Subject: Re: asynchronous timers
Posted by Doug on Mon, 18 Aug 2014 16:27:51 GMT
View Forum Message <> Reply to Message

On 8/16/14, 2:28 AM, superchromix wrote:
> On Saturday, August 16, 2014 12:51:23 AM UTC+2, Doug wrote:
>> All,
>>
>>
>>
>> For IDL 8.4, we have done a couple of things to timers:
>>
>>
>>
>> * 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.
>>
>>
>>
>> * For when there still are nasty surprises, there will be the "block"
>>
>> and "unblock" methods to enable the programmer to specify when they
>>
>> don't want code to be interrupted. This is useful if currently
>>
>> executing code is writing/reading to data that a timer will read/write.
>>
>> Place "block" and "unblock" around code that shouldn't be interrupted.
>>
>>
>>
>> As for pseudo-multithreading, one really doesn't get any benefit since
>>
>> there's still only one IDL interpreter. For that you'll need another
>>
>> IDL, of course.
>>
>>
>>
>> Cheers,
>>
>> Doug
>
>
> Thanks for the info. When you say "they won't interrupt widget event handlers",

what do you mean exactly? Suppose I have a GUI application, which is event driven.
Essentially ALL routines are handling events in some sense. Does this mean the
asynchronous timer wouldn't fire at all in my application? Or do you mean only
that the timer wouldn't interrupt the "top level" event handler? (ie the procedure
that is specified when starting Xmanager?)
>
> best,
> Mark
>

Hi Mark,

Asynchronous timers can have their callbacks invoked at two points:

(1) While IDL is idle (waiting for the next IDL statement or native event (which often get translated into an IDL event))
(2) While IDL is executing PRO code (with the exceptions mentioned previously)

The first one is what widget-based timers were limited to. The new timers still fire there as well, so if one has an event-driven widget program the new timer's callbacks will be invoked between widget events.

If a timer pops while IDL is busy processing a callback, invocation of the timer's callback is postponed until IDL is able to handle it.

Cheers,
Doug
