Intel® HID Event Filter

Release Notes and Bring Up Guide

December 2017

Revision 1.9

Intel Confidential

Contents

Revision History

Revision Number	Description	Revision Date
0.6	• Skylake Windows* 10 Beta Release	March, 2015
0.8	• Skylake Windows* 8.1 Production Candidate (PC) Release	May, 2015
0.9	• Intel® HID Event Filter Windows* 10 Certified Driver	June, 2015
1.0	• Intel® HID Event Filter Windows* 8.1 Certified Driver	June, 2015
1.1	• Intel HID Event Filter Windows* 10 TH2 Certified Driver • Supports Windows* 10 TH2 Device Guard Feature requirement.	October, 2015
1.2	• Intel® HID Event Filter Windows* 10 TH2 HotFix1 Certified Driver ver1.1.0.311 • Supports Windows* 10 TH2 Device Guard Feature requirement.	November, 2015
1.3	• Intel® HID Event Filter Windows* 10 TH2 Certified Driver ver 1.1.0.313 • Supports Windows* 10 TH2 Device Guard Feature requirement.	February, 2016
1.4	• Intel® HID Event Filter Windows* 10 Certified Driver ver 1.1.0.317 • Supports Windows* 10 TH2 Device Guard Feature requirement. • Updated Section 7: Closed issues list.	July, 2016
1.5	• Intel® HID Event Filter Windows* 10 RS1 certified ver 1.1.0.317 • Hardware Requirement in Section 1.1 updated	September 2016
1.6	• Intel® HID Event Filter Windows* 10 RS1 certified ver 1.1.0.318 • Hardware Requirement in Section 1.1 updated	November 2016
1.7	• Intel® HID Event Filter Windows* 10 RS2 certified ver 2.2.1.349	August 2017
1.8	• Intel® HID Event Filter Windows* 10 RS3 certified ver 2.2.1.364	November 2017
1.9	• Intel® HID Event Filter Windows* 10 RS3 certified ver 2.2.1.369	December 2017

1 Introduction

1.1 Purpose and Scope of Document

This document provides installation instructions and general usage of the driver as well as release information, such as release kit summary, important notes, resolved issues and known issues. This document is intended to help OEM and ODM customers setup their platform as they prepare for validation and debug.

Intel® HID Event Filter driver allows the SBIOS to send Intel® HID messages and button events to the operating system for various key presses. The driver supports the following operating system and platform:

Operating System:

- Windows* 8.1 Operating System
- Windows* 10 Operating System

Hardware Requirement:

- Skylake Platforms
- Kaby Lake Platforms
- Cannon Lake Platforms
- Coffee Lake Platforms
- Gemini Lake Platforms
- Apollo Lake Platforms

1.2 Acronyms and Terminology

Term	Description
ACPI	Advanced Configuration and Power Interface
ASL	ACPI Source Language
BTNE	Button Enable
BTNS	Button Status
BTNC	Button Control

Term	Description
BTNL	Button Load
BDW	Broadwell
BSW	Braswell
BYT	Bay Trail
EC	Embedded Controller
HSW	Haswell
HID	Human Interface Devices
MSFT	Microsoft
RVP	Reference Validation Platform
SKL	Skylake
KBL	Kabylake
SBIOS	System BIOS
GPIO	General Purpose IO
HDMM	Intel® HID Driver Mode Method
HDSM	Intel® HID Driver Status Method
HDEM	Intel HID Driver Event Method
HDDM	Intel® HID Driver Descriptor Method
PC	Production Candidate
PV	Production Version

1.3 Reference Documents

Document	Document No./Location
Handling_Buttons_and_Indicators_on_Microsoft_Windows_10	556574
BIOS Enabling Guide for Windows* 10	557130

2   Release Kit Summary

2.1   Release Kit Details

Kit Name: Intel® HID EventFilterDriver_v2.2.1.369_Win 10_RS3 Signed Driver Release

