

---

Subject: Re: Should the pick semantics be changed?

Posted by [mirek](#) on Wed, 05 Mar 2014 07:54:57 GMT

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

---

piotr5 wrote on Tue, 04 March 2014 17:14 have you tried

```
struct Bar {  
    int foo;  
    Vector<int> bar;  
    Bar(Bar&&)=default;  
};?
```

Aah, I have to admit I have missed this one...

10 years ago I was discussing && with its original author (Howard Hinnant) over usenet (<http://www.archivum.info/comp.lang.c++.moderated/2005-08/00708/Re-Rvalue-references--a-done-deal.html>) and at that time, the sentiment was against providing this. I am glad it got through in the end.

Well, in this case I believe && could really replace pick\_ soon. I will be really be only glad for this to happen... It will be safer and more in accord with C++ mainstream.

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).

The only issue I can see now is that for pick types, you have to define twice as much constructors and operator= to keep the current behaviour, you need

```
T(&)  
T(&&)  
operator=(&)  
operator=(&&)
```

But that is only a small price to pay...

The other bad thing is that I unfortunately still have to support the "old" C++ now...

That said, it would perhaps be possible to have U++ "dual-mode", using move constructors with C++11 and pick\_ with old C++.

Mirek

---