
Subject: Where O Where Di My Pointer Go?
Posted by [phillipbitzer](#) on Tue, 15 Jul 2008 19:23:49 GMT
[View Forum Message](#) <> [Reply to Message](#)

Gurus-

I've battling with this for a week now, and I think it needs a fresh pair of eyes. I am working on code that pops up a Modal Widget, does some manipulation, and pass some information back to the user. For reference, I am using D Fanning's general approach to this problem. I've put some code at the bottom.

Here's how the program works:

The main routine is "display_fcm_data_GUI_filter_data". This calls the dialog routine, where I put the widgets in a base, realize the widget, define a structure of pointers (called filter_dialog_info), and put a this into the widget. I then register the widget (and it's event handler) with Xmanager. Everything works in the event handler until I try to get out of the widget. When I reach the END statement of the event handler, I can still access filter_dialog_info. If I try .stepover from this point, I enter into the xmanager.pro routine. At this point I can no longer access any of filter_dialog_info, but the pointers are still in memory. What in the world am I doing wrong?

Thanks, PMB

CODE:::::::::::
PRO display_fcm_data_GUI_filter_dialog_events, event
;;;;;;event handler
but_make = WIDGET_INFO(event.top, FIND_BY_UNAME='but_make_plots')
but_freq1 = WIDGET_INFO(event.top, FIND_BY_UNAME='but_freq1')

CASE event.id OF
but_make : BEGIN
 ;do some stuff with filter_dialog_info - this works
 END
but_freq1 : BEGIN
 WIDGET_CONTROL, filter_dialog,
 GET_UVALUE=filter_dialog_info, /NO_COPY
 *filter_dialog_info.filter_data =
 *filter_dialog_info.filter_data1 ;ASSIGN A NEW VALUE
 WIDGET_CONTROL, event.top, /DESTROY
 END
 ELSE: ;do nothing
ENDCASE

END;;;;;;,;, CAN

STILL ACCESS filter_dialog_info

FUNCTION display_fcm_data_GUI_filter_dialog, times, data, Group_Leader
= group_leader

;;;;;;;;;;;;;;pops up the widget

IF N_ELEMENTS(group_leader) EQ 0 THEN \$
 tlb = WIDGET_BASE(Title='Filter Stuff', /MODAL,
 Group_leader=group_leader, UNAME='filter_dialog') \$
ELSE tlb = WIDGET_BASE(Title='Filter Stuff', ROW=4,
 UNAME='filter_dialog')

;other widgets in tlb go here

make_plotsID = WIDGET_BUTTON(tlb, UNAME='but_make_plots', VALUE='Make
Plots')

keep_butID = WIDGET_BASE(tlb)
 but_orig = WIDGET_BUTTON(keep_butID, UNAME='but_freq1',
 VALUE='Keep Filtered Data', Sensitive=0)

WIDGET_CONTROL, tlb, /realize

filter_dialog_info = {filter_times:PTR_NEW(times), \$
 filter_data:PTR_NEW(data), \$
 filter_data1:PTR_NEW(data), \$
 filter_data2:PTR_NEW(data), \$
 filter_data3:PTR_NEW(data)}

WIDGET_CONTROL, tlb, SET_UVALUE=filter_dialog_info, /No_copy

XMANAGER, 'display_fcm_data_GUI_filter_dialog', tlb, \$
event_handler='display_fcm_data_GUI_filter_dialog_events'

filtered_data = *filter_dialog_info.filter_data ;IDL doesn't know
what filter_dialog_info is!!!!

PTR_FREE, filter_dialog_info.filter_times
PTR_FREE, filter_dialog_info.filter_data
PTR_FREE, filter_dialog_info.filter_data1
PTR_FREE, filter_dialog_info.filter_data2
PTR_FREE, filter_dialog_info.filter_data3

RETURN, filtered_data

END;;;;;;;;;;;;

PRO display_fcm_data_GUI_filter_data, event

;;;;;;;;;;;this calls the pop up widget

```
out = display_fcm_data_GUI_filter_dialog( *info.times, *info.data,  
group_leader = event.top)
```

```
;do something with out
```

```
END;.....
```
