Subject: Re: destroying graphics objects
Posted by Brad Gom on Tue, 04 Apr 2000 07:00:00 GMT
View Forum Message <> Reply to Message

David Fanning wrote:

> Brad Gom (b_gom@hotmail.com) writes:
>
>> I'm having a little conceptual difficulty with managing graphics
>> objects. I am making a plot object which can contain an arbitrary number
>> of IDLgrPlot objects. I need to be able to add and remove them from the
>> plot at will (and keep track of them in my code). Since this is going
>> to be an object-widget, the less things to keep track of in the code the
>> better.
>
> Sounds like a job for a LIST object. :-)
> You can find one on my web page, if you are interested.

I have a 'stack' object that would also do the trick, but I'd rather properly
use the features of the IDL objects..

>> My question is this: Is is better to store all the IDLgrPlot objects in
>> an IDLContainer object for ease of access, or to just search for them in
>> the IDLgrModel that they are linked to?
>
> A model *is* a container object. That is, a model has inherited
> the CONTAINER object. So anything you add to a model is automatically
> destroyed when the model is destroyed. I'd search for the object in
> the model.

As an aside, did IDL 5.0 have a bug with the
IDLgrModel->get(isa=object_class_name) method? When I use this to return an
object reference for something in a model, it returns the wrong object in IDL
5.0, but seems to work in version 5.3.

>> Should I remove an object from a model
>> before I destroy the object, or does it matter?
>
> I haven't tested this, but I can imagine that a model would have
> a great deal of difficulty rendering something that was no longer
> in existence. Knowing what I do about computer languages, I would
> probably be willing to place a rather largish wager than it *would*
> make a difference. I'd remove any object from the model before
> I destroyed it.

I know this is better form, but it would simplify things for me if I just had
a conainer for just the plot objects and nothing else, and delete the

container when necessary without worrying about what else is in the model.

With attached code, it looks like the model is rendered properly and returns the right valid objects when I delete objects without previously removing them from the model. So long as there are no hidden problems or memory leaks with this technique, then I'll use it.

Thanks,

Brad

File Attachments

1) objtest.pro, downloaded 107 times