What can I do about timer build-up when waiting for COM outbound calls to complete?

devblogs.microsoft.com/oldnewthing/20211008-50

October 8, 2021

Raymond Chen

Some time ago, I talked about how COM synchronous calls can result in timer message build-up So what can you do about it?

This is one of those situations where life puts you between a rock and a hard place, and you just have to create your own escape route.

One solution I've seen employed is to peek out the timer messages before they can build up, and save them for later. To simulate timer coalescing, keep only the most recent instance of any timer (identified by matching the window and timer ID). After the COM outbound call returns, re-post the saved timer messages.

It's a simple idea, but the hard part is actually doing it. Before you start making outbound COM calls, you need to call `CoRegisterMessageFilter` to install a custom message filter. That message filter forwards all calls to the previous message filter (or requests default behavior if there is no previous message filter), but it contains extra work in the `Message-Pending` method to peek out any timer messages and coalesce them.

After you finish making COM outbound calls (and before you pump messages), post the saved timer messages back to their original windows. Some of the `PostMessage` calls may fail if the window no longer exists, or if the timer is no longer active.

There is a risk here that during the COM outbound call, the timer may have been stopped and then restarted. In that case, you may be reposting an old timer message that the window had thought it had previously stopped. If you hadn't applied this workaround, those leftover timer messages would have remained in the queue anyway. So you're not making things any worse than they already were.

One tiny detail that does change is that if the `WM_TIMER` message handler calls `Get-MessagePos`, they will get the mouse cursor position at the time the timer message was re-posted, rather than the time the timer message was originally generated. On the other hand, if COM hadn't forced the generation of those timer messages, they would have been generated at the time the COM call returned anyway, so you are just replacing one reality with another equally-valid one.

The people who have tried it report that it worked well enough.

Raymond Chen

Follow