

---

Subject: Re: Important object lesson

Posted by [Mark Hadfield](#) on Tue, 30 Jun 1998 07:00:00 GMT

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

---

David Fanning wrote in message ...

> Phillip David (pdavid@earthling.net) writes:

>

>> I've been doing some more playing around with objects, and learned that there

>> is a HUGE difference between creating an object with a null constructor and

>> destructor and creating an object with no constructor or destructor.

>> ...

> ...

>

> Actually, I think this is a function of how many directories you have

> on your path. If the INIT and CLEANUP methods are not compiled when

> IDL has to use them, it looks for a file named myclass\_\_init.pro or

> myclass\_\_cleanup.pro in the directories specified by the !PATH

> system variable. If you have a lot of files there, it can take

> a long time!

I imagine that this applies also to the Draw method of objects that are inherited from IDL graphics classes.

Whenever one calls

oDest->Draw, oView

(where oDest is a destination object & oView a view object) , IDL scans through the models & atoms inside the view and for each one (say oAxis) it calls

oAxis->Draw, oDest, oView.

So if oAxis is a member of a class MyAxis, inherited from IDLgrAxis, IDL presumably looks first for MyAxis::Draw, and only if it can't find it anywhere in the search path does it settle for IDLgrAxis::Draw. (I haven't verified that this is the case, but it seems to follow from the behaviour reported by Phillip for Init & Cleanup.)

So perhaps, for performance reasons, one should include an explicit Draw method in any derived graphics class. But it isn't even documented (except for IDLgrModel). What about all the other methods that we want to inherit

--

Mark Hadfield, [m.hadfield@niwa.cri.nz](mailto:m.hadfield@niwa.cri.nz) <http://www.niwa.cri.nz/~hadfield/>  
National Institute for Water and Atmospheric Research

PO Box 14-901, Wellington, New Zealand

>  
>> [Code with and without constructors and destructors snipped.]  
>  
>> When I actually timed these two methods, I found that there was a factor  
of  
>> about 200,000 times between the two. WOW! I had time to create and  
destroy  
>> 200,000 test1 objects for every test2 object I created.  
>>  
>> The moral of this story is: ALWAYS define your constructors and  
destructors  
>> when defining IDL objects.  
>  
> 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/>

---