Subject: Re: Common block conumdrum
Posted by J.D. Smith on Wed, 05 Mar 1997 08:00:00 GMT
View Forum Message <> Reply to Message

Struan Gray wrote:

> This works but has it's own problems. I often need to have the
> state information when the widget is killed so that I can save
> parameters and variables to disk or pass a response to another widget.
> David's technique works well if you can be sure that the user will
> only ever kill the widget with 'close'/'quit'/'cancel' buttons, but if
> the widget can be killed another way (with the xmanager tool or via a
> close box provided by the operating system's window manager) I find
> that by the time my widget knows it is being killed only the top level
> base remains and my state information is lost.

XMANAGER's CLEANUP callback mechanism gives you control of your dying widget after almost all of it has been killed. This is problematic, since you often can't then get to your state info.

A technique I often use to prevent this behaviour is the KILL_NOTIFY mechanism. Its use is strongly discouraged in various IDL references, but the thrust of the warning is "don't assign a kill_notify procedure for a top level base". As long I avoided this, I haven't had problems. The key recognition is that the callback procedure specified by KILL_NOTIFY only has access to the user value of that widget for which it was specified -- no other widgets are gauranteed to be alive, so widget_control doesn't let you even try to access them. The specified widget can't be the top level base, since XMANAGER wouldn't like this. What is needed is a widget that contains the state info but is *not* the top level base. I therefore use the the base's first child's uvalue to store my state structure (the de facto for compound widgets), and assign the callback procedure to the *child* widget. E.g. I might say:

```
widget_control,widget_info(base,/CHILD),SET_UVALUE=state,/NO _COPY, $
  KILL_NOTIFY='WIDGET_KILL_PROCEDURE'
```

The strength of this technique lies in the fact that when the callback procedure is called, the state is still defined, since it's in the user value of the widget... so you can free handles, save info, etc. This works whether the user exits with the DONE button, or otherwise. In fact, it's also convenient for application development, since even a crashed widget (after you retall and xmanager) will call its kill_notify, freeing handles and preventing memory leakage.

Perhaps there is substance to the warnings, and problems with using XMANAGER jointly with KILL_NOTIFY do exist, but I haven't experienced them, as long as I stayed away from the TLB.

JD
