
Subject: Re: Newbie questin regarding Widget timers
Posted by [David Fanning](#) on Fri, 29 Aug 2008 13:31:41 GMT
[View Forum Message](#) <> [Reply to Message](#)

Mike Potter writes:

> The reason the WIDGET timer stuff seemed superfluous was:
>
> 1) the line:
>
> oWindow->SetEventMask, TIMER_EVENTS=3D1
>
> appears in the branch of the case statement that handles play_button
> events in the event procedure
>
>
> 2) the line:
> oWindow->AddWindowEventObserver, oObserver
>
> (oObserver is an instance of the custom-defined "timer_observer"
> class) appears in the main "animation_doc" procedure,
> and
>
> 3) there is a timer_observer::OnTimer method defined with oWindow as
> it's only argument containing code that sets the ACTIVE_POSITION
> within the olmages IDLgrMODEL instance, then calls oWindow->Draw.
>
> It's all part of the standard IDL distribution, so if you have
> that set up you should be able to try it yourself. I just commented
> out the "WIDGET_CONTROL, wBase, TIMER=3D1" line in the main procedure,
> and commented out the lines in the event procedure that checked to see
> if the event received was a WIDGET_TIMER event (beginning with "IF
> TAG_NAMES(sEvent, /STRUC) eq", running from lines 107-111, at
> the top of the event procedure).
>
> So, in essence, to these inexperienced eyes, there seems to be two
> timer schemes operating simultaneously, one which just fires off a
> timer event every one second no matter what, and the other which
> drives the animation at a rate that are user-selectable.

Well, you are correct that the timer mechanism is not observable to you. It is buried in the IDLitWindow object and the timer behavior is controlled by methods to that object. Specifically, using the SetEventMask method to turn TIMER_EVENTS either on or off. Once turned on they fire at a user-defined rate until turned off.

Presumably the IDLitWindow object receives the timer event,

notifies the timer observer or observers of it, then fires the next timer event. I say "presumably" because this is the only iTool function on the planet that is hidden from end users. I have no idea why that is the case. But it is an internal routine, rather than a routine written in the IDL language and made available to you.

This is actually quite nice behavior, although it doesn't particularly help the new user understand how timers work, because it takes some of the drudgery out of writing the timer mechanism yourself. This is the only timer program I've ever seen that is written like this, and of course, it only works in object graphics. I would bet well over 90% of the programs written outside the confines of ITTVIS are written in direct graphics, so that may be why it hasn't come to our attention yet.

It would be useful to have a direct graphics equivalent of this behavior and I can see immediately how to add it to my Catalyst Library of direct graphics widget objects. Then we could all use it. Well, we could if I ever get around to writing a User Guide for it. :-)

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Sepore ma de ni thui. ("Perhaps thou speakest truth.")
