Intel® Intelligent Power Node Manager 5.0

External Interface Specification Using IPMI

August 2018

Revision 1.02

Intel Confidential

Notice: This document contains information on products in the design phase of development. The information here is subject to change without notice. Do not finalize a design with this information.

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.

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.

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.

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest Intel product specifications and roadmaps.

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.

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

Intel, the Intel logo, and Xeon 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.

Contents

Tables

Figures

Revision History

Document Number	Revision Number	Description	Date
575576	0.5	Initial version based on Intel® NM 4.0 External Interface Specification.	June 2017
575576	0.7	Document edited to support Mehlow platform.	August 2017
575576	0.9	Updated reference documents list. Updated list of sensors supported on Mehlow platform.	November 2017
575576	1.0	Updated "Get Device ID" command with supported SPS 4.0 platforms. Added IPMI OEM Intel® ME Firmware Update backward compatible commands chapter. Added "Set Intel® ME FW USB Port Override" command.	January 2018
575576	1.01	Added information on Completion Codes that stop events resending in appendix B.2. Updated "Set Intel® ME FW USB Port Override" command. Updated "Get Intel® Boot Guard Health" command. Updated "Platform Characterization Launch Request" command. Updated "Get PMBus Device Energy" command. Updated Table 5-1 with commands supported in SiEn SKU. Updated Table B-4 with IPMI sensors supported on Mehlow platform. Updated "Read/Write Memory SMBus" commands with compatibility notes for Mehlow platform. Updated Table B-3 with IPMI sensors supported in SiEn SKU.	April 2018
575576	1.02	Added list of commands recommended to be disabled for BMC bridging. Added note that BMC address is configurable with spsFITc tool. Updated note in Table 1-1 Intel® NM Operational Capabilities for Enable Condition of the Power Limiting Capability. Added note that response may take longer than 250 ms for commands that are accessing the flash. Updated "Update Reading Package" command.	August 2018

§

1 Introduction

1.1 Purpose of this Document

This document assists BMC vendors and external management software vendors in supporting Intel® Intelligent Power Node Manager 5.0. It documents the Intelligent Platform Management Interface (IPMI) commands that can be sent to the Intel® Management Engine (Intel® ME). This document also describes IPMI sensors implemented to ensure the correct and reliable operation of the platform.

1.2 Scope

This document describes the details on the IPMI commands used by Intel® Node Manager (Intel® NM) component running on an Intel platform. It focuses only on BMC-Assist SKU, aka Intel® NM SKU. Depending on Intel platform type the following Intel® NM 5.0 functionalities are provided:

- SI Enabling (no manageability, provides essential platform functionality because the FW is a mandatory component of every server platform and platform will not be fully operational if Intel® Server Platform Services FW is not installed on flash).
- Intel® Node Manager 5.0 Base - platform power management, power telemetry, platform domain only power limiting, Intel® NM Power and Thermal Utility (Intel® NM PTU) and Telemetry Hub.
- Intel® Node Manager 5.0 Segment Defined Features (SDF) - features defined additionally to Intel® Node Manager 5.0 Base: per domain power limiting, Compute Usage per Second (CUPS) and other advanced features. Those features are optional for platform operation and are included individually based on segment requirements.

IPMI command supported with Intel® Node Manager 5.0 Base and Intel® Node Manager 5.0 Segment Defined Features required for specific Intel® Node Manager 5.0 platforms are listed in Table 5-1.

1.3 Compatibility Notes

IPMI commands supported by Intel® Node Manager 5.0 defined in this document are compatible with [IPMI] and [IPMB] specifications. The length of the IPMI frames supported by Intel® Node Manager 5.0 is 80 bytes.

1.4 General Conventions

By default, all the values occupying more than one byte are LS Byte first encoded if not specified differently in their descriptions.

In not specified explicitly, all indexed values mentioned in this specification start from index 0.

1.5 System States and Power Management

Acronym or Term	Definition
S0	A system state where power is applied to all HW devices and system is running normally.
S1, S2, S3	A system state where the host CPU is not running however power is connected to the memory system.
S4	A system state where the host CPU and memory is not active.
S5	A system state where all power to the host system is off however power cord is still connected.
Sx	All S states that are different than S0/S1.
OS Hibernate	OS state where the OS state is saved on the hard drive.
Standby	OS state where the OS state is saved on memory and resumed from the memory when mouse/keyboard is clicked.
Shut Down	All power is off for the host machine however the power cord is still connected.

