

# The life story of the SwitchToThisWindow function

 [devblogs.microsoft.com/oldnewthing/20111107-00](http://devblogs.microsoft.com/oldnewthing/20111107-00)

November 7, 2011



Raymond Chen

Commenters Mick and Nick (you guys ever considered teaming up and forming a morning radio show?) are [interested in the life story of the SwitchToThisWindow function](#).

The `SwitchToThisWindow` was originally added in enhanced mode Windows 3.0 in order to support switching out of fullscreen MS-DOS sessions. Recall that enhanced mode Windows 3.0 was actually three operating systems in one: There was a 32-bit virtual machine manager, and inside one virtual machine ran a copy of standard-mode Windows,<sup>1</sup> and inside all the others ran a copy of MS-DOS. This meant that when you pressed a key when in an MS-DOS session, the keyboard interrupt went to the MS-DOS program and not to Windows.

When you pressed `Alt + Tab`, some crazy magic had to happen. The virtual machine manager had to “un-press” the `Alt` key in the MS-DOS program, then synchronize the shift states of the Windows virtual machine to match the one from the MS-DOS virtual machine. (For example, if you had the shift key down in the MS-DOS virtual machine, it had to simulate pressing the shift key in the Windows virtual machine so they two shift states were back in sync.) And then it could simulate pressing the `Tab` key, at which point the Windows virtual machine would see the `Alt + Tab` sequence and put up the `Alt + Tab` interface.

That’s how things worked if you were running in a windowed MS-DOS session. But if you were in a fullscreen MS-DOS session, things worked differently. Switching back to Windows would mean a display mode reset (which can take a second or longer), and then all the applications on your desktop had to redraw themselves (and probably paging quite a bit in order to do so). This definitely failed to meet the responsiveness people expected from `Alt + Tab`, so the virtual machine manager pulled a trick: If you pressed `Alt + Tab` while in a fullscreen MS-DOS session, then instead of switching back to the Windows virtual machine, the virtual machine manager displayed a text-mode version of the `Alt + Tab` interface.

I will stop to let the craziness of that sink in: The virtual machine manager had its own `Alt + Tab` interface built out of text mode.

Anyway, when you finally released the `Alt` key and completed the `Alt + Tab` sequence, the virtual machine manager needed to tell Windows, “Hey, like, pretend that an `Alt + Tab` thingie just happened, okay?”

That is what the `SwitchToThisWindow` function was for. It was the function the virtual machine manager called to tell Windows to switch to a window as if the user had selected it via `Alt + Tab` (because that is, in fact, what the user did, just via the text-mode interface rather than the graphical one).

A similar thing happened if you pressed `Alt + Esc` (or `Alt + Shift + Esc` in a fullscreen MS-DOS session. That’s why there’s a second parameter to indicate whether the switch should be done “in the style of `Alt + Tab`” or “in the style of `Alt + Esc`.”

The function was undocumented because it existed only for the virtual machine manager to call in order to coordinate its actions with Windows user interface so that you had one big happy `Alt + Tab` family.

The text-mode `Alt + Tab` interface disappeared in Windows 95, but the `SwitchToThisWindow` function hung around because it wasn’t causing anybody any harm, and there was at the time no formal process in place to deprecate and eventually remove an API, not even an internal undocumented one.

In the Windows XP SP1 timeframe, a bunch of lawyers decided that some functions in Windows needed to be documented. The precise rules for determining which functions needed to be documented and which didn’t need to be documented were rather complicated. (Some people applied an algorithm different from the ones those lawyers used and came up with a list of functions that are “missing”, when all that they really came up with is a list of functions different from the list those lawyers came up with.)<sup>2</sup>

Anyway, the `SwitchToThisWindow` function got caught in the dragnet, so it got documented. Mind you, like it says right at the top of the documentation, there is no guarantee that the function will continue to exist; it can vanish at any time. Although there is documentation, it has the logical status of an internal function, and internal functions have a tendency to change or vanish entirely. Perhaps someday a new chapter will be added to the life story of `SwitchToThisWindow`: “The `SwitchToThisWindow` was removed in Windows Q” for some value of Q.

## Footnotes

<sup>1</sup> Not true, but true enough. Don’t make me bring back the Nitpicker’s Corner.

<sup>2</sup> I will delete any comments on the subject of the algorithm by which those lawyers determined which functions needed to be documented, or on the documentation itself.

**Bonus chatter:** As far as I can determine, `SwitchToThisWindow` just does a `Set-ForegroundWindow` on the window you're switching to, possibly posting it a `WM_SYSCOMMAND / SC_RESTORE` message, and moving the previous foreground window to the bottom of the Z-order if switched via `Alt + Esc`. It doesn't provide any special secret sauce for bypassing the normal foreground activation rules. The process that calls `SwitchToThisWindow` still requires foreground-change permission.

Raymond Chen

**Follow**

