
Subject: Re: Which is the biggest drawback of U++ "unpopularity"?

Posted by [mirek](#) on Sun, 30 Nov 2008 22:41:12 GMT

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

Didier wrote on Sun, 30 November 2008 17:09

The main drawback I see to U++ is its code editor and the apparent complexity to build a U++ graphical application without using the editor (sorry, I haven't looked very much).

I understand the sentiment. However, theide was created for purpose.

The main focus was managing modular dependencies and do it across platforms. There is nothing comparable available AFAIK.

Quote:

I'm perfectly aware of the time and energy spent creating the code editor (and it's debugger) but it's far from being perfect and why create what already exists free and much more complete (I'm thinking about Eclipse CDT for example).

If Writing and debugging the code editor wasn't necessary, much more things could be done for the part that really matters, I 'meen:

- The U++ core library and concepts : FANTASTIC
- The Graphical editor and the class inheritance mechanism
- Additional evolutions / libraries.

BTW, there was quite limited development of theide in 2007 and 2008. Maybe that is the reason why it lagged. It is the focus now, if you have not checked the latest svn version (note that current svn is on google-code), maybe do so first.

I see no tool able to solve problems theide was designed to solve.

Quote:

My bet is that with few modification U++ could spread out very easily:

1 - Have an eclipse plugin that manages the graphical part.
(CDT is starting to be a very popular C/C++ code editor)

2 - Change the build process to something more classical

You can switch BLITZ off with single click It is off for release builds by default anyway - nothing special about build process then.

You can also export the whole project with makefile and build that way. Frankly, I was forced to add this feature for the exact reason:

Quote:

(blitz++ is perfect as far as i'm concerned but when you try to convince you're boss or a work colleague the usual reaction is Uhhh? a specific compilation process !?! Can we trust it? is it bug

free ? Is there support for it ? And finally they say : "Forget it". and I understand that position).

If they see the code buildable with makefile in svn repo, they do not care what tool I have used to edit it. Frankly, they do not even care that the project was created with U++ - it is just regular C++ code with no strings attached.

All that said - I will repeat myself, but I strongly support the idea of "library version". This was discussed here a couple of times. I am only not going to do it myself - for me, there is no benefit (except more popularity, of course). I am offering all the support for the project - but in the same time I have to say that currently I do not see how to solve some issues, especially how to regroup current highly modular structure of U++ packages into something more orthodox.

Mirek

P.S.: Maybe, after (eventually) checking the latest svn version, you can write a list of issues you dislike about theide. Even if you are not going to like the idea of its further development, I would like to see it.

Mirek
