Subject: Re: Yet another object graphics question
Posted by btt on Thu, 24 Feb 2005 14:40:10 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.

Hi,

I'm sure there are plenty of paradigms out there (and I am quite curious to see how others handle this), but I have fallen into a simple one. Here's the definition of a graphic axis similar to one I have used...

```
struct = {My_Axis , $
 INHERITS IDLgrAxis, $
 LocalFont: 0, $
 Font: OBJ_NEW()}
```

And here is the cleanup statement...

```
If (Self.LocalFont NE 0) AND (Obj_Valid(self.Font)) Then Obj_Destroy, self.Font
```

I have a method SETFONT that accepts either and IDLgrFont or a string description of the font (just like in the INIT of an IDLgrFont object.) In the latter case, the font is considered 'local' and will be cleanup up by this object, otherwise I assume that the font is being managed 'globally' and that some other object will be cleaning it up. I assume that the title and ticktext share the same font; keeping a reference to it in the ticktext, the title and in the class definition isn't very "heavy" if you know what I mean.

Cheers,
Ben