1.6 Reference Documents

Table 1-1 Reference Documents

Ref	Document Name	File/Location
[Addr]	*IPMB v1.0 Address Allocation*, 1998.	http://www.intel.com/design/servers/ipmi/spec.htm
[DCMI]	*Data Center Management Interface Specification* version 1.5, draft revision 0.7	http://www.intel.com/technology/product/dcmi/specification.htm
[EDS]	*8th Generation Intel Core Families External Design Specification Volume 1 of 2, Revision 2.4*	My Intel, Document Number: 570805
[PDG]	*Coffee Lake H Platform Design Guide Rev 1.8*	My Intel, Document Number: 571391
[PCH EDS]	*Intel® 300 Series and Intel® C240 Series Chipset Family Platform Controller Hub External Design Specification Volume 1 of 2 Revision 2.2*	My Intel, Document Number: 571182
[EPS]	*Intel® Server Platform Services 5.0 Firmware for Mehlow, Idaville and Jacobsville - External Product Specification*	My Intel, Document Number: 572822
[INT_GUIDE]	*Intel® Server Platform Services 5.0 Services Integration Guide*	My Intel, Document Number: 575469
[IPMB]	*Intelligent Platform Management Bus Specification*, version 1.0, 1999.	http://www.intel.com/design/servers/ipmi/spec.htm
[IPMI]	*Intelligent Platform Management Interface Specification*, version 2.0, 2004.	http://www.intel.com/design/servers/ipmi/spec.htm
[ME_BIOS_INT]	*Intel® Server Platform Services 5.0 Firmware for Mehlow, Idaville and Jacobsville Intel® Management Engine-BIOS Interface Specification*	My Intel, Document Number:575642
[PET]	*IPMI Platform Event Trap Format Specification*, version 1.0, 1999.	http://www.intel.com/design/servers/ipmi/spec.htm
[Platform IG]	*Intel® Server Platform Services Mehlow Platform Integration Guide*	My Intel, Document Number: 574653

§

2 Intel® Management Engine IPMI Interface

This chapter contains IPMI commands and sensor devices provided by Intel® Management Engine (Intel® ME). BMC shall use these commands and sensors to control Intel® NM firmware running on Intel® ME. All commands listed in this chapter are mandatory and will be implemented by Intel® NM firmware.

2.1 SEL Device Commands

Table 2-1 SEL Device Commands

Net Function = Storage (0Ah)			
Code	**Command**	**Request, Response Data**	**Description**
48h	Get SEL Time	Request None	This is standard IPMI 2.0 command. Intel® Node Manager firmware responds to this command returning internal clock value. Intel® Node Manager is synchronizing periodically its internal clock with system RTC. Intel® Node Manager is validating time read from system RTC. Valid time needs to be in range 1 January 2010 to 31 December 2079. In case Intel® Node Manager is not able to get valid time from system RTC FFFFFFFFh is returned as Present Timestamp value.
		Response Byte 1 – Completion Code = 00h – Success (Remaining standard Completion Codes are shown in Section 2.15) Bytes 2:5 - Present Timestamp value.	

2.2 IPMI Device “Global” Commands

Table 2-2 IPMI Device “Global” Commands

Net Function = App (06h) LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
02h	Cold Reset	Request None	This is standard IPMI 2.0 command. Reboots Intel® ME without resetting host platform. This command may not be effective when CPU is not present. Platform or global reset may be required to recover.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in Section 2.15)	
01h		Request	This is standard IPMI 2.0 command.

Net Function = App (06h) LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
	Get Device ID	None	Intel® Server Platform Services FW version is coded as follows: **Major Version Number** Byte 4 [6:0] – Major Firmware Revision, binary encoded. **Minor Version Number** Byte 5 – Minor Firmware Revision, BCD encoded: X.Y X maps to Minor Version Number **Milestone Version Number** Byte 5 – Minor Firmware Revision, BCD encoded: X.Y Y maps to Milestone Version Number **Build Version Number** Byte 14 – Firmware build number, BCD encoded: A.B Byte 15 [7:4] – Firmware build number, BCD encoded: C Build Version Number = 100*A + 10*B + C
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in Section 2.15) Byte 2 – Device ID =50h – Intel® Management Engine (Intel® ME) Byte 3 – Device Revision =0 – For Intel® NM, Silicon Enabling and Recovery boot-loader device does not provide Device SDRs [6:4] reserved. Return as 000b. [3:0] Device Revision, binary encoded. = 1 Byte 4 - Major Firmware Revision [7] Device available: =0 – normal operation =1 – device firmware update or self-initialization in progress or firmware in the recovery boot-loader mode [6:0] Major Firmware Revision, binary encoded = 5 Byte 5 - Minor Firmware Revision, BCD encoded. Byte 6 - IPMI Version. Holds IPMI Command Specification Version. BCD encoded. =00h - Reserved. [7:4] hold the Least Significant digit of the revision [3:0] hold the most significant digits. =02h to indicate revision 2.0. Byte 7 - Additional Device Support. Lists the IPMI 'logical device' commands and functions that the controller supports that are in addition to the mandatory IPM and Application commands. For Intel® NM and Silicon enabling SKU byte 7 is set to: [7] = 0 Not a chassis Device [6] = 0 Not a Bridge [5] = 1 IPMB Event Generator [4] = 0 Not a IPMB Event Receiver [3] = 0 Not a FRU Inventory Device [2] = 0 Not a SEL Device [1] = 0 Not a SDR Repository Device [0] = 1 Sensor Device	

