

Trying to recover from enhancement software that generates fake input incorrectly

devblogs.microsoft.com/oldnewthing/20160715-00

July 15, 2016

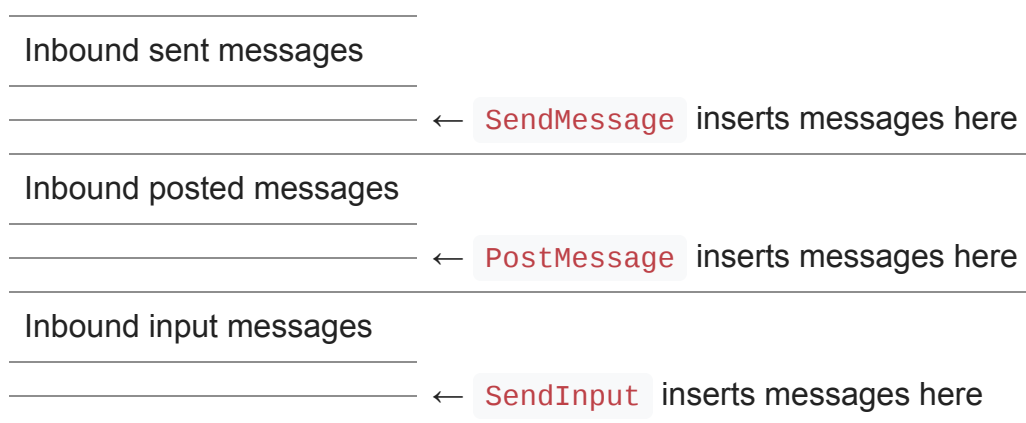


Raymond Chen

A customer had a program that supported the mouse wheel, and they found that some laptops have trackpad enhancement software which supports a gesture for turning the mouse wheel. When the enhancement software recognizes the gesture, it *sends* a `WM_MOUSEWHEEL` message to the foreground window.. This causes problems for the program because there are various things that are not allowed when processing an inbound sent message. Fortunately, they can detect whether they are in this situation, but it's not clear how they can recover from it. "We have been brainstorming and thinking that if we see this problem, we will use `Post-Message` to put the message 'back' into our queue."

The trackpad software is supposed to be using `SendInput` so that the wheel message orders correctly with the other messages in your input queue. The `SendMessage` delivers the message immediately (or as immediately as possible), and messages posted with the `Post-Message` function are processed ahead of input, both of which are wrong when you are trying to simulate input. Even your attempt to delay processing by calling `PostMessage` won't help because posted messages are processed ahead of input.

Here's a diagram which is inaccurate but may help to get the point across:



Basically, the enhancement software already screwed you with respect to message ordering. You won't be able to make a perfect recovery; all you can do is try to make the best of a bad situation.

One idea is to use the `ReplyMessage` function in your message handler. The `Reply-Message` function says, "Hey, like, act like I returned from this message for the purpose of inter-thread `SendMessage` bookkeeping, such as unblocking the sender, but let me keep running anyway." This may be enough to get the parts of the system that normally say "No, you can't do that from inside a sent message" to realize "Oh, wait, the synchronous part is over. Carry on."

If that doesn't work, then you can use the `SendInput` message to generate a wheel message back into the input queue. The wheel input will be a bit late (by the amount of time it took your window procedure to receive the message), but it'll probably be close enough to correct that most people won't notice. There's most likely already a lag in the gesture recognition in the enhancement software, so a little more lag probably isn't the end of the world.

Raymond Chen

Follow

