
Subject: Re: Running time display
Posted by [davidf](#) on Fri, 21 Feb 1997 08:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

David Gunter writes:

> I would like to have an elapsed-time display in a widget application but I
> can't figure out how to update the display. Placing the update-routine in the
> event handler only works as long as events are being generated. Has anyone
> else attempted this before?

Oh, many's the time, David. :-)

This used to be the compound widget building exercise in my first IDL widget programming classes. I expect many of the people on this newsgroup still have this program hanging around in a subdirectory somewhere. :-)

Here is mine, dusted off, looking a little worse for wear, but still servicable. The secret is the TIMER. In this case I cause a timer event to be generated by the label widget (which normally **never** generates events). I catch those events in a separate event handler.

It's a compound widget, so you can include it in your programs just the way you would a button, or label, etc. There is a tiny test program at the end of the code, so you can run it and see what it does.

You will want to fix this up and make it more sophisticated, but this will get you going. Try making the timer faster or slower. This is about right on my machine. It may be too slow on yours. If the label flashes, try making the clock a text widget instead of a label widget.

Enjoy,

David

PRO CW_TIMER_UPDATE, event

 ; This event handler handles the timer events set on the clock.

Widget_Control, event.id, Get_UValue=stateHolder
Widget_Control, stateHolder, Get_UValue=state

 ; Calculate the elapsed time.

```

elapsedTime = SysTime(1) - state.startTime

; Update the clock.

strTime = String(elapsedTime, Format='(F6.1)')
Widget_Control, state.clock, Set_Value=strTime

; If the stop flag is clear, set another timer event.

IF state.stopper NE 1 THEN Widget_Control, state.clock, Timer=0.2
END ; ***** **

PRO CW_TIMER_BUTTONS, event

; This event handler handles the START and STOP buttons.

child = Widget_Info(event.handler, /Child)
Widget_Control, child, Get_UValue=state

; Which button event?

Widget_Control, event.id, Get_UValue=thisButton

CASE thisButton OF

'START': BEGIN
    state.stopper = 0 ; Clear stop flag.
    state.startTime = SysTime(1) ; Starting time.
    strValue = String(0.0, Format='(F6.1)')

    ; Get the first timer event going.

    Widget_Control, state.clock, Timer=0.02, Set_Value=strValue
END

'STOP': BEGIN
    state.stopper = 1 ; Set stop flag.
END

ENDCASE
Widget_Control, child, Set_UValue=state
END ; ***** **

FUNCTION CW_TIMER, parent, UValue=uvalue
IF N_ELEMENTS(uvalue) EQ 0 THEN uvalue=0

```

; Define widgets.

```
tlb = Widget_Base(parent, Column=1, UValue=uvalue, $
    Event_Pro='CW_TIMER_BUTTONS', Frame=1)
holderbase = Widget_Base(tlb, Column=1) ; To hold "state".
starter = Widget_Button(holderbase, Value='Start Clock', UValue='START')
quitter = Widget_Button(holderbase, Value='Stop Clock', UValue='STOP')
strValue = String(0.0, Format="(F6.1)")
clock = Widget_Label(holderbase, Value=strValue, $
    Align_Center=1, Event_Pro='CW_TIMER_UPDATE', UValue=holderbase)
state = {stopper:1, clock:clock, tlb:tlb, startTime:-1D}
Widget_Control, holderbase, Set_UValue=state
RETURN, tlb
END ; ***** **
```

```
PRO TEST_EVENT, event
WIDGET_CONTROL, event.top, /Destroy
END
```

```
PRO TEST
tlb = Widget_Base(/Column)
clock = CW_Timer(tlb)
quitter = Widget_Button(tlb, Value='Quit Program')
Widget_Control, tlb, /Realize
XManager, 'test', tlb
END
```

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>