Net Function = App (06h) LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
		If Recovery boot-loader image is loaded byte 7 is set to: [7] = 0 Not a chassis Device [6] = 0 Not a Bridge [5] = 1 IPMB Event Generator [4] = 0 Not a IPMB Event Receiver [3] = 0 Not a FRU Inventory Device [2] = 0 Not a SEL Device [1] = 0 Not a SDR Repository Device [0] = 0 Not a Sensor Device Bytes 8:10 – Manufacturer ID = 57h, 01h, 00h. Byte 11 – Product ID Minor Version =00h – Tylersburg platform =01h – Bromolow platform =02h – Romley platform =03h – Denlow platform =04h – Brickland platform =05h – Grantley platform =06h – Grangeville platform =07h – Groveport Self Boot platform =08h – Groveport Leverage Boot platform =09h – Greenlow platform =0Ah – Purley platform =0Bh – Harrisonville platform =0Ch – Purley-EPO platform =0Dh – Monte Vista platform =0Eh – Bakerville platform =0Fh – Whitley platform =10h – Mehlow platform Byte 12 – Product ID Major Version = 0Bh Bytes 13:16 – Auxiliary Firmware Revision Information Byte 13 – (Binary encoded) Implemented version of Firmware [7:4] Implemented DCMI version =0 – DCMI not implemented/enabled =1 – DCMI Revision 1.0 =2 – DCMI Revision 1.1 =3 – DCMI Revision 1.5 [3:0] Implemented Intel® Node Manager IPMI interface version =0 – Intel® NM not implemented/enabled =1 – Intel® NM Revision 1.5 =2 – Intel® NM Revision 2.0 =3 – Intel® NM Revision 3.0 =4 – Intel® NM Revision 4.0 =5 – Intel® NM Revision 5.0 Note: For Silicon Enabling SKU byte 13 will be set to all zeros. Byte 14 – Firmware build number, BCD encoded Byte 15 – Firmware build number and patch number, BCD encoded	

Net Function = App (06h) LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
		Byte 16 – Image flags [7:2] – reserved. Return as 000000b [1:0] – image type =00b – recovery image =01b – operational image 1 =10b – operational image 2 =11b – unspecified: flash error indication	
04h	Get Self-Test Results	Request None Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in Section 2.15) =D5h – Returned if self-tests is not finished yet. Byte 2 =55h – No error. All Self-Tests Passed. =56h – Self Test function not implemented in this controller. =57h – Corrupted or inaccessible data or devices. =58h – Fatal hardware error (system should consider BMC inoperative). This will indicate that the controller hardware (including associated devices such as sensor hardware or RAM) may need to be repaired or replaced, or that multiple software exceptions occurred. =80h – PSU Monitoring service error see Byte 3 for error description only if Intel® ME Firmware directly monitors PMBUS PSU. PMBUS PSU Monitoring service will return the current status of all defined PSUs on 'Get Self-Test Results' call. Note: The error code is continuously updated in runtime in S0/S1 host power states by the Monitoring Service. Additionally, the test will be performed in any host power state if Manufacturing Test On Command is issued. =81h – Firmware entered Recovery boot-loader mode =82h – HSC Monitoring service error see Byte 3 for error description only if Intel® ME Firmware directly monitors HSC. PMBUS HSC Monitoring service will return the current status of all defined HSCs on 'Get Self-Test Results' call. Note: The error code is continuously updated in runtime in S0/S1 host power states by the Monitoring Service. Additionally, the test will be performed in any host power state if Manufacturing Test On Command is issued. =83h – Firmware entered non-UMA restricted mode of operation =FFh – reserved.	This is standard IPMI 2.0 command. Note: In case the platform supports configuration with both HSC and PSU devices, reporting of HSC failures have precedence over the PSU errors.

