Intel® HID Event Filter

Release Notes and Bring Up Guide

November 2023

Revision 2.2.2.6_Prod

Intel Confidencial

Document: XXXXXX

You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein. You agree to grant Intel a non-exclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at Intel.com, or from the OEM or retailer.

No computer system can be absolutely secure. Intel does not assume any liability for lost or stolen data or systems or any damages resulting from such losses.

The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at intel.com, or from the OEM or retailer.

All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest Intel product specifications and roadmaps.

Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or visit www.intel.com/design/literature.htm.

By using this document, in addition to any agreements you have with Intel, you accept the terms set forth below.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Intel and Intel logo are trademarks of Intel Corporation in the U.S. and/or other countries.

*Other names and brands may be claimed as the property of others.

Copyright © 2023, Intel Corporation. All rights reserved.

Contents

1 Introduction 6
1.1 Purpose and Scope of Document 6
1.2 Acronyms and Terminology 6
1.3 Reference Documents 8

2 Release Kit Summary 9
2.1 Release Kit Details 9
2.2 Kit Contents 9

3 Architecture 10
3.1 Intel® HID Event Filter Driver on Windows* 8.1 versus Windows* 10 11
3.2 Intel® HID Event Filter Driver Requirements 11
3.2.1 ACPI ID's 12
3.2.2 _DSM Function ID Definitions 13
3.3 Operating Modes 15

4 Important Notes 18
4.1 New Features: 18
4.2 Button Support: 18
4.3 BTNE (Button Enable): 19
4.4 BTNS (Button Status): 20

5 Driver Installation 22
5.1 Driver Installation via Installer 22
5.2 Silent Driver Installation via Installer 22
5.3 Checking the Driver Version 22
5.4 Uninstalling the Driver via Control Panel 23

6 Intel® Virtual Buttons Feature merged to Intel® HID Event Filter Driver 24

7 Known Issues 25

8 Closed Issues 26

Figures

Figure 1: Button/Indicator Implementation in SKL/KBL/APL 10
Figure 2. Uninstall a program 23
Figure 3. Flow diagram of Canon Lake and beyond Intel® HID Event filter Driver 24

Tables

Table 1: DSM Function ID 13
Table 2: Supported keys 15
Table 3: Button notification code meaning 18
Table 4: Button notifications 19

Revision History

Revision Number	Description	Revision Date
2.2.2.5	• Intel® HID Event Filter Windows* 10, Windows* 11 RS5,19H1, 20H1, 21H2, 22H2 certified ver 2.2.2.5_Prod	June, 2023
2.2.2.6	• Intel® HID Event Filter Windows* 10, Windows* 11 RS5,19H1, 20H1, 21H2, 22H2 certified ver 2.2.2.6_Prod	November, 2023

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
- Windows* 11 Operating System

Hardware Requirement:

- Kaby Lake Platforms
- Amber Lake Platforms
- Cannon Lake Platforms
- Coffee Lake Platforms
- Gemini Lake Platforms
- Apollo Lake Platforms
- Whiskey Lake Platforms
- Comet Lake Platforms
- Icelake Platforms
- Jasper Lake
- Tiger Lake Platforms
- Rocket Lake
- Alder Lake
- Raptor Lake
- Meteor lake
- Lunar Lake
- Arrow Lake
- Nova Lake
- Wildcat Lake

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
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

Term	Description
MTL	Meteor Lake
PC	Production Candidate
PV	Production Version
RPL	Raptor Lake
WHL	Whiskey Lake
ICL	IceLake

1.3 Reference Documents

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

2 Release Kit Summary

2.1 Release Kit Details

Kit Name: Intel® HID EventFilterDriver_v2.2.2.6_22H2_21H2_20H1_19H1_RS5 Signed Driver Release

- Intel® HID Event Filter driver_version_2.2.2.6_Prod

2.2 Kit Contents

The contents of this release kit include: Intel® HID Event Filter.

The driver installer compose of the following modules:

- Driver INF files
- Driver CAT files
- Driver SYS files
- Intel® HID Event Filter Driver Release Notes and Bring Up Guide Software License Agreement
- PV Intel OBL Software License Agreement

