Subject: Re: Should the pick semantics be changed?
Posted by piotr5 on Tue, 04 Mar 2014 22:14:55 GMT
View Forum Message <> Reply to Message

have you tried

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

Array<Ctrl> is imho a bad idea to make public, so you just need to define a pick-constructor and delete the copy-constructor. I think with the syntactic sugar of c++11 the original reasons for the current design of pick semantics are becoming obsolete.

with the suffix stuff I am referring to the operator""_suffix() syntax. to call that function you just write any literal and add _suffix at the end. of course you can replace _suffix by whatever name. (additionally there is the constexpr keyword to create objects at compile-time.) so I imagine instead of writing T_("text") next version of Core could also accept "text"_T to perform translations...

I experimented a bit with c++11 and I noticed the programming-style is quite different: std::array sounds more useful than the old c-style arrays, I keep using std algorithms more frequently because function-objects are easier to write, and most object-initialization I am doing with initializer-lists. but I cannot say for sure how this language will evolve. maybe someone will have a nice idea for new language-features by using the new stuff...