

---

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!

---