Subject: Re: U++ as .lib
Posted by sergei on Mon, 24 Sep 2007 11:00:04 GMT
View Forum Message <> Reply to Message

mr_ped wrote on Mon, 24 September 2007 11:54makefiles are firstly make-dependent.
The system/compiler is issue which can be solved by creating some configuration script and a modular universal makefile. (but it's not a simple thing to do)
Common linux way of building binaries is "configure && make && make install", i.e. the first step is to set up makefile for current system, and to switch on/off modules as you wish it.

So actually you really are reinventing makefiles.

What platforms do you want libs for?
So far mingw+linux+OSX/X11 can work with same (nasty) universal autoconf+makefile.
The other one is needed for MSC.

pkggen.exe looks to me less portable. Some makefile+platforms guru would do this very likely in shorter time.

pkggen.exe is written in U++, source attached. So I believe it should be possible to precompile it as binary for the main platforms.

I agree that CMake or some other make system, if cross-platform/compiler and working, would be a better solution. Yet the configure/make script will look rather ugly, right? Plus, pkggen scans .upp sources for file lists and dependencies. E.g. unless a package is added/removed, nothing has to be modified, neither pkggen.exe nor pkggen.txt. With CMake/configure, I think any modification to a package (add/remove file/dependency) will have to be represented in the script.

The solution IMHO should be a "minimal maintenance" one - one that would allow working on U++ with little regard to the libs. Unless CMake/configure can handle change of filelist, they don't really qualify. A combination of them + .upp scanner could work, though.

Note: I never used CMake/configure, so I might be wrong.