Subject: Re: NTL - "deep copy semantics"?
Posted by nixnixnix on Sun, 09 Sep 2007 00:36:51 GMT
View Forum Message <> Reply to Message

Thanks Mirek,

The fact that you think that what I was thinking was possible was very hard if at all possible, reassured me that I was barking up the wrong tree

Thanks Mr Ped,

I can't share instances in this case as what I need to do is to create independent objects which can share base functionality (I do share instances elsewhere in the same app). However, most of what you said is spot on. I can add additional data to the derived class and still use the base functionality. I just need to accept that the derived class has to manage its specific data which is obvious really when you think about it. I think I was getting just a bit too carried away with the awesome power and beauty of OOP.

Now I've implemented what I originally thought was the "messy" way I see that it is still incredibly elegant and that my derived classes only need to handle their own data and can override the base functionality or not as desired. The only "mess" is two arrays to handle what are in effect the same objects and even then there is only one array explicitly declared in the base class and then the extra array specified in the derived class to hold the additional data.

Thanks to both of you for answering and for letting me know when I'm starting to run up against the edges of the ocean of OOP rather than just my usual small sand-bank in the fog

Nick