

---

Subject: Re: How would you design a good copy/move semantics system?

Posted by [mirek](#) on Mon, 12 Jan 2015 18:06:21 GMT

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

---

cbpporter wrote on Mon, 12 January 2015 11:21I was interested in this in general, not necessarily in the context of C++, but in the context of value based programming languages.

Well, ideologically speaking, I would say normally there are 2 types of entities:

- those than can be (deeply) copied, usually called "concrete types". Typically, not polymorphic, no abstraction. E.g. Color, Font, Value...
- those that do not have easy meaningful copy - say "identity types". Usually all polymorphic types fall into this category. E.g. File, Ctrl, ...

I like to say that U++ adds to the list 3rd type:

- containers. For these it makes sense to have both 'clone' (in case that they contain concrete types) and 'pick' semantics. 'pick', besides being optimization, nicely adds RAI compatible and MT friendly way to move identity types around.