Subject: Re: Copying (Duplicating) Objects
Posted by Paul Van Delst[1] on Thu, 06 Jun 2002 13:44:38 GMT
View Forum Message <> Reply to Message

Mark Hadfield wrote:
>
> "M. Katz" <MKatz843@onebox.com> wrote in message
> news:4a097d6a.0206050708.7dc1b091@posting.google.com...
>
>> After 4 months away from IDL, I can't remember how to duplicate an
>> object variable.
>
> That's OK. After 10 years with it, I can't remember it either.
>
>> I know that the simple command, b = a, only makes b and a have the
>> same reference--they point to the same object.
>
> Indeed.
>
>> What's the command to make b an entirely new object but identical to
>> a.
>
> There isn't one, at least not built-in. The simplest & most robust
> (though not foolproof) method of doing this is to SAVE the object to
> disk then RESTORE it.

Really? Wouldn't writing your own copy/assign function be more, uh, OO-y? When I create definitions for structures with pointer components in Fortran the very next thing I do is write the Initialize() [nullifies the pointers], Destroy() [disassociates the pointers], Allocate() [associates the pointers], and Copy() [COPIES the entire structure, including the allocation of memory for the pointer components] functions (or methods, to use the obfuscatory OO terminology).

paulv

p.s. And in Fortran, if you wanted to, you could overload an Copy() subroutine with the "=" so that

```
b = a
```

does the same thing as

```
CALL Copy( b, a )
```

Some people have expressed a wish that IDL could do something similar to be truly OO-ey. The fact that fortran can do this (to some extent has *always* done it wrt intrinsic functions) has always made me wonder why this capability was thought of as an OO characteristic.

Anyway...

--
Paul van Delst Religious and cultural
CIMSS @ NOAA/NCEP/EMC purity is a fundamentalist
Ph: (301)763-8000 x7274 fantasy
Fax:(301)763-8545 V.S.Naipaul
