Subject: Re: Can Timer interrupt widget callbacks?
Posted by Helder Marchetto on Mon, 30 Mar 2015 09:00:34 GMT
View Forum Message <> Reply to Message

On Sunday, March 29, 2015 at 9:50:33 PM UTC+2, David Grier wrote:
> On Sunday, March 29, 2015 at 2:58:36 PM UTC-4, Helder wrote:
>> On Sunday, March 29, 2015 at 2:45:33 PM UTC+2, David Grier wrote:
>>> Dear Folks,
>>>
>>> A change in the behavior of Timer callbacks from IDL 8.3 to IDL 8.4 has broken one of my
>>> applications, and I could use some help in fixing it.
>>>
>>> Under IDL 8.3, the firing of an asynchronous Timer preempts widget callbacks.
>>> This behavior appears to have been reversed in IDL 8.4, so that callbacks now take
precedence.
>>>
>>> Here's the issue ...
>>>
>>> My application has a widget_draw object that is supposed to update at regular intervals while

>>> the user drags graphics objects across the screen. I'm using Timer events to trigger the
updates.
>>> Under IDL 8.3, the widget_draw's animation is smooth. Under IDL 8.4, the animation stops
>>> updating during drag events, which defeats the purpose of the animation.
>>>
>>> Is there any way to restore the old behavior so that firing a Timer interrupts a widget
>>> callback, perhaps as an option to Timer::Set()?
>>>
>>> If there's no way to make the Timer "dominant", does anyone have suggestions for modifying
>>> my widget callbacks so that they can check for pending timer events and handle them?
>>>
>>> All the best,
>>>
>>> David
>>
>> Hi David,
>> not sure if this is what you are looking for... could it be: !DEBUG_PROCESS_EVENTS = 0
>>
>> From the IDL help (http://www.exelisvis.com/docs/Whats_New_8_3.html):
>> Event handling while debugging
>> In the past, IDL would not sent widget events when you were stopped within a routine. Now,
by default, IDL sends widget events even when stopped within a routine. This allows you to use
graphics and widget applications while debugging.
>> There is a new system variable, !DEBUG_PROCESS_EVENTS, that can be set to 0 to
disable this behavior, or to 1 to enable this behavior. The default value is 1.
>>
>> I hope it helps.
>>

>> Cheers,
>> Helder
>
> Hi Helder,
>
> This is a great idea, but not apparently what's going on in my program. I tried setting
> !DEBUG_PROCESS_EVENTS = 0
> but did not see any change in performance. Under IDL 8.3 the program displays the animation
> smoothly, and under IDL 8.4, the animation pauses during drag events. This makes sense
> because !DEBUG_PROCESS_EVENTS was introduced in IDL 8.3 (where my program works)
> and does not seem to have changed in IDL 8.4 (where it doesn't).
>
> I'm guessing that the difference I'm seeing is due to a change in the behavior of asynchronous
> timer events because there's a notation to that effect in the Version History of the
documentation
> for Timer.
>
> Even so, knowing about !DEBUG_PROCESS_EVENTS will be very useful for building widget
programs
> in the future -- thanks for the pointer!
>
> All the best,
>
> David

Hi David,
I was trying a quick answer to you question... I now realize, that !debug_process_events has nothing to do with your problem.

However, I have a maybe the answer to "why" and a question:
Answer: timers have been modified "under the hood" according to this post:
https://groups.google.com/d/msg/comp.lang.idl-pvwave/NLy0qn0 JVVs/1smQwF_YPu0J
cite:
[* They no longer fire in the middle of system callbacks. For example,
they won't interrupt widget event handlers and object cleanup methods.
This is better since there are fewer nasty surprises.]
If you read further down, Doug explains in what conditions in 8.4 async timers will not interrupt the pro code in the widget event handling:
"async timer will not fire until the event handler finishes. As long as execution is in the "scope" of the handler, timers will not be processed."

My guess: It seems that you're going to have to modify your code. Maybe you could pass on a variable to eventually fire the timer within the widget handling (as suggested by Doug).

Question: I never used async timers, but I work a lot with draw widgets and mouse dragging operations. I normaly use widget events. *Why* don't you use those to update your video interface? Sorry if it's a stupid question, but I would have gone (out of ignorance) for the widget event handling and some widget timers.

Mondays are always good days to learn something new...

Cheers and good luck with those timers...
Helder
