Subject: How do I compile umk.exe without theide (freebsd)?
Posted by rxantos on Mon, 16 Jun 2014 11:37:34 GMT
View Forum Message <> Reply to Message

How does one compile umk on a freebsd virtual machine without using theide to do so?

Does anyone have a make or cmake files?

Subject: Re: How do I compile umk.exe without theide (freebsd)?
Posted by dolik.rce on Mon, 16 Jun 2014 16:48:42 GMT
View Forum Message <> Reply to Message

rxantos wrote on Mon, 16 June 2014 13:37How does one compile umk on a freebsd virtual machine without using theide to do so?

Does anyone have a make or cmake files?
Hi rxantos,

There is several options

1) Simplest solution: download nightly tarball from http://www.ultimatepp.org/www$uppweb$nightly$en-us.html. It contains a makefile, which should work on freebsd.

2) Generate a makefile in theide on different machine. This would be actually the same makefile as in 1), just without the sources. This way is probably not as practical, but you can be used for other packages as well...

3) Use FreeBSD ports. There used to be U++ port in devel/upp. It'll be probably some older version, but it should be rather easy to update or you could use the old theide to build a newer one.

4) Last possible way is to use the universal makefile. It should be able to build any package. It is a long time since I haven't test it on any *BSD, but it used to work year or two ago.

Hopefully some of that will work for you

Best regards,
Honza

Subject: Re: How do I compile umk.exe without theide (freebsd)?
Posted by Mindtraveller on Wed, 18 Jun 2014 15:50:19 GMT
View Forum Message <> Reply to Message

I do make U++ based software under FreeBSD for some time. The best way I found is the following scheme:
1) ssh connection to production server (no X, just plain FreeBSD)
2) virtual machine (vmware in my case) with almost the same FreeBSD version as in production server, but with X+Gnome and other development software.
3) The development process and debugging is done under virtual machine. After code is ready, it is exported with Makefile.
4) Using rsync, exported sources are updated in production server.
5) Using ssh, an updated binary is built from new sources in production server.

It actually takes less than 30 seconds to export sources, rsync them to production and start rebuilding there (which is usually quick too).

P.S. Please consider switching from default GCC 4.2.1 to GCC 4.6/4.7 for better U++ compatibility.
