

Hi everyone!

A long post follows. If you are lazy to read, just scroll down for the short version

Almost exactly two years ago, I got inspired by extraordinary features of GNU make to create universal makefile, that could be used to build U++ apps without TheIDE. This beast then evolved and was rewritten few times. It was used used for building packages for ubuntu, and it worked well enough. But few days ago, there came Max and his improvements to gdb interface and broke the builds (no offense Max, it's great work). The reason is that the universal makefile does not support including .brc files. So it is time to take next step in evolution and reveal a new makefile, that I have worked on for quite some time. I didn't want to publish it just yet, as some of the features are not 100% finished and some are even missing, but it builds TheIDE, even with .brc files, so it will save us for now. It can probably be called a beta version.

The biggest difference from the old one is that it does not rely on GNU make features. I tried as hard as possible to keep everything POSIX compatible, so that BSD make and others work with it as well. This resulted in complete change in philosophy, instead of some macro-magic parsing upp files there is now C++ parser embeded in the makefile. It is stored in compressed form and extracted and compiled upon first usage. This parser then reads and analyzes the upp files and creates temporary makefile, that builds the application.

Now good news for fans of other build systems than make: the parser is written with those in mind. The code can be found in uppbox/UppBuilder and whoever is interested can try to write an alternative backend that would generate something else than Makefile, e.g. cmake files or scons script. It would be even possible to create a backend that builds the application instead of just describing how to build it. One strange thing about the parser is that it is NOT written in U++. There are two reasons for this: it must be simple to build and it must be fast to be build. The former is achieved by having all the code in single file and relying on minimum of dependencies (no libs, only 6 standard headers). The later requirement forced me to use almost no templates (only 1 class and 1 function) and compiling without optimizations. The result is ~15kB of code, that is compiled in less than 1s on any modern hardware (tested on my lazy Intel Atom netbook;)), that can parse all the packages necessary to build TheIDE and generate the makefile in about 0.2s. This is definitely an improvement, compared to ~20s that it took with the old makefile. Nice bonus is that the code is IMHO quite well documented, as well as it's usage.

Here are the major changes from the old makefile:

- C++ parser is used instead of make+bash magic
- Better POSIX compatibility and hence greater portability
- Faster... much faster
- Similar, but slightly improved interface (the make variables)
- Handles brc files
- Can perform some custom build steps (experimental)
- Interactive mode: user can be asked for some details, e.g. what configuration to use
- The file counter eye-candy was dropped (it was often broken anyways)

Colorized output - new eye-candy

You can see the generated makefile in uppbbox/lpbuild2/mkfile, if you don't want to examine the UppBuilder too closely.

TL;DR version:

There is new, better universal makefile here, if you used the old one you might want to switch. If you intend to build U++ sources using other build systems than TheIDE or make, you might want to check new UppBuilder package.

Best regards,
Honza
