Subject: Re: Yet another object graphics question
Posted by Antonio Santiago on Thu, 24 Feb 2005 14:38:48 GMT
View Forum Message <> Reply to Message

Michael Wallace wrote:
> Say that you want to write a utility function which will create a basic
> plot. Let's say this function returned an IDLgrModel with some
> IDLgrPlots on the the inside and various axes and the like. Because all
> of the objects are in a single tree, destroying them is a snap -- just
> destroy the top level object and the destruction cascades down.
>
> Now what do you do if you want to create an IDLgrFont or other "helper"
> object inside the utility function? You can't destroy the helper
> because you'll get an invalid object reference where it had been used
> and once you fall out of the function, you won't have a named variable
> reference to the helper object. The helper will still be present on the
> heap, but there isn't any name to pass obj_destroy. Once I finish using
> the IDLgrModel returned from the function, I can destroy it, but the
> helpers are left dangling.
>
> Is there any rule of thumb ya'll follow for cases like this? I don't
> want to have heap_gc commands in my code just to clean up after myself.
> :-)
>
> -Mike

You can use an IDL_Container object to contains all helper object (like IDLgrFont).
In my case, i store the reference in an IDL_Conteiner. At the moment of the destruction, one called to
OBJ_DESTROY, container
destroy all its associated object.

In my particular case, I use IDL objects to work with Object Graphics and many times stores references to helper objects as class attributes. Perhaps it will be usefull for you.

Bye.
Antonio.