Subject: Re: Modal dialog - returning values
Posted by JD Smith on Mon, 21 Mar 2005 16:28:01 GMT
View Forum Message <> Reply to Message

On Mon, 21 Mar 2005 07:03:51 -0700, David Fanning wrote:

> Dave Roberts writes:
>
>> Have read lots of solutions to this problem but none seem to apply? Im
>> new so forgive me.
>> Here is my code
>>
>> pro Create_Images, Event
>> Image_Create,GROUP_LEADER=Event.top
>> end
>>
>> (Image_Create brings up a modal dialog box that was created using the
>> automatic IDL gui generation i.e. I put the buttons on and it wrote the
>> Image_Create.pro and Image_Create_eventcb.pro for me)
>>
>> Ive read I can store a pointer in the top level but have already got
>> something stored there - the code below shows what I put there
>>
>> imagestore=CREATE_STRUCT(imagestore,'invertedsmallarray',inv ertedsmallarray)
>> wDraw=WIDGET_INFO(Event.top,FIND_BY_UNAME='Draw')
>> WIDGET_CONTROL,wDraw,SET_UVALUE=imagestore
>>
>> Therefore if I put a pointer in the top level base will I loose
>> imagestore?
>
> No, because imagestore is not stored in the the top-level base. It is
> stored in the user value of the draw widget.
>
>> I have a modal dialog box that just returns a float so i used a COMMOM
>> block (but believe this is not advisable). I cant do this now beacuse
>> I want to return an array (structure) generated in the modal dialog.
>>
>> Also should I be storing stuff as above - it was the only way I found I
>> could pass information around my gui program (i.e. between procedures)
>
> No, common blocks are NOT the preferred solution in widget programs,
> because by using them you automatically restrict yourself to just one
> copy of the program running at any one time.

Actually, you *could* get multiple running copies using a single
common block. You'd just have to implement a matched store of
multiple state structures in a single common block --- say a pointer
to a list of pointers to state structures, indexed by the TLB widget

ID or name (or either). But this is sufficiently similar to the work that the object system does for you behind the scenes, that you may as well just use object widgets from the get go. However, the common block approach might be a clever, if hard to maintain, way to provide notification-free inter-application communication, if *parts* of the state structure were shared, and the rest were multiplexed in the way described.

JD