Net Function = App (06h) LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
		Byte 3 For byte 2 = 55h, 56h, FFh: =00h For byte 2 = 58h: Exception type. For byte 2 = 57h: Self-test error bit field. [7] – SEL access error. [6] – SDR access error. [5] – FRU access error. [4] – Reserved. [3] – PIA access error. [2] – SDR repository empty. [1] – Firmware boot error. [0] – Operational Image or Factory Presets checksum error. For byte 2 = 80h: PSU monitoring error bit field, where each bit corresponds to one of the PSU's in order. If bit[N] is set to 1b PSU[N] is not responding. PSU order is set by factory presets. For byte 2 = 81h: byte 3 contains reason =00h – recovery entered due to recovery jumper being asserted =01h – recovery entered due to Security strap override jumper being asserted =02h – recovery mode entered by IPMI command "Force ME Recovery" =03h – Invalid flash configuration, either: - flash master access permissions are wrong - VSCC entry is missing or wrong - flash erase block size in Intel® ME region is wrong =04h – Intel® ME internal error Intel® ME could not start operational mode. For byte 2 = 82h: HSC monitoring error bit field, where each bit corresponds to one of the HSC's in order. If bit[N] is set to 1b HSC[N] is not responding. HSC order is set by factory presets. Bit[7]=1 may indicate some problem with at least one PSU For byte 2 = 83h: byte 3 contains reason =00h – 3 consecutive UMA errors =01h – DWR flow =02h – Pre-Go-S1 flow	

2.3 Sensor Device Commands

Table 2-3 Sensor Device Commands

Net Function = S/E (04h) LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
00h	Set Event Receiver	Request Byte 1 – Event Receiver Slave Address. = FFh disables Event Message Generation, Otherwise: [7:1] – IPMB (I2C) Slave Address [0] – always 0b when [7:1] hold I2C slave address Byte 2 [7:2] – reserved. Write as 000000b. [1:0] – Event Receiver LUN Note: Depending on the Factory preset “Default Event Receiver Address,” - if 00h is set in the factory presets Intel® ME Firmware will not send any event until Set Event Receiver command will be sent by BMC on platform startup from G3 or on Global Platform Reset (see definition in PCH EDS) - if 20h is set in the factory presets Intel® ME Firmware will not wait for BMC to send Set Event Receiver command before starting events generation. BMC can still use the command to regenerate all the active events.”	Note: Value set by Set Event Receiver command is not stored in the persistent storage so it should be send on each platform startup from G3 and on Global Platform Reset (see definition in PCH EDS). Note: This command is used by Intel® NM to determine if BMC starts properly. If this command would not be received in configured time special policy would be triggered to limit power. For more details, see Set Intel® NM Policy IPMI command description for trigger type Time After Host Reset.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in Section 2.15)	
01h	Get Event Receiver	Request None	
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in Section 2.15) Byte 2 – Event Receiver Slave Address. =FFh indicates Event Message Generation has been disabled, otherwise: [7:1] – IPMB (I2C) Slave Address. [0] – Always 0b when [7:1] hold I^2C slave address. Byte 3 [7:2] – Reserved. Return as 000000b. [1:0] – Event Receiver LUN.	
26h	Set Sensor Thresholds	For command description see [IPMI]	This is standard IPMI 2.0 command.
27h	Get Sensor Thresholds	For command description see [IPMI]	This is standard IPMI 2.0 command.

Net Function = S/E (04h) LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
28h	Set Sensor Event Enable	For command description see [IPMI]	This is standard IPMI 2.0 command.
29h	Get Sensor Event Enable	For command description see [IPMI]	This is standard IPMI 2.0 command.
2Ah	Re-arm Sensor Events	For command description see [IPMI]	This is standard IPMI 2.0 command.
2Bh	Get Sensor Event Status	For command description see [IPMI]	This is standard IPMI 2.0 command.
2Dh	Get Sensor Reading	For command description see [IPMI]	This is standard IPMI 2.0 command. Note: If sensor scanning is disabled for example using Flash Image Tool Get Sensor Reading command will return: - Completion Code 00h - last reading or 00h if there was not reading - and bit [6] of byte 3 set to 1.

2.4 Intel® ME Firmware Heartbeat Event

Intel® Server Platform Services firmware is capable of generating so called Heartbeat event to the BMC. The Heartbeat event is an IPMI message which is sent to the BMC periodically every second. The command code of the heartbeat message is configured by OEM using FITC.

2.4.1 Heartbeat over IPMI

Intel® Server Platform Services firmware can be configured to send to BMC the following IPMI command every 1 second.

Table 2-4 Heartbeat over IPMI command

Net Function = 30h LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
00h (May be enabled by setting different value in spsFITC)	OEM Intel® Server Platform Services Heartbeat	Request Bytes 1:4 – Intel® ME Firmware Status #1 Bytes 5:8 – Intel® ME Firmware Status #2 (The definition of Intel® ME Firmware Status #1 and Intel® ME Firmware Status #2 is provided in [ME_BIOS_INT])	This command is optional and may be implemented by the BMC. The Heartbeat via IPMI mechanism is supported in Operational mode only. In Recovery mode Intel® Server Platform Services Firmware shall not send the command to BMC. Note: Heartbeat over IPMI must be enabled in spsFITC for Intel® ME to send Heartbeat every 1 second.