Version: Windows* 10 Certified Driver Release

- Intel® HID Event Filter driver_version_2.2.1.369

2.2   Kit Contents

The contents of this release kit include:

Intel® HID Event Filter. The driver installer compose of the following modules:

- Intel® HID EventFilterDriver_v2.2.1.369_Win 10_RS3 Signed Driver Release
- Intel® HID Event Filter Driver Release Notes and Bring Up Guide Software License Agreement

3 Architecture

The path to Intel® HID Event handling starts with a platform specific Hardware Event and continues up the software stack until it is serviced by the operating system. With the exception of the hardware event and the specific Intel® HID Message to service, a complete solution is being provided to support Intel HID Messaging while using PS2 Keyboard Controllers.

Intel® Event Handling:

1. Operating System Services Intel® HID
2. Hardware Event is generated.
3. Hardware Event is sent to SBIOS
4. SBIOS passes the event notification to Intel® HID event filter driver
5. Intel® HID Event Filter Driver passes Intel® HID Message onto

The below picture explains the buttons and indicator implementation on Windows* 8.1 vs Windows* 10.
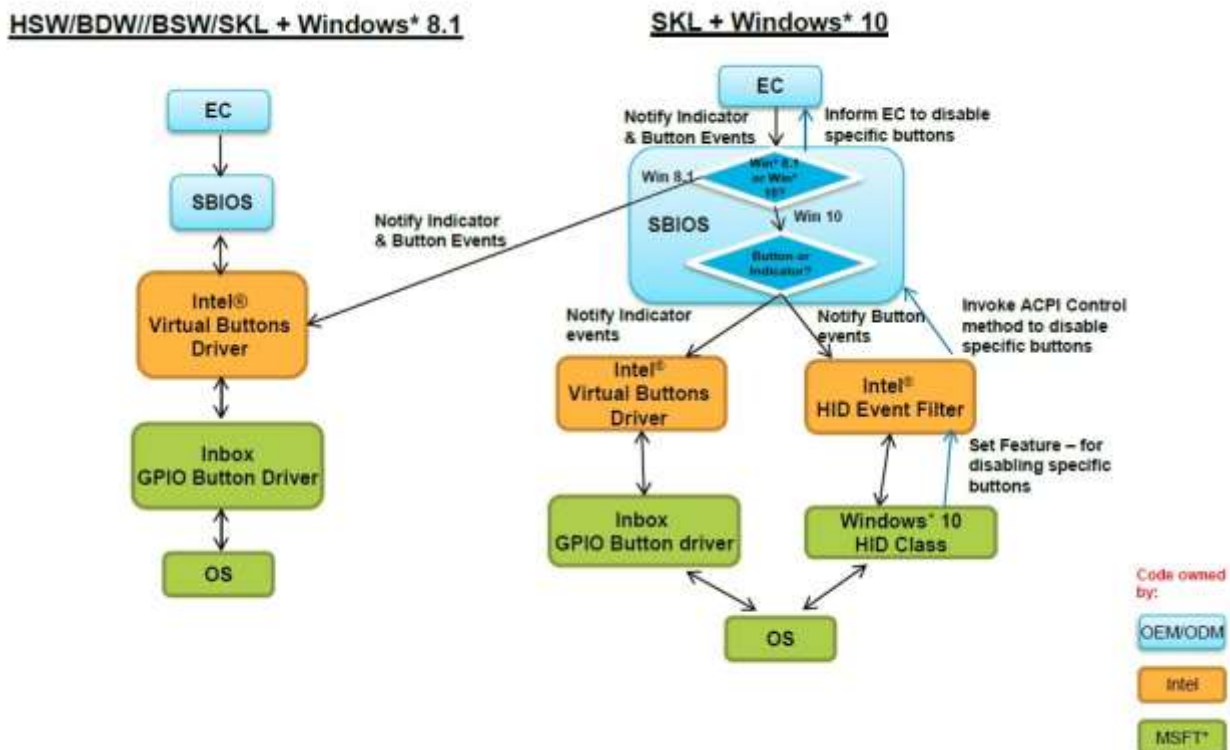
The BIOS receives EC notifications for buttons and indicator events, the OS version is then checked by the BIOS and the buttons/indicators events are dispatched to the respective driver. Below table depicts the notification flows to Intel® HID Event Filter driver versus Intel® Virtual Buttons Driver depending on the OS version detected by the BIOS.

