

---

Subject: Re: New Core

Posted by [cbpporter](#) on Wed, 18 May 2016 19:53:21 GMT

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

---

cbpporter wrote on Thu, 12 May 2016 04:04 Which is the last "old core" version. For bookmarking :).

And not just for bookmarking. For bug-fixing and submitting test-cases. I'm a bit stuck. I need to patch both my code and submit one for TheIDE to detect better older compiler versions. I know that it is legacy mode and no longer supported, but there is still autodecting code, which doesn't work as good as it should. So either remove it or fix it. I really need to take my ugly code and write a package that autodects really well the compiler versions and add it to bazaar and as a dependency in my code.

Then I'm finding bugs left and right, some probably just because I'm doing something wrong, but at least one is in U++, but I can't compile yet with the new core.

Then I'm stuck since my command line project is not supposed to support C++11, but it is also used as a library by the GUI, which doesn't care about C++ version. So if I update, I need to make my code old school C++ and use U++ Core in a way that compiles at least on both.

mirek wrote on Wed, 18 May 2016 21:44

Anyway, while this is sound idea, as I said 10% in size is not worth the trouble. So from the next release on, it will be just speed.

Speed and size is fine.

Actually, I'm having this problem too. I need to support MSC, several versions, TDM, Clang overrides for them, Linux G++ and Linux Clang.

And they hate to agree upon what option means what :). So I'm thinking of only supporting two options, Debug and Release, and to have compiler dependent options as overrides. For MSC: O1, O2, Ox and Od. For G++: O0-O3 and Os. So I'm thinking of adding these options to the GUI based on compiler and no even trying to give them meaningful names. Debug and Release come with sensible defaults and if you touch the Onn options, it means you know what you are doing.

---