

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

Subject: Re: Universal makefile & UppBuilder
Posted by [Mindtraveller](#) on Wed, 08 Feb 2012 21:23:18 GMT
[View Forum Message](#) <> [Reply to Message](#)

Very cool!
Did you test its compatibility with *BSD?

Subject: Re: Universal makefile & UppBuilder
Posted by [Didier](#) on Wed, 08 Feb 2012 22:24:55 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Honza,

I'm quite impressed, you went all this way just to enhance UPP build wow
But managing all the portability issues with the make facilities is a pretty tough shot !

I've been thinking for some while that boost-build is very well adapted to Upp package hierarchy and it works on WIN/LINUX/BSD/MAC/.... everywhere there is a decent C compiler.

boost-build is a build tool that has it's own syntax (quite simple) and that manages package dependencies (as in Upp) and it supports many compilers on many OS.
It is written in C ==> if you have a C compiler/linker and you can build it and then use it to build whatever you want ==> you don't depend on presence of make/gmake/imate/xxttools/...whatever.

On big gain is that the same "makefile" works on all platforms !! and that most common compilation options are available in generic form (optimisations, 32/64 bits, ...)

Subject: Re: Universal makefile & UppBuilder

Posted by [dolik.rce](#) on Wed, 08 Feb 2012 22:33:16 GMT

[View Forum Message](#) <> [Reply to Message](#)

Mindtraveller wrote on Wed, 08 February 2012 22:23: Very cool!

Did you test its compatibility with *BSD?

I did tests on PCBSD in early stages of development, but I didn't have time to test it properly in this final form. So there might be some errors on BSD, but probably easy to fix.

Honza

Subject: Re: Universal makefile & UppBuilder

Posted by [dolik.rce](#) on Wed, 08 Feb 2012 22:44:53 GMT

[View Forum Message](#) <> [Reply to Message](#)

Didier wrote on Wed, 08 February 2012 23:24: Hi Honza,

I'm quite impressed, you went all this way just to enhance UPP build. wow. It's my long-term project, I like to see how I can improve on the same topic as my skills are leveling up. Also, I'm actively using it to build packages in launchpad, to avoid chicken-egg problem with TheIDE, so that is a good motivation.

Didier wrote on Wed, 08 February 2012 23:24: But managing all the portability issues with the make facilities is a pretty tough shot. I tried to use only documented, POSIX compatible features, so as long as the tools on other platforms follow POSIX, it should work. Some of the POSIX specifications are linked in documentation, if you're interested.

Didier wrote on Wed, 08 February 2012 23:24: I've been thinking for some while that boost-build is very well adapted to Upp package hierarchy and it works on WIN/LINUX/BSD/MAC/... everywhere there is a decent C compiler.

boost-build is a build tool that has its own syntax (quite simple) and that manages package dependencies (as in Upp) and it supports many compilers on many OS.

It is written in C ==> if you have a C compiler/linker and you can build it and then use it to build whatever you want ==> you don't depend on presence of make/gmake/imate/xxtools/...whatever.

On big gain is that the same "makefile" works on all platforms !! and that most common compilation options are available in generic form (optimisations, 32/64 bits, ...)

Hi Didier,

I've never seen boost-build in action, but I believe that if necessary I could almost match its portability just by writing another backend in some common language. It should be possible to implement the builds in shell script or even in C and avoid make completely. But for now, I believe that supporting systems with make is enough. When the code is stable and bug-less, we can play with other possibilities.

Honza

Subject: Re: Universal makefile & UppBuilder
Posted by [Didier](#) on Wed, 08 Feb 2012 22:59:54 GMT
[View Forum Message](#) <> [Reply to Message](#)

Quote:I've never seen boost-build in action

Well I tried it out a few times mostly at home, but I used it once at work and it saved my ass when I had to turn over the project to another person in another company... which only had a linux station with no make tool installed (and on which it was forbidden to install anything !!!!)

So I just recompiled boost-build, added the 32bit compiler option (the new linux station was 64bits while mine was 32bits) and the app was running fine again.

Subject: Re: Universal makefile & UppBuilder
Posted by [dolik.rce](#) on Thu, 09 Feb 2012 06:35:40 GMT
[View Forum Message](#) <> [Reply to Message](#)

Didier wrote on Wed, 08 February 2012 23:59Quote:I've never seen boost-build in action

Well I tried it out a few times mostly at home, but I used it once at work and it saved my ass when I had to turn over the project to another person in another company... which only had a linux station with no make tool installed (and on which it was forbidden to install anything !!!!)

