks for your prompt reply. I gave it a small try yesterday to U++ + TheIDE and the feelings are quite good so far. It looks like it is working well and it is battle tested.

It is true that it is not native rendering, but for my purposes it will not make a big difference (it is mainly for internal tooling).

1. What would be the effort involved (maybe man-hours?) in, let us say, have Meson wraps for the GUI part of U++?
https://mesonbuild.com/Wrap-dependency-system-manual.html

If you are not familiar with Meson wraps, these are basically overlays (patched builds) + meson.build files written to be able to consume libraries as subprojects in a standard way, think of something like Conan. I would be interested in having something that can make use of Meson standard workflow and be able to consume U++. Conan would also be ok if there is more interest, since Conan can be used from both CMake and Meson AFAIK.

Why this workflow? Basically because I already use emacs + lsp and CLion for code completion.
I would still use TheIDe as the layout designer. I am really impressed that it worked so well. U++ looks like polished and the API nicer to use than wxWidgets, since everything ends up embedded in normal C++, even layouts.

As something out of my reach now, I think that if you have few resources, and unless you are under some kind of restriction, it would be nice to make U++ consumable by CMake + Meson. The library is all worth. The missing part would be a layout designer out of TheIDE. Nowadays IDEs have a really good completion engine via compile_commands.json and/or CMake but it seems that U++ is locked down under theIDE. This would free resources to focus on:

1. integrating with Conan to be able to consume from CMake/Meson --> this could bring new users
2. free from TheIde maintainance, unless it has reasons to exist still. Nowadays there are really powerful IDEs (the layout designer should be kept somewhere, as a plugin for other software or from theIde anyway).

I would be interested in porting U++, in a modular way to be consumed by Meson (Conan is ok, so Meson + CMake is viable). Thank you.

1. I like that
2. I do not have time nor energy to do that

Anyway, it should not really be hard to do. Core U++ libraries are pure C++, basically all you need to do is to put them into the project...

One haunting problem is that you will need to "demodularize" modular structure (decide what modules to combine). But maybe with meson, you can do it the right way, it looks like it provides something quite similar to U++ package dependencies.

Mirek

Subject: Re: Ultimate++ vs Wxwidgets for desktop linux/windows/mac app
Posted by mirek on Fri, 25 Jun 2021 05:28:02 GMT
View Forum Message <> Reply to Message

germandiago wrote on Thu, 24 June 2021 16:56
1. What would be the effort involved (maybe man-hours?) in, let us say, have Meson wraps for the GUI part of U++?
https://mesonbuild.com/Wrap-dependency-system-manual.html

3 hours < (time needed for germandiago to do Meson wraps for the U++ GUI with U++ community support) < 1 week

Mirek

Subject: Re: Ultimate++ vs Wxwidgets for desktop linux/windows/mac app

Posted by germandiago on Fri, 25 Jun 2021 10:14:36 GMT
View Forum Message <> Reply to Message

It seems something reasonable. I could give it a try.

Actually, I am interested in having Conan packages using the current build system if that is possible.
This would kill two birds with one stone and make it consumable from both CMake and Meson.

I would give that higher priority.

As for Meson wraps, I also want to do that, but maybe as a second step.

The good thing about creating Conan packages is that I think, at least in theory, that the current build system for U++ does not need to be changed.

Subject: Re: Ultimate++ vs Wxwidgets for desktop linux/windows/mac app
Posted by mirek on Fri, 25 Jun 2021 13:49:27 GMT
View Forum Message <> Reply to Message

Some links you have probably already found, but to be sure:

https://www.ultimatepp.org/app$ide$upp$en-us.html
https://www.ultimatepp.org/app$ide$PackagesAssembliesAndNest s$en-us.html
https://www.ultimatepp.org/app$ide$umk$en-us.html
https://www.ultimatepp.org/app$ide$Flags$en-us.html

https://www.ultimatepp.org/app$ide$Files$en-us.html
- here a note: "icpp" extension was a problem when trying to compile U++ outside of theide/umk. It is not a problem anymore as its use in core libraries is replaced by another mechanism and deprecated.

