Subject: Re: Objects with Widgets, Save/Restore
Posted by John-David T. Smith on Fri, 03 Aug 2001 16:55:47 GMT
View Forum Message <> Reply to Message

Brent Griffith wrote:

> I am using a handful of home-made objects (thanks to RCK's book) in a
> application with lots of widgets. I am trying to use SAVE and RESTORE
> to get my objects back to where they were in a previous run of the
> application. I have it sort of working but would appreciate seeing
> some examples of an "object's restore method".
>
> For example, is there an elegant way to deal with the widget IDs and
> values? I am storing widgetIDs in the object for widget_control
> statements, but the newly created widgets generally have different IDs
> than the old Saved-object. I find myself repeatedly writing the same
> statements for each and every widget that might get its value changed
> -- see example statements below (* and 1 indicate multiplicity).

You might check older postings of mine which comment on this very topic ad naseum. The basic idea is to detach those portions of the object data structure which aren't relevant to a saved version, before saving. This requires a bit of organization, but is pretty easy. Here are a few messages I found regarding this method:

 http://groups.google.com/groups?selm=361AC6B3.436DB6E%40astr
osun.tn.cornell.eduhttp://groups.google.com/groups?selm=3847
128F.B532A27F%40astro.cornell.edu
 http://groups.google.com/groups?selm=37B4F36B.E1D4BCEC%40ast rosun.tn.cornell.edu

Here's an example, excerpted from a complicated class of mine:

```
***************
  ;; Detach the stuff we don't want to save!
  detInfo=self.wInfo & self.wInfo=ptr_new() ;don't save the info
  detMsgList=self.MsgList & self.MsgList=ptr_new() ;or the message list
  ;; Here we detach the data to avoid duplication, if they want it.
  if self.SpaceSaver eq 1b then begin
     detDatas=(*self.DR).DATA  ;or the data
     (*self.DR).DATA=ptrarr(self->N_Records())
     detTotals=(*self.DR).TOTAL ;or the totals
     (*self.DR).TOTAL=ptrarr(self->N_Records())
  endif else self->RestoreAll

  oldchange=self.Changed       ;we want the file written to have
changed=0!
  self.Changed=0b             ;but save the old preference incase we
```

```
fail
  catch, serr
  if serr ne 0 then begin      ;it failed!
    catch,/CANCEL
    self.wInfo=detInfo        ;reattach them
    self.MsgList=detMsgList
    if self.SpaceSaver eq 1b then begin ;reconnect the data
      (*self.DR).DATA=detDatas
      (*self.DR).TOTAL=detTotals
    endif
    self.Changed=oldchange    ;reassign our old changed status
    wmessage,PARENT_GROUP=pg, $
     'Error Saving to File: '+pname
    return
  endif
  save,self,FILENAME=pname
  catch,/CANCEL
  ;; Reattach
  self.wInfo=detInfo
  self.MsgList=detMsgList
  if self.SpaceSaver eq 1b then begin
    (*self.DR).DATA=detDatas
    (*self.DR).TOTAL=detTotals
  endif
***************
```

Notice the error handling, especially. You don't want to lose your running widget hierarchy if the save fails. I basically remove the entire widget info structure (which is nicely organized under a single pointer) among a few other things, and then on restoration, I can rebuild the widgets there again... something like:

```
***************
  ;; make the info structure if we need it.
  if NOT ptr_valid(self.wInfo) then begin
    self.wInfo=ptr_new({scoreProj_wInfo})
  endif
  self.wInfo.base=widget_base(...
***************
```

You probably get the point.

Good luck,

JD