Subject: Re: processing widget events from multiple hierarchies
Posted by ronn on Tue, 21 Aug 2001 01:27:04 GMT
View Forum Message <> Reply to Message

in article 998348877.720166@emperor.labs.agilent.com, Rick Baer at
rick_baer@agilent.com wrote on 8/20/01 7:07 PM:

> I would like to use a menu command to invoke a modal dialog box. If I
> create and realize the widgets that make up the dialog box without invoking
> xmanager, the events aren't handled. If I invoke "xmanager" with the new top
> level base as an argument, the events are handled by the default event
> handler ("AppName_event"), but the handlers that I have specified (e.g.
> event_pro="AppName_eventX") are not used. How can I make the message loop
> aware of the new event handlers?
>

If all the widgets use the Appname_eventX handler then you could try using
the event_handler keyword to xmanager. This overides the default
AppName_event way of doing it. But your original way should still work, any
chance of showing a small section of code?

-Ronn

--
Ronn Kling
KRS, inc.
email: ronn@rlkling.com
"Application Development with IDL"ï¿½ programming book updated for IDL5.4!
"Calling C from IDL, Using DLM's to extend your IDL code" NEW BOOK!
http://www.rlkling.com/

Subject: Re: processing widget events from multiple hierarchies
Posted by Rick Baer on Wed, 22 Aug 2001 18:15:38 GMT
View Forum Message <> Reply to Message

```
; Code fragment illustrating problems with event_pro specification.
; Events go to XWidTest_event( ), not XWidTest_Param( ),
XWidTest_ParamAccept( ), XWidTest_Param2( ).

;///////////////////////////////////////////////////////////////
;/// modal dialog ///

pro XWidTest_Action3, event

    widget_control, event.top, Get_UValue=info, /No_Copy

   ; If this widget already exists, then don't duplicate it
```

```
    if (NOT widget_info(info.baseAID, /valid_id)) then begin
       info.baseAID = widget_base(column = 1, title = 'parameters')

       baseA1ID = widget_base(info.baseAID, column = 2)
       labelA1ID = widget_label(baseA1ID, value = "parameter 1")
       editA1ID = widget_text(baseA1ID, value="edit 1", uvalue = "PARAM1",
/EDITABLE, event_pro =    XWidTest_Param)

      ; .... other base/label/text commands omitted for clarity

      buttonAID = widget_button(info.baseAID, value="accept", event_pro =
XWidTest_ParamAccept)

      widget_control, info.baseAID, /realize
   endif   ; (widget already exists test)

   xmanager, 'XWidTest', info.baseAID, /NO_BLOCK, event_handler =
XWidTest_Param2

   ; restore the user data handle
   widget_control, event.top, Set_UValue=info, /NO_COPY

end

;////////////////////////////////////////////////////////
;/// main program ///

pro XWidTest

    base1ID = widget_base(column = 1, title = 'base 1')
    menuID = WIDGET_BUTTON(base1ID, VALUE='Test', /MENU)
    action3ID=WIDGET_BUTTON(menuID, VALUE='pop-up dialog', $
       UVALUE='ACTION3', EVENT_PRO="XWidTest_Action3")

    widget_control, base1ID, /realize

    ; place holder for dialog widget
    baseAID = -1L

    ; pass pointer to window data
    info = {baseAID:baseAID}
    widget_control, base1ID, set_uvalue=info, /NO_COPY

    xmanager, 'XWidTest', base1ID, /NO_BLOCK

end
```

Subject: Re: processing widget events from multiple hierarchies
Posted by ronn on Wed, 22 Aug 2001 21:02:24 GMT
View Forum Message <> Reply to Message

Hi Rick,

It looks like the only mistake was that the event_pro keyword must be a string. The fix is below...

-Ronn

--
Ronn Kling
KRS, inc.
email: ronn@rlkling.com
"Application Development with IDL"ï¿½ programming book updated for IDL5.4!
"Calling C from IDL, Using DLM's to extend your IDL code" NEW BOOK!
http://www.rlkling.com/

```
-------------cut here-------------------------------
; Code fragment illustrating problems with event_pro specification.
; Events go to XWidTest_event( ), not XWidTest_Param( ),
;XWidTest_ParamAccept( ), XWidTest_Param2( ).

;////////////////////////////////////////////////////////
;/// modal dialog ///
pro xWidTest_param,event
print,'you pressed enter in the text widget'
return
end

pro XWidTest_ParamAccept,event
print,'you pressed the accept button'
return
end

pro XWidTest_Action3, event

    widget_control, event.top, Get_UValue=info, /No_Copy

   ; If this widget already exists, then don't duplicate it
    if (NOT widget_info(info.baseAID, /valid_id)) then begin
       info.baseAID = widget_base(column = 1, title = 'parameters')

       baseA1ID = widget_base(info.baseAID, column = 2)
       labelA1ID = widget_label(baseA1ID, value = "parameter 1")
       editA1ID = widget_text(baseA1ID, value="edit 1", uvalue = "PARAM1",$
        /EDITABLE, event_pro ='XWidTest_Param')
    ; .... other base/label/text commands omitted for clarity

    buttonAID = widget_button(info.baseAID, value="accept", $
      event_pro ='XWidTest_ParamAccept')

    widget_control, info.baseAID, /realize
  endif   ; (widget already exists test)

  xmanager, 'XWidTest', info.baseAID, /NO_BLOCK, event_handler =
'XWidTest_Param2'

  ; restore the user data handle
  widget_control, event.top, Set_UValue=info, /NO_COPY

end

;////////////////////////////////////////////////////////
;/// main program ///

pro XWidTest

   base1ID = widget_base(column = 1, title = 'base 1')
   menuID = WIDGET_BUTTON(base1ID, VALUE='Test', /MENU)
   action3ID=WIDGET_BUTTON(menuID, VALUE='pop-up dialog', $
     UVALUE='ACTION3', EVENT_PRO="XWidTest_Action3")

   widget_control, base1ID, /realize

   ; place holder for dialog widget
   baseAID = -1L

   ; pass pointer to window data
   info = {baseAID:baseAID}
   widget_control, base1ID, set_uvalue=info, /NO_COPY

   xmanager, 'XWidTest', base1ID, /NO_BLOCK

end
```