
Subject: Re: why no 'Ctrl* Ctrl::Clone() const = 0' (virtual constructor)

Posted by [kohait00](#) on Thu, 09 Sep 2010 08:56:46 GMT

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

i tried it with the following approach, it works in MSC9 but crashed in TDMGCC, any idea why?
(maybe the brute cast is inappropriate..alignment of vtable wrong or sth..)

```
//copyable interface, implementing the Copy function, used by PolyDeepCopyNew
```

```
template<class T, class B = EmptyClass>
```

```
class Copyable : public B
```

```
{
```

```
public:
```

```
    virtual ~Copyable() {}
```

```
    virtual T* Copy() const { return new T(); }
```

```
};
```

```
//this is a nice helper, meant to be used like i.e. PolyCopying<EditCtrl>
```

```
//assigning Copy'ed instances to PolyCopying<Ctrl>* with a cast, as Ctrl is direct base class of  
EditCtrl
```

```
template<class T>
```

```
class PolyCopying : public Copyable< PolyCopying<T>, PolyDeepCopyNew<PolyCopying<T>, T>  
> {};
```

```
////
```

```
PolyCopying<EditInt> abc;
```

```
PolyCopying<Ctrl>* pabc = (PolyCopying<Ctrl>*)abc.Copy();
```

```
PolyCopying<Ctrl>* pabc2 = pabc->Copy();
```

```
Add(pabc2->HSizePos().TopPos(0,100)); //crashes here
```

```
delete pabc2;
```

thats basicly why i need to have the Copy function in Ctrl..it makes factory cloning easy..
