Subject: Cross compile on ubuntu for windows
Posted by mdelfede on Wed, 06 Mar 2024 21:16:50 GMT
View Forum Message <> Reply to Message

After some trial and error, I managed to have a working (and simple) cross compile environment.
Instructions :

1- Download windows UPP package (just tested with upp-win-17185.7z)
2- Unpack it into a folder, example UPP-WIN
3- Wipe all binary files inside root folder, keep just the subfolders
4- Install clang and tools on ubuntu (if you have a working clang on theide it's ok, I guess)
5- Inside the folder UPP-WIN create the file 'myclang' with following content:

```
#!/bin/bash
SCRIPT=$(realpath "$0")
PTH=$(dirname "$SCRIPT")

/usr/bin/clang-14 \
-Qunused-arguments \
-nodefaultlibs -nostartfiles -target x86_64-w64-windows-gnu -fuse-ld=lld \
-L"$PTH/clang/x86_64-w64-mingw32/lib" \
-L"$PTH/clang/lib/clang/14.0.0/lib/windows" \
"$PTH/clang/x86_64-w64-mingw32/lib/crt2.o" \
"$PTH/clang/x86_64-w64-mingw32/lib/crtbegin.o" \
-isystem "$PTH/clang/include/c++/v1" \
-isystem "$PTH/clang/lib/clang/14.0.0/include" \
-isystem "$PTH/clang/include" \
$@ \
-Wl,-Bstatic \
-Wl,--start-group \
-lc++ -lmingwthrd -lmingw32 -lclang_rt.builtins-x86_64 -lunwind -lmoldname -lmingwex -lmsvcrt \
-Wl,--end-group
```

And make it executable.
I user clang-14 ad it's the one used by UPP package... I guess other versions should go as well.
You can try replacing 'clang-14' with 'clang' inside the script, to use latest available clang compiler on your system.

For build method you can create the 'CLANG_WINDOWS64.bm' file along the others:

```
BUILDER = "CLANG";
COMPILER = "myclang";
COMMON_OPTIONS = "-Wall ";
COMMON_CPP_OPTIONS = "-Wno-logical-op-parentheses -std=c++14 -Wno-narrowing
-Wno-unknown-pragmas -Wno-ignored-pragma-intrinsic -Wno-ignored-attributes
-Wno-pragma-pack -Wno-deprecated-declarations -Wno-macro-redefined
-Wno-nonportable-include-path -Wno-mismatched-tags -Wno-overloaded-virtual
-Wno-bitwise-instead-of-logical";
```

COMMON_C_OPTIONS = "";
COMMON_LINK = "";
COMMON_FLAGS = "WIN32";
DEBUG_INFO = "2";
DEBUG_BLITZ = "0";
DEBUG_LINKMODE = "0";
DEBUG_OPTIONS = "-O0";
DEBUG_FLAGS = "";
DEBUG_LINK = "";
RELEASE_BLITZ = "0";
RELEASE_LINKMODE = "0";
RELEASE_OPTIONS = "-O3 -ffunction-sections -fdata-sections";
RELEASE_FLAGS = "";
RELEASE_LINK = "";
DEBUGGER = "gdb";
ALLOW_PRECOMPILED_HEADERS = "0";
DISABLE_BLITZ = "0";
PATH = "/home/massimo/sources_TOOLS/CLANG_WIN";
INCLUDE =
"/home/massimo/sources_TOOLS/CLANG_WIN/SDL2/include;/home/massimo/sources_TOOLS/CLANG_WIN/pgsql/x64/include;/home/massimo/sources_TOOLS/CLANG_WIN/mysql/include;/home/massimo/sources_TOOLS/CLANG_WIN/llvm";
LIB =
"/home/massimo/sources_TOOLS/CLANG_WIN/SDL2/lib/x64;/home/massimo/sources_TOOLS/CLANG_WIN/pgsql/x64/lib;/home/massimo/sources_TOOLS/CLANG_WIN/mysql/lib64;/home/massimo/sources_TOOLS/CLANG_WIN/llvm";
LINKMODE_LOCK = "0";

(here my unpack path is "/home/massimo/sources_TOOLS/CLANG_WIN", replace with your own!)

or just use the gui to add it.

Tested with a complex gui app (www.timberstruct.it) and it works perfectly.

Ciao

Massimo

---