

Why do I get spurious WM_MOUSEMOVE messages?

 devblogs.microsoft.com/oldnewthing/20031001-00

October 1, 2003



Raymond Chen

In order to understand this properly, it helps to know where `WM_MOUSEMOVE` messages come from.

When the hardware mouse reports an interrupt, indicating that the physical mouse has moved, Windows determines which thread should receive the mouse move message and sets a flag on that thread's input queue that says, "The mouse moved, in case anybody cares." (Other stuff happens, too, which we will ignore here for now. In particular, if a mouse button event arrives, a lot of bookkeeping happens to preserve the virtual input state.)

When that thread calls a message retrieval function like `GetMessage`, and the "The mouse moved" flag is set, Windows inspects the mouse position and does the work that is commonly considered to be part of mouse movement: Determining the window that should receive the message, changing the cursor, and determining what type of message to generate (usually `WM_MOUSEMOVE` or perhaps `WM_NCMOUSEMOVE`).

If you understand this, then you already see the answer to the question, "Why does my program not receive all mouse messages if the mouse is moving too fast?"

If your program is slow to call `GetMessage`, then multiple mouse interrupts may arrive before your program calls `GetMessage` to pick them up. Since all that happens when the mouse interrupt occurs is that a flag is set, if two interrupts happen in succession without a message retrieval function being called, then the second interrupt will merely set a flag that is already set, which has no effect. The net effect is that the first interrupt acts as if it has been "lost" since nobody bothered to pick it up.

You should also see the answer to the question, "How fast does Windows deliver mouse movement messages?"

The answer is, "As fast as you want." If you call `GetMessage` frequently, then you get mouse messages frequently; if you call `GetMessage` rarely, then you get mouse messages rarely.

Okay, so back to the original question, "Why do I get spurious WM_MOUSEMOVE messages?"

Notice that the delivery of a mouse message includes lots of work that is typically thought of as being part of mouse movement. Often, Windows wants to do that follow-on work even though the mouse hasn't actually moved. The most obvious example is when a window is shown, hidden or moved. When that happens, the mouse cursor may be over a window different from the window it was over previously (or in the case of a move, it may be over a different part of the same window). Windows needs to recalculate the mouse cursor (for example, the old window may have wanted an arrow but the new window wants a pointy finger), so it *artificially sets the "The mouse moved, in case anybody cares" flag*. This causes all the follow-on work to happen, a side-effect of which is the generation of a spurious `WM_MOUSEMOVE` message.

So if your program wants to detect whether the mouse has moved, you need to add a check in your `WM_MOUSEMOVE` that the mouse position is different from the position reported by the previous `WM_MOUSEMOVE` message.

Raymond Chen

Follow

