Subject: Pick overloaded by Rvalue?
Posted by bytefield on Tue, 11 Nov 2008 22:44:30 GMT
View Forum Message <> Reply to Message

Hi,
Will pick concept be available with new C++0x standard or it will disappear letting his place to Rvalue references?

Taking the next simple example, it behave like using pick for the returned data of the function, no copy-constructor or = op. is involved.

```
// rval.cpp
#include <iostream>
using namespace std;

class X
{
public:
 X()
 {
  cout << "Def ctor\n";
 }
 ~X()
 {
  cout << "Def dtor\n";
 }
 X(const X& )
 {
  cout << "Def cpyctor\n";
 }
 void Msg()
 {
  cout << "Showing a message\n";
 }
 X& operator=(X& )
 {
  cout << "= op\n";
  return *this;
 }
};

X fun()
{
 cout << "In fun(), before X x;\n";
 X x;
 cout << "In fun(), after X x;\n";
 return x;
}

int main()
{
 X&& x = fun();
 x.Msg();
 return 0;
}
```

Quote:g++ rval.cpp -o rval -std=c++0x

If that, have MS introduced(or will) rvalues in their compiler?
Seems g++ is ready for rvalue;

I think because rvalues references are implemented in languages they will be faster than pick concept(which still need copy constructor).

Also i don't know how utf-8 strings will be treated in new standard but if they will be then which string implementation will Upp use it's own or one provided by standard? Porting upp to "new C++" will require some effort .