
Subject: Re: U++ state

Posted by [dolik.rce](#) on Sat, 09 Jan 2010 10:35:17 GMT

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

Hi Andrei,

Just my quick humble opinions about some of the topics:

andrei_natanael wrote on Sat, 09 January 2010 01:34 IMO (probably I'm wrong here) U++ choose bad operator = for PICK, it should do what it say "equality" and that means that what is in one side is in other side too, i would use <=<= (deep copy?) operator for PICK so you should not have to invent hacks to avoid picking if you didn't want to use it (is that done to have picking for function returned value?) A lot of languages (maybe most of them) actually uses the symbol "=" for "assignment" (and "==" for "equality"). Picking is a type of assignment too and I (and U++ devs probably too) think that it's used most of the time, therefore the short operator - it saves time writing and makes the code faster without programmer thinking about it. If you look for example at python, you'll see that it makes shallow copy on "=" and if you need deep copy you need much longer syntax. So U++ is not really that special in this approach. Also the 'length' of "<=<=" operator IMHO nicely suggest that it takes longer to perform the operation

andrei_natanael wrote on Sat, 09 January 2010 01:34

Even if macros make our work easier(to acquire RAD) i think there are too much macros in a modern framework as U++ and they hide portions of code making it less readable. I'm pro readability even if that means writing 10 chars or more to get it, let's count some macros: THISBACK, PTEBACK, INITBLOCK, EXITBLOCK, __countof, NTL_MOVEABLE, FN*, ONCELOCK, INTERLOCKED, CH_STYLE, CH_COLOR, GUI_APP_MAIN, CONSOLE_APP_MAIN and all these macros are from developer space not from U++ core developer space which contain more macros which make core unreadable in some portions (i.e. code responsible for IML files, LAY files, DLI), IMO there are nicer solutions to solve problems. Some of those macros hide platform differences (e.g. GUI_APP_MAIN) which is IMO very good reason for macro. Others (e.g. THISBACK) increase a readability of the code a lot, which is also very helpful.

Apart from those two points I mostly agree with you. U++ is a great project and helps me a lot even though it still has few glitches.

Best regards,
Honza
