

# Why does my mouse/touchpad sometimes go berzerk?

 [devblogs.microsoft.com/oldnewthing/20040917-00](http://devblogs.microsoft.com/oldnewthing/20040917-00)

September 17, 2004



Raymond Chen

Each time you move a PS/2-style mouse, the mouse send three bytes to the computer. For the sake of illustration, let's say the three bytes are x, y, and buttons.

The operating system sees this byte stream and groups them into threes:

x y b x y b x y b x y b

Now suppose the cable is a bit jiggled loose and one of the “y”s gets lost. The byte stream loses an entry, but the operating system doesn't know this has happened and keeps grouping them in threes.

x y b x b x y b x y b x

The operating system is now out of sync with the mouse and starts misinterpreting all the data. It receives a “y b x” from the mouse and treats the y byte as the x-delta, the b byte as the y-delta, and the x byte as the button state. Result: A mouse that goes crazy.

Oh wait, then there are mice with wheels.

When the operating system starts up, it tries to figure out whether the mouse has a wheel and convinces it to go into wheel mode. (You can influence this negotiation from Device Manager.) If both sides agree on wheeliness, then the mouse generates **four** bytes for each mouse motion, which therefore must be interpreted something like this:

x y b w x y b w x y b w x y b w

Now things get **really** interesting when you introduce laptops into the mix.

Many laptop computers have a PS/2 mouse port into which you can plug a mouse on the fly. When this happens, the built-in pointing device is turned off and the PS/2 mouse is used instead. This happens **entirely within the laptop's firmware**. The operating system has no idea that this switcheroo has happened.

Suppose that when you turned on your laptop, there was a wheel mouse connected to the PS/2 port. In this case, when the operating system tries to negotiate with the mouse, it sees a wheel and puts the mouse into “wheel mode”, expecting (and fortunately receiving) four-byte packets.

Now unplug your wheel mouse so that you revert to the touchpad, and let's say your touchpad doesn't have a wheel. The touchpad therefore spits out three-byte mouse packets when you use it. Uh-oh, now things are really messed up.

The touchpad is sending out three-byte packets, but the operating system thinks it's talking to that mouse that was plugged in originally and continues to expect four-byte packets.

You can imagine the mass mayhem that ensues.

Moral of the story: If you're going to hot-plug a mouse into your laptop's PS/2 port, you have a few choices.

- Always use a nonwheel mouse, so that you can plug and unplug with impunity, since the nonwheel mouse and the touchpad both use three-byte packets.
- If you turn on the laptop with no external mouse, then you can go ahead and plug in either a wheeled or wheel-less mouse. Plugging in a wheel-less mouse is safe because it generates three-byte packets just like the touchpad. And plugging in a wheeled mouse is safe because the wheeled mouse was not around for the initial negotiation, so it operates in compatibility mode (i.e., it pretends to be a wheel-less mouse). In this case, the mouse works, but you lose the wheel.
- If you turn on the laptop with a wheel mouse plugged in, **never unplug it** because once you do, the touchpad will take over and send three-byte packets and things will go berzerk.

Probably the easiest way out is to avoid the PS/2 mouse entirely and just use a USB mouse. This completely sidesteps the laptop's PS/2 switcheroo.

[Raymond Chen](#)

**Follow**