Net Function = 30h LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in Section 2.15).	

2.5 IPMI OEM Device Commands

Table 2-5 IPMI OEM Device Commands

Net Function = 2Eh-2Fh LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
DDh	Set Intel® ME FW Capabilities	Request Byte 1:3 = Intel Manufacturer ID – 000157h, LS byte first. Byte 4:12 – Reserved. Should be set to 0. Byte 13 – Assist Modules [7:6] Performance Assist Module =00b – reserved =01b – disable =10b – reserved =11b – enable [5:4] RAS Assist Module =00b – reserved =01b – disable =10b – reserved =11b – enable [3:2] BIOS Assist Module =00b – reserved =01b – disable =10b – reserved =11b – enable [1:0] Power and Thermal (Intel® NM) Assist Module =00b – reserved =01b – disable =10b – reserved =11b – enable	This command can be used to disable Node Manager. This command requires a reset of Intel® ME subsystem in order to boot with new settings. Note: Performance Assist Module and BIOS Assist Module are not supported.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in Section 2.15). Byte 2:4 = Intel Manufacturer ID – 000157h, LS byte first. Byte 5:8 – Intel® ME FW version. The full version number has the following format: "[Major].[Minor].[Milestone].[ABC]" where ABC is the build firmware number. Byte 5 – Major FW version [7] Reserved	

Net Function = 2Eh-2Fh LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
		[6:0] [Major] binary encoded = 5 Byte 6 – Minor FW version [Minor] and milestone [Milestone] FW version BCD encoded Byte 7 – Build version [AB] part BCD encoded Byte 8 – Build revision [C] part and patch number BCD encoded. Byte 9 – IPMI version =01h – IPMI 1.0 =02h – IPMI 2.0 Other – reserved. Byte 10:11 – IPMI Message Size Supported (bytes). Value includes encapsulation. Byte 12 – Intel® ME FW Update and State Control Version =01h – v1.0 =02h – v2.0 Other – reserved. Byte 13 – Proxies supported by Intel® ME FW [7:6] – MIC Discovery =00b – not supported =01b – supported, not enabled =10b – reserved =11b – supported and enabled [4:5] – IPMB Proxy (aka MIC Proxy) =00b – not supported =01b – supported, not enabled =10b – reserved =11b – supported and enabled [2:3] – PMBUS proxy =00b – not supported =01b – supported, not enabled =10b – reserved =11b – supported and enabled [1:0] – PECI proxy =00b – not supported =01b – supported, not enabled =10b – reserved =11b – supported and enabled	

Net Function = 2Eh-2Fh LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
		Byte 14 – Assist Modules [7:6] Performance Assist Module =00b – not supported =01b – supported, not enabled =10b – reserved =11b – supported and enabled [5:4] RAS Assist Module =00b – not supported =01b – supported, not enabled =10b – reserved =11b – supported and enabled [3:2] BIOS Assist Module =00b – not supported =01b – supported, not enabled =10b – reserved =11b – supported and enabled [1:0] Power and Thermal (Intel® NM) Assist Module =00b – not supported =01b – supported, not enabled =10b – reserved =11b – supported and enabled	
DEh	Get Intel® ME FW Capabilities	Request Byte 1:3 = Intel Manufacturer ID – 000157h, LS byte first. Byte 4:7 – Reserved. Should be set to 00000000h Byte 8 – IPMI version supported by BMC =01h – IPMI 1.0 =02h – IPMI 2.0 Other – reserved. Byte 9:10 – IPMI Message Size supported by BMC (in bytes). The value includes encapsulation. Byte 11 – Proxies supported by BMC [7:6] – MIC Discovery =00b – not supported =01b – supported, not enabled =10b – reserved =11b – supported and enabled. [4:5] – IPMB Proxy (aka MIC Proxy) =00b – not supported =01b – supported, not enabled =10b – reserved =11b – supported and enabled [2:3] – PMBUS proxy =00b – not supported =01b – supported, not enabled =10b – reserved =11b – supported and enabled [1:0] – PECI proxy =00b – not supported =01b – supported, not enabled =10b – reserved =11b – supported and enabled	Note: This command returns a current working SKU info. It does not provide information about the SKU which will be active after next reset. Note: Performance Assist Module and BIOS Assist Module are not supported.
		Response	

