My window has the WS_EX_NOACTIVATE extended style, but it got activated anyway

devblogs.microsoft.com/oldnewthing/20240919-00

September 19, 2024

If you set the WS_EX_NOACTIVATE extended window style, Windows won't activate the window when it is clicked on, Windows will skip the window when looking for a new window to activate when the active window is destroyed, and the window won't show up in Alt+Tab or the taskbar by default. You can still activate it programmatically, usually by calling Set-ActiveWindow or SetForegroundWindow.

If you override the default and make the window appear in Alt+Tab and the taskbar, then those UI surfaces will call SetForegroundWindow when the window is selected, so that's one way an end-user can activate a no-activate window.

But there's one oft-overlooked way the user can activate a no-activate window: Use the "Activate a window by hovering over it with the mouse" setting (known programmatically as "active window tracking"), and hover over the window.

When this happens, the "active window tracking" code doesn't check the WS_EX_NOACTIVATE style.[1] It goes directly to sending the window a WM_MOUSEACTIVATE message to ask whether the window should be activated, and DefWindowProc will say "Sure, activate me!"

If you want to prevent the window from being activated by mouse hover, you need to handle the WM_MOUSEACTIVATE message and return MA_NOACTIVATE or MA_NOACTIVATEANDEAT, depending on what you want to happen to the mouse event.

[1] This is arguably an oversight in the activate window tracking feature code when it was originally written.