When can a thread receive window messages?

devblogs.microsoft.com/oldnewthing/20040608-00

June 8, 2004

Raymond Chen

Everybody who has messed with window messaging knows that GetMessage and PeekMessage retrieve queued messages, which are dispatched to windows via DispatchMessage. Most people also know that GetMessage and PeekMessage will also dispatch nonqueued messages. (All pending nonqueued messages are dispatched, then the first queued message is returned.) But apparently not many people realize that SendMessage will also dispatch messages! If one thread T1 send a message to a window that belongs to another thread T2, the sending thread T1 is put to sleep until the receiving thread replies to the message. But if somebody else sends a message to thread T1, thread T1 **is woken to process the message**, then is returned to sleep. Why is that? Well, when two threads T1 and T2 are working together, it's common that thread T1 may send a message to thread T2, and while handling the message, thread T2 will send messages back to thread T1 before it returns to T1. Therefore, thread T1 must be ready to accept incoming sent messages. For example, thread T1 may send a message saying, "Tell me about all the X's that you know." Thread T2 will then send one message back to thread T1 saying, "Here's an X", and then another message to say "Here's another X", and so on, until it has finished telling thread T1 about all the X's, at which point it returns. Thread T1 now knows, when the original message returns, that it has received the entire list of X's from thread 2. This back-and-forth is how DDE service discovery works. Another case is that thread T1 sends a message to thread T2, and thread T2 needs to ask thread T1 for help before it can finish the operation. This isn't as strange as it sounds. You do something similar all the time without realizing it when you respond to a WM_NOTIFY message by sending messages back to the control that sent the notification. (For example, you may respond to a LVN_ITEMACTIVATE by sending back a LVM_GETITEM to get information about the item that was activated.)

So remember: Any time you send a message, there is a potential for re-entrancy.

Raymond Chen

Follow