Subject: U++ libraries and TheIDE converted to Meson build system
Posted by germandiago on Sun, 17 Jul 2022 21:56:40 GMT
View Forum Message <> Reply to Message

Hello everyone.

I have been long interested in using U++ to be able to compare it to WxWidgets. The code patterns that U++ uses look nice to me and the environment fully mature. The problem for me has been consuming it from other build system, in my case from Meson.

So I took my weekend and converted U++ to compile with Meson build system. Currently it is building under Linux and with static libraries (as I saw from what the Makefile is doing). I am using the 2022.2 tag from the github repo since my target is to make something consumable as a library to systems foreign to TheIde environment.

Beware that this is work in progress and right now it is working in Linux (others coming) with a static library setup tested.

For the conversion I extracted all files from the Makefile and started to code, one by one, a library per directory, so that it has basically the same structure as the original. I also ripped off some internal dependencies such as zstd and libpng. Now they are consumed from the package manager or Meson wraps if absent in subprojects/ (see Meson subprojects for more info) This allows to change those dependencies more easily when new versions come up and also allows for the use of Meson wraps, which is the package management system of Meson. It also flattens internal dependencies and makes them explicitly visible.
I did not compile absolutely all the code but the one necessary to run TheIde and start it.

My own plan is to use the layout designer but use Meson for package consumption and Emacs/CLion and for application creation.

There are some improvements I can think of (and some mistakes maybe, I would need some review) along the way to convert it. I would like to ask if you would be interested in me contributing this work. My plan is:

- make the Meson build system work in Windows
- make the Meson build system work in Mac
- make Meson generate .pc and .cmake packages out of the box. There is support for it in Meson.
- publish packages for all libraries (I suspect the ones in the ide/ directory are private and the others are meant to be for public cinsumption?)
- publish a wrap Meson file based on this work to be able to download the libraries from Meson and compile
them.

- publish a Conan recipe based on this work as well, which would allow the libraries to be consumed from many more sources.

Some highlights about the current conversion:

- It is easy to obtain debug/release and any setup you like, including ASAN and other builds since Meson provides all that.
- You can compile also shared libraries (I just use neutral library targets, you can choose) BUT it has not been tested in shared setup so far. I just tested the static library variant so far.
-This opens the possibility of compiling plugins dynamically?
- Getting rid of all "customized" nested dependencies is something not fully done yet, but doable. (Done for zstd, pcre, lzma, lzma sdk, zlib so far)

I will publish a branch in the next few days.

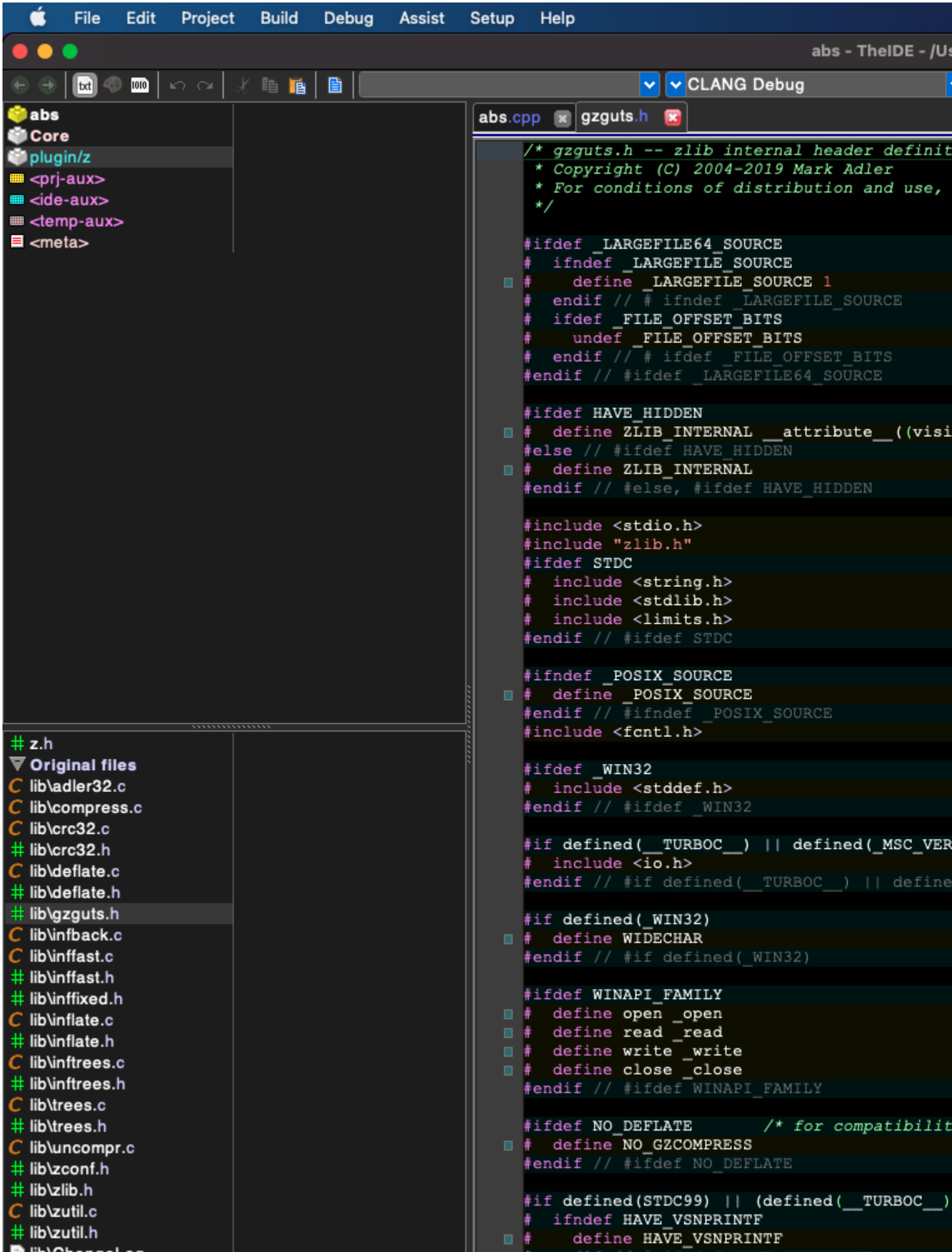
EDIT: Just pusblished it at https://github.com/germandiagogomez/ultimatepp/tree/feature/meson-bu ild-system-based-on-2022.2
Feedback is and will be very welcome.

