

Filtering out fake keyboards from the `GetRawInputDeviceList` function

 devblogs.microsoft.com/oldnewthing/20220303-00

March 3, 2022



Raymond Chen

Last time, we used the `GetRawInputDeviceList` to enumerate all the input devices, with a focus on keyboards. You may notice that on some systems, the function reports a lot of phantom keyboards. What's up with that?

The phantom keyboards are devices that report themselves to HID as keyboard devices, even though they aren't keyboards in a traditional sense. They might be a separate numeric keypad, or buttons on the chassis for volume or brightness control that report themselves as keyboard keys. You can see all the keyboard devices in Device Manager under *Keyboards*, including the phantoms. Can we filter out the phantoms?

I don't know, but I'm going to try.

```
bool SmellsLikeARealKeyboard(HANDLE hDevice)
{
    RID_DEVICE_INFO info;
    UINT size = sizeof(info);
    UINT actualSize = GetRawInputDeviceInfoW(
        hDevice, RIDI_DEVICEINFO, &info, &size);
    if (actualSize == (UINT)-1 || actualSize < sizeof(info))
    {
        // Weird failure.
        return false;
    }
    assert(info.dwType == RIM_TYPEKEYBOARD);
    return info.keyboard.dwNumberOfKeysTotal >= 30;
}
```

I'm going to say that a keyboard device smells like a real keyboard if it has at least 30 keys. This will rule out most devices that pretend to be a keyboard just so they can provide a handful of hardware buttons. (There is also a field that gives the keyboard type, but I've found that to be largely useless. One of my real keyboards reports as *Unknown*, and I have a phantom keyboard that reports as *Generic 101*.)

We can use this function to try to filter out phantom keyboards:

```

int main(int argc, char** argv)
{
    auto devices = GetRawInputDevices();
    int mouseCount = 0;
    int keyboardCount = 0;
    int otherCount = 0;
    for (auto const& device : devices) {
        switch (device.dwType)
        {
        case RIM_TYPEKEYBOARD:
            if (SmellsLikeARealKeyboard(device.hDevice)) {
                keyboardCount++;
            }
            break;
        case RIM_TYPEMOUSE: mouseCount++; break;
        default: otherCount++; break;
        }
    }
    printf("There are %d keyboards, %d mice, and %d other things\n",
        keyboardCount, mouseCount, otherCount);
    return 0;
}

```

It helps a little. Some of the phantom keyboards disappear. But at least on my machine, there's one phantom keyboard left: It describes itself as a *Virtual HID Framework (VHF) HID device*. This appears to be a facility for creating virtual devices.

I don't know enough about Windows device management to find an easy way to filter out these virtual keyboards. Even KeyboardCapabilities.KeyboardPresent appears to be faked out by them. Maybe somebody can pick up the ball here.

Raymond Chen

Follow

