Subject: Killing Widget Objects
Posted by David Fanning on Fri, 21 Mar 2014 17:48:18 GMT
View Forum Message <> Reply to Message

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.")

Subject: Re: Killing Widget Objects
Posted by Matthew Argall on Fri, 21 Mar 2014 18:06:00 GMT

View Forum Message <> Reply to Message

I did the exact same thing a few weeks ago. I got around it by only destroying the top level base object or widget.

In the draw widget object, the only cleanup I have is
    if obj_valid(self._oTLB)   then self._oTLB -> Destroy

In the cleanup method for the top level base object I have
    if widget_info(self._ID, /VALID_ID) then widget_control, self._id, /DESTROY

So, now the top level base object and its widget are destroyed. When the widget is destroyed, XManager kills all of its children. All of the children have a Kill_Notify procedure/method that destroy their associated object.

For a while, I tried to kill the draw widget inside the draw object's cleanup method as well. This caused XManager to try to kill the draw widget twice and gave me the same error you are getting.

Subject: Re: Killing Widget Objects
Posted by David Fanning on Fri, 21 Mar 2014 18:15:26 GMT

View Forum Message <> Reply to Message

Matthew Argall writes:

> I did the exact same thing a few weeks ago. I got around it by only destroying the top level base object or widget.
>
> In the draw widget object, the only cleanup I have is
>   if obj_valid(self._oTLB)   then self._oTLB -> Destroy
>
> In the cleanup method for the top level base object I have
>   if widget_info(self._ID, /VALID_ID) then widget_control, self._id, /DESTROY
>
> So, now the top level base object and its widget are destroyed. When the widget is destroyed, XManager kills all of its children. All of the children have a Kill_Notify procedure/method that destroy their associated object.

>
> For a while, I tried to kill the draw widget inside the draw object's cleanup method as well. This caused XManager to try to kill the draw widget twice and gave me the same error you are getting.

Well, the VERY same program works with Catalyst objects. Since I copied
Catalyst objects (nearly directly, I thought) when I made these new
objects, I'm really at a loss to explain this. I'm in the process of
looking more closely at things now.

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.")

Subject: Re: Killing Widget Objects
Posted by David Fanning on Fri, 21 Mar 2014 19:11:54 GMT
View Forum Message <> Reply to Message

David Fanning writes:

> Well, the VERY same program works with Catalyst objects. Since I copied
> Catalyst objects (nearly directly, I thought) when I made these new
> objects, I'm really at a loss to explain this. I'm in the process of
> looking more closely at things now.

OK, weird! The problem is suddenly GONE! I didn't DO anything, expect
put print statements in the code so I could see what was happening.
Sheesh!

```
IDL> .go
ID in gliTLBWidget:           1
ID in gliDrawWidget INIT:           2
% Compiled module: GLIKILLNOTIFY.
ID in gliKillNotify:           2
In gliDrawWidget CLEANUP
ID in gliKillNotify:           1
In gliTLBWidget CLEANUP

IDL> .go
ID in TOPLEVELBASE INIT method:           3
ID in DrawWidget INIT method:           4
ID in CatKillNotify:           4
In DrawWidget CLEANUP method:
```

ID in CatKillNotify: 3
In TopLevelBase CLEANUP method:

Whoa! I've been fooling around with this all week. Suddenly the problem is gone!? This is really too weird, even for computers. :-(

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.")