
Subject: Modal widgets (Re: Widgets and research)

Posted by [Stein Vidar Hagfors H](#) on Mon, 23 Jun 1997 07:00:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

David Foster wrote:

>
> In IDL 5.0 the implementation of modal widgets is completely different
> than it was previously, and this has major implications for how
> widget applications work together when one or more of them are modal.
> In our lab we often have a series of apps that are called
> successively, and I'm finding that all the "rules" are different
> as far as what combinations are possible.
>
> For example, the following scenario is ok:
>
> MODAL_WID --> NON_MODAL_WID --> MODAL_WID
>
> (modal widget calls non-modal widget etc.).
>
> But the following is NOT ok:
>
> MODAL_WID --> MODAL_WID --> NON_MODAL_WID --> MODAL_WID
>
> When you do this the last modal widget does not generate any events.
> This use to work in IDL 4.0.1, now it doesn't. I'm finding other
> similar problems related to modal widgets. If you're lucky, you
> don't use modal widgets often, and so won't notice these problems.

I agree that something's amiss here, though I don't fully understand what you wrote. To make a modal widget "the version 5 way" you have to have a group leader (why! see below), so your widget tree cannot start with a modal one. It has to be

NON_MODAL --> NON_MODAL --> MODAL (works ok)

and

NON_MODAL --> MODAL --> NON_MODAL [---> MODAL]

The brackets around the last MODAL widget instance indicates that it's impossible, since the last NON_MODAL one doesn't process events.

And this is a problem (read: bug)! Let's say I want to let the user start e.g., a general on-screen calculator to calculate some input to a modal widget. I **don't** necessarily want this general calculator to be modal (disappearing with the answer before the modal parent widget will accept it!), nor a group member, but according to RSI I have to!

I think this is caused by the fact that the people at RSI only had tiny, fashionable "dialog" widgets in mind when rewriting the modal widget stuff. Modal widgets are modal mainly because further processing by their parent process(es) is meaningless without some result from the modal widget - this does **not** imply that all modal widgets are simple dialogs, or that all other activities should be blocked.

I tend to think of IDL widgets as a potential "operating system" that allows various (not necessarily related) processes to coexist, but RSI seems to think that this should instead be implemented by having more than one IDL session - sigh :-}

And: Why no modal widgets without group leaders? What's **wrong** with allowing an essentially non-widgety routine (that doesn't care about group leaders) pop up an occasional widget with a question or warning in a modal widget? With version 5, this is not possible without either sending a group leader ID all the way down to the bottom level routines, or creating a dummy non-modal TLB because modal widgets must have group leaders...

I do like the way a modal widget stays on top of it's group leader in version 5, but I'm a bit puzzled about other aspects of modal widgets...

This is what I would **like** to be able to do, though:

1. Creating a modal widget **with** a group leader ID should block events in **that** widget hierarchy (maybe all the way up to the topmost group leader and all of it's group members) - **not** blocking events in totally unrelated widgets that are not at all related.
I mean - what would you feel like if your email program froze up all your applications when you're composing an email!

2. Creating a modal widget **without** group leader should be **allowed**, and it should block events in **all** widgets.

3. Please..make it possible to turn off this warning:

% XMANAGER: The MODAL keyword to the XMANAGER procedure is obsolete.
It is
superseded by the MODAL keyword to the WIDGET_BASE function.

Below is a pretty minimal demonstration program that highlights the possibilities (and **impossibilities**) of version 5 widgets.

(David F.: I hope this one's got no "viruses" :-)

Regards,

Stein Vidar Haugan

PRO witest_event,ev

widget_control,ev.id,get_uvalue=uvalue

CASE uvalue OF

"QUIT":BEGIN

widget_control,ev.top,/destroy

return

ENDCASE

"MODAL_GROUP":BEGIN

witest,/modal,group_leader=ev.top

ENDCASE

"NONMODAL_GROUP":BEGIN

witest,group_leader=ev.top

ENDCASE

"MODAL_NOGROUP":BEGIN

witest,/modal

ENDCASE

"NONMODAL_NOGROUP":BEGIN

witest

ENDCASE

END

END

PRO witest,modal=modal,group_leader=group_leader

IF n_elements(group_leader) EQ 0 THEN group_leader=0L

IF n_elements(modal) EQ 0 THEN modal=0L

text1 = "I'm a "+(["nonmodal","modal"])(modal NE 0)+"* widget"

text2 = "I have "+(["a","no*"])(group_leader EQ 0)+" group leader"

IF 1 and !version.release EQ '5.0' THEN BEGIN

base = widget_base(/column,group_leader=group_leader,modal=modal)

```
    modal = 0
END ELSE BEGIN
    base = widget_base(/column,group_leader=group_leader)
END

dummy = widget_label(base,value=text1)
dummy = widget_label(base,value=text2)

quit = widget_button(base,value='Quit',uvalue='QUIT')

mgchild = widget_button(base,value='Modal member',$
                        uvalue='MODAL_GROUP')
nmgchild = widget_button(base,value='Nonmodal member',$
                        uvalue='NONMODAL_GROUP')
mgchild = widget_button(base,value='Modal nonmember',$
                        uvalue='MODAL_NOGROUP')
nmgchild = widget_button(base,value='Nonmodal nonmember',$
                        uvalue='NONMODAL_NOGROUP')

widget_control,base,/realize

xmanager,'witest',base,modal=modal
END
```
