
Subject: Port of U++ core to windows library publishing

Posted by [hans](#) on Wed, 18 Jun 2014 18:55:06 GMT

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

Hi,

I have developed over the years a port of the U++ core as windows only library.

The name of the library is UPPW.

Rationale is I have only develop for windows and have a thin class library for GUI too which I also want to publish. And NTL is fun!

Now I think to make the port public.

The port consists of only the U++ core (without TCP/HTTP...) with some changes in the String and Stream classes. No IDE, no GUI, nothing else.

Pros would be U++ would be known on larger audience, and more people with multi-platform demands would know about it. Fixes for bugs found in the port could flow back to U++.

What do others, especially the active U++ developers, think?

Subject: Re: Port of U++ core to windows library publishing

Posted by [mirek](#) on Sun, 22 Jun 2014 14:24:05 GMT

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

[hans](#) wrote on Wed, 18 June 2014 20:55Hi,

I have developed over the years a port of the U++ core as windows only library.

The name of the library is UPPW.

Rationale is I have only develop for windows and have a thin class library for GUI too which I also want to publish. And NTL is fun!

Now I think to make the port public.

The port consists of only the U++ core (without TCP/HTTP...) with some changes in the String and Stream classes. No IDE, no GUI, nothing else.

Pros would be U++ would be known on larger audience, and more people with multi-platform demands would know about it. Fixes for bugs found in the port could flow back to U++.

What do others, especially the active U++ developers, think?

Well, spending most of my professional time developing in U++/Win32, I might be a bit curious what is advantage of Win32 U++ Core over Multiplatform U++ Core... :)

But I believe that the point is to ditch theide and just use Visual Studio, am I correct?

As such, it is a good effort. I would perhaps suggest to think about making the "export" of normal U++ Core to UPPW somewhat automatic, perhaps add some `#ifdefs` into Core so that trunk sources can be used.

Thin GUI library is also sounds interesting.

Mirek

Subject: Re: Port of U++ core to windows library publishing

Posted by [hans](#) on Sun, 22 Jun 2014 15:33:38 GMT

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

Hi Mirek,
you are right im using Visual Studio as IDE.

But main reason was getting an "almost complete" class library for Visual Studio Express, where I have no MFC.

The other reason while I greatly honor the effort of making GUI portable, it is for me better to use the GUI of the OS (not to draw the GUI myself) as I'm not programming for Linux.

So not too much hope for the GUI lib, it is basically a WTL-like thin wrapper library on the Win32 system.

For your suggestion to auto-sync with U++, I'm thinking about.

But first I would see how much users it will get. Think I'll publish on CodeProject. But as you can imagine writing good article will take time too.

Hans
