

# Hotkeys involving the Windows logo key are reserved by the system

 [devblogs.microsoft.com/oldnewthing/20071130-00](http://devblogs.microsoft.com/oldnewthing/20071130-00)

November 30, 2007



Raymond Chen

Hotkeys involving the Windows logo key are reserved by the system. New ones are added over time. For example, Windows 98 added Win+D to show the desktop. Windows 2000 added Win+U to call up the Utility Manager. Windows XP added Win+B to move keyboard focus to the taskbar notification area. And a whole bunch of new hotkeys were added in Windows Vista, such as Ctrl+Win which is used by speech recognition. (It's also a bit of a pun: "Control Windows", get it?) The fact that these hotkeys are reserved is hard to find. It's buried in a bullet point a third of the way down the Guidelines for Keyboard User Interface Design document. It's also highlighted in the WHQL keyboard requirements right in the section title: "Windows Logo Key Support (Reserved for Operating System Use)". But even if you didn't find the documentation (and I don't blame you), the history of Windows clearly indicates that new Windows logo hotkeys arrive on a pretty regular basis. If your program uses one of those hotkeys, there's a good chance it'll conflict with a future version of Windows. Why isn't this mentioned in the `RegisterHotKey` documentation? Perhaps it should. But historically, function documentation merely told what a function does, not how it should be used. How a function should be used is a value judgement, and it was traditionally not the role of function documentation to make value judgements or specify policy. If you go to a programming language's specification document, you'll find a discussion of what the language keywords do, but no guidance as to how you should use them. Like a language specification, function documentation historically limited itself to how something is done, not whether it ought to be done. Recommendations traditionally belong in overviews, white papers, and other guidance documents. After all, the `CreateFile` documentation doesn't say "Do not create application data in the My Documents folder." The `CreateFile` function doesn't care; its job is to create files. Whether a file is created in the correct location is a guideline or policy decision at a higher level.

Note: I don't know whether that is still the position of the function documentation team, that it limit itself to the facts and avoid value judgements and guidance. My goal today is to draw attention to the guidance.

Raymond Chen

**Follow**

