Subject: Re: Destroying Multiple Widgets
Posted by David Fanning on Tue, 04 Sep 2007 13:36:58 GMT
View Forum Message <> Reply to Message

2d@onetel.com writes:

```
>  Could anyone help me with a widget program to create multiple widgets
>  which are not linked to a single base widget, so I can use the space
>  more economically on the screen?
>
>  All of the widget programs I have seen link descendent widgets to a
>  base widget via the widget_base commands, and then quit all widgets by
>  destroying the base widget.
>
>  The problem with this is that having all widgets within a single base
>  widget space means there is lots of unused space in the base widget.
>  Better would be to have each widget on its own so there is no unused
>  space in the widgets windows.
>
>  To do this a tried creating multiple base widgets and this created the
>  widget effectively. However, when I tried to destroy them with widget
>  control, IDL complained that I was supplying an invalid widget ID. As
>  I printed out the widget ID, I knew that it was not invalid.
>
>  I have transferred the widget ID of the second base widget to be
>  destroyed through a common block, as below. In programme (omitting
>  some lines for clarity) I had:
>
>  ; Widget event handler module
>  pro widgets_quit, ev
>  common imager,  base_id2
>  widget_control, ev.top, /destroy
>  widget_control, base_id2, /destroy
>  end
>
>  ; Widget definition module.
>  pro widgets
>  common imager, base_id2
>  base_id    = widget_base(title = 'Control',  /column)
>  base_id2  = widget_base(title = 'Control2', /column)
>  quit_id      = widget_button(base_id, value='finish,
>  event_pro='widgets_quit')
>  widget_control, base_id, /realize
>  widget_control, base_id2, /realize
>  xmanager, 'widgets', base_id
>  xmanager, 'widgets', base_id2
>
>  Problem is when I run this I get an error: % XMANAGER: Invalid widget
```

> ID.
>
> Any suggestions? For example is multiple xmanager calls in the widget
> definition module allowed.

I think your error message is caused by your second XMANAGER
call not being executed (since the first XMANAGER blocks)
until after you have destroyed the base_id2 widget. Either
set a NO_BLOCK or JUST_REG keyword on the first XMANAGER call
and the error will probably go away.

But this is the wrong way to do this, anyway. :-)

It relies on your user ending the program with the QUIT
button, which users seldom do. Instead, they kill the
widget with their mouse. What you want to do is use
a CLEANUP procedure to find out when a widget dies. (You
can use a CLEANUP keyword on the XMANAGER call to assign
the cleanup procedure.) And you want to use a GROUP_LEADER
keyword, which ITTVIS is always shortening to GROUP, to
assign group leaders to your widgets. When a group leader
dies, any widget belonging to that group will die with
it. You could have something like this.

```
; Widget event handler module
pro widgets_quit, ev
widget_control, ev.top, /destroy
end



; Widget definition module.
pro widgets
base_id    = widget_base(title = 'Control', XSIZE=100)
base_id2  = widget_base(title = 'Control2', XSIZE=100, $
   group_leader=base_id)
widget_control, base_id, group_leader=base_id2
quit_id = widget_button(base_id, value='finish', $
    event_pro='widgets_quit')
widget_control, base_id, /realize
widget_control, base_id2, /realize
xmanager, 'widgets', base_id, /just_reg
xmanager, 'widgets', base_id2
end
```

Cheers,

David
--

David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")