Subject: Re: U++ libraries and TheIDE converted to Meson build system
Posted by germandiago on Mon, 18 Jul 2022 18:55:49 GMT
View Forum Message <> Reply to Message

Good progress, just got Mac working with a tiny more amount of work compared to Linux.

File Attachments

1) Captura de pantalla 2022-07-19 a las 1.49.07.png, downloaded 535 times


File Edit Project Build Debug Assist Setup Help
abs - TheIDE - /Us
CLANG Debug
abs
Core
plugin/z
<prj-aux>
<ide-aux>
<temp-aux>
<meta>
abs.cpp
gzguts.h
/* gzguts.h -- zlib internal header definit
* Copyright (C) 2004-2019 Mark Adler
* For conditions of distribution and use,
*/
#ifdef _LARGEFILE64_SOURCE
ifndef _LARGEFILE_SOURCE
define _LARGEFILE_SOURCE 1
endif // # ifndef _LARGEFILE_SOURCE
ifdef _FILE_OFFSET_BITS
undef _FILE_OFFSET_BITS
endif // # ifdef _FILE_OFFSET_BITS
#endif // #ifdef _LARGEFILE64_SOURCE
#ifdef HAVE_HIDDEN
define ZLIB_INTERNAL __attribute__((visi
#else // #ifdef HAVE_HIDDEN
define ZLIB_INTERNAL
#endif // #else, #ifdef HAVE_HIDDEN
#include <stdio.h>
#include "zlib.h"
#ifdef STDC
include <string.h>
include <stdlib.h>
include <limits.h>
#endif // #ifdef STDC
#ifndef _POSIX_SOURCE
define _POSIX_SOURCE
#endif // #ifndef _POSIX_SOURCE
#include <fcntl.h>
#ifdef _WIN32
include <stddef.h>
#endif // #ifdef _WIN32
#if defined(__TURBOC__) || defined(_MSC_VER
include <io.h>
#endif // #if defined(__TURBOC__) || define
#if defined(_WIN32)
define WIDECHAR
#endif // #if defined(_WIN32)
#ifdef WINAPI_FAMILY
define open _open
define read _read
define write _write
define close _close
#endif // #ifdef WINAPI_FAMILY
#ifdef NO_DEFLATE /* for compatibilit
define NO_GZCOMPRESS
#endif // #ifdef NO_DEFLATE
#if defined(STDC99) || (defined(__TURBOC__)
ifndef HAVE_VSNPRINTF
define HAVE_VSNPRINTF
z.h
Original files
lib\adler32.c
lib\compress.c
lib\crc32.c
lib\crc32.h
lib\deflate.c
lib\deflate.h
lib\gzguts.h
lib\infback.c
lib\inffast.c
lib\inffast.h
lib\inffixed.h
lib\inflate.c
lib\inflate.h
lib\inftrees.c
lib\inftrees.h
lib\trees.c
lib\trees.h
lib\uncompr.c
lib\zconf.h
lib\zlib.h
lib\zutil.c
lib\zutil.h

Subject: Re: U++ libraries and TheIDE converted to Meson build system
Posted by Klugier on Mon, 18 Jul 2022 19:42:49 GMT
View Forum Message <> Reply to Message

Hello German,

The general idea with this change is good. We need to deliver binaries that could be consume with other build systems. So, any project that aims to achieve this goal deserves recognition. If we for example manage to be consume by Conan package manager and CMake, then it will be a big gain for us. I remembered the discussion on the Redit and the main complain about Upp was that it requires specific environment. So, braking that silo in the long term is needed to achieve bigger reorganization and success.

