
Subject: Re: Deepcopying One container
Posted by [mirek](#) on Tue, 03 Nov 2009 18:45:39 GMT
[View Forum Message](#) <> [Reply to Message](#)

dolik.rce wrote on Tue, 03 November 2009 08:21luzr wrote on Mon, 02 November 2009 11:14
Surprisingly, yes, we can provide polymorphic copies - by overloading DeepCopyNew.

That can be simplified by using PolyDeepCopyNew, using virtual Copy method.

In reality, I have never really used polymorphic deep copy, it looks a little bit tricky to me. What is your usage scenario?

Mirek

PolyDeepCopyNew works like a charm! Thank you very much Mirek, the solution was trully genial in its simplicity.

Just for future reference, the solution is something like this:#include <Core/Core.h>
using namespace Upp;

```
class A: public PolyDeepCopyNew<A>{
public:
    virtual void DoSmthng(){Cout()<<"I'm A.\n";};
    virtual A* Copy()const{return new A;}
};
class B:public A{
public:
    virtual void DoSmthng(){Cout()<<"I'm B.\n";};
    virtual B* Copy()const{return new B;}
};
```

```
CONSOLE_APP_MAIN{
    One<A> a=new A;
    One<A> b=new B;
    One<A> c;
```

```
    c<<=a;
    c->DoSmthng();
```

```
    c<<=b;
    c->DoSmthng();
}
```

Honza

IMO, the example is a very little bit misleading - Copy should deepcopy some content...

