Subject: Building applications
Posted by normvcr on Wed, 12 Aug 2020 00:03:25 GMT
View Forum Message <> Reply to Message

After untarring upp-posix-14429.tar and building with ./install, I get the theide and umk binaries, but there are no .a or .so libraries for building applications. There are lots of .o files in the .cache hierarchy, however. How does one build the libraries and move them to a common folder? How does one collect the header files into a common folder? (I see there is a umk build utility, but I would like to be able to fall back to compilers and linkers, directly. Also, I have found a description of the umk command line, but not a high level description of umk for beginners)

Subject: Re: Building applications
Posted by Xemuth on Wed, 12 Aug 2020 01:49:40 GMT
View Forum Message <> Reply to Message

Hello Normvcr,

To speak about UMK we must first speak about packages. The thing is, an U++ application is made of "packages"

From U++ documentation Packages are centric to U++. An executable application is built from a package. A package can also build into a dynamic link library. A package can be used by other packages. A package corresponds to a single directory with the directory name being the name of the package. The package directory contains the package definition file (a plain text file with a .upp extension), which always has the same name as the package directory. The package definition file contains a list of the source files that make up the package, plus information on what type of package it is, how it should be built and what other packages it uses. The source files for the package are normally located in the package directory and its subdirectories, but they may be in any desired location.

By example, most famous packages of Upp are:
-Core (which also include the plugin/z package) the core package is one of pillars of Upp framework, it provide a lot of interesting think
-CtrlLib (which also include CtrlCore(which also include ...)) the CtrlLib package will provide most of features you need to do real interesting GUI

Package can be configured to allow some option on compilation/linking depending on flags(by example depending on the OS or the build type)

UMK is a tool which will link a compilation toolchain (GCC / CLANG / MSVS) with all the U++ logic (packages, assemblies etc...)
so, when you will need to build an U++ application, instead having to specify to your compiler wich source code he should look for or wich librairy need to be link, Umk will read package information and will use the compiler you chose to build your application / dll ...

The way you link UMK to a compilation toolchain is by creating Build method which can be seen in TheIDE :

As you can see, in the build method editor window of TheIDE you can specify most of thing a compilator need.

Later when you will compile your package (which can contain some others packages), TheIDE will invoque UMK and use your current build method to build your app.
The current build method is here in TheIDE :

When you want to compile your package by command line (imagine you develop your application under windows then compile it on linux without any Graphical interface)
then you must call UMK (as you have seen in the doc) by specifing your build method, package, assemblies etc... (https://www.ultimatepp.org/appideumk$en-us.html)

I'm not sure I answered your question, but maybe with a more precise vision about UMK your problem will be resolve

Xemuth

Subject: Re: Building applications
Posted by Xemuth on Wed, 12 Aug 2020 01:56:22 GMT
View Forum Message <> Reply to Message

Also, to just awnser your question about .a:

.h work the same but located under the "include directories" tab

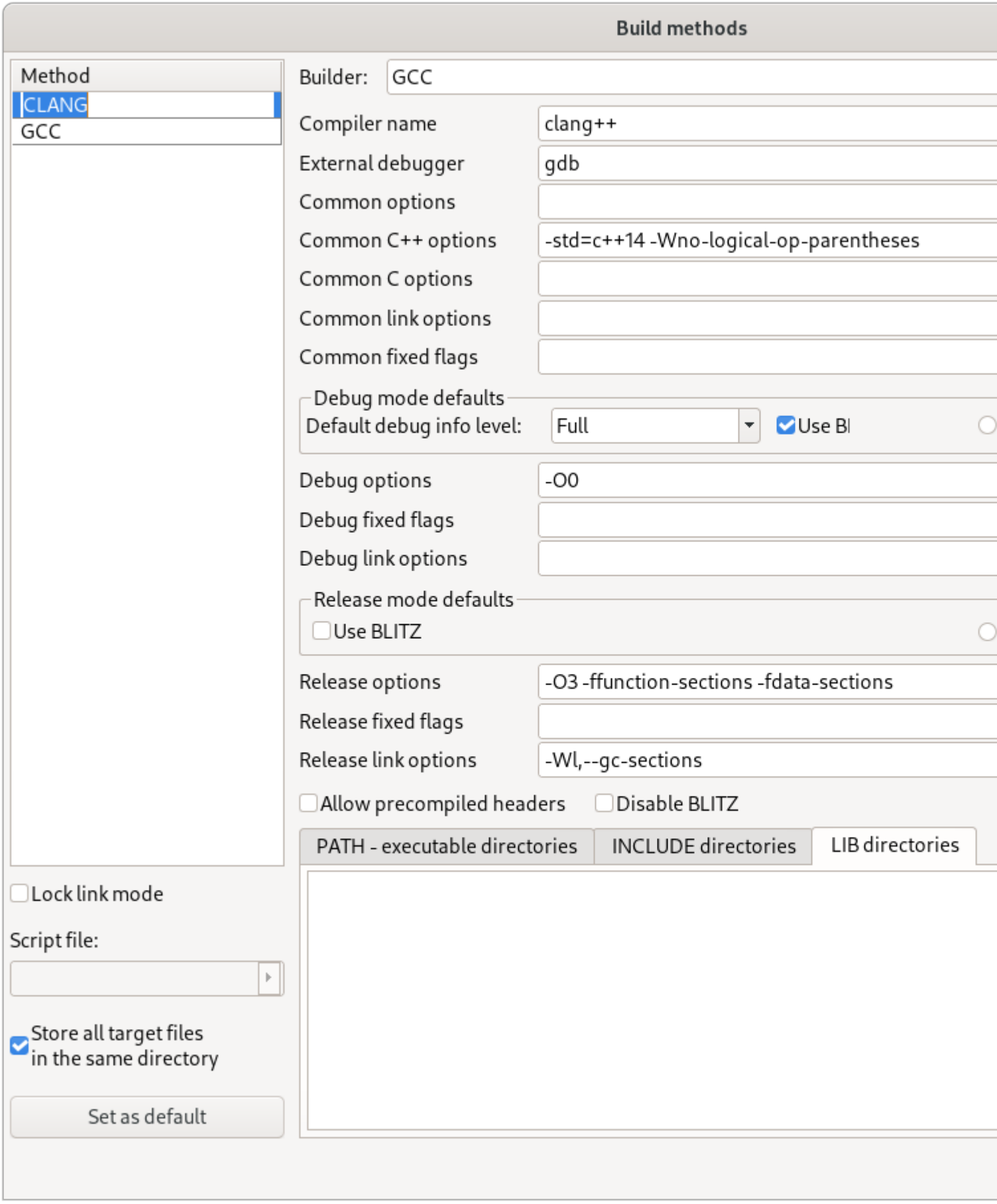
---

Subject: Re: Building applications
Posted by normvcr on Wed, 12 Aug 2020 02:17:03 GMT
View Forum Message <> Reply to Message

Thank you for the replies! Here is a screen shot of my Build Methods. As you can see, there are no folders listed under LIB directories. The folders that you are showing are under C:\Upp\Clang, these do not seem to be the implementation libraries for Ultimate++. For example, why are the SDL libraries stored under C:\Upp? Are these not generic SDL implementation libraries? It seems to me that Ultimate++ does not provide a way to create the Ultimate++ libraries....?? Or, do I need to run umk on each package, individually, to create these libraries?

File Attachments

1) Screenshot from 2020-08-11 19-01-57.png, downloaded 740 times


Build methods
Method
CLANG
GCC
Builder:
GCC
Compiler name
clang++
External debugger
gdb
Common options
Common C++ options
-std=c++14 -Wno-logical-op-parentheses
Common C options
Common link options
Common fixed flags
Debug mode defaults
Default debug info level:
Full
Use Bl
Debug options
-O0
Debug fixed flags
Debug link options
Release mode defaults
Use BLITZ
Release options
-O3 -ffunction-sections -fdata-sections
Release fixed flags
Release link options
-Wl,--gc-sections
Allow precompiled headers
Disable BLITZ
PATH - executable directories
INCLUDE directories
LIB directories
Lock link mode
Script file:
Store all target files
in the same directory
Set as default

Subject: Re: Building applications
Posted by Xemuth on Wed, 12 Aug 2020 02:28:24 GMT
View Forum Message <> Reply to Message

All my lib are located under a C:/Upp/.... because I'm working on windows, and when you download the windows version of upp you also donwload some libs (which has not been done by Upp but is used by it).

Since you are working on linux, I can't help you anymore, I will let another U++ member who know the linux env better than me help you

Quote:Or, do I need to run umk on each package, individually, to create these libraries?

no, thoses libs have been compiled and provided by the entity wich own it (by example SDL2 have been recovered from here :https://www.libsdl.org/download-2.0.php)
Since Upp dont have the source code of this lib I don't see how he could compile it (don't forget umk is just an interface between Upp logic and compilation toolchain)

Subject: Re: Building applications
Posted by koldo on Wed, 12 Aug 2020 06:02:29 GMT
View Forum Message <> Reply to Message

Dear normvcr

Your Build methods dialog seems to be right, and apparently your install is correct.
I would advice you first to compile some demos. You can choose any going to File/Open main package/examples, choosing one of the packages and compiling it.
As Xemuth explained, U++ structure is through packages, any U++ library is a package, a package can include more packages, and the main application is also a package.
If you feel you need more support you can ask in our Tutoring plan, that is totally free.

Subject: Re: Building applications
Posted by mirek on Wed, 12 Aug 2020 13:39:00 GMT
View Forum Message <> Reply to Message

normvcr wrote on Wed, 12 August 2020 04:17Thank you for the replies! Here is a screen shot of my Build Methods. As you can see, there are no folders listed under LIB directories. The folders that you are showing are under C:\Upp\Clang, these do not seem to be the implementation libraries for Ultimate++. For example, why are the SDL libraries stored under C:\Upp? Are these not generic SDL implementation libraries?

U++ for Win32 ships with a couple of external libraries, that is what you see there.

Quote:
It seems to me that Ultimate++ does not provide a way to create the Ultimate++ libraries....??

That is about correct. Not that it would not be possible, but they are not needed. Nobody cares too much.

In this regard, please consider the situation similar to e.g. Python - would you expect library files for Python libraries?

Any intermediate files are considered the implementation detail (.a files are sometimes produced, sometimes not, whichever fits the required build better).

That frankly means that you are required to use at least umk to build U++ projects. OTOH, you will not have to deal with library paths, build setup, include paths etc.. anymore.

Mirek
