Subject: Re: Building & using U++ without TheIDE
Posted by sergei on Sun, 28 Oct 2007 00:04:13 GMT
View Forum Message <> Reply to Message

There aren't any legal issues with BSD licences - you only have to mark them as modified (and open-source them too, but that will happen anyway).

I agree that as a general practice that would be bad, but I'm only talking about these few:

zlib - used in Core, thus simply has to be modified for SCU to work. Last update in 2005 - unlikely to change.
jpg - Often used. Last update in 1998 - won't change.
png, tif, bz2 - these are still rather active, so changes could be submitted to them. However, I doubt my modifications will be accepted - they are neither bugfixes nor functionality improvements, simply modifications for the plugin to make it work in another way.

I think that only z and png are used in all GUI apps. Thus they are a must. Rest aren't vital for trying out / working with U++. Modification to the (old) png version are simply adding casts. I'll try the latest version later. As I said, "big" ones like sqlite could be compiled in a separate lib.

In short - vital is zlib and png (hopefully new version will just work).

I still have no idea how to solve ReplaceText. That's really a technical challenge - how to make it work with headers and a single CPP file that has to include all of U++. Any help would be welcome.

Options:
1) The way I do it now - #include all packages headers into a single header (UppBase.h) included into program file. #include first all headers, then all sources of all packages into a single CPP file (UppBase.cpp). On MinGW that makes ReplaceText declared in UppBase.h become ReplaceTextA, and implementation in UppBase.cpp becomes ReplaceTextA too. On MSVC, ReplaceText in UppBase.h stays ReplaceText, yet implementation in UppBase.cpp becomes ReplaceTextA - error.
2) The way I did it before - #include everything into a single header (UppBase.h) - can be done per-package (pkg1hdr, pkg1src, pkg2hdr, pkg2src, ...). Then it will work on MinGW and MSVC (all will become ReplaceTextA). But then, every build, even a tiny change in program, will trigger full rebuild of U++ source (in first option that didn't happen since U++ source was in a never-changing UppBase.cpp).
3) Combine option 1 with option 2 - #include U++ per-package into UppBase.cpp, and only the headers into UppBase.h. ReplaceText will become ReplaceTextA in UppBase.cpp in both declaration and implementation, both MinGW and MSVC. However, in UppBase.h, ReplaceText will become ReplaceTextA only on MinGW, and will stay ReplaceText in MSVC - error.

Funny, but polluting the global namespace helps SCU

Hmm, this could also be hacked by #defining ReplaceText ReplaceTextA in RichEdit.h... Is it better than changing the function name?