CORE PLATFORMS	Windows* 8.1	Windows* 10 (Threshold)
Broadwell	**Buttons and Indicators:** Intel® Virtual Button Driver	**Buttons and Indicators:** Intel® Virtual Button Driver
Haswell	**Buttons and Indicators**: Intel® Virtual Button Driver	**Buttons and Indicators**: Intel® Virtual Button Driver
Skylake	**Buttons and Indicators**: Intel® Virtual Button Driver	**Buttons**: Intel® HID Event Filter Driver **Indicators**: Intel® Virtual Button Driver

3.1 Intel® HID Event Filter Driver on Windows* 8.1 versus Windows* 10

Windows* 8.1:

- EC notifies System BIOS of buttons and indicator events. The System BIOS uses ACPI control method to send notifications to communicate with the Virtual GPIO Buttons Driver.

- Driver services ACPI notifications from System BIOS and passes button/indicator State changes to the OS inbox buttons driver via exposed interface.

Windows* 10:

- EC notifies System BIOS of buttons and indicator events. The system BIOS will check for the OS version.
- If it is Windows* 10, the Indicator events are sent to Intel® Virtual Buttons Driver and the driver services ACPI notifications from System BIOS and passes indicator state changes to the OS inbox buttons.
- The Button events are sent to Intel® HID Event Filter driver and the driver services. ACPI notifications from System BIOS and passes button events to the Windows* 10 Intel® HID Class Driver.

3.2 Operating Modes

Intel® HID Event Filter will support Simple Mode:

Mode 0 (Simple Mode)

This mode will be simple for the OEM to implement in the BIOS ASL but will support only a limited number of keys that can be passed through to the OS. While running in Mode 0 (Simple), the BIOS will send an ASL

notification to the driver when it has a key that it is ready to send through. The driver will then call back to the BIOS to get the index that matches the key in the following table. For this specific set of keys, the driver will take care of the Intel® HID related reports, usage pages, and usage ids. The driver will build a static table in memory to get any other needed information at run time. The index is used to locate the correct row in the static table.

Supported Keys: Indexes 0 to 20 are optional and Index 27 is mandatory.

Index	Usage ID	Usage Name	Notes
0	-	-	Index 0 is not supported.
1	0xE3	Keyboard Left GUI (windows button)	Requires BIOS delay due to <Ctrl><Alt><Shift> being held for hotkey.
2	0xE3 + 0x69	Rotation Lock	Requires BIOS delay due to <Ctrl><Alt><Shift> being held for hotkey.
3	0x53	Num Lock	Requires BIOS delay due to <Ctrl><Alt><Shift> being held for hotkey.
4	0x4A	Home	Requires BIOS delay due to <Ctrl><Alt><Shift> being held for hotkey.
5	0x4D	End	Requires BIOS delay due to <Ctrl><Alt><Shift> being held for hotkey.
6	0x4B	Page Up	Requires BIOS delay due to <Ctrl><Alt><Shift> being held for hotkey.
7	0x4E	Page Down	Requires BIOS delay due to <Ctrl><Alt><Shift> being held for hotkey.
8	0xC6	Wireless Radio Button	Implemented. Spec says this is an OOC (On/Off Control). So, we should send '0' to turn off and '1' to turn on. However, on Windows 8 it works like all of the other OSC's and it toggles when the request is sent. Windows also turns airplane mode on/off when this HID request is sent to the OS. For now the driver always sends a '1'. The behavior is the same even when sending a '0'.
9	0x81	System Power Down	
11	0x82	System Sleep	

