Subject: Re: Interrupting widget applications
Posted by davidf on Sun, 26 Jan 1997 08:00:00 GMT
View Forum Message <> Reply to Message

Sam Haimov <haimov@uwyo.edu> quotes M. Hegde with respect to interrupting loops in widget programs:

>> 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.

He goes on to write:

> I have similar experience.
>
> My application uses widgets for a real-time radar display. It can show several different
> images on one or two windows and includes various widget controled features. The main
> reason I am using widget_event to handle widgets is the fact that my main loop is controled
> by the radar data acquisition system (non-IDL program) and I want the interrupts to be
> based on the DAQ behaviour rather than on time intervals by TIMER.

Here, in my opinion, is just about the *only* justification for using WIDGET_EVENT to manage IDL events rather than XMANAGER. Sam is exactly right. If the locus of control is in an external program, then I think you want to use WIDGET_EVENT to get IDL widget events and process them appropriately.

But suppose the locus of control was the IDL widget program. How could you get the program to respond to real-time events that were being monitored by an external program?

One possible solution would be to use TIMER events. I typically set timers on widgets that don't ordinarily receive events. Usually I use a sub-base, whose only other purpose is to organize my widget layout. I attach an event handler to that base to handle the timer event. It looks like this:

```
subbase = WIDGET_BASE(tlb, EVENT_PRO='Timer_Event', COLUMN=2)
```

When I am ready to check my external program, I want to get into the TIMER_EVENT event handler, so I set the timer to go off immediately, like this:

```
WIDGET_EVENT, subbase, TIMER=0.0
```

Now I am inside my TIMER_EVENT event handler. Here I can do whatever it is I need to do. For example, I can check my external program to see if any new data has come in from my instruments. If it has, I can plot it.

When I am finished doing whatever it is I need to do, I am set to exit my event handler. If I want to come back periodically I need to set the next timer event before I exit. Suppose I wanted to check for new data every 1.5 seconds, then I would probably have code like this at the bottom of my TIMER_EVENT event handler:

```
IF info.stop NE 1 THEN WIDGET_CONTROL, event.id, TIMER=1.5
```

The info.stop variable is a stop flag that is usually set by a STOP or INTERRRUPT or CANCEL button of some sort. In 1.5 seconds, I get back into this event handler for the next round of doing whatever it is I do. But meanwhile, I can be processing whatever other events are being generated by my widget program, in the order in which they are being generated. This will include EXIT buttons, READ DATA buttons, etc.

If you want to see a good example of a TIMER event in action, look at the program XMOVIE on my web page. This program shows you how to do an animation correctly in a widget program. It is possible to interrupt the animation because the STOP and START buttons just cue up the TIMER events.

Hope this gives people some ideas.

David

David Fanning, Ph.D.
Fanning Software Consulting
2642 Bradbury Court, Fort Collins, CO 80521
Phone: 970-221-0438 Fax: 970-221-4762
E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com
