Enhanced Wheel Support in Windows

November 10, 2010

Abstract

Starting with Windows Vista®, the Windows® operating system offers enhanced support for next-generation wheel mouse devices. This paper provides guidelines for independent hardware vendors who are developing such input devices and for application developers who want to take advantage of wheel support in Windows.

This information applies to the following operating systems:   
 Windows 7

Windows Vista

The current version of this paper is maintained on the web at:   
 <http://www.microsoft.com/whdc/device/input/wheel.mspx>

References and resources discussed here are listed at the end of this paper.

For the latest information, see:   
 <http://www.microsoft.com/whdc/>

|  |
| --- |
| **Disclaimer**: This document is provided “as-is”. Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it.  Some examples depicted herein are provided for illustration only and are fictitious. No real association or connection is intended or should be inferred.  This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.  © 2010 Microsoft Corporation. All rights reserved. |

#### Revision History

|  |  |
| --- | --- |
| **Date** | **Change** |
| 10/22/2010 | Updated for Windows 7. |
| 10/16/2007 | * Corrected definitions of SPI\_SETWHEELSCROLLLINES and SPI\_SETWHEELSCROLLCHARS. * Corrected mistake in sample report descriptor. * Corrected descriptions of HID registry values. |
| 6/12/2006 | Created |

Contents

[Introduction 3](#_Toc275428616)

[Sample Report Descriptor 3](#_Toc275428617)

[Device Initialization and Reporting 6](#_Toc275428618)

[Initialization Phase 6](#_Toc275428619)

[Reporting Phase 6](#_Toc275428620)

[HID Registry Values 7](#_Toc275428621)

[Windows Wheel Support FAQ 7](#_Toc275428622)

[Resources 8](#_Toc275428623)

# Introduction

Starting with Windows Vista®, the Windows® operating system offers enhanced wheel support for human interface device (HID)-based pointing devices, including:

* Horizontal scrolling that is enabled by default in the operating system.
* Scrolling based on wheel movement that is smoother in both horizontal and vertical directions.
* More effective navigation of high-resolution and high-dpi (dots-per-inch) screens.
* Finer scrolling granularity.

Starting with Windows Vista, horizontal scrolling support is an extension of the vertical scrolling support that was first implemented natively in Windows 98. This implementation led to the standardization of wheel data reporting mechanisms for mouse devices and to the standardization of window messages related to vertical scrolling. Standardized window messages allow input device vendors to provide compatibility with applications without creating custom libraries for their devices or mapping custom libraries to legacy Windows libraries or messages. Windows Vista and Windows 7 use the **AC** Pan usage to support horizontal scrolling on HID devices that implement that feature.

Earlier versions of Windows don’t necessarily fully use the wheel’s ability to report movement. Starting with Windows Vista, the operating system's support for smooth scrolling takes full advantage of wheel movement granularity. However, vendors should ship their devices with smooth scrolling disabled, to make sure that the device works as expected on versions of Windows that do not support smooth scrolling. Windows Vista and Windows 7 automatically enable smooth scrolling for any device that supports it.

This paper describes the enhanced wheel support in Windows Vista Windows 7 for HID-based pointing devices. The information in this paper is for hardware engineers, driver developers, test managers, and independent hardware vendors. For information about proper implementation of mouse movement, see "Resources" at the end of this paper.

# Sample Report Descriptor

Starting with Windows Vista, a device takes advantage of the enhanced wheel by indicating its capabilities in its report descriptor. This section shows an example of a report descriptor that has been tested on Windows Vista. Vendors are not required to use this exact report descriptor.

#### Report Descriptor

The following example shows a complete report descriptor for a mouse that supports both horizontal and vertical wheel scrolling. This descriptor also allows for controlling the resolution multiplier by using features.

USAGE PAGE (Generic Desktop)

USAGE (Mouse)

COLLECTION (Application)

USAGE PAGE (Generic Desktop)

USAGE (Mouse)

COLLECTION (Logical)

REPORT\_ID (0x01)

USAGE (Pointer)

COLLECTION (Physical)

USAGE PAGE (Buttons)

USAGE MINIMUM (1)

USAGE MAXIMUM (5)

REPORT COUNT (5)

REPORT SIZE (1)

LOGICAL MAXIMUM (1)

INPUT (Var)

REPORT SIZE (3)

REPORT COUNT (1)

INPUT (Constant)

USAGE PAGE (Generic Desktop)

USAGE (X)

USAGE (Y)

REPORT COUNT (2)

REPORT SIZE (8)

LOGICAL MINIMUM (-127)

LOGICAL MAXIMUM (127)

INPUT (Var, Rel)

COLLECTION (Logical)

REPORT ID (0x02)

USAGE (Resolution Multiplier)

REPORT COUNT (1)

REPORT SIZE (2) ;OS requires > 1

LOGICAL MINIMUM (0)

LOGICAL MAXIMUM (1)

PHYSICAL MINIMUM (1) ;OS requires > 1

PHYSICAL MAXIMUM (4)

FEATURE (Var)

REPORT ID (0x01)

USAGE (Wheel)