Index	Usage ID	Usage Name	Notes
12	0xB5	Scan Next Track	
13	0xB6	Scan Previous Track	
14	0xB7	Stop	
15	0xCD	Play/Pause	
16	0xE2	Mute	
17	0xE9	Volume Increment	
18	0xEA	Volume Decrement	
19	0x6F	Display Brightness Increment	
20	0x70	Display Brightness Decrement	
27	0x83	System Wake	

The hotkeys are supported in our test BIOS. Use <Ctrl><Alt><Shift><Hotkey> to activate.

4 Important Notes

4.1 New Features:

Below are the list of new features implemented in the Intel® HID Event Filter Driver for Windows* 10.

4.2 Button Support:

For Win10, we added the MSFT defined 5 button array report descriptor supporting the buttons in the HID Event Filter Driver. The 5 button events supported are Windows Home, Volume Increment, Volume Decrement, Rotation Lock and System power down.

Intel HID Event filter will process the button press and release events from BIOS and send corresponding input reports to OS

BIOS to define Press and Release Events for this 5 button array

Button Notification Value	Meaning
0xC2	Windows Home Button has been pressed
0xC3	Windows Home Button has been released
0xC4	Volume Up Button has been pressed
0xC5	Volume Up Button has been released
0xC6	Volume Down Button has been pressed
0xC7	Volume Down Button has been released
0xC8	Rotation Lock Button has been pressed
0xC9	Rotation Lock Button has been released
0xCE	Power Button has been pressed
0xCF	Power Button has been released

Usage ID	Usage Name	Notes
0xE3	Windows Home	BIOS defined ACPI notifications processed for press and release events
0xCA	Rotation Lock	BIOS defined ACPI notifications processed for press and release events
0x81	System Power Down	BIOS defined ACPI notifications processed for press and release events
0xE9	Volume Increment	BIOS defined ACPI notifications processed for press and release events
0xEA	Volume Decrement	BIOS defined ACPI notifications processed for press and release events

Important Note: The SKL BIOS ASL code and EC reference code for Skylake/Kabylake will include all the details on how to implement the new features mentioned in the previous sections. Customer can also implement the same EC/BIOS changes in their HSW/BDW EC and BIOS if they want to take advantage of the new features.

4.1.2 BTNE (Button Enable):

Intel® HID Event Filter driver has an additional support for Set feature request - Initiated by OS HID Stack to disable specific buttons. This invokes ACPI control method (BTNE - Button Enable) to disable specific buttons

BIOS sends ACPI control method (BTNE - Button Enable) to HID Event Filter driver enable/disable specific buttons.

0: Disabled

1: Enabled

Bits [5-31] - Reserved
Bits [4] – Rotation Lock Button
0 = Rotation Lock Button needs to be disabled
1 = Rotation Lock Button needs to be enabled
Bits [3] – Volume Down Button
0 = Volume Down Button needs to be disabled
1 = Volume Down Button needs to be enabled
Bits [2] – Volume Up Button
0 = Volume Up Button needs to be disabled
1 = Volume Up Button needs to be enabled
Bits [1] – Windows Button
0 = Windows Button needs to be disabled
1 = Windows Button needs to be enabled

Bits [0] – Power Button
0 = Power Button needs to be disabled
1 = Power Button needs to be enabled

4.1.3 BTNS (Button Status):

Intel® HID Event Filter also supports the 'Get feature request' - Initiated by OS HID Stack to query if a button is enabled/disabled. Intel® will Query Button status during driver load and CS/S3/S4 resume using ACPI control method (BTNS - Button Status)

BIOS implements ACPI control method (BTNS - Button Status) to query specific buttons that are enabled/disabled (BTNS needs to return status of all buttons as enabled on initial boot). This method returns the button status as an integer with the following button bit-level definition.

