Subject: Re: U++ 2017 beta
Posted by MrSarup on Wed, 28 Dec 2016 19:53:37 GMT
View Forum Message <> Reply to Message

Hello Mirek and cbpporter,

Both of you have NOT REALLY CAPTURED the crux of the story, although you understand what I have been talking about. I have used both methods, rpm and make. The question of using either of these methods only relates to building theIDE. But if I do not want to build it, I could simply compile a console application.

No one have told me how to compile a simple console application and which commands I need to use it to compile it UNTIL TODAY! In the other thread of FPM, I requested to the poster to let me know what modules i.e. additional cpp files I need to compile with GCC to make a simple console application of SocketServer from reference.

There are no docs. There is no mention anywhere on the website that gives me this information.

What is then this U++ ideal for, to mislead users like me and let them stand alone? I could not work and cannot work until today. So I give up and move forward. This is a classic example of a new comer, who is either not getting help from docs or community or things does not work fundamentally.

I DO NOT WANT TO USE theIDE on Centos 7 but want to code only console applications. Nothing more. So I do not even need to compile with make or rpm, right? Had I known, that this does not work on Centos 7 at all, then I would have saved 20 hours. I needed to read the website, docs, forums, setups, installs, etc.

---
Where does it say on the website that the U++ is not working with Centos 7 and Fedora?
Was it tested before at all by someone?
---

I wasted hours and hours to find out this as a new comer trying to compile a little ServerSocket.cpp on Centos! On windows it worked perfect. Lets see what in ServerSocket.cpp console application: Does U++ have many core modules or functions that A MINI CONSOLE APPLICATION FAILS TO COMPILE or run on Centos/Fedora?

Most likely this IS NOT THE CASE! I suspect that there are functions in the core that may not be needed in there to run a mini console application.

Developers should make some testing on platforms and declare on which platform/version it has been tested. Did the developers know that it is not possible to make a tiny console application on Centos or Fedora?

Currently, I am coding on Laravel PHP Framework. U++ is in principle in the same direction of Laravel Framework on php. They both have similar philosophy. In Laravel, there is composer which works on Mac, windows and Linux. Then, after installing, it does all the work beautifully, like

make, make install for php and configure the application or project. It does not have the usual coding techniques and uses its own convention, just like U++ does. This is the reason why Laravel became the best framework. It installs, configures, coding like charm.

However, here I appreciate U++ very much, as I immediately could recognize its quality, and find that U++ will take a long time to increase its user base or grow its community. This will never work, or at least it will take many years of delay.

To solve this, one - as developers - HAS to make sure to release applications that are stable and easy to handle. If this does not work, the sorry is over before  it began.

And that's the sad future of U++ I see. I am sorry to take a distance with U++ because the docs are terrible, there are not many examples to handle some complex situations, there is less cross-platform compatibility and the community is tiny.

I remember of making an exe in 2010 on windows. I had intended to make it on Mac too. That did not work. After release, I had heard tons of complains from Mac users.

Cross-platform means that most users should be able to use it on variety of platform. Why would I restrict myself with U++, when I can achieve a users group three times bigger by coding on C++. Here, the story ends if U++ restricts me by imposing Handcuffs of coding techniques.

Then I need to say goodbye to U++. And that's the crux of this story.

So, it is certainly not the question of being harsh. It is the question of usability of a software. If one cannot use it or if the use is difficult, then one cannot use it. As simple.

Apart from that, you will not find many honest ones, who are criticizing so openly like myself, take their time and offer a feedback. I merely take the opportunity to discuss here as I do think that people have not understood certain things and live in their little world surrounded by a tiny community.

I also think that the philosophy of U++ coding is more powerful to have a huge community that what it has achieved until today. The fact that it is extremely powerful and has not progressed so much shows that there needs to be a change somewhere, strategically speaking.