andrei_natanael wrote on Fri, 08 January 2010 19:34
[*] 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?)

The problem there is you have to use copy constructor in order to make pick work for function
returns. No way around it.

I am not 100% happy with it. But that is what C++ gives us.

Maybe some day we could try &&. But there is still that ugly issue with composition rules (&& does
not automatically get generated for classes). Maybe we could find a way around.

Quote:
[*] 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

If macro saves time and (first of all) errors, let us use it. Just because sometimes it is argued that
(over)using macros in some contexts(!) is bad does not automatically makes them all bad. IMHO.

Quote:
IMO there are nicer solutions to solve problems.

Depends on definition of nicer...

Quote:
I know that everyone have limited time and I don't expect any change to come from someone but
I'm putting these here to know what to work on in future to have a better U++.

Sure, you are welcome. I have only commented points where I disagree:)