

# On the almost-feature of floppy insertion detection in Windows 95

 [devblogs.microsoft.com/oldnewthing/20090403-00](http://devblogs.microsoft.com/oldnewthing/20090403-00)

April 3, 2009



Raymond Chen

Gosh, [that floppy insertion article](#) generated a lot of comments.

First, [to clarify the table](#): The table is trying to say that if you had a *Style A* floppy drive, then issuing the magic series of commands would return 1 if a floppy was present, or 0 if the floppy was not present. On the other hand, if you had a *Style B* floppy drive, then issuing the magic series of commands would return 0 if a floppy was present, or 1 if the floppy was not present. That's what I was trying to say in the table. The answer was consistent within a floppy style, but you first had to know what style you had.

The downside of [waiting until the user uses a floppy for the first time](#) is that you have the *it sometimes works and sometimes doesn't* problem. Dad buys a new computer and a copy of the *Happy Fun Ball* game for his son. Dad turns on the computer, and then follows the instructions that come with the *Happy Fun Ball* package: "Just insert the floppy and follow the instructions on the screen." Dad inserts the floppy and... nothing happens because this is the first time Dad used the floppy, and he was expecting autodetection to work.

Dad says, "Stupid program doesn't work."

Dad complains to his co-workers at work. "He loves this game *Happy Fun Ball* when he visits his cousin's house, so I bought a computer and a copy of *Happy Fun Ball*, and it doesn't work!"

Dad tries again that evening and this time it works, because in the meantime, he inserted a floppy to do something else (say, create an emergency boot disk). Bizarre. This just reinforces Dad's impression that computers are unpredictable and he will never understand how to use them.

One could say that a feature that mysteriously turns itself on and off is worse than a feature that simply doesn't work. At least when it doesn't work, it predictably doesn't work. Human beings value predictability.

You can't perform the test "the first time the drive is installed" because there is no way to tell that a drive has been installed. (Classic floppy drives are not Plug-and-Play.) Even worse, you can't tell that the user has replaced the *Style A* floppy drive with a *Style B* floppy drive. The user will see that floppy insertion detection stopped working and return the drive to the store. "This drive is broken. Floppy insertion detection doesn't work."

It is also not the case that the ambiguity in the specification indicated a flaw in the specification. The C++ language specification, for example, leaves a lot of behaviors at the discretion of the implementation. This allows implementations to choose a behavior that works best for them. The *no-spin-up floppy presence detection* algorithm relied on several behaviors which were covered by the specification, and one that was not. It was not part of the original charter for the floppy specification committee to support spinless presence detection; it's just something that my colleague discovered over a decade after the specification was written.

But the main reason for not bothering is that the benefit was minuscule compared to the cost. Nobody wants floppy drives to spin up as soon as a disk is inserted. That just makes them think they've been attacked by a computer virus. It'd all just be a lot of work for a feature nobody wants. And then you'd all be posting, "I can't believe Microsoft wasted all this effort on floppy insertion detection when they should have fixed *insert favorite bug here*."

Raymond Chen

**Follow**

