

What's the story with the parameters to the WM_INPUT_DEVICE_CHANGE message?

devblogs.microsoft.com/oldnewthing/20110902-00

September 2, 2011



Raymond Chen

A customer found these strange macros in `winuser.h`:

```
#if (_WIN32_WINNT >= 0x0601)
#define GET_DEVICE_CHANGE_WPARAM(wParam) (LOWORD(wParam))
#elif (_WIN32_WINNT >= 0x0501)
#define GET_DEVICE_CHANGE_LPARAM(lParam) (LOWORD(lParam))
#endif /* (_WIN32_WINNT >= 0x0601) */
```

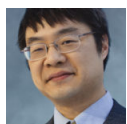
According to the documentation for the `WM_INPUT_DEVICE_CHANGE` message, the `wParam` is the operation code and the `lParam` is a handle to the device that changed. Given that definition, the correct macro would be `GET_DEVICE_CHANGE_WPARAM`. What's up with the bogus `GET_DEVICE_CHANGE_LPARAM` macro?

The macro was incorrectly defined in Windows Vista. In the Windows 7 version of the Platform SDK, the correct macro was added, but in order to avoid introducing a breaking change to existing code, the old broken macro remains in place in order to retain bug-for-bug compatibility with existing code.

Even though the macro didn't work, there is a good possibility that code exists which relied on it anyway. For example, people may have read the documentation, read the macro, realized that the macro was wrong, and worked around the bug like this:

```
case WM_INPUT_DEVICE_CHANGE:
    return OnInputDeviceChange(GET_DEVICE_CHANGE_LPARAM(wParam),
                               (HANDLE)lParam);
```

To avoid breaking this code, the old broken definition remains in the header file. But at least it's defined only if you say that you want the Windows Vista version of the header file, so at least people won't use the bad macro going forward.



Raymond Chen

Follow