This one is just for completness, you will not need it for GUI libraries, this is IMO sort of U++ mechanism for tasks you are using Meson (?):

https://www.ultimatepp.org/app$ide$importext$en-us.html

Please, I would really be happy if your effort would succeed. Do not be shy to ask question! And if you setup github repo for this, even better...

Mirek

Subject: Re: Ultimate++ vs Wxwidgets for desktop linux/windows/mac app
Posted by germandiago on Fri, 25 Jun 2021 14:04:14 GMT
View Forum Message <> Reply to Message

Hello again Mirek,

I did not know all the links. That is exactly what I need in order to prepare packages, I think.

I have little time to invest, hence my question for the man-hours effort. I will try some experiment but I cannot commit dates yet, I will keep in touch if I have something to show, and be sure that if I have, I will package it conveniently for others to be able to use it.

The best return on investment is to prepare a Conan recipe to be able to consume U++ (at least the GUI subset and the dependencies it drags) from CMake/Meson.

Reasons to choose this strategy of Conan packaging (mentioned in previous post), but I repeat (and add) here:

- it does not need changes to the build system (or it would need minimal changes, if it does).
- it does not need additional Meson build system maintainance, only a recipe on top of the existing build system.
- it makes things available in CMake/Meson and even MsBuild/autotools (modulo bugs) because Conan tries to integrate with all of those
  and, as far as my knowledge goes, tries to be build-system agnostic for the ones it has no integration with.

So, yes, I will give it a try but I cannot commit dates. It would be, hopefully, some time during summer.

I have previous experience creating Conan recipes but I have zero experience with U++ build system, it is there where I would have some questions to be able to package a recipe in conditions to be consumed.

I would do this:

- target the latest release (2021.1?) since using a static version avoids doing work on top of a moving target. All via Github would be ok?
- basic debug/release packages. The more conventional (and widely consumable), the better.
- upload to conan center or bincrafters pre-packaged artifacts.

From there, Conan can build more stuff for specific needs/flags, but I would leave that as a second thought if it creates bad interactions with the current build system and focus on the basics. Better 2 things working well than 5 mid-working. :)

Thanks for your feedback!

Subject: Re: Ultimate++ vs Wxwidgets for desktop linux/windows/mac app
Posted by mirek on Fri, 25 Jun 2021 14:50:39 GMT
View Forum Message <> Reply to Message

germandiago wrote on Fri, 25 June 2021 16:04Hello again Mirek,

I did not know all the links. That is exactly what I need in order to prepare packages, I think.

I have little time to invest, hence my question for the man-hours effort. I will try some experiment but I cannot commit dates yet, I will keep in touch if I have something to show, and be sure that if I have, I will package it conveniently for others to be able to use it.

The best return on investment is to prepare a Conan recipe to be able to consume U++ (at least the GUI subset and the dependencies it drags) from CMake/Meson.

Reasons to choose this strategy of Conan packaging (mentioned in previous post), but I repeat (and add) here:

- it does not need changes to the build system (or it would need minimal changes, if it does).
- it does not need additional Meson build system maintainance, only a recipe on top of the existing build system.
- it makes things available in CMake/Meson and even MsBuild/autotools (modulo bugs) because Conan tries to integrate with all of those
  and, as far as my knowledge goes, tries to be build-system agnostic for the ones it has no integration with.

As you wish, but I would like to comment that getting core U++ GUI to work is really just an act of compiling all .cpp files into something. There are no additional tools involved not even ./configure is necesarry (U++ is using compiler macros instead for the same task). The only thing that you really need is to provide a very limited set of -D defines on compiler commandline to describe the environment.

Subject: Re: Ultimate++ vs Wxwidgets for desktop linux/windows/mac app
Posted by germandiago on Fri, 25 Jun 2021 15:19:10 GMT
View Forum Message <> Reply to Message

Good to know.

