
Subject: Re: base widgets growing uncontrollably.... ?
Posted by [Paul van Delst](#) on Fri, 27 Jul 2001 14:48:32 GMT
[View Forum Message](#) <> [Reply to Message](#)

Marc Schellens wrote:

>
>> WOOHOO! That did it! Inserting the UPDATE in the following code fixed the problem.
>>
>> You're a star! This problem was annoying the crap outta me to the point of tossing and
>> turning at night.
>>
>> Onya!
>>
>> paulv
>
> Thanks,
> I have to pass the compliment to Stein Vidar Hagfors Haugan,
> actually I think I got it long time ago from Davids page Alex
> mentioned below (but didn't remeber from where I had it).
>
> Anyway, good that it helped you!
>
> But was astonishes me is, that you use the update BEFORE
> the base is realized.
> I didn't know that update makes a difference then.
>
> What I do not understand in your program is,
> why you first unmap the base during creation
> and then map it when realizing?
> Until a base is realized (= brought to screen) mapping didn't matter
> (at least for the tlb).

I unmap the top level base right at the start so I don't see the final GUI being created widget by widget in front of me - it was interesting to watch but didn't look very nice. Unmapping then mapping at the end makes the complete widget just pop up when it has finished being created/assembled.

> OR: Might it be that in the real program the realizing is done before
> and
> so the realizing there is unnecessary (which would also explain the
> astonishing effect of update to non realized widgets)?

Each separate compound widget function realises *and* registers the widget using XMANAGER. I did this so that when I killed the top level base, the cleanup routines for all the child compound widgets would be called. If I don't do a

XMANAGER, id, 'widget_name', /JUST_REG, CLEANUP = 'widget_name_cleanup'

in the compound widget creation functions, then I am left with a bunch of dangling pointers - that used to be in the compound widget's top-level base user value - hanging about afterwards. The XMANAGER call in *each* compound widget creation function was the only way I could get stuff cleaned up in a heirarchial-type of way. I want to keep the information structure for each compound widget separate in it's own top-level base user value (rather than shoving everything in the god-base uvalue) as I envisage these routines to be usable on their own, not just as a component of a container GUI.

If anyone has a better method of doing this please let me know. I couldn't figure out how to make the child widget cleanup routines (for the compound widgets) "visible" unless I put in separate XMANAGER calls.

Maybe I should just stick everything in the god-base id..... I gotta sit down and think about this a bit more....

paulv

--

Paul van Delst A little learning is a dangerous thing;
CIMSS @ NOAA/NCEP Drink deep, or taste not the Pierian spring;
Ph: (301)763-8000 x7274 There shallow draughts intoxicate the brain,
Fax:(301)763-8545 And drinking largely sobers us again.
 Alexander Pope.