In context of WxWidgets, you are right that Upp is better designed. I have never liked WxWidgets and the applications that use it. Upp just works faster and better. So, I am glad you want to invest your time in our framework.

I am not familiar with Meson, so I can not tell if it is right choice to achieve that. But, from what I reading it looks good, so far. I see one drawback at the moment of meson, you need to specify sources explicitly. It adds additional maintenance cost, which is not good. Sources should be deduced by parsing .upp file. There should be one source of truth, which files belong to project. We have explicit files list in our Makefiles, but this file are generated, not written by hand.

Emacs/CLion - I totally understand the people who would like to use different ide's. We should allow people to do that. In addition to that, I would like to add that Mirek is right now working on integrating libclang to TheIDE, so our ide should be much more better in near future. However, still you should't be force to use and you should be able to use ide of your chocie.

Thanks for that contribution. We will monitor your progress.

Klugier

Subject: Re: U++ libraries and TheIDE converted to Meson build system
Posted by Oblivion on Mon, 18 Jul 2022 20:37:33 GMT
View Forum Message <> Reply to Message

My thoughts:

Adding an optional meson support is a good idea (generally), at least on POSIX, as it is a modern and very easy to use (yet very capable) build system.

However, I think it might be better to expose a generic builder api in TheIDE/umk for making build tool scripts... (instead of a hardcoded make/cmake script creator we have...)

And this can and should be done with TheIDE/umk + Esc-based macros (then it can be implemented and shipped as different "build-script maker" macros without a costly maintenance burden or broken code in theIDE's code base...

This way, we can even collect different build script creator macros into a single, externally curated package (say, in UppHub)

Best regards,
Oblivion

Subject: Re: U++ libraries and TheIDE converted to Meson build system
Posted by germandiago on Mon, 18 Jul 2022 21:33:22 GMT
View Forum Message <> Reply to Message

Klugier wrote on Mon, 18 July 2022 21:42Hello German,

The general idea with this change is good. We need to deliver binaries that could be consume with other build systems. So, any project that aims to achieve this goal deserves recognition. If we for example manage to be consume by Conan package manager and CMake, then it will be a big gain for us. I remembered the discussion on the Redit and the main complain about Upp was that it requires specific environment. So, braking that silo in the long term is needed to achieve bigger reorganization and success.

In context of WxWidgets, you are right that Upp is better designed. I have never liked WxWidgets and the applications that use it. Upp just works faster and better. So, I am glad you want to invest your time in our framework.

I am not familiar with Meson, so I can not tell if it is right choice to achieve that. But, from what I reading it looks good, so far. I see one drawback at the moment of meson, you need to specify sources explicitly. It adds additional maintenance cost, which is not good. Sources should be deduced by parsing .upp file. There should be one source of truth, which files belong to project. We have explicit files list in our Makefiles, but this file are generated, not written by hand.

Emacs/CLion - I totally understand the people who would like to use different ide's. We should allow people to do that. In addition to that, I would like to add that Mirek is right now working on integrating libclang to TheIDE, so our ide should be much more better in near future. However, still you should't be force to use and you should be able to use ide of your chocie.

Thanks for that contribution. We will monitor your progress.

Klugier

Thank you Klugier. Very glad to see you like the idea.

My plan is not TheIde-related right now, though it could make sense to integrate Meson in the backend as well for TheIde to be able to use that as a backend for your builds in TheIde in the future, even if you keep the interface via .upp. This is not an explicit goal in the first stage, but in a second stage it could be done. It would be nice if you can emit Meson build files even if the interface in TheIde is still the same via upp files or whatever. I am not very familiar so I could be saying something silly.

As for wxWidgets, well, it does the job actually, and it always has that native-looking thing that is really attractive from the POV of a user using software. But coding-wise Ultimate++ looks way better and it has advanced features as well

As you mention, Meson does not do wildcard source expansion, since it seems to not be reliable according to the author. However, there is a trick that can do it. Relevant explanation here: https://mesonbuild.com/FAQ.html#why-cant-i-specify-target-fi les-with-a-wildcard

For the record, my plan is that a Conan package can follow after having a solid conversion to Meson build system to compile libraries and TheIde. Note that Meson itself can already create `.pc` config files and `.cmake` files for consumption. Conan would stretch package consumption by other generators,
but only with Meson you would already have those two at least.