Net Function = 2Eh-2Fh LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
		Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in Section 2.15). Byte 2:4 = Intel Manufacturer ID – 000157h, LS byte first. Byte 5:8 – Intel® ME FW version. The full version number has the following format: "[Major].[Minor].[Milestone].[ABC]" where ABC is the build firmware number. Byte 5 – Major FW Revision [7] Reserved [6:0] [Major] binary encoded = 5 Byte 6 – Minor FW version [Minor] and milestone [Milestone] FW version BCD encoded Byte 7 – Firmware build number [AB] part BCD encoded Byte 8 – Firmware build number [C] part and patch number BCD encoded Byte 9 – IPMI version =01h – IPMI 1.0 =02h – IPMI 2.0 Other – reserved. Byte 10:11 – IPMI Message Size Supported (bytes). Value includes encapsulation. Byte 12 – Intel® ME FW Update and State Control Version =01h – v1.0 =02h – v2.0 Other – reserved. Byte 13 – Proxies supported by Intel® ME FW [7:6] – MIC Discovery =00b – not supported =01b – supported, not enabled =10b – reserved =11b – supported and enabled [4:5] – IPMB Proxy (aka MIC Proxy) =00b – not supported =01b – supported, not enabled =10b – reserved =11b – supported and enabled [2:3] – PMBUS proxy =00b – not supported =01b – supported, not enabled =10b – reserved =11b – supported and enabled [1:0] – PECI proxy =00b – not supported =01b – supported, not enabled =10b – reserved = 11b – supported and enabled	

Net Function = 2Eh-2Fh LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
		Byte 14 – Assist Modules [7:6] Performance Assist Module =00b – not supported =01b – supported, not enabled =10b – reserved =11b – supported and enabled [5:4] RAS Assist Module =00b – not supported =01b – supported, not enabled =10b – reserved =11b – supported and enabled [3:2] BIOS Assist Module =00b – not supported =01b – supported, not enabled =10b – reserved =11b – supported and enabled [1:0] Power and Thermal (Intel® NM) Assist Module =00b – not supported =01b – supported, not enabled =10b – reserved =11b – supported and enabled	
DFh	Force Intel® ME Recovery	Request Byte 1:3 = Intel Manufacturer ID – 000157h, LS byte first. Byte 4 – Command =01h Restart using Recovery Firmware (Intel® ME FW configuration is not restored to factory defaults) =02h Restore Factory Default Variable values and restart the Intel® ME FW =03h PTT Initial State Restore	With parameter Command = 01h Intel® ME FW resets and prevents loading the regular operational FW code – Intel® ME FW stops in Recovery Boot Loader. After issuing this command, the direct FW update is available even after End-of-POST reception. With parameter Command = 02h Intel® ME FW restores all variables stored in nonvolatile memory to its factory defaults (as set using FIT in the factory). This requires two Intel® ME FW resets: Intel® ME FW first resets and temporarily stays in the recovery boot loader code while the factory defaults are restored, and then another FW reset happens and Intel® ME FW comes back with the factory settings restored. Note: Intel® ME FW will always stay in recovery boot loader for other reasons, such as recovery jumper asserted or security strap override asserted.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in Section 2.15). =81h – Unsupported Command parameter value in the Byte 4 of the request. Byte 2:4 = Intel Manufacturer ID – 000157h, LS byte first.	
E0h	Get Intel® ME Factory Presets Signature	Request Byte 1:3 = Intel Manufacturer ID – 000157h, LS byte first.	

Net Function = 2Eh-2Fh LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in Section 2.15). Byte 2:4 = Intel Manufacturer ID – 000157h, LS byte first. Byte 5:8 – Vendor Label 4 bytes long data assigned by platform vendor at Intel® ME image creation time. Byte 9:N – Factory defaults signature of length between 1..32 bytes.	The signature length is Intel® ME Firmware implementation specific and fixed for given Intel® ME firmware release, but may be different in subsequent releases. Bytes 5:N exist in response frame only if Completion Code (Byte 1) = 00h.
E1h	Set Intel® ME FW USB Port Override	Request Byte 1:3 = Intel Manufacturer ID – 000157h, LS byte first. Byte 4 – USB Port Identifier. Byte 5 – USB Port Override Action. =00h – Disable USB port. =01h – Enable USB port. =02h – FFh – Reserved.	This command can be used to enable or disable USB port(s). This command requires a reset of host subsystem in order to take effect. **Compatibility Note:** On Mehlow platform up to 12/14 USB 2.0 ports for C242/C246 Intel Cannon Lake PCH SKUs respectively are supported. The Mehlow platform supports also up to 6/10 USB 3.x ports. Typically USB 3.x port identifiers are assigned after the last USB 2.0 port. Note: port numbers: 13-14 and 15-16 for C242/C246 respectively are reserved on Mehlow server platform, which means that USB3.x ports are expected to start from #15 for C242 and #17 for C246. For details on port numbering scheme and general USB port topology, refer to [PDG] and [PCH EDS].
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in Section 2.15). Byte 2:4 = Intel Manufacturer ID – 000157h, LS byte first.	
E6h	Get Exception Data	Request Byte 1:3 = Intel Manufacturer ID – 000157h, LS byte first. Byte 4 – Dump log part index =00h – use it for first time	This command retrieves exception data which may be helpful in resolving the reason for an Intel® ME Firmware Exception. Thus sequence of this commands should be issued to get full data from Exception Log – starting from dump log part index 00h until you get FFh in index of next dump log in response. Note: The whole IPMB response frame may be up to 80 bytes long (including all protocol overhead – addresses, checksums, header) but may be shorter as well.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in Section 2.15). Byte 2:4 = Intel Manufacturer ID – 000157h, LS byte first. Byte 5:N-1 – Exception data (this field may not exist if no exceptions were recorded). Byte N – Index of next dump log part index =FFh – no more parts	