PHYSICAL MINIMUM (0) ;Resets the global value

PHYSICAL MAXIMUM (0)

LOGICAL MINIMUM (-127)

LOGICAL MAXIMUM (127)

REPORT SIZE (8)

INPUT (Var, Rel) ;Vertical wheel scroll

END COLLECTION (Logical)

COLLECTION (Logical)

REPORT ID (0x02)

USAGE (Resolution Multiplier)

REPORT SIZE (2) ;OS requires > 1

LOGICAL MINIMUM (0)

LOGICAL MAXIMUM (1)

PHYSICAL MINIMUM (1)

PHYSICAL MAXIMUM (4)

FEATURE (Var)

PHYSICAL MINIMUM (0) ;Resets the global value

PHYSICAL MAXIMUM (0)

REPORT SIZE (4)

FEATURE (Constant)

REPORT ID (0x01)

USAGE PAGE (Consumer Devices)

LOGICAL MINIMUM (-127)

LOGICAL MAXIMUM (127)

REPORT SIZE (8)

USAGE (AC Pan)

INPUT (Var, Rel) ;Horizontal wheel scroll

END COLLECTION (Logical)

END COLLECTION (Physical)

END COLLECTION (Logical)

END COLLECTION (Application)

#### Input Report

The following input report is based on the example report descriptor shown earlier in this section.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Input Report (Non-Boot)** | | | | | | | | |
| **Input** | **D7** | **D6** | **D5** | **D4** | **D3** | **D2** | **D1** | **D0** |
| **Byte 0** | Report ID (0x01) | | | | | | | |
| **Byte 1** | 0 | | | FB | BB | MB | RB | LB |
| **Byte 2** | X | | | | | | | |
| **Byte 3** | Y | | | | | | | |
| **Byte 4** | Wheel | | | | | | | |
| **Byte 5** | AC Pan | | | | | | | |

**Legend:** FB Forward Button  
 BB Back Button  
 MB Middle Button  
 RB Right Button  
 LB Left Button

#### Feature Report

The following feature report is based on the example report descriptor shown earlier in this section.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Feature Report (Mouse Axis Resolution in Non-Boot)** | | | | | | | | |
| **Input** | **D7** | **D6** | **D5** | **D4** | **D3** | **D2** | **D1** | **D0** |
| **Byte 0** | Report ID (0x02) | | | | | | | |
| **Byte 1** | Reserved (0) | | | | AC Pan  Resolution Multiplier | | Wheel Resolution Multiplier | |

#### Collection Hierarchy

The collection hierarchy for the example report descriptor shown earlier in this section should look as follows.

Mouse Application Collection

Mouse Logical Collection

Pointer Physical Collection

Buttons (Input Report 0x01)

X (Input Report 0x01)

Y (Input Report 0x01)

Logical Collection

Resolution Multiplier (Feature Report 0x02)

Wheel (Input Report 0x01)

End Logical Collection

Logical Collection

Resolution Multiplier (Feature Report 0x02)

AC Pan (Input Report 0x01)

End Logical Collection

End Physical Collection

End Logical Collection

End Application Collection

As this collection hierarchy shows, without the innermost logical collections, the X, Y, Wheel, and **AC** Pan usages would be at the same level as the two resolution multipliers. There would be no way to tell which multiplier was associated with which axis or axes. Note that a report ID field is necessary only if you need it to associate more than one feature with another feature.

# Device Initialization and Reporting

Starting with Windows Vista, when the operating system identifies a HID-based device that reports a horizontal wheel, it initializes smooth scrolling support for the device. The device then reports whether it supports wheel scrolling or smooth scrolling.

## Initialization Phase

Windows Vista and Windows 7 identify whether a HID device supports vertical and horizontal scrolling and smooth scrolling by analyzing the device's report descriptor for Wheel and **AC** Pan usages.

If the report descriptor indicates that the HID device supports smooth scrolling, Windows sends the following Set\_Report request to the device to initialize smooth scrolling.

Setup Data: 21 09 02 03 00 00 02 00

------------------------------------

Direction: Host-to-device

Type: Class

Recipient: Interface

------------------------------------

bRequest: SET\_REPORT

wValue: Feature Report, Report ID 02

wIndex: Interface 0

wLength: 2

Data stage (2 bytes) :

------------------------------------

Report:(bytes 0-1) :

02 05

Windows currently selects the maximum resolution multiplier value. Future implementations may adopt other schemes.

Vendors should ship their devices with smooth scrolling disabled and allow Windows to enable it. This ensures that the device works like a regular HID device on legacy operating systems that do not support smooth scrolling.

## Reporting Phase

During the reporting phase, the HID device reports wheel scrolling or smooth scrolling. When the high-resolution multiplier is set, the value sent to applications is determined by multiplying the report value by 120 and dividing by the resolution multiplier’s physical maximum value. For example, assume that the mouse reports the following resolution multiplier values:

Logical minimum: 0   
Logical maximum: 1   
Physical minimum: 1  
Physical maximum: 4

During initialization, Windows Vista and later versions of Windows set the resolution multiplier to the logical maximum (1). When the mouse sends a wheel usage report with a resolution multiplier of x, Windows performs the following calculation:

(Application Value) = x \* 120 / 4

In this example, if the mouse reports a resolution multiplier of 1, an application receives a value of 30.

