

---

Subject: Re: Child widget group leader of its own TLB?

Posted by [nhbknich](#) on Fri, 04 Jul 1997 07:00:00 GMT

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

---

David Foster (foster@bial1.ucsd.edu) wrote:

: Michael -

: I think you may be confused about the concept of GROUP\_LEADER and  
: it's effect on widget destruction. When you create a widget heirarchy,  
: any and all widgets that are "below" the TLB will be destroyed  
: automatically when you destroy the TLB. By "below" I mean that you can  
: trace back from a widget up through the heirarchy and get to the  
: TLB. So there is no need for a cascade as you mention above.

In the actual case, there is some need. The widget application passes  
the Id of a (child) text widget outside, rather than its top level Id.

The reason is, that I want to be able to manipulate this text widget from  
outside in more ways than possible by FUNC\_GET\_VALUE and PRO\_SET\_VALUE.

Things like dis- or enabling editing, dis- or enabling events created by  
the text widget or establishing a separate event handler.

But one possible manipulation is also destroying the text widget. In  
this case, there would only remain a kind of ruin of a mini window frame  
containing meaningless buttons and labels.

I intended the whole application to quit in such a situation as if "Done"  
had been pressed.

: The GROUP\_LEADER keyword is intended for situations where you have  
: one widget program up and running, and then you wish to call another  
: widget (a program or a modal popup) from within that first program.  
: In the second program you specify the TLB of the first widget as the  
: GROUP\_LEADER. When you do this, the second widget will be destroyed  
: if the first is destroyed. You can have a whole series of programs  
: up, and have them all destroyed at once. This is also useful when  
: a program has several (or many!) widgets that all must  
: go when the parent widget is destroyed.

I know, this is the most common use of widget groups. But is it restricted  
to this use? Neither the online help nor the printed manuals state something  
like that. Furthermore, wouldn't this imply, that only top level bases can  
reasonably be group leaders as well as \_group members\_? If so, why do all  
widget creation functions, including WIDGET\_TEXT, WIDGET\_BUTTON, WIDGET\_LABEL  
etc., which can never create top level widgets, accept the GROUP\_LEADER  
keyword?

: Hope this helps.

Yes it does. Maybe my goal is a bit too exotic. But I'm not yet really  
convinced :-)

