Subject: Re: How would you design a good copy/move semantics system?
Posted by mirek on Thu, 08 Jan 2015 10:23:21 GMT
View Forum Message <> Reply to Message

cbpporter wrote on Thu, 18 December 2014 12:55What I'm mostly interested in is what are the rules for guaranteeing a maximum invocation of move constructors in implicit situations, like when assigning or returning values.

U++ chose to have default copy to be move for this reason, right?

Originally, yes. But please note that the whole thing changed last year.

Now you have to be explicit, either 'pick' or 'clone' the source, with some exceptions (e.g. returning temporary). (Enforced in C++11, in C++0x it is backward compatible).

IMO, C++11 way, where move constructor is called "sometimes" based on context leads to using deep copy in unexpected situations.