
Subject: Re: Should the pick semantics be changed?

Posted by [piotr5](#) on Wed, 05 Mar 2014 09:27:29 GMT

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

mirek wrote on Wed, 05 March 2014 08:54

But I still believe it is a good idea to keep "pick" behaviour as default (regardless it is implemented as && or ugly pick_ macro) and operator<=< for deep copy. Also, I found over years extremely useful to maintain the source in picked state (not to clear it, as is AFAIK common for C++11 usage).

I agree, for now when compiling in c++11 the operator=(&) should produce a compilation error whenever instantiated, informing the programmer that picking is default and right side either must be enclosed in std::move (or maybe rather some customized Upp::Pick) or use operator<=< for deep copy. would break some old code (but only when compiling c++11) but is more user-friendly. however, for constructor, in c++11 please let us add a (deep-)copy-constructor and initialization list constructor, preferably both enclosed in a single macro, maybe included in the std-compatibility-stuff...

that way backwards-compatibility wouldn't break, instead c++11-users would encounter a different interface to U++, one where deep-copy constructor is default. or should a similar error-message be issued for the copy-constructor too?