So I just recompiled boost-build, added the 32bit compiler option (the new linux station was 64bits while mine was 32bits) and the app was running fine again.

Ok, so I've just looked at it and it looks as a nice tool. As long as it is installed on the targeted system. It is binary tool, just as make, so there is not much difference. The portability is nice, but I still think that POSIX compatible shell backend in UppBuilder could be even more portable But it is becoming kind of off-topic, I just wanted to announce that there is a new makefile, if anyone wants to use it

Honza

Subject: Re: Universal makefile & UppBuilder
Posted by [unodgs](#) on Thu, 09 Feb 2012 07:05:43 GMT
[View Forum Message](#) <> [Reply to Message](#)

That's great you put so much effort to get working upp packages on arch linux! But I have a one question. Wouldn't be better to simply compile umk first and then use it to build theide?

Subject: Re: Universal makefile & UppBuilder
Posted by [dolik.rce](#) on Thu, 09 Feb 2012 07:38:52 GMT

[View Forum Message](#) <> [Reply to Message](#)

unodgs wrote on Thu, 09 February 2012 08:05 That's great you put so much effort to get working upp packages on arch linux! But I have a one question. Wouldn't be better to simply compile umk first and then use it to build theide?
How could I compile umk, if you don't have umk to compile it with?

Umk consists of 200+ source files, that require correct flags to be set and correct libraries to be linked. OTOH UppBuilder parser is single file (actually there is 2, but they are merged for the purpose of makefile), which doesn't depend on any flags nor libraries. That simplifies things a lot

Honza

PS: Also, last time I checked, the U++ web said that "if the reinvented wheel spins about 4 times faster, reinventing is not that bad idea..."

Subject: Re: Universal makefile & UppBuilder
Posted by [unodgs](#) on Thu, 09 Feb 2012 07:49:58 GMT
[View Forum Message](#) <> [Reply to Message](#)

dolik.rce wrote on Thu, 09 February 2012 02:38
Umk consists of 200+ source files, that require correct flags to be set and correct libraries to be linked. OTOH UppBuilder parser is single file (actually there is 2, but they are merged for the purpose of makefile), which doesn't depend on any flags nor libraries. That simplifies things a lot

Ok, that's explain a lot.. I thought that recently separated umk is much simpler to compile

Subject: Re: Universal makefile & UppBuilder
Posted by [chickenk](#) on Sat, 11 Feb 2012 20:11:27 GMT
[View Forum Message](#) <> [Reply to Message](#)

Great work Jan!

Just a very small suggestion while I look at the code: it would be interesting to add the '-v' option to hexdump in brc.sh ; I let you read the manual page to understand.

I got stuck with the same problem with waf (i.e. brc files), and I'll try to reuse parts of your new framework, of course!

Cheers
Lionel

Subject: Re: Universal makefile & UppBuilder
Posted by [dolik.rce](#) on Sun, 12 Feb 2012 11:18:49 GMT
[View Forum Message](#) <> [Reply to Message](#)

chickenk wrote on Sat, 11 February 2012 21:11 Just a very small suggestion while I look at the code: it would be interesting to add the '-v' option to hexdump in brc.sh ; I let you read the manual page to understand. Good catch Lionel, hexdump -v is definitely good idea to prevent trouble

Thanks,
Honza

Subject: Re: Universal makefile & UppBuilder
Posted by [sam_](#) on Thu, 06 Sep 2012 20:49:21 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Jan

I've got small bug report for you. On Linux Mint 13 Xfce 64bit (Ubuntu 12.04 based) makefile from trunk r5321 fails:

```
sam@herkales ~/.upp_makefile $ make PKG=Bombs FLAGS="GCC"
/bin/sh: 5: Syntax error: "then" unexpected (expecting "done")
make: *** [do-build] Error 2
```

This is caused by /bin/sh linking to /bin/dash. After

```
sam@herkales ~/.upp_makefile $ sudo rm /bin/sh
[sudo] password for sam:
sam@herkales ~/.upp_makefile $ sudo ln -s /bin/bash /bin/sh
```

makefile works like a charm.

Thanks for your excellent work.
Lubos

Subject: Re: Universal makefile & UppBuilder
Posted by [dolik.rce](#) on Fri, 07 Sep 2012 07:54:02 GMT
[View Forum Message](#) <> [Reply to Message](#)

Thank for report Lubos.

I'll try to fix that as soon as I get home from vacation. Note that you can also force another shell by executing make like this make SHELL=/bin/bash PKG=...

Best regards,