But I guess that in order to draw in Linux/Windows/Mac it will need additional dependencies to link against (OpenGL/Cocoa/Whatever).

So you are basically suggesting that it is maybe better to write the build files directly in another build system? It is a possibility if that is not difficult.

1. do a port to Meson.
2. compile and create a conan recipe from the new Meson build system.

What I wonder is that if U++ developers (I assume you are one of them) keep using the assemblies/packages/nests, etc. the "U++ way", Meson build system would get out of sync.

This would create maintainance burden, that is why I was thinking of using what is updated by developers directly.

Another possibility is adopt a more mainstream build system (CMake/Meson come to my mind, I have a strong preference for Meson) and provide Conan packages to consume by any other build system. But this, as far as I understand, makes TheIDE packaging system "obsolete" in the sense that TheIDE consumes, as of today, the packages in the format expected by TheIDE, so if some package is added, what happens to that from the point of view of availability for consumers from CMake/Meson/MsBuild/Autotools/Whatever?

FWIW my own personal use would be to use a mainstream IDE (CLion) or Emacs, a more mainstream build system and the layout designer from TheIDE.

What do you think?

Subject: Re: Ultimate++ vs Wxwidgets for desktop linux/windows/mac app
Posted by mirek on Fri, 25 Jun 2021 17:03:32 GMT
View Forum Message <> Reply to Message

germandiago wrote on Fri, 25 June 2021 17:19Good to know.

But I guess that in order to draw in Linux/Windows/Mac it will need additional dependencies to link against (OpenGL/Cocoa/Whatever).

So you are basically suggesting that it is maybe better to write the build files directly in another build system? It is a possibility if that is not difficult.

1. do a port to Meson.
2. compile and create a conan recipe from the new Meson build system.

What I wonder is that if U++ developers (I assume you are one of them) keep using the assemblies/packages/nests, etc. the "U++ way", Meson build system would get out of sync.

This would create maintainance burden, that is why I was thinking of using what is updated by

developers directly.

Another possibility is adopt a more mainstream build system (CMake/Meson come to my mind, I have a strong preference for Meson) and provide Conan packages to consume by any other build system. But this, as far as I understand, makes TheIDE packaging system "obsolete" in the sense that TheIDE consumes, as of today, the packages in the format expected by TheIDE, so if some package is added, what happens to that from the point of view of availability for consumers from CMake/Meson/MsBuild/Autotools/Whatever?

FWIW my own personal use would be to use a mainstream IDE (CLion) or Emacs, a more mainstream build system and the layout designer from TheIDE.

What do you think?

When I was thinking about producing "regular libs", I planned to simply parse .upp files and make it into some makefile or something (btw, you actuall CAN convert U++ project to makefile already, umk can do conversion - there are limits, but it would do for basic GUI libs).

Hm, now thinking about it, it might be even possible to add something like "export to meson" option to umk... The only problem is that my knowledge of meson is nil.

Mirek

Subject: Re: Ultimate++ vs Wxwidgets for desktop linux/windows/mac app
Posted by germandiago on Fri, 25 Jun 2021 17:11:32 GMT
View Forum Message <> Reply to Message

I can handle Meson myself and Conan recipes.

But the problem is not to create a Meson package, but whether
it is convenient or not to add that build system itself, since the goal (unless you have a wishlist) is to be able
to consume U++ from other build systems.

What I am trying to figure out in this discussion is the best way to go before starting.

Nice traits would be to be able to create consumable packages as U++ evolves, that is why I put in the table the possibility of using what is already working and being maintained, since I expect it to be maintained in the future and that would not add a burden to current developers. Or a full migration to another build system, but I am not sure that is neither viable nor convenient, since TheIDE seems to be built around this system.

I am focusing on the "consume U++ packages from normal build systems" use case. Adding a build system by adding an extra build system means an added maintainance burden IMHO.

Maybe an exporter in umk is more work than that, not sure.

Thanks for your feedback.