Net Function = 2Eh-2Fh LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
E7h	Unlock ME Region	Request Byte 1:3 = Intel Manufacturer ID – 000157h, LS byte first. Byte 4 – Flash Access Request [0] – ME Region Lock/Unlock Request =0 – Lock ME Region =1 – Unlock ME Region [7:1] – Reserved	This command is used to request SPI Flash Master write/read access to the ME region. Access will be granted only to the master associated with the request: IE or BMC. This command is available only in recovery mode.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in Section 2.15). =D1h – ME is already in FW Update mode. This may be due to upgrade initiated on another interface. =D5h – Command not supported in present state. This indicates flash is already in requested state or another agent (IE/BMC/Host) has been granted access to ME region. Byte 2:4 = Intel Manufacturer ID – 000157h, LS byte first. Byte 5 – Flash Access Response [0] – ME Region Request Status =0 – ME Region Locked =1 – ME Region Unlocked [7:1] – Reserved	
E8h	Restart MCTP Discovery	Request Byte 1:3 = Intel Manufacturer ID – 000157h, LS byte first.	This command restarts discovery of MCTP endpoints by INTEL® SERVER PLATFORM SERVICES FW as MCTP bus owner. All previously discovered endpoints will be removed from bus owner routing table. INTEL® SERVER PLATFORM SERVICES FW MCTP driver will execute full MCTP endpoint discovery process, starting with sending "Prepare for Endpoint Discovery" command, which will cause endpoints to reset internal Discovery Flag.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in Section 2.15). =D5h – MCTP discovery cannot be executed since system is not in S0 state =FFh – Unspecified Error Byte 2:4 = Intel Manufacturer ID – 000157h, LS byte first.	

2.6 IPMI Intel® Platform Trusted Technology (Intel® PTT) Status Reporting Commands

The IPMI Intel® PTT commands allow to check the Intel® PTT status, health or force ME Recovery.

Table 2-6 IPMI PTT Status Reporting Commands

Net Function = 2Eh-2Fh LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
70h	Get Intel® Platform Trust Technology Version	Request Byte 1:3 – Intel Manufacturer ID – 000157h, LS byte first.	Get Intel® Platform Trust Technology firmware version. Major Firmware revision and Minor Firmware revision unambiguously identify firmware release. For every release, at least one of these numbers changes. This command would be executed regardless of responder LUN value. Common setting is returned for all LUNs.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in Section 2.15). Byte 2:4 – Intel Manufacturer ID – 000157h, LS byte first. Byte 5 – Intel® PTT version =01h – Supported Intel® PTT gen 3.0 =02h – FFh – Reserved for future use. Byte 6 – IPMI interface version =01h – Intel® PTT IPMI version 1.0 Byte 7 – Security Version Number (SVN) =01h – FEh – Security Version Number (SVN = 1 - 254). =FFh – Not available in the current state, try again. Byte 8 – Patch version (binary encoded). Byte 9 – Major Firmware revision (binary encoded) – identifies current build of the code –and should contain the same value as the Get Device Id command's response byte 4 [6:0] – Major firmware revision. Byte 10 – Minor Firmware revision (BCD encoded) – identifies current build of the code and should contain the same value as the "Get Device Id" command response byte 5 Minor firmware revision.	
71h	Get Intel® Platform Trust Technology Capabilities	Request Byte 1:3 – Intel Manufacturer ID – 000157h, LS byte first.	Get Intel® Platform Trust Technology Capabilities. This command would be executed regardless of responder LUN value. Common setting is returned for all LUNs.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in Section 2.15). Byte 2:4 – Intel Manufacturer ID – 000157h, LS byte first. Byte 5 – Intel PTT Support [7:6] =00b – Not supported =01b – Supported but not enabled =10b – Reserved =11b – Supported and enabled [5:0] – Reserved	
72h	Get Intel® Platform Trust Technology Health	Request Byte 1:3 – Intel Manufacturer ID – 000157h, LS byte first.	Get Intel® Platform Trust Technology Health.

