Subject: Re: U++ 2017 beta
Posted by MrSarup on Mon, 02 Jan 2017 05:09:20 GMT
View Forum Message <> Reply to Message

Hello Mirek,

HAPPY NEW YEAR TO MIREK AND THE COMMUNITY OF U++.

mirek wrote on Sun, 01 January 2017 22:03OK, so here is the plan for 2017:
Windows:

Here is my wishlist for 2017:

1) INSTALL AND COMPILE SCRIPTS ON CONSOLE

From Linux, we know that there are shell script-installers available. It guides the whole process by choosing numbers. It also detects if certain dependencies are installed and, if not, does the needful. Release an installer that does all the work on console with bash scripts stable.

For e.g. ./umk.sh could offer a menu to choose, if one wants to install or compile. If one chooses compile, it could offer assemblies as choices after reading the local directories. Choosing one of them, it could offer packages to be compiled.

2) DEVELOPING U++ FOR OTHER PLATFORMS

Change the status of U++ to beta UNTIL IT WORKS ON ALL PLATFORM (and do not laugh on this...). Make distribution available that works on all platforms!

3) MAKE COMPLEX EXAMPLES AVAILABLE

It would be helpful to provide examples related to workflows and multiple windows.

In examples, I have not seen (on windows GUI) that multiple windows could be easily created and attached to workflows. However, I could be - as a new comer - wrong.

After server, cloud hosting, virtual machines, etc. technology became popular and stable, so many people use c++ for their applications. U++ can provide an alternative in this area too.

Here, more examples, complex ones, are needed to show this and help an user to begin with. This could include examples of console applications on client and servers, how they could interact with each other with different methods, like SSH, Web sockets, Proxy services, etc.

4) DISTRIBUTION PACKAGES

Amrein-Marie has compiled one rpm and tested on Fedora and Centos. In the spec file, I found that he did several years ago, and thereafter not. Well, now its there, it could be made available on Sourceforge.

Cross-platform packages needs to be used, like FPM or any other such technologies that detects dependencies and does the needful.

I do not agree with Mirek on his idea that the responsibility of a developer ends by providing a tarball. For an overloaded single developer, he is - in fact - right to say this. But no developer group would say this, if they are working as a developers community.

5) EXTENSIVE DOCUMENTATION FOR CROSS-PLATFORM

From the other thread, you could see that there was no documentation on building umk on console. Everything is based on building theIDE on GUI. Thus umk remained under-developed.

Hello Mindtraveller ,

Mindtraveller wrote on Wed, 28 December 2016 23:12MrSarup wrote on Wed, 28 December 2016 22:53The fact that it is extremely powerful and has not progressed so much shows that there needs to be a change somewhere, strategically speaking.
What exactly do you suggest?
BAN MIREK FROM PARTICIPATING IN THIS FORM! In particular Mirek should not participate in normal postings and spend his time here at the cost of development of U++. Other community members should take this responsibility instead .

Beyond that, my suggestions are above.

One should create an ACTION GROUP to achieve cross-platform quality, where everyone contributes a little so that the sole developer Mirek is not left alone.

Without this, the Usability of U++, on the international platform, is limited. With this understanding of how incompatible U++ on cross-platform is, I question if the time spent on further development is in proportion to its usability.