0: Disabled

1: Enabled

Bits [5-31] - Reserved
Bits [4] – Rotation Lock Button status
0 = Rotation Lock Button is disabled
1 = Rotation Lock Button is enabled
Bits [3] – Volume Down Button status
0 = Volume Down Button is disabled
1 = Volume Down Button is enabled
  Bits [2] – Volume Up Button status
0 = Volume Up Button is disabled
1 = Volume Up Button is enabled
Bits [1] – Windows Button status
0 = Windows Button is disabled
1 = Windows Button is enabled **Bits**
  [0] – Power Button status
0 = Power Button is disabled
1 = Power Button is enable

Note:

- BIOS implements ACPI control method (BTNC - Capability Query) to query buttons implemented on the platform
- BIOS also implements ACPI control method (BTNL - Button Load) - This method will be called upon loading of the HID Event Filter Driver. For enabling 10s Power Button, this method is responsible for sending the 10s Power Button command to the EC.
- EC to support a method for BIOS to inform EC to disable specific buttons: home, volume up, volume down, rotation lock.

5 Driver Installation

Note: A supported Operating System must be installed prior to the installation of the Intel® HID Event Driver.

There are two different methods to install the Intel® HID event filter driver for this release:

1. Driver Installation via Installer

2. Silent Driver Installation via Installer

5.1 Driver Installation via Installer

To install the Intel® HID Event Driver following steps must be taken:

1. Update the test system with BIOS that supports the INT33D5 ACPI device.

2. Install a new copy of Windows* 8.1 or Windows* 10.

3. Copy the installation package to the test machine.

4. Run the setup.exe program from within the Driver_Installer folder from the release package and follow the steps in the following windows.

5.2 Silent Driver Installation via Installer

Follow the steps listed below for silent driver installation via installer:

1. Open a Command Prompt (cmd.exe) with administrator rights (i.e. Run as Administrator). Click on 'Yes' button in User Account Control pop-up window.

2. Switch to the Intel HID installer

3. Setup.exe –s

5.3 Checking the Driver Version

To check the Intel® HID Event Driver version, follow the below instructions:

1. Open Device Manager.

2. In View, select "show hidden devises" Click on Human Interface Devices.

3. Double click on "Intel® HID Event Filter"

4. Select the "Driver" tab and the Driver Version will be listed.

5.4 Uninstalling the Driver via Control Panel

Follow the steps listed below to uninstall the driver via the Control Panel:

1. Open the Control Panel window.

2. If the Control Panel window is shown in 'Category' view, then select "Uninstall a program" as shown in Figure 6. Otherwise if the Control Panel window is shown in 'icon' view, then select "Programs and Features".

3. Follow the next windows to continue the uninstallation process. System will restart the after clicking "Finish" in the last step.

Figure 6. Control Panel – Uninstall a program

6 Known Issues

Issue #	Description
NA	

7   Closed Issues

Issue #	Description	Resolution
NA	MUP issue fixed	Intel® HID Installer update
NA	Device Guard Support	Implemented in Win* 10 TH2 Release versions 1.1.0.311 and newer.
1604178088	Button functionality not working when system booted from hybrid shutdown + G3	Fix implemented in Win* 10 Release version 1.1.0.317
NA	Intel® HID Event Filter Driver v1.1.0.317 certified with Windows* 10 RS1 WHCK build	NA
NA	Intel® HID Event Filter v1.1.0.318 driver has the installer fix for Arabic language	Fix implemented in Win* 10 Release version 1.1.0.318 RS1 certified
NA	When all buttons are disabled except wireless button, wireless button functionality was missing. The new release fixes this issue.	Fix implemented in Win* 10 Release version 2.2.1.349 RS2 certified
220703155	Incorrect Ctrl+Alt+Del log on message with HID Event filter driver	Fix implemented in Win* 10 Release version 2.2.1.364 RS3 certified
2006694467	Physical Buttons are no function after install Intel 2016 HID Event Filter Driver	Fix implemented in 2.2.1.369