

How can I prevent the mouse from moving in response to touch input?



Raymond Chen

A customer had a program that responded to touch input, but they found that when the user touched the screen, the mouse jumped to the touch point. How can they prevent that?

What you can do is make your program `WM_POINTER`-aware: Process the various `WM_POINTER` messages directly, and don't let them go to `DefWindowProc`. It is the `DefWindowProc` function that takes unprocessed pointer messages and turns them into equivalent mouse activity.

You can take our [scratch program](#) and make these changes:

```
case WM_POINTERDOWN:
case WM_POINTERUPDATE:
case WM_POINTERUP:
case WM_POINTERWHEEL:
case WM_POINTERHWHEEL:
{
    auto pointerId = GET_POINTERID_WPARAM(wParam);
    POINTER_INPUT_TYPE type;
    if (GetPointerType(pointerId, &type) && type == PT_MOUSE) {
        return DefWindowProc(hwnd, uiMsg, wParam, lParam);
    }
    /* here is where you process the pointer message directly */
    return 0;
}
```

This program checks whether the pointer message came from a mouse. If so, then it lets the message go through and be processed normally.¹ Otherwise, it handles the message. Or at least, it *would* handle the message once you replace that comment with code that processes the message.

The mapping between pointer messages and mouse messages is

Pointer	Mouse
----------------	--------------

WM_POINTERDOWN	WM_*BUTTONDOWN
WM_POINTERUPDATE	WM_MOUSEMOVE
WM_POINTERUP	WM_*BUTTONUP
WM_POINTERWHEEL	WM_MOUSEWHEEL
WM_POINTERHWHEEL	WM_MOUSEHWHEEL

There are also corresponding nonclient pointer and mouse messages, but I'm going to let those be processed normally so you can use touch to drag the window by its title bar.

¹ Mouse messages by default don't even come in as `WM_POINTER` messages, but you can change that with `EnableMouseInPointer` .

[Raymond Chen](#)

Follow