3 Architecture

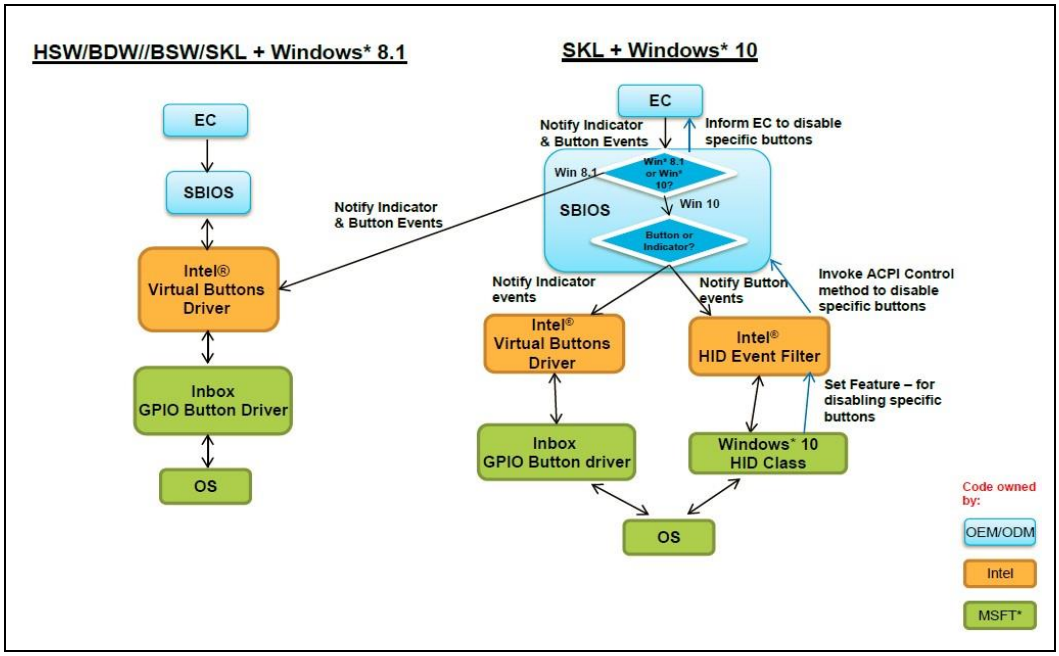
The path to Intel® HID Event handling starts with a platform specific Hardware Event and continues up the software stack until it is serviced by the operating system. With the exception of the hardware event and the specific Intel® HID Message to service, a complete solution is being provided to support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

Figure 1: Button/Indicator Implementation in SKL/KBL/APL



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 Intel® HID Event Filter Driver Requirements

The following settings should be implemented:

- Define input reports for 5 button array (Volume up/down buttons, power button, Windows button and Rotation lock/unlock button). Refer to Table 3 for code and definition.
- Process the button events from BIOS and send corresponding input reports to Windows* 10 OS HID Class Filter Driver.
- Apart from processing the button events, the Intel® HID Filter Driver for Windows* 10 will also support:
    - A "Set Feature" which will enable/disable specific buttons:
        - Add support for Set Feature request which is initiated by OS HID stack to disable specific button(s)
        - Invoke ACPI control method (BTNE – Button Enable) to enable/disable specific buttons

- A "Get Feature" request which will query status of specific buttons:
  - Add support for Get Feature request w
  - hich is initiated by OS HID stack to query status of specific button(s)
  - Query button status during driver load and Modern Standby/S3/S4 resume. ACPI control method (BTNS – Button Status) to query status of specific buttons

3.2.1 ACPI ID's

The following are the ACPI ID's supported for the Intel HID Event Filter Driver

- INT33D5
  - AML, GLK, CNL, WHL, ICL, CML
- INTC1051
  - TGL
- INTC1054
  - RKL
- INTC1070
  - ADL
- INTC1076
  - JSL
- INTC1077
  - MTL
- INTC1078
  - RPL
- INTC107B
  - LNL
- INTC10CB
  - ARL
- INTC10CC
  - PTL

3.2.2 _DSM Function ID Definitions

The following functions shall be supported for the Intel HID Event Filter Driver _DSM

- Function 0: Returns a buffer with a bit-field representing the supported function IDs:Table 1 Supported keys

Table 1: DSM Function ID

Function Index	ASL Object
1	BTNL Button Load Method: This method will be called upon loading of the HID Event Filter Driver. The intent of this method is to have a mechanism by which to seamlessly switch from the 4s Power Button behavior to the 10s Power Button behavior. This is a parameterless control method. If the 10s Power Button is enabled, then this method is responsible for sending the 10s Power Button command to the EC.
2	HDMM HID Driver Mode Method. BIOS to return the mode -> 0 = Simple
3	HDSM HID Driver Status Method. The driver will call this ASL function during D0 entry and D0 exit to let the platform know its current status. 0 - Driver not available. 1 - Driver available.
4	HDEM HID Driver Event Method. BIOS to return the index of a mode 0 supported BTNEkey. Refer to table (1) below for the index to be returned.
5	BTNS Button Status Method to query if specific buttons are enabled/disabled. BIOS to return a 32 bit value with the following definition, for button status **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 enabled
6	BTNE Button Enable/Disable Method To Enable/Disable specific buttons. 32 bit value with the following definition, sent to BIOS to enable/disable buttons **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
7	HEBC (v1) To query buttons implemented on the platform. BIOS to return a 32 bit value with the definition marked in table (2) below.

8	VGBS Virtual GPIO Button Status: Returns status of buttons and indicators. This method returns an integer with the following bit-level definition: **Bits [7] – Docking Indicator status** 0 = Docking Indicator is in undocked mode 1 = Docking Indicator is in docked mode **Bits [6] – Convertible Indicator status** 0 = Convertible Indicator is in slate mode 1 = Convertible Indicator is in Laptop (clamshell) mode **Bits [0 - 5] – Reserved**
9	HEBC (v2) To query buttons implemented on the platform. BIOS to return a 32 bit value with the definition marked in table (3) below.

3.3 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.

Table 2: Supported keys

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 +	Rotation Lock	Requires BIOS delay due to

Index	Usage ID	Usage Name	Notes
	0x69		<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	
12	0xB5	Scan Next Track	
13	0xB6	Scan Previous Track	

Index	Usage ID	Usage Name	Notes
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.

Table 3: Button notification code meaning

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

Table 4: Button notifications

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.

4.3 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

4.4 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 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

2 = Windows Button needs to be disabled

3 = Windows Button needs to be enabled

Bits [0] – Power Button

2 = Power Button needs to be disabled

3 = Power Button needs to be enabled

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:

a. Driver Installation via Installer
b. 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 2. 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 2. Uninstall a program

6 Intel® Virtual Buttons Feature merged to Intel® HID Event Filter Driver

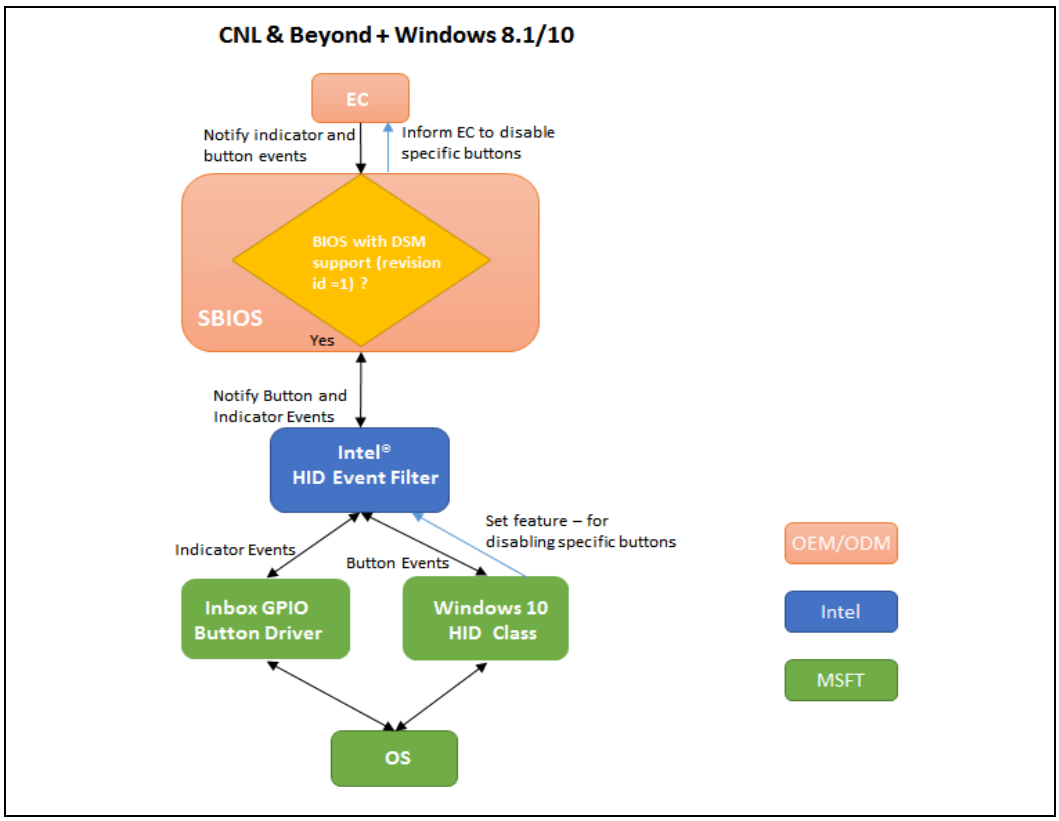
Both Button and indicatior notifications are handled by Intel® HID Event filter Driver from CNL and beyond for Windows 8.1/10.

For Sky lake/Kaby lake/Apolo lake there were two separate driver for Button and Indicator events

- Indicator events are handled by Intel® Virtual Button Driver
- Button events are handled by Intel® HID Event filter Driver

From Cannon Lake and beyond both events are solely handled by Intel® HID Event filter Driver. Refer **Table 4** for more details on button events.

Figure 3. Flow diagram of Canon Lake and beyond Intel® HID Event filter Driver



7 Known Issues

Issues	Description

8 Closed Issues

Usage ID	Usage Name	Notes