
Subject: Re: Important object lesson

Posted by [davidf](#) on Mon, 29 Jun 1998 07:00:00 GMT

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

Mark Hadfield (m.hadfield@niwa.cri.nz) writes:

- > Interestingly enough, this does not seem to apply to subclasses. If one
- > creates a class
- > MyClass that inherits from SomeOtherClass, then IDL attempts to resolve
- > calls to
- > obj_new('MyClass') by first searching for MyClass::Init & if it can't find
- > that it settles for SomeOtherClass::Init. But as far as I can tell it does
- > this once only, either at the time MyClass is first defined or (more likely)
- > the first time it creates a MyClass. After that the rule "when initialising
- > MyClass call SomeOtherClass::Init" seems to be lodged in its memory banks,
- > with the result that if you later compile a MyClass::Init it will be
- > ignored.

Yes, it must be that when the first object definition is made that the two lifecycle methods are "registered" for the object. Probably much the way keywords are registered for procedures and functions when they are compiled. If you inadvertently forget to define an INIT or CLEANUP method, and you have already made an object of that class, then you must exit IDL and start over for those INIT and CLEANUP methods to be recognized. This does NOT apply to other methods, however, which you can add in the same way you add other procedures and functions to programs. As far as I know, this behavior is not documented anywhere.

- > This explains some confusing experiences I have had with IDL objects.

Indeed. I am beginning to wonder if it is not one source for the slowness of objects in general, apart from the problems of just manipulating this huge 3D space.

- > It also suggests a solution (though not a very elegant one): make all your
- > objects subclasses of something, if only a dummy class, and make sure one of
- > the superclasses has explicit Init and Cleanup methods.

I feel certain you are going to try this, Mark. Could you fill us in on the result? :-)

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting
E-Mail: davidf@dfanning.com
Phone: 970-221-0438
Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

[A copy of this news article was also sent to the author.]
