
Subject: Re: U++ 2017 beta

Posted by [cbpporter](#) on Tue, 03 Jan 2017 10:21:46 GMT

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

mirek wrote on Sun, 01 January 2017 23:03

- fix include paths issue (if cbpporter will be kind enough to provide info)

I would gladly help, but now, on the first official work day on my main dev machine with 10625 the paths are detected and the build method labeled as MSC14. But this machine worked before, so it should work now.

And if memory serves me correctly, the same paths are detected. I'm starting to suspect that it depends from machine to machine and/or number of MSC versions you have installed and Windows versions. It looks like U++ detected the correct include paths, but the MSC installer did not add all the ucrt or um files there, only some.

Anyway, I'll check the computer I used for the beta review to see the paths.

As for 10625, MSC14 is detected and labelled and works. A fresh install will wait a bit after you click "Accept" until the IDE pops up, just enough to think that it crashed. A simple popup with an oscillating progress bar and "preparing for first launch" would alleviate these problems, but it is not necessary.

Quote:OK, so here is the plan for 2017:

Windows:

- rename MSC15 to MSC14
- return to registry check of MSC. If not found, suggest full directory scan (with prompt first, to avoid long scans).
- update MINGW to the point that debugger works.
- fix include paths issue (if cbpporter will be kind enough to provide info)

Tarballs:

- provide separate 'umk only' make option
- info on using umk to compile
- ? remove .spec ?
- if possible, option to use clang instead of gcc (for distros using buggy 4.8.* GCC)

Looks like a solid plan! U++ code is great right now and except for new features I doubt it has much room to improve (maybe even more move inside the lib?), but do feel free to prove me wrong! :)

So the focus should be on usability. Nothing too fancy, but the package should work out of the box in 99% of the cases on Windows. It is a bit of a self defeating process to have a bit site (this is a pretty big site for a project "nobody" uses) that claims up and down the merits and productivity gains with U++ and then you spend 30 minutes getting it to work. Or 1 minute in my case since I know what is wrong since I had to fix it so many times before. I believe you on the site claims, but newcomers will not.

Anyway, as I said before, part of this is my fault. I have little free time and when I have some, once I track down a bug and report it once, the time is all gone and don't have time to keep checking back, making sure it is fixed.

I'll try my best to improve upon this in 2017. I have at least 6 bugs, 2 minor enhancements and one major one that I have to port into each new U++ version, so I hate updating to a new nightly.

Must use a lot of Redmine...