What I would need from your side from experts in U++ is to review carefully what depends on what to do it properly in Meson build system itself.
I would like to improve it to be explicit about the private vs public dependencies of each package.

I was able to compile Linux and Mac already. Windows coming soon.

Also, as for using TheIde, I do not know how good code completion is right now but in that department there are very advanced tools nowadays compared
to years back. Meson generates a compile_commands.json file that can be used from many IDEs and provides code completion even for Ultimate++ code compiled
with Meson. That said, I understand you want to maintain the tool and experience for whom is valuable and I agree with that.

I could even collaborate helping integrating the build system in TheIde, but I think it would be a big win to let people use Ultimate++ from outside as you said.

Thanks for your support, I am excited to be able to contribute to offer U++ to a wider audience, which includes me also, since I was scratching my own itch

---

Subject: Re: U++ libraries and TheIDE converted to Meson build system
Posted by germandiago on Mon, 18 Jul 2022 21:42:33 GMT
View Forum Message <> Reply to Message

Oblivion wrote on Mon, 18 July 2022 22:37My thoughts:

Adding an optional meson support is a good idea (generally), at least on POSIX, as it is a modern and very easy to use (yet very capable) build system.

However, I think it might be better to expose a generic builder api in TheIDE/umk for making build tool scripts... (instead of a hardcoded make/cmake script creator we have...)

And this can and should be done with TheIDE/umk + Esc-based macros (then it can be implemented and shipped as different "build-script maker" macros without a costly maintenance burden or broken code in theIDE's code base...

This way, we can even collect different build script creator macros into a single, externally curated package (say, in UppHub)

Best regards,
Oblivion

Meson works fine in Windows also.

I do not understand what the macros are about. Not sure if an UppHub would attract many people honestly. It can be good to have and nice in theory, but people go to Conan or Vcpkg for packages to be realistic. However, I am not the one who take this kind of decisions and if it is good for Ultimate++ project I do not have anything against it, it is just my two cents.

I just would like to contribute at least the possibility of consuming the packages in a standard way. With the compilation of the code in Meson TheIde executable is also provided. Conan package manager and Meson wraps can provide and serve programs, not just libs, so this would spread also the number of people that
can try TheIde. They can also do packaging and more and streamline releases if needed.

Thank you for your feedback!

Subject: Re: U++ libraries and TheIDE converted to Meson build system
Posted by Oblivion on Mon, 18 Jul 2022 22:12:56 GMT
View Forum Message <> Reply to Message

Hello germandiago,

Quote:I do not understand what the macros are about. Not sure if an UppHub would attract many people honestly. It can be good to have and nice in theory, but people go to Conan or Vcpkg for packages to be realistic. However, I am not the one who take this kind of decisions and if it is good for Ultimate++ project I do not have anything against it, it is just my two cents.

What I mean is: We can have a collection of macros that will create build scripts for major build

tools such as make, cmake, meson, etc. You (or any maintainer) run the preferred generator once, then deploy the generated build script with your U++ based app, without the need for theide or umk. (We already have this with make, but it is hard-coded ATM.)

So, all in all, it is meant to simplify what you already aim.

Best regards,
Oblivion

Subject: Re: U++ libraries and TheIDE converted to Meson build system
Posted by germandiago on Tue, 19 Jul 2022 05:48:32 GMT
View Forum Message <> Reply to Message

Oblivion wrote on Tue, 19 July 2022 00:12Hello germandiago,

Quote:I do not understand what the macros are about. Not sure if an UppHub would attract many people honestly. It can be good to have and nice in theory, but people go to Conan or Vcpkg for packages to be realistic. However, I am not the one who take this kind of decisions and if it is good for Ultimate++ project I do not have anything against it, it is just my two cents.

What I mean is: We can have a collection of macros that will create build scripts for major build tools such as make, cmake, meson, etc. You (or any maintainer) run the preferred generator once, then deploy the generated build script with your U++ based app, without the need for theide or umk. (We already have this with make, but it is hard-coded ATM.)

So, all in all, it is meant to simplify what you already aim.

Best regards,
Oblivion

Nice! Now I understand, thanks.

Subject: Re: U++ libraries and TheIDE converted to Meson build system
Posted by germandiago on Thu, 21 Jul 2022 13:06:55 GMT
View Forum Message <> Reply to Message

As I go with the Windows conversion, I think I found a couple of problems (I guess bugs?).

Basically Iml::Image is marked as moveable.

But it holds std::atomic which is not movable, for loaded.
It also holds a NoMove buffer for data and the reference count is another atomic variable.

However, another part of the code is trying to use it inside a Vector, which asserts in the destructor
that things contained should be movable.
