
Subject: Re: Widget question

Posted by [Liam Gumley](#) on Fri, 02 Jul 1999 07:00:00 GMT

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

Harald Frey wrote:

>
> Sorry if I missed that point somewhere in the very interesting
> discussion about "Passing info and destroying widgets". I followed the
> suggestion by Liam Gumley to create my widget structure but now I have a
> problem.
>
> I have 3 levels in my widget structure, main, parent and child. What I
> do is:
>
> - Create main widget
> - Create info structure
> - store info structure in the main level base
> - start Xmanager with event handler
>
> A certain button on the main widget calls the parent process where I
> - get the info structure
> - create the parent widget
> - create a pointer to the info structure
> - store pointer in parent base
> - start second Xmanager with second event handler
>
> Now a button on the parent widget calls a process on the child level
> where I
> - get the pointer
> - get the info structure via pointer
> - manipulate the info
>
> So far everything is fine but now I want to store the manipulated info
> from the child level back into the main widget and destroy the parent
> widget. How do I do this?

I use the following method in event handlers:

PRO MYPROG_EVENT, EVENT

;- Get the info pointer

widget_control, event.top, get_uvalue=info_ptr

;- Get the info structure

info = *info_ptr

;- Manage widget events, and change info structure...

;- Save the info structure

*info_ptr = info

END

If I create child widgets which are spawned from the top level widget, then I pass the ID of the top level widget (event.top) to the child, e.g.

PRO MYPROG_CHILD, TOP

;- Get the info structure

widget_control, top, get_uvalue=info_ptr

;- Get the info structure

info = *info_ptr

;- Modify items in the info structure...

;- Save the info structure

*info_ptr = info

As long as you remember to get/save the info structure, you won't have any problems. This method is a holdover from the days of handles, where managing info structures in this way was mandatory.

Cheers,
Liam.

--

Liam E. Gumley
Space Science and Engineering Center, UW-Madison
<http://cimss.ssec.wisc.edu/~gumley>
