Subject: Re: Object Widgets
Posted by m218003 on Tue, 16 Nov 1999 08:00:00 GMT
View Forum Message <> Reply to Message

In article <38306F7E.F5371DDE@astro.cornell.edu>,
"J.D. Smith" <jdsmith@astro.cornell.edu> writes:
>
> I often do all sorts of things in the Start method. Usually they are
> along the
> lines of setting up variables (like window id's), that don't yet exist
> until a
> subsidiary object is Started. Also for drawing to widget_draw's, and similar
> things which require the widgets already to have been started up. But it's
> not
> just for widget issues.
>
> For instance, I have a color manager object which other object widgets
> inquire
> for various colors to set themselves up. Until this object is initialized,
> it
> makes no sense to inquire things of it. If there were only one object
> utilizing
> it... no problem -- just make sure it's initted first. But when many objects
> being initted in many different places might be involved, this is *much*
> easier.
>
> Another nice thing about having a standard method "Start" is that a
> controlling
> class can automatically "Start" all of it's composited objects, without
> knowing
> the details of what they're doing (be it widget or otherwise).
>

The last couple of days I experimented a litle with a hierarchy of objects
defining a rectangular box (superclass), a page (one subclass branch)
and a [plotting] frame (another subclass branch). Halfway through it occured
to me that probably the best way to initialize the object is to call its
own SetProperty method from the Init method (of course you need to make
sure that the inherited methods are called as well and you must set
default values for all possible keywords). I then recalled that this is what
was proposed in onebook about OOP that I read (wait - 8 years? ago).
Upon second thought, this method (pun intended) of course only works
for the public properties of your object, i.e. those that are "visible"
to the SetProperty method. The advantage of this approach is that you
don't need to check your keywords twice for correctness.

Here's a quick example:

```
pro bogus::SetProperty, afloat=afloat, anintarr=anintarr

   ; make sure arguments are correct
   if n_elements(afloat) eq 1 then self.afloat=float(afloat)
   if n_elements(anintarr) gt 0 then self.anintarr = fix(anintarr)

end


function bogus::Init, afloat=afloat, anintarr=anintarr

   ; need only set default values here
   if n_elements(afloat) eq 0 then afloat = 0.
   if n_elements(anintarr) eq 0 then anintarr = intarr(1)

   self -> SetProperty,afloat=afloat, anintarr=anintarr

end
```

Any thoughts about this?

Regards,
Martin

--
[[[[[[[[[
[[Dr. Martin Schultz   Max-Planck-Institut fuer Meteorologie   [[
[[Bundesstr. 55, 20146 Hamburg   [[
[[phone: +49 40 41173-308   [[
[[fax: +49 40 41173-298   [[
[[martin.schultz@dkrz.de   [[
[[[[[[[[[