Net Function = 2Eh-2Fh LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in Section 2.15). Byte 2:4 – Intel Manufacturer ID – 000157h, LS byte first. Byte 5 – Intel® PTT Health =00h – PTT is operational. =01h – PTT is in FAILURE_MODE. =02h – PTT is in FAILURE_MODE due to ME being in recovery. =03h – PTT is in FAILURE_MODE due to SVN downgrade. =04h – FEh – Reserved. =FFh – PTT internal error. Byte 6:70 – PTT in FAILURE_MODE root cause log, only full entry will be stored in the log - up to 8 entries. To decode particular record, see error code definition for TPM2_GetTestResult command in TPM2.0 Spec Part 3.	This command would be executed regardless of responder LUN value. Common setting is returned for all LUNs.

2.7 IPMI Boot Guard Status Reporting Commands

The IPMI BtG commands allow to check the boot guard version, health and capabilities.

Table 2-7 IPMI Boot Guard Status Reporting Commands

Net Function = 2Eh-2Fh LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
80h	Get Boot Guard Version	Request Byte 1:3 – Intel Manufacturer ID – 000157h, LS byte first.	Get Intel® Boot Guard firmware version. Major Firmware revision and Minor Firmware revision unambiguously identify firmware release. For every release, at least one of these numbers changes. This command would be executed regardless of responder LUN value. Common setting is returned for all LUNs.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in Section 2.15). Byte 2:4 – Intel Manufacturer ID – 000157h, LS byte first. Byte 5 –BtG version =01h – Supported Boot Guard 1.0 =02h – FFh – Reserved for future use. Byte 6 – IPMI interface version =01h – Boot Guard IPMI version 1.0 (version defined in this document). Byte 7 – Patch version (binary encoded).	

Net Function = 2Eh-2Fh LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
		Byte 8 – Boot Guard ACM Security Version Number (ACMSVN). Byte 9 – Major Firmware revision (binary encoded) – identifies current build of the code –and should contain the same value as the Get Device Id command's response byte 4 [6:0] – Major firmware revision. Byte 10 – Minor Firmware revision (BCD encoded) – identifies current build of the code and should contain the same value as the "Get Device Id" command response byte 5 Minor firmware revision.	
81h	Get Intel® Boot Guard Capabilities	Request Byte 1:3 – Intel Manufacturer ID – 000157h, LS byte first.	Get Intel® Boot Guard Capabilities. This command would be executed regardless of responder LUN value. Common setting is returned for all LUNs.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in Section 2.15). Byte 2:4 – Intel Manufacturer ID – 000157h, LS byte first. Byte 5 – Boot Guard Support [7:6] =00b – Not supported =01b – Supported but not enabled =10b – Reserved =11b – Supported and enabled [5:0] – Reserved.	
82h	Get Intel® Boot Guard Health	Request Byte 1:3 – Intel Manufacturer ID – 000157h, LS byte first.	Get Intel® Platform Trust Technology Health. This command would be executed regardless of responder LUN value. Common setting is returned for all LUNs.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in Section 2.15). Byte 2:4 – Intel Manufacturer ID – 000157h, LS byte first.	

Net Function = 2Eh-2Fh LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
		Byte 5 – Intel® Boot Guard Health [0] – Enforcement Flow. This bit set indicates that Intel® ME FW has started enforcement. [1] – Sx Resume Type =0b – Platform resuming from S4 or S5 =1b – Platform resuming from S3 or deep S3 [2] – All TPMs Disconnected. This bit set indicates Intel® ME FW has executed TPM disconnection flow. It happens on ACM Active timeout. [3] – Boot Guard Self-Test. This bit set indicates that profile loaded from FPF/NVAR is correct. [4] – Boot Guard ACM Active STS (ACM_ACTIVE). This bit set indicates that ACM is executing. [5] – Result Code Source (RCS). This bit indicates the source of the value in ESC. =0b – Source is ACM =1b – Source is CPU [6] – Boot Guard ACM DONE STS (ACM_DONE). This bit set indicates completion of ACM. [7] – Reserved. Byte 6 [0:6] – Startup Module Timeout Count (SMTC). These bits provide the Protect BIOS Environment Timer (PBET) timeout value. This timer is effective only when ACM DONE STS bit is set and Enforcement Flow bit is not set. [7] – S-CRTM Indicator. This bit set indicates TPM commands are from locality 3. When set Intel® PTT acts on any TPM2 commands as coming from locality 3. Byte 7 [0:3] – Increment Boot Guard ACM Security Version Number (INC_ACM_SVN). Contains the new value of ACM SVN (Security Version Number) count. If there is no increment required, this field shall be set to 0 by BTG ACM. [4:7] – Increment Key Manifest Security Version Number (INC_KM_SVN). Contains the new value of the Key Manifest SVN (Security