
Subject: Re: Passing info and destroying widgets...
Posted by [mallors](#) on Mon, 21 Jun 1999 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

In article <376E5BF5.622A1540@ssec.wisc.edu>,
Liam Gumley <Liam.Gumley@ssec.wisc.edu> writes:
> Dirk Fabian wrote:
>> Here's something i can't figure out... I'm following the advice of DF and
>> communicating between my widgets with info structures. All is well, but now i
>> want to pass the info structure from the event handler back to the parent widget
>> with
>>
>> WIDGET_CONTROL, event.top, SET_UVALUE=lines, /NO_COPY
>>
>> and then destroy the widget. But you can't do this, because WIDGET_CONTROL (i
>> think) dereferences event.top so that
>>
>> WIDGET_CONTROL, event.top, /DESTROY
>>
>> fails since it doesn't know where to look. (you can't even put in a dummy to hold
>> the event.top number, the widget itself is gone from that id)
>>
>> Unfortunately, you can't /DESTROY the top widget first and expect to set it's
>> UVALUE later, either. So what do i do here? I tried putting a flag in my
>> info structure to trigger the base widget destruction back in the widget
>> definition level (not in the event handler), but i can't figure out when the
>> program would be able to look at that newly inserted flag.
>
> Dirk,
>
> If I understand your question correctly, you are trying to figure out
> how to pass a value from a widget event manager back to the calling
> program (i.e. the one that invoked XMANAGER) after the top level widget
> has been destroyed. The answer in IDL5 is pointers.
>
> In a widget which does not need to return any information, you store the
> info structure in the top level base user value, e.g.
>
> ;- Create widgets...
>
> ;- Create info structure
> info = {name:'test', value:indgen(10)}
>
> ;- Store info structure in top level base
> widget_control, tlb, set_uvalue=info
>
> ;- Start XMANAGER...
>

```

> However when the event manager must pass back information to the program
> which invoked XMANAGER, use a pointer to store the info structure, and
> store the *pointer* in the top level base user value, e.g.
>
> ;- Create widgets...
>
> ;- Create info structure and store via pointer
> info = {name:'test', value:indgen(10)}
> ptr = ptr_new(/allocate_heap)
> *ptr = info
>
> ;- Store pointer in top level base
> widget_control, tlb, set_uvalue=ptr
>
> ;- Start XMANAGER...
>
> and then in the event manager, get the contents of the info structure
> from the pointer, e.g.
>
> ;- Get pointer
> widget_control, event.top, get_uvalue=ptr
>
> ;- Get info structure
> info = *ptr
>
> When the top level base is destroyed, the *pointer* still exists, thus
> in the calling program you can retrieve it's value.

```

I believe you must use XMANAGER in blocking mode for this technique to work.

Regards,

-bob

--

```

~~~~~
Robert S. Mallozzi                                256-544-0887
                                           Mail Code SD 50
Work: http://gammaray.msfc.nasa.gov/   Marshall Space Flight Center
Play: http://cspar.uah.edu/~mallozzir/   Huntsville, AL 35812
~~~~~

```
