How can I know when a window has processed a message that I posted to it?

devblogs.microsoft.com/oldnewthing/20241125-00

November 25, 2024

A customer was writing diagnostic code to monitor another application. They found that under certain conditions (which they can detect by other means), the program would stop responding to incoming posted messages, though it would respond to incoming sent messages.[1] Is there version of `PostMessage` that puts the message in the message queue and either waits for the message to be retrieved (like `SendMessage`) or calls you back when the retrieval occurs (like `SendMessageCallback`)?

No, there is no such version. When a message is posted, it is placed in destination queue, and that's the end of it. There is no further tracking of the message.

If you can change the target program, you can post it a custom message like `WM_HEARTBEAT` and implement the handler for that message by posting an acknowledgement back.

```
// Monitoring program posts a heartbeat
PostMessage(hwndAppBeingMonitored, WM_HEARTBEAT,
    (WPARAM)hwndMonitor,
    MAKELPARAM(replyMessage, replyMessageData));

// Program being monitored replies with a heartbeat
case WM_HEARTBEAT:
    PostMessage((HWND)wParam, LOWORD(lParam),
                HIWORD(lParam), 0);
```

If you cannot make changes to the app being monitored, you could install a `WH_GETMESSAGE` hook onto the UI thread of the app you are monitoring, post the window a harmless message like `WM_NULL`, and have your hook respond when it sees that message.

Another option is to post a harmless message that has some observable effect, like `WM_CONTEXTMENU` and then use accessibility or UI Automation to detect when the menu appears. (You could also use accessibility or UI Automation to dismiss the menu.)

¹ This can happen if the thread is inside its own `SendMessage` call to another thread. While waiting for `SendMessage` to complete, the thread can process inbound sent messages, but inbound posted messages go into the queue and are not processed until the thread calls `GetMessage` or `PeekMessage`.