

How do I disable the press-and-hold gesture for my window?

 devblogs.microsoft.com/oldnewthing/20170227-00

February 27, 2017



Raymond Chen

A customer had a program which responded to left mouse clicks, but they found that when used with a touch screen, when users touched the screen, the `WM_LBUTTONDOWN` message didn't arrive until the users lifted their fingers from the screen. They wanted to know whether this was by design, and also whether there was a way to get the `WM_LBUTTONDOWN` message as soon as the finger touches the screen.

Yes, this behavior is by design. The system is waiting to see whether the user is making a press-and-hold gesture. If so, then the touch events are converted to right-mouse-button messages (`WM_RBUTTONDOWN` and `WM_RBUTTONUP`). But if the finger does not remain in contact for a long time, then the touch events are converted to left-mouse-button messages (`WM_LBUTTONDOWN` and `WM_LBUTTONUP`).

The customer's program was targeting Windows 7, so they were looking for solutions that would work on that platform.

Take our [scratch program](#) and add the following:

```

#include <strsafe.h> // StringCchPrintf
#include <tpcshrd.h> // WM_TABLET_QUERYSYSTEMGESTURESTATUS

BOOL
OnCreate(HWND hwnd, LPCREATESTRUCT lpcs)
{
    g_hwndChild = CreateWindow(TEXT("listbox"), NULL,
        LBS_HASSTRINGS | WS_CHILD | WS_VISIBLE | WS_VSCROLL,
        0, 0, 0, 0, hwnd, NULL, g_hinst, 0);
    return TRUE;
}

void
OnSize(HWND hwnd, UINT state, int cx, int cy)
{
    if (g_hwndChild) {
        MoveWindow(g_hwndChild, 0, 0, cx, cy/2, TRUE);
    }
}

LRESULT CALLBACK
WndProc(HWND hwnd, UINT uiMsg, WPARAM wParam, LPARAM lParam)
{
    ...
    case WM_LBUTTONDOWN:
    case WM_LBUTTONUP:
    case WM_RBUTTONDOWN:
    case WM_RBUTTONUP:
    {
        TCHAR buffer[80];
        StringCchPrintf(buffer, 80, TEXT("%04x %d %d"), uiMsg,
            GET_X_LPARAM(lParam), GET_Y_LPARAM(lParam));
        ListBox_AddString(g_hwndChild, buffer);
    }
    break;

    case WM_TABLET_QUERYSYSTEMGESTURESTATUS:
        return TABLET_DISABLE_PRESSANDHOLD;
    ...
}

```

Most of this code is just creating a logging window so we can see the message traffic. (Note that we divide `cy` by 2 in the `OnSize` function so that there is room at the bottom of the window for touch activity.)

The interesting part is adding a handler for the `WM_TABLET_QUERYSYSTEMGESTURESTATUS` message and responding that we want to disable press-and-hold.

This successfully disables the press-and-hold gesture on Tablet PC (remember that?), allowing the left-button messages to be generated immediately upon contact. But it doesn't help for Windows 7 and above. For that, we need something else:

```
BOOL
OnCreate(HWND hwnd, LPCREATESTRUCT lpcs)
{
    g_hwndChild = CreateWindow(TEXT("listbox"), NULL,
        LBS_HASSTRINGS | WS_CHILD | WS_VISIBLE | WS_VSCROLL,
        0, 0, 0, 0, hwnd, NULL, g_hinst, 0);

    GESTURECONFIG config;
    config.dwID = 0;
    config.dwWant = 0;
    config.dwBlock = GC_ALLGESTURES;
    SetGestureConfig(hwnd, 0, 1, &config, sizeof(config));

    return TRUE;
}
```

This time, we disable all gestures using `SetGestureConfig`. This takes care of Windows 7 and higher.

So there are your options: There's a "Windows XP and Windows Vista" solution, and there's a "Windows 7 and higher" solution. Or you can just play it safe and use both.

[Raymond Chen](#)

Follow

