
Subject: Re: widget_problem

Posted by [d.poreh](#) on Wed, 23 Jul 2008 14:01:08 GMT

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

On Jul 22, 5:52 pm, d.po...@gmail.com wrote:

> On Jul 17, 8:32 pm, Paul van Delst <Paul.vanDe...@noaa.gov> wrote:

>

>

>

>> Justus Skorps wrote:

>>>> Justus

>>>> stil can not fix it. i have Liam E.Gumley's book but

>>>> Cheers

>>>> Dave

>

>>> after u load your arrays (lets call them A) store them with

>

>>> widget_control, event.top, set_uvalue=A, /nocopy

>

>>> You have to change 'event.top' that it fits your program...

>>> In your second button you can now load the arrays with

>

>>> widget_control, event.top, get_uvalue=A, /nocopy

>

>> but don't forget to put them back into your uvalue when you're done!

>

>>> It is useful to

>

>>> -store the data in the main widget

>>> -use a structure to store every data you want

>

>> I also tend to use procedures to get the Info state:

>

>> ; Routine to get the Info state

>> PRO GetState, ID, Info, No_Copy = No_Copy

>> ; -- Get pointer

>> WIDGET_CONTROL, ID, GET_UVALUE = InfoPtr

>> IF (PTR_VALID(InfoPtr) EQ 0) THEN \$

>> MESSAGE, 'State Information pointer is invalid'

>

>> ; -- Get state information structure

>> IF (N_ELEMENTS(*InfoPtr) EQ 0) THEN \$

>> MESSAGE, 'State information structure is undefined'

>> IF (KEYWORD_SET(No_Copy)) THEN BEGIN

>> Info = TEMPORARY(*InfoPtr)

>> ENDIF ELSE BEGIN

>> Info = *InfoPtr

>> ENDELSE

```

>> IF ( Info.Debug EQ 1 ) THEN PRINT, 'GetState'
>> END
>
>> and to set the info state
>
>> ; Routine to set the Info state
>> PRO SetState, ID, Info, No_Copy = No_Copy
>> ; -- Get pointer
>> WIDGET_CONTROL, ID, GET_UVALUE = InfoPtr
>> IF ( PTR_VALID( InfoPtr ) EQ 0 ) THEN $
>> MESSAGE, 'State information pointer is invalid'
>
>> ; -- Set state information structure
>> IF ( N_ELEMENTS( Info ) EQ 0 ) THEN $
>> MESSAGE, 'State information structure is undefined'
>> IF ( KEYWORD_SET( No_Copy ) ) THEN BEGIN
>> *InfoPtr = TEMPORARY( Info )
>> ENDIF ELSE BEGIN
>> *InfoPtr = Info
>> ENDELSE
>> IF ( (*InfoPtr).Debug EQ 1 ) THEN PRINT, 'SetState'
>> END
>
>> My widget event handlers then do something like:
>
>> FUNCTION ComponentTest_LogLin_Event, Event
>> ; -- Get main info state
>> GetState, Event.Top, Info
>
>> ; -- Print debug statement if required
>> IF ( Info.Debug EQ 1 ) THEN PRINT, 'ComponentTest_LogLin_Event'
>> ; -- Set the selected variable number index
>> Info.LogLin_Index = Event.Value
>
>> ; -- Save info state
>> SetState, Event.Top, Info
>
>> ; -- Display the result
>> ComponentTest_Display, Event.Top
>> RETURN, 0
>> END
>
>> Note how I call GetState and then SetState. Because I tend not to use /no_copy, it's
>> really only an issue when I update a component of the info state (like in my example
>> above). But, if you *do* use /no_copy, then I think you have to call SetState again to
>> replace the info pointer.
>
>> cheers,

```

>
>> paulv
>
> Thanks Justus and Paulv
> You help me very much. And it was very useful.
> Cheers
> Dave

Hi Justus

I encounter a now problem:

Say I have set and get 2 arrays by this method:

widget_control, event.top, set_uvalue=A, /nocopy

...

widget_control, event.top, get_uvalue=A, /nocopy

and from another button:

widget_control, event.top, set_uvalue=B, /nocopy

...

widget_control, event.top, get_uvalue=B, /nocopy

Now I want to get both A&B in another button like this:

widget_control, event.top, get_uvalue=A, /nocopy

widget_control, event.top, get_uvalue=B, /nocopy

But widget just accepts one array. How I can take this two arrays simultaneously in another button?

Any help?

Cheers

Dave
