Subject: Re: Q: Clearing Widget Events
Posted by steinhh on Thu, 15 Aug 1996 07:00:00 GMT
View Forum Message <> Reply to Message

In article <4utfr2$av4@news1.ucsd.edu>, David Foster <foster@bial1.ucsd.edu> writes:
|> I've tried using
|>
|> WIDGET_CONTROL, /RESET
|>
|> to destroy all active widgets, but this doesn't seem to clear the
|> error condition, even if "retall" and "xmanager" are issued afterwards.
|> The next widget that is created is realized, but then you find yourself
|> at the IDL> prompt! [Any suggestions here?]
|>

What usually works fairly well in this situation is simply

  IDL> xmanager

Since you have a fresh, newly registered widget that is initialized
for event processing (but the first xmanager was a fall-through one),
you seldom have problems with your widget being in an inconsistent
state.

I found another thing that works is to make your new widget (after
the crash, retall, reset) MODAL, this makes the widget program's
own xmanager call work, even though it believes that there's
another call to xmanager (the one it tries to fall back on if
non-MODAL operation was requested).

The problem with this is if you're developing a suite of
widget applications that really shouldn't be modal.

I've observed that if you do the following:

1 Start a widget
2 Crash it
3 retall, widget_control,/reset, etc. etc.
4 Restart a widget, ending up at the IDL> prompt
5 Type XMANAGER to process events for the new widget
6 Press the "done" button (or whatever) on the new widget,
   for a graceful death.

then

7 the next restart of the widget has no problems.

So if we can make a cleanup file that does steps 3-6

automatically, we have a cleanup file that takes care of this type of problem.

The difficult thing is step 6, but one could probably do this automatically through the use of a TIMER event.

Haven't done this yet, but it might be worth a try.

Stein Vidar
