Subject: Re: Copying an object
Posted by davidf on Tue, 31 Aug 1999 07:00:00 GMT
View Forum Message <> Reply to Message

Pavel Romashkin (promashkin@cmdl.noaa.gov) writes:

> Since object structures are heap variables, couldn't a wizard like David or
> Liam hack a small piece of code to allow unauthorized access to object
> structures directly through pointers? Something like
> a = ptr_valid(Obj_heap_var_number, /cast) ? This would be dangerous but
> probably useful because then duplication would be a one liner.

Well, probably Liam can, but I'm left scratching my head over what seem to me to be insurmountable problems. Randy Frank, who wrote much of this object code when he was at RSI, has sent me a private e-mail outlining even more problems than I was aware of on my own. For example, he points out that many times the model objects (which is what we were discussing earlier this morning) have parents. What should be done about those? Should the copy have the *same* parents, or any parents? What implications does this have if the parents are destroyed or changed in some way? You would have to answer this question for yourself. It is unlikely that a general algorithm could be written.

Of course, if the GetProperties and SetProperties methods of the object are well written, you could simply do something like this:

```
   copyObj = Obj_New('OurObjectClass')
   oldObj->GetProperty, All=allSettableProperties
   copyObj->SetProperty, _Extra=allSettableProperties
```

But in practical applications (at least the ones I've written) this doesn't always work.

You can also try SAVEing and RESTOREing your objects, but this has the problem that sometimes the methods of restored objects can't be found. (See JD Smith's comments on this on my web page.)

And yet, I find that I need a copy of the object quite often. I haven't found a perfect solution yet, but something like the solution outlined on my web page is where I usually start from. So far, I've been able to hack my way around some kind of solution in every case.

Cheers,

David

P.S. Let's just say if Liam *can* write something, I'd be
very interested in seeing it. :-)

--
David Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438 E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155