## HID Registry Values

Wheel scrolling and smooth scrolling are controlled by a set of DWORD registry values. Vendors should add or modify these values as needed by providing a custom INF file with appropriate AddReg directives.

The values are associated with the following registry key:

HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Enum\HID\

Vid\_xxxx&Pid\_yyyy\<instance\_ID>\Device Parameters

The following list shows the available values. They are set only if they are required by the device.

FlipFlopHScroll

Reverses the direction for horizontal scrolling. It is similar to the FlipFlopWheel value that was used with early prototype mouse devices to reverse the vertical scrolling values. The default value is zero, and a positive value reverses the direction.

VScrollPageOverride and VScrollUsageOverride

Override the page or usage that Windows uses for vertical scrolling.

HScrollPageOverride and HScrollUsageOverride

Override the page or usage that Windows uses for horizontal scrolling.

VScrollHighResolutionDisable

Disables high-resolution vertical scrolling. Third-party software can set this value to disable the Windows default behavior and control high-resolution vertical scrolling through third-party applications or drivers. The default value is zero, and a positive value disables high-resolution vertical scrolling.

HScrollHighResolutionDisable

Disables high-resolution horizontal scrolling for a device. Third-party software can set this value to disable the Windows default behavior and control high-resolution horizontal scrolling through third-party applications or drivers. The default value is zero, and a positive value disables high-resolution horizontal scrolling.

# Windows Wheel Support FAQ

The following section provides answers to frequently asked questions for vendors who are developing HID hardware and software for Windows Vista and Windows 7.

***Q. Can a user set or change the value of a resolution multiplier?***

No. Hardware reports the resolution multiplier in the HID device’s report descriptor, and Windows does not provide a mechanism for users to modify this setting. This could change with future versions of Windows.

Windows does allow users to enable or disable the mouse wheel vertical and horizontal scrolling option as a system-wide setting. To allow users to enable or disable these scrolling options, independent hardware vendors can create a vendor-specific **Wheel** tab for the Control Panel Mouse application that replaces the generic mouse wheel tab.

***Q. Does the Control Panel Mouse application allow users to enable or disable smooth scrolling support?***

No. Windows automatically enables smooth scrolling for mouse devices that report this functionality. There are currently no plans to provide system-wide or user-specific ways to disable smooth scrolling.

With Windows and later, third-party Control Panel applications can adjust smooth scroll settings by setting system-wide accessibility parameters. To do so, call the SystemParametersInfo function and set uiAction to one of the following values:

* SPI\_SETWHEELSCROLLLINES, which specifies the number of lines to scroll when the vertical mouse wheel is moved.
* SPI\_SETWHEELSCROLLCHARS, which specifies the number of characters to scroll when the horizontal mouse wheel is moved.

Use the function’s uiParam parameter to specify the number of lines or characters.

***Q. Is it necessary to modify legacy applications to support horizontal scrolling?***

Windows applications that base their user interface on standard Windows controls such as the Edit or ComboBox controls should not require modification. Those controls should be able to take advantage of native operating system support. However, hardware and applications should be thoroughly tested on the latest release of Windows to verify that they behave correctly.

Applications that do not support horizontal scrolling should be modified to handle the WM\_MOUSEHWHEEL message, as described in "Mouse Input."

***Q. Is it necessary to modify legacy applications to support smooth scrolling?***

Windows applications that handle window messages for vertical scrolling should not require modification to support smooth scrolling if the application acts on the mouse wheel delta and does not assume a message of 120. However, hardware and applications should be thoroughly tested on the latest release of the Windows operating system to verify that they behave correctly. For more information, see "Best Practices for Supporting Microsoft Mouse and Keyboard Devices."

# Resources

The following resources provide further information about mouse wheel support. For questions about enhanced wheel support in Windows Vista and Windows 7, send email to [inputdev@microsoft.com](mailto:inputdev@microsoft.com) with "Enhanced Wheel Support in Microsoft Windows Vista" in the Subject line.

#### General

Best Practices for Supporting Microsoft Mouse and Keyboard Devices

<http://msdn2.microsoft.com/en-us/library/ms997498.aspx>

Input and HID – Architecture and Driver Support

[http://www.microsoft.com/whdc/device/input/default.mspx](http://www.microsoft.com/whdc/device/input/default.mspx%20)

USB HID Specifications

<http://www.usb.org/developers/devclass_docs>

#### Platform SDK

Windows Controls

<http://msdn2.microsoft.com/en-us/library/bb773173.aspx>

Mouse Input

<http://msdn2.microsoft.com/en-us/library/ms645533.aspx>

Setup API

<http://msdn.microsoft.com/en-us/library/cc185682(VS.85).aspx>

#### Windows Driver Kit (WDK)

Overview of Device Installation

<http://msdn.microsoft.com/en-us/library/ff549455.aspx>