

# Re: Identify USB Keyboard Device Instance For Each Character Dynamical

---

*Source:*

<http://www.tech-archive.net/Archive/Development/microsoft.public.development.device.drivers/2006-01/msg00699.1>

---

- *From:* "Doron Holan [MS]" <[doronh@xxxxxxxxxxxxxxxxxxxxxxxx](mailto:doronh@xxxxxxxxxxxxxxxxxxxxxxxx)>
  - *Date:* Tue, 17 Jan 2006 20:48:31 -0800
- 

IoOpenDeviceRegistryKey is API. in the INF, HKR is the location under the device install section in which you would write to.

d

---

Please do not send e-mail directly to this alias. this alias is for newsgroup purposes only.

This posting is provided "AS IS" with no warranties, and confers no rights.

"Liang Fu" <[liang\\_fu\\_dr@xxxxxxxxxxxx](mailto:liang_fu_dr@xxxxxxxxxxxx)> wrote in message [news:OHqpmG4GGHA.376@xxxxxxxxxxxxxxxxxxxxxxxx](mailto:news:OHqpmG4GGHA.376@xxxxxxxxxxxxxxxxxxxxxxxx)

> Doron,

> I am also interested in this topic. Your solution sounds great and seems well taken by the OP. But as a novice, I could not follow completely.

> You wrote:

>> to determine which device you are filtering, in your AddDevice routine

>> you

>> run your algorithm. Typically this would be looking at a devnode registry

>> key that your INF put down during installation that tells you the type of

>> device. you store this value in the device extension of your devobj. in

>> the callback routine, you are passed teh devobj, so it is a simple core

>> at

>> that point in time.

>

> Can you provide more information on how to interact with registry to find

> the information of the device being filtered? Is there code example for

> this?

>

> Thanks in advance.

>

> "Doron Holan [MS]" <[doronh@xxxxxxxxxxxxxxxxxxxxxxxx](mailto:doronh@xxxxxxxxxxxxxxxxxxxxxxxx)> wrote in message

> [news:ufj71NUGGHA.3984@xxxxxxxxxxxxxxxxxxxxxxxx](mailto:news:ufj71NUGGHA.3984@xxxxxxxxxxxxxxxxxxxxxxxx)

>> look at the kbfiltr example in the DDK, your life will be much easier.

> you

>> can easily hook into the scan code reporting mechanism by using the

## Re: Identify USB Keyboard Device Instance For Each Character Dynamical

> service  
>> callback as demonstrated in the sample. no irps, just an array of  
> packets.  
>> from there you can decide which elements of the array you report up to  
>> the  
>> class driver.  
>>  
>> to determine which device you are filtering, in your AddDevice routine  
>> you  
>> run your algorithm. Typically this would be looking at a devnode registry  
>> key that your INF put down during installation that tells you the type of  
>> device. you store this value in the device extension of your devobj. in  
>> the callback routine, you are passed teh devobj, so it is a simple core  
>> at  
>> that point in time.  
>>  
>> finally, if you still have control of the hw, put the barcode stuff into  
>> a  
>> separate top level collection. that is the whole point of HID, you  
> separate  
>> out different functions into separate pieces.  
>>  
>> d  
>>  
>> --  
>> Please do not send e-mail directly to this alias. this alias is for  
>> newsgroup purposes only.  
>> This posting is provided "AS IS" with no warranties, and confers no  
> rights.  
>>  
>>  
>> "cdavis999" <cdavis999@xxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote in message  
>> [news:141B0008-9679-441E-B434-69C77437D7FA@xxxxxxxxxxxxxxxxxxxx](mailto:news:141B0008-9679-441E-B434-69C77437D7FA@xxxxxxxxxxxxxxxxxxxx)  
>> >I have two USB keyboard devices attached to my target WinXP/SP2 box.  
>> >One  
>> > device is actually a USB barcode scanner that functions as a keyboard  
> and  
>> > one  
>> > actually is a USB keyboard. My debugger output below shows the  
>> > devstack  
>> > for  
>> > each device object under my kbdclass driver object. (The reason you  
>> > see  
>> > the  
>> > SysInternals Ctrl2cap filter driver is because I've been using the code  
> in  
>> > the Ctrl2capReadComplete function to peek at the scan codes.)  
>> >  
>> > What I'm trying to understand is whether it is possible to  
>> > programmatically  
>> > distinguish the IRPs from the barcode scanner from the IRPs from the

