It says that I should use USB usage page 1 and usage 6 to get raw keyboard data, but what if I have a PS/2 keyboard?

devblogs.microsoft.com/oldnewthing/20160926-00

September 26, 2016

Raymond Chen

Some time ago, I wrote about raw input and noted that if you want to receive raw keyboard input, you set the usage page to 1 and the usage to 6, noting that these numbers come from the USB HID specification. Commenter onodera asks, “Does this work for PS/2 input devices as well?”

Let’s try it. I happen to have a laptop within easy reach, so I plugged in an external USB wireless keyboard, then went to Device Manager. It says

Keyboards
Standard PS/2 Keyboard
Wireless Keyboard Filter Device

Then I ran the sample raw keyboard input program from the same article and pressed the space bar a few times on the external USB wireless keyboard, then the space bar a few times on the built-in PS/2 keyboard:

```
04E00817, msg=0100, vk=0020, scanCode=39, press
04E00817, msg=0101, vk=0020, scanCode=39, release
04E00817, msg=0100, vk=0020, scanCode=39, press
04E00817, msg=0101, vk=0020, scanCode=39, release
04E00817, msg=0100, vk=0020, scanCode=39, press
04E00817, msg=0101, vk=0020, scanCode=39, release
0001003D, msg=0100, vk=0020, scanCode=39, press
0001003D, msg=0101, vk=0020, scanCode=39, release
0001003D, msg=0100, vk=0020, scanCode=39, press
0001003D, msg=0101, vk=0020, scanCode=39, release
0001003D, msg=0100, vk=0020, scanCode=39, press
0001003D, msg=0101, vk=0020, scanCode=39, release
```

The entries from device `04E00817` are from the external USB wireless keyboard. The entries from device `0001003D` are from the built-in PS/2 keyboard.

So there you go: It works even for PS/2 keyboards.

The fact that the input is described in terms of HID usage pages and HID usages is just a convenient way of expressing what kind of input you want, because it generalizes in the obvious way to to HID devices. Non-HID devices are mapped to the corresponding HID usages.

Bonus chatter: Raw input also captures synthesized input. Here's the result after performing a `SendInput` to simulate pressing the space bar:

```
00000000, msg=0100, vk=0020, scanCode=39, press
00000000, msg=0101, vk=0020, scanCode=39, release
```

Notice that this was a completely synthetic keypress, but it showed up anyway, despite not having come from a USB device (or indeed any device at all). Note also that the device handle is null.

Raymond Chen

Follow