Can I force a WM_TIMER message to be generated when the timer comes due, even if the message queue is not idle?

devblogs.microsoft.com/oldnewthing/20191108-00

November 8, 2019

Raymond Chen

Normally, WM_ TIMER messages for due timers are generated only if the message queue is empty, which puts them very close to the bottom of the message priority list. But what if you want to give the timer message the same priority as posted messages?

You may have a background task that, under periods of high activity, posts a lot of work to the UI thread. During those busy periods, WM_ TIMER messages won't be generated because the message queue is never empty. But you want the timer to run so you can update some UI, like,`say, a progress meter.

Some time ago, we discussed the problem of a message queue filled with unprocessed WM_TIMER messages. We can take that understanding of how these timer messages can be forced to be created, and we can use that power for good, rather than for evil.

The work items that are posted to the UI thread can force a timer message to be generated if a timer is due:

```
void ForceTimerMessagesToBeCreatedIfNecessary()
{
    MSG msg;
    PeekMessage(&msg, nullptr, WM_TIMER, WM_TIMER, PM_NOREMOVE);
}
```

The filtered PeekMessage will look for a WM_ TIMER message in the queue. If one is found, then it is copied to the msg (which we discard), but it is not removed from the message queue (PM_ NOREMOVE) so it will be processed by the main message pump.

If one is not found, the system will see if any timers are due, and if so, it will synthesize a WM_ TIMER message and insert it into the queue, and then return a copy.

If there are no timer messages in the queue, and no timer is due, then nothing happens.

Note that this mechanism does not suffer from timer message build-up because it generates a timer message only if there isn't one already. It could trigger systimer message build-up, but that won't be a problem provided that your main message pump is pumping *all* messages.

The message build-up problem exists when you force the synthesis of messages and steadfastly refuse to process them. So don't do that.

Raymond Chen

Follow