Subject: Re: Interrupting widget applications
Posted by hegde on Thu, 23 Jan 1997 08:00:00 GMT
View Forum Message <> Reply to Message

Thanks for your suggestion. Although it works for a simple for loop, this type of implementation goes out of hand for multiple loops or recursive operations. And if it is required to monitor within a time consuming loop, I have to find another way.

My need is to have an interrupt button in the main application window and it interrupt anything going on in other windows (display windows, printer gizmos etc.,).

So far the most convenient method I have found out is to monitor the event queue with WIDGET_EVENT(interrupt button) wherever the interrupt is desired and proceed accordingly. Any suggestion is appreciated.

Thanks,

-M. Hegde
hegde@neptune.gsfc.nasa.gov

William Thompson (thompson@orpheus.nascom.nasa.gov) wrote:
: What you need to do is to break your application down into discrete events, and
: then use TIMER events to run the events in sequence. For example, if your
: program uses a FOR loop, then you could instead make each run through the loop
: a discrete event, and store the loop counter between events.
:
: For example, you might define your main base with a line like
:
: MAIN_BASE = WIDGET_BASE(TITLE='CDS FITS Generation Software', $
: /FRAME, /COLUMN, UVALUE='Timer')
:
: and then in the event handler you would have code that would read
:
: WIDGET_CONTROL, EV.ID, GET_UVALUE=VALUE
: IF VALUE EQ 'Timer' THEN BEGIN
: ... do one piece of the complete operation ...
:
: It's important to restart the timer after every timed event, to keep it going.
: What I like to do is to keep a parameter called RUNNING which can be either 0
: or 1. A start button generates an event which does some initial preparation
: (e.g. setting the loop counter to 0), and sets RUNNING=1. Then at the bottom
: of the event handler routine I put a line like
:
: IF RUNNING THEN WIDGET_CONTROL, MAIN_BASE, TIMER=0.1
:
: A stop button simply sets RUNNING=0.

:
: Bill

--
test