Subject: Re: copying of objects
Posted by JD Smith on Wed, 09 Oct 2002 21:25:47 GMT

View Forum Message <> Reply to Message

On Tue, 08 Oct 2002 05:40:52 -0700, David Fanning wrote:

> Sebastian Loebbert (sebaaihb@peach.zrz.TU-Berlin.DE) writes:
>
>> how can I copy an object?
>> E.g.:
>> v0 = OBJ_NEW('IDLgrVolume')
>> v1 = v0
>>
>> v1 only contains only a reference to v1, so I cannot use v0 and v1 in
>> different model trees, which is what I need. I was looking for
>> something like
>> v1 = OBJ_NEW(v0), but this doesn't work.
>
> Copying objects is one of those things that come up over and over again
> in object programming. In practice it seems difficult, because you have
> the problem of "deep" copying (I.e., pointers to pointers, other objects
> with pointers, etc.) You start thinking about recursion, then--of
> course--you get totally confused.
>
> You have a page full of diagrams with arrows going every which way, your
> migraine headache starts up again, and you begin to think, "Oh, hell,
> I'll just go back to thinking about where to invest my money in the
> stock market." It seems so much easier.
>
> Even if you call up RSI and ask the experts there, they give you the
> same old song and dance, and you work on it some more until your eyes
> cross and your backhand has disappeared from disuse.
>
> But it still never works right...until you discover the secret. :-)
>
> Here it is (already published in this newsgroup space, I guess, although
> I must have missed it...

Hi David,

Since one of your own pages
(http://www.dfanning.com/tips/saved_objects.html) documents one
potential peril encountered when using this very technique, I'm quite
surprised you weren't aware of it! Come to think of it, I don't spend
to much time reading my own web pages either;).

Here's the first reference I can find to it, about 4 years ago (ok, it was me).

http://groups.google.com/groups?selm=36152C3D.3C8A3C73%40ast rosun.tn.cornell.edu

In any case, some advice for using this save/restore idea for copying, saving, and reverting objects:

- For quick copying in a single IDL session, there's no need to worry about the method-restoration stuff. If, however, you are restoring a saved copy of an object into a new IDL session, you must deal with the fact that the method definitions might be missed, and the class structure might be incorrect (read the copious posts and David's page on the issue).

- Remember that whatever variable you used to save the object is overwritten upon restore. You can also get the objects with judicious use of the RESTORED_OBJECTS keyword.

- Often, you don't want to copy certain data for which the operation isn't relevant, e.g. temporary widget ID's, etc. You can "detach" these before copying, and "reattach" them to the original by using null pointers (or objects). See my many posts on this method.

- The "self" variable can be overwritten! Hard to believe, but true. This is simultaneously dangerous and useful. Use it to easily restore a saved version of an object right on top of itself (e.g. "Revert to saved version..."), within one of the object's methods. Do not forget to free the discarded version. Search on "transmogrification" (a nomenclature for which I received not a small amount of flack) for more on this possibility.

JD