Subject: Re: Widget Event_Pro question
Posted by Spon on Mon, 31 Mar 2008 14:14:46 GMT
View Forum Message <> Reply to Message

On Mar 31, 2:45 pm, David Fanning <da...@dfanning.com> wrote:
>
> This will depend entirely on what kind of IDL programmer you
> are. :-)
>
> You can't have two different event handlers assigned to a
> widget at the same time, but there is nothing that prevents
> that single event handler from being an event dispatcher.
> The event comes in. The event handler figures out what kind
> of event it is (e.g., button up, button down, motion,
> expose, etc.), and then the event is passed onto some other
> IDL procedure or function for further processing. Often,
> the info pointer is also needed, along with the event structure,
> to completely handle the event.
>
>   PRO MyProg_DrawWidgetEventProcessing, event
>
>   ; Get info pointer.
>   Widget_Control, event.top, GET_UVALUE=infoPtr
>
>   ; What kind of event is this?
>   kind =
> ['DOWN','UP','MOTION','VIEWPORT','EXPOSE','CH','KEY','WHEEL']
>   CASE kind[event.type] OF
>   'DOWN': MyProg_HandleButtonDownEvents, event, infoPtr
>   'UP': MyProg_HandleButtonUpEvents, event, infoPtr
>   'MOTION': MyProg_HandleButtonMotionEvents, event, infoPtr
>   ELSE: ; Don't care.
>   ENDCASE
>
>   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.")

Hey, that's a neat trick, thanks David! Looks like it could keep
things a little more organised in my event handler structure; and God
knows I need all the help I can get when it comes to being

organised! :-)

As it happened, I just chickened out and went for for an
  IF Event.Release THEN BEGIN
statement in this case, but it'd only be a short cut-'n'-paste from there to calling a separate handler programme to do what I've shoehorned in after the BEGIN...

...which includes the following, and has me struggling with widget geometry (and this is _before_ I'm even going near cross-platform stuff :-\ )

```
    ; Prepare a second widget base:
   GraphBase = WIDGET_BASE(/COLUMN, $
     GROUP_LEADER = Event.Top, $
     TITLE = "Press 'Done' to close window.", $
     /ALIGN_RIGHT)
```

to which I add a simple 'done' button, a couple of labels to remind me where in the image I'd clicked to get this graph, and a draw widget to hold my graph.

Now, if instead of Group_Leader I go for Event.Top as my *parent*, the graph window appears in a sort of added-on blob to my original window, with the result that the 'Done' button kills the whole shebang, instead of just closing my graph window and letting me get on with wasting CPU processing power dragging my mouse over the image window.

But if I keep Event.Top as my Group_Leader, then the /ALIGN_RIGHT keyword doesn't seem to do a whole lot, and my graph widget sits smack bang on top of my image window widget.

Typically, I want to have my cake *and* eat it ;-)
I'd like my second widget base to be independent enough of its leader to be a separate entity, but to be *aware* enough of the leader to know where I want it to sit, namely glued to its right edge. Is this possible?

Thanks for the help so far,
regards,

Chris