Subject: Re: killing non-modal widgets
Posted by btt on Fri, 09 Jun 2006 16:35:00 GMT
View Forum Message <> Reply to Message

David Fanning wrote:
> Ben Tupper writes:
>
>> Yes, I just double checked that each of the widgets have TLBs with the
>> parent widget argument but *not* the GROUP_LEADER keyword set to valid
>> widgets.
>
> OK, are these KILL_NOTIFY callbacks or CLEANUP callbacks?
> It *used* to be the case that you couldn't use KILL_NOTIFY
> with TLBs that were directly managed by XMANAGER. I see
> RSI code that violates this concept all the time, but I've
> always stuck to it and never had any problems. The intermittent
> nature of the problem suggests to me a problem with the
> various event loops that get set up to manage these autonomous
> widgets. It has never been clear to me in which order callbacks
> are executed, but I know it is different on different operating
> systems. I wonder if switching to CLEANUP callbacks would help.
>

Ah! That's the lead I am looking for. I'll switch it all over to
CLEANUP specified for XMANAGER rather than the KILL_NOTIFIES.

The documentation is pretty clear that the CLEANUP callback routine
supercedes the KILL_NOTIFY routine.

From XMANAGER online help...

CLEANUP
Set this keyword to a string that contains the name of the routine to be
called when the widget program dies. If this keyword is not specified,
the routine (if any) specified for the program's top-level base by the
KILL_NOTIFY keyword to WIDGET_BASE or WIDGET_CONTROL is used.
The routine specified by CLEANUP becomes the KILL_NOTIFY routine for the
widget application, overriding any cleanup routines that have been set
previously via the KILL_NOTIFY keyword to WIDGET_BASE or WIDGET_CONTROL.

Thanks for the help. I post something if that works.

Cheers,
Ben