Subject: Re: U++ 2017 beta
Posted by cbpporter on Wed, 04 Jan 2017 09:06:05 GMT
View Forum Message <> Reply to Message

MrSarup wrote on Wed, 04 January 2017 09:47
Sure, here is the link:
http://http://www.wxwidgets.org/

What I meant is a tool that takes out the hassle out of building packages for each platform, not an individual package that may or may not have managed to achieve high Linux distribution penetration.

Quote:
On the main page it declares the following:

"wxWidgets is a C++ library that lets developers create applications for Windows, Mac OS X, Linux and other platforms with a single code base."

This is 100% the same for U++. One code base that works out of the box on some platforms (some temporary problems not withstanding), works with minor hassle on a lot more and can be made to work on anything with a C++11x (or normal C++ if you revert to old versions) with a couple of hours of tweaking. At the end of the day, all you need is a C++ compiler and at msot a dozen of dependencies, most of these related to X.

That's all you need.

And yes, this should be improved. U++ is very well supported for *.deb based distros, but even that fails often. Then, you need to take the tarball and run make. And completely necessary and off-putting fact, but after you do this, you have a fully working U++.

Quote:
I see no reason why I should work on research of what and how U++ distribution needs to be done.

I see where you are coming from, but I'm afraid you can't avoid all the hassle since CentOS is not listed on the download page as a supported platform. I'm sure I could get it to work anywhere from 10 minutes to at most 1h if I ever install CentOS, but end users should have to do this.

Quote:
As a new comer, I see no reason why I should convince you, or any one else in this community, of what is good and what is bad. I came here to use this excellent source code.

The fact remains unchanged: I could not easily work with it. I am more than sure that I will have hundreds of difficulties later on.

As said, there are problems. But they are mostly related to the install, building and library location

process. After this is done, you are very unlikely to encounter difficulties later, except for:
 - GDB integration is not that great, so debugging works but is not the best you have ever seen
 - some more rarely used methods from rarely used classes might have minor bugs. These usually get fixed if reported here, or even better on Redmine.
 - nightly downloads are very stable, but still not as stable as some releases

Quote:
Consequently, the problem starts with U++ on every other level. This is a bottleneck, which is embedded within the system of U++, where many things are required by its special quality that needs to be made easy.

The only real problem I see is TheIDE. With U++ it is TheIDE or the highway. Or umk. I never wanted or liked using TheIDE, but you kind of have to and I have gotten used to it. Trying to not use it is a major hassle that eventually leads everybody to give up on this endeavor. I'm sure that a lot of the user base, if U++ worked out of the box with Visual Studio, would never boot up TheIDE again.

But at the end of the day, it works. Been using it for who knows how many years now (7+) and it works. I view it as a tax: you want Core and the widgets? Well TheIDE is your tax!

I remember in my old job, we got some senior on one of the U++ projects. Very talented guy and great coder. Didn't want to use use TheIDE, but said he'll use it for a day or two to see what's the deal with the packages and learn the code and the move over to Visual Studio. Soon, he became very angry, cursing TheIDE. I asked what's wrong. He way very explicitly cursing it, saying what kind of an IDE does not have an "Open file" option. He just wanted to open a file to edit it and couldn't do it. I showed him that the option was "Edit file", not "Open file". He never touched TheIDE again, nor U++ or the project.

Quote:
And that's not the case. But that's my problem. I need to choose the right open source coding that provides me a good start. After spending many hours, the only thing I could do is compile the theIDE!

I'm afraid that's on you. Getting TheIDE to compile and work, is the one major obstacle and blocking point. After you can compile that and are willing to use TheIDE, compiling anything becomes trivial. All you need to do is provide it with some include paths and lib references and 99% of the code out there should work.