



# **Intel® HID Event Filter**

## **Release Notes and Bring Up Guide**

---

**January 2025**

**2.2.2.12**

**Intel Confidential**



You may not use or facilitate the use of this document in connection with any infringement or other legal analysis. 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.

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

All product plans and roadmaps are subject to change without notice.

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 technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Check with your system manufacturer or retailer or learn more at intel.com.

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.

Altering clock frequency, voltage, or memory interface speeds may void any product warranties and reduce stability, security, performance, and life of the processor and other components. Intel has not validated processor running memory above Plan-Of-Record (POR) speed. DRAM/DIMM devices should support desired speed, check with DRAM/DIMM vendors for details. System manufacturers are responsible for all validation and assume the risk of any stability, security, performance, or other functional issues resulting from such alterations.

© Intel Corporation. Intel, <Intel launch names, if mentioned in the slides below> the Intel logo, and other Intel marks are trademarks of Intel Corporation or its subsidiaries.

\*Other names and brands may be claimed as the

property of others. Copyright© 2025, Intel

Corporation. All rights reserved.

# Contents

---

Contents	3
1	Release Kit Summary ..... 6
2	General Information..... 7
3	Architecture..... 8
	3.1 Intel® HID Event Filter Driver on Windows* 10/Later ..... 9
	3.2 Intel® HID Event Filter Driver Requirements..... 9
	3.2.1 ACPI ID's ..... 9
	3.2.2 _DSM Function ID Definitions .....10
	3.3 Operating Modes .....13
	3.4 Button Support: .....14
	3.5 BTNS (Button Status): .....15
	3.6 Driver Installation .....16
	3.6.1 Install the Driver .....16
	3.6.2 Check the Driver Version .....17
	3.6.3 Uninstall the Driver .....17
	3.7 Intel® Virtual Buttons Feature merged to Intel® HID Event Filter .....17



# Revision History

Revision Number	Description	Revision Date
2.2.2.5	Intel® HID Event Filter release_Preprod	November 2023
2.2.2.7	Intel® HID Event Filter release_Preprod_sign_updated	January 2024
2.2.2.8	Intel® HID Event Filter release_Preprod	February 2024
2.2.2.9	Intel® HID Event Filter release_Production	February 2024
2.2.2.10	Intel® HID Event Filter release_Production	May 2024
2.2.2.11	Intel® HID Event Filter release_Production	September 2024
2.2.2.12	Intel® HID Event Filter release_Production	January 2025



# 1 *Release Kit Summary*

---

## 1.1 Release Kit Details

<b>Kit Release</b>	Intel® HID Event Filter Driver 2.2.2.12
<b>Target Platform</b>	<ul style="list-style-type: none"> <li>• Kaby Lake Platforms</li> <li>• Amber Lake Platforms</li> <li>• Coffee Lake Platforms</li> <li>• Gemini Lake Platforms</li> <li>• Apollo Lake Platforms</li> <li>• Whiskey Lake Platforms</li> <li>• Comet Lake Platforms</li> <li>• Ice Lake Platforms</li> <li>• Jasper Lake</li> <li>• Tiger Lake Platforms</li> <li>• Rocket Lake</li> <li>• Alder Lake</li> <li>• Raptor Lake</li> <li>• Meteor lake</li> <li>• Lunar Lake</li> <li>• Arrow Lake</li> <li>• Panther Lake</li> <li>• Wildcat Lake</li> </ul>
<b>Contents</b>	<ul style="list-style-type: none"> <li>• Hideeventfilter.cat</li> <li>• HidEventFilter.inf</li> <li>• HidEventFilter.sys</li> <li>• Intel® HID Event Filter Driver Release Notes and Bring Up Guide Software License Agreement</li> </ul>

## 2 *General Information*

---

### 2.1 **Important Notes**

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.

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

### 2.2 **Intended Audience**

This document is intended to help OEM and ODM customers setup their platform as they prepare for validation and debug.

### 2.3 **Supported OS**

OS Version
Windows 10
Windows 11

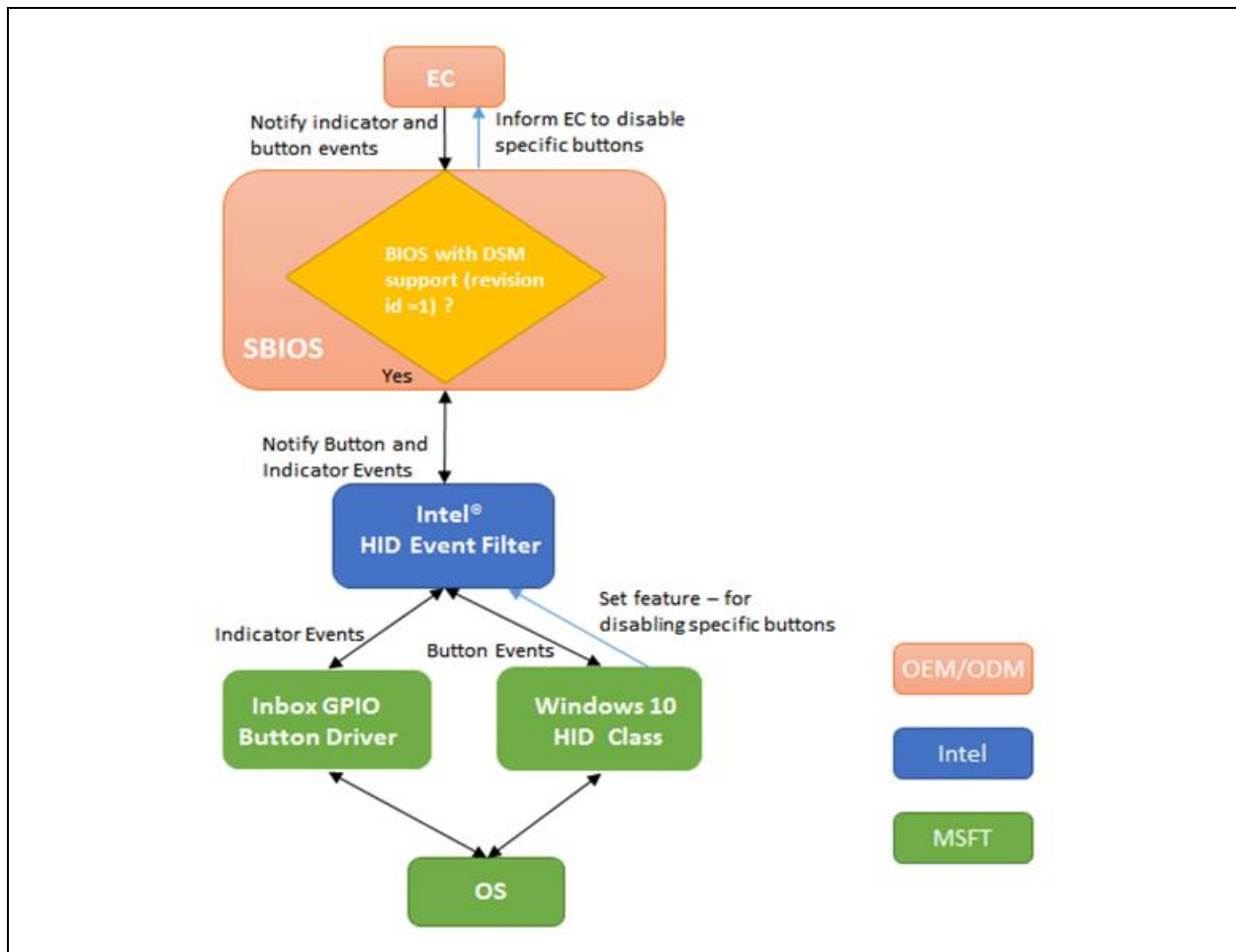
## 3 Architecture

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 picture explains the buttons and indicator implementation.







## 3.1 Intel® HID Event Filter Driver on Windows\* 10/Later

### 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 and later:

- 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\* 10/11 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).
- Process the button events from BIOS and send corresponding input reports to Windows\* 10 and later OS HID Class Filter Driver.
- Apart from processing the button events, the Intel® HID Filter Driver for Windows\* 10 and later 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 which 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, WHL, ICL, CML



- INTC1051
  - TGL
- INTC1054
  - RKL
- INTC1070
  - ADL
- INTC1076
  - JSL
- INTC1077
  - MTL
- INTC1078
  - RPL
- INTC107B
  - LNL
- INTC10CB
  - ARL
- INTC10CC
  - PTL
- INTC10F1
  - WCL
- INTC10F2
  - NVL
- INTC110E
  - RZL

### 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

**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	<p>HDMM HID Driver Mode Method. BIOS to return the mode -&gt; 0 = Simple</p>
3	<p>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.</p>
4	<p>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.</p>
5	<p>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</p> <p>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</p>

6	<p>BTNE Button Enable/Disable Method To Enable/Disable specific buttons. 32 bit value with the following definition, sent to BIOS to enable/disable buttons</p> <p>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</p>
7	<p>HEBC (v1) To query buttons implemented on the platform. BIOS to return a 32 bit value with the definition marked in table (2) below.</p>
8	<p>VGBS Virtual GPIO Button Status: Returns status of buttons and indicators. This method returns an integer with the following bit-level definition:</p> <p>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</p>
9	<p>HEBC (v2) To query buttons implemented on the platform. BIOS to return a 32 bit value with the definition marked in table (3) below.</p>



## 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 + 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	



Index	Usage ID	Usage Name	Notes
11	0x82	System Sleep	
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.

## 3.4 Button Support:

Starting from 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

Button Notification Value	Meaning
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

#### 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

## 3.5 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.

## 3.6 Driver Installation

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

### 3.6.1 Install the Driver

There is only inf. Installation supported for Intel® HID Event Driver.



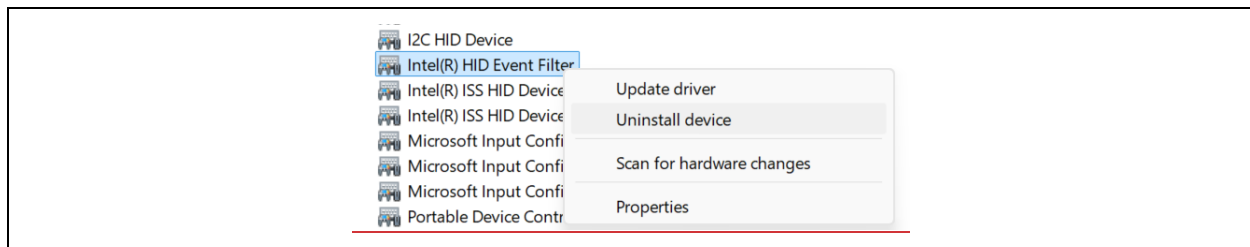
## 3.6.2 Check the Driver Version

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

1. Open Device Manager.
2. Double click on "Intel® HID Event Filter"
3. Select the "Driver" tab and the Driver Version will be listed.

## 3.6.3 Uninstall the Driver

1. Open Device Manager.
2. Right click on "Intel® HID Event Filter" and choose "Uninstall device".
3. Follow the steps in the next Windows to continue the uninstallation process.



## 3.7 Intel® Virtual Buttons Feature merged to Intel® HID Event Filter

Both Button and indicator notifications are handled by Intel® HID Event filter Driver beyond CNL for Windows 8.1/10/11.

For Kaby lake there were two separate drivers for Button and Indicator events

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