
Subject: No, no, not again!

Posted by [steinhh](#) on Mon, 16 Nov 1998 08:00:00 GMT

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

<Rant mode on>

As John Krist mentioned, one particularly painful aspect of new IDL version releases is the way they almost invariably break some existing, well-functioning code (even when written in the most kosher way for previous versions), especially widgets.

Sadly, I just discovered (yet) another one of these instances. The "new and improved" handling of modal widgets is the culprit in this case.

Modality used to be the method of choice whenever a widget program needed to return something - this made sure the widget program didn't "fall through" the XMANAGER call and return before the answer was finished - no matter how complex the task.

(Note that for such applications, it's not very meaningful to switch modality on or off through a keyword - if you're returning data through return values or output parameters, you **should never** return to the caller before you're finished).

With the old system (xmanager,...,/modal), modal widgets could have any level of complexity - no questions asked. With the "new and improved" system (widget_base(...,/modal)), RSI has changed the meaning of modality!

Modality is now apparently meant exclusively to make braindead **dialogs**. When I say braindead, I mean braindead as in "less than an application". Indeed, more like just an "Abort/Retry/Fail?" question than an application!

Let's say I have application A, which may call application B in order to massage some of the data "owned" by A. Naturally, I want to be able to shut off application A until B is finished, since the state of the data in A may be "undefined" while B is working on them. So, application B should be modal. Fine.

The problem is that if application B needs to have "auxiliary" widgets, those widgets now have to be modal! (Since their group leader is modal - check the doc's for widget_base/group_leader keyword).

But I don't **want** those to be modal! I want to have my analysis program work, **and** have the auxiliary widget launched from it work at the same time. I want my auxiliary widget to exist in peace and harmony

with application B, letting the user control the state of the color scaling method continuously while using application B!

I can't see any way of doing this the new way, and it's a serious flaw in the "new and improved" modal method (widget_base(...,/modal))!

But the **worst** thing is that this could be done in version 4 with the old modality (xmanager,...,/modal), and now it's ^%#^%# broken! If I start a non-modal widget alongside a modal application, the non-modal widget doesn't register any events. It can't even use the window manager to close the window! This is on { alpha OSF unix 5.1.1 Jul 20 1998}.

(Though through some **weird** mechanism, clicking and dragging the mouse in a draw window in the non-modal widget a few times somehow manages to "wake" it up for a while, but I guess this is a bug somewhere, not a feature!)

Below is an example program. To demonstrate the effect, start it with "IDL> test", press "Push to start new (modal)", then press "Push to start new (nonmodal)" on the new widget. The last widget appearing (#3) is all "dead" on my machine, while #2 is alive. #3 does disappear though, when I exit #2. One funny thing is that #3 dies with #2, even when I do **not** propagate the information about the group leader from the caller (#2)...??

I do hope that RSI will give the old functionality of the xmanager,...,/modal call back (5.2 ??? I guess I'm too late). It's just too silly not to be able to write modal widgets of any level of complexity. I know that it's **possible** to write workarounds for this (like doing manually the things xmanager used to do when I want widgets to be modal in the now unsupported way), but that's too silly as well. My programs used to work, and now they don't!

<Rant mode off>

Regards,

Stein Vidar

Test program: -----

```
PRO test_event,ev
  widget_control,ev.id,get_uvalue=uval
  CASE uval OF
    'NEWN': test,group=ev.top
    'NEWM': test,/modal,group=ev.top
    'QUIT': widget_control,ev.top,/destroy
```

```
END  
END
```

```
PRO test,modal=modal,group=group
```

```
IF n_elements(group) NE 1 THEN b = widget_base(/column) $  
ELSE                          b = widget_base(/column,group_leader=group)
```

```
bb = widget_button(b,value='Push to start new (modal)',uvalue='NEWM')  
bb = widget_button(b,value='Push to start new (nonmodal)',uvalue='NEWN')  
bb = widget_button(b,value='Push to exit',uvalue='QUIT')
```

```
widget_control,b,/realize
```

```
IF keyword_set(modal) THEN xmanager,'test',b,/modal $  
ELSE                      xmanager,'test',b  
END
```