## Re: Identify USB Keyboard Device Instance For Each Character Dynamical

>>> actual  
>>> keyboard. I would like to intercept and redirect the barcode input  
> before  
>>> it  
>>> passes into the Windows keyboard input processing and is tied to  
>>> interpretation as messages by a windows procedure with keyboard focus.  
> So  
>>> I  
>>> need to evaluate the DO origin for each character.  
>>>  
>>> Can a specific (IRP that contains keystroke data) be associated somehow  
>>> with  
>>> the specific device object that originated it? (In my case, the  
>>> KeyboardClass2 DO with the Device Instance string  
>>> "HID\Vid\_065a&Pid\_0001\6&223cc4d1&0&0000" is my barcode scanner.)  
>>>  
>>> I would be grateful for any insight into possible approaches to solving  
>>> this  
>>> problem. I would also like to know it's infeasible or crazy. Also, as  
> a  
>>> newcomer to driver development I would appreciate any correction to any  
>>> obvious misunderstandings on my part.  
>>>  
>>> Thanks very much!  
>>>  
>>> Debugger Output:  
>>>  
>>> kd> !drvobj kbdclass  
>>> Driver object (866fd5e8) is for:  
>>> \Driver\Kbdclass  
>>> Driver Extension List: (id , addr)  
>>>  
>>> Device Object list:  
>>> 860a5030 86697030 866fe030  
>>>  
>>> kd> !devstack 860a5030  
>>> !DevObj !DrvObj !DevExt ObjectName  
>>> 8621d020 \Driver\Ctrl2cap 8621d0d8  
>>>> 860a5030 \Driver\Kbdclass 860a50e8 KeyboardClass2  
>>> 86535020 \Driver\kbdhid 865350d8  
>>> 864e20f8 \Driver\hidusb 864e21b0 0000005d  
>>> !DevNode 86699518 :  
>>> DeviceInst is "HID\Vid\_065a&Pid\_0001\6&223cc4d1&0&0000"  
>>> ServiceName is "kbdhid"  
>>>  
>>> kd> !devstack 86697030  
>>> !DevObj !DrvObj !DevExt ObjectName  
>>> 86277020 \Driver\Ctrl2cap 862770d8  
>>>> 86697030 \Driver\Kbdclass 866970e8 KeyboardClass1  
>>> 864e75a8 \Driver\kbdhid 864e7660  
>>> 86792920 \Driver\hidusb 867929d8 0000005b

Re: Identify USB Keyboard Device Instance For Each Character Dynamical

```
>>> !DevNode 86276468 :  
>>> DeviceInst is "HID\ Vid_413c&Pid_2003\6&15c34ccb&0&0000"  
>>> ServiceName is "kbdhid"  
>>>  
>>>  
>>  
>>  
>  
>
```

---

• **References:**

- ◆ **Re: Identify USB Keyboard Device Instance For Each Character Dynamical**  
    ◇ From: Doron Holan [MS]
- ◆ **Re: Identify USB Keyboard Device Instance For Each Character Dynamical**  
    ◇ From: Liang Fu
  
- Prev by Date: **Re: How to prevent "New hardware found" – wizard from starting?**
- Next by Date: **Re: WDK – new driver framework**
- Previous by thread: **Re: Identify USB Keyboard Device Instance For Each Character Dynamical**
- Next by thread: **what is "associated IRP" ?**
- Index(es):
  - ◆ **Date**
  - ◆ **Thread**