Folks,

OK, I am trying to follow the excellent advice I gave Wayne earlier in the week about how to clean up widget-objects, but I find nothing but trouble in doing so.

I have a simple system: a top-level base widget and a draw widget, both written as objects. When the widget is created inside the object, a KILL_NOTIFY callback is registered for both objects. In the callback, I destroy the object (as I explained to Wayne), which puts me into the CLEANUP routine for the object. In the CLEANUP routine, I look to see if the objects have valid widgets associated with them, and if they do I destroy the widgets.

But...

When I put the widgets on the display and kill them with the mouse, I am immediately dumped into the draw widget Kill_Notify procedure. So, I kill that object, which puts me into the Cleanup routine for the draw object. The draw widget is still hanging in there, not dead yet, so the Cleanup routine tries to kill it again, which puts me back in the Kill_Notify procedure, where I get the dreaded error:

  % Internal error: attempt to delete nonexistent widget
    from hash table.

If I try to kill the draw widget the first time it goes into the Kill_Notify procedure (Widget_Control, id, /Destroy), the same thing happens. I can't kill the damn thing! At least it doesn't seem to die until I exit the first Kill_Notify procedure. Logically, I can understand this as a series of calls deeper and deeper into the call stack. But, come on guys, I need to kill this damn widget! Is there some way to choke it to death in the Kill_Notify method?

This is the way I wrote the Catalyst Library, so I'm pretty sure this is NOT the way IDL worked in IDL 7. And, oddly, the Catalyst Library still appears to work in IDL 8, which is also a mystery to me.

Any ideas on what might be going on here?

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thue. ("Perhaps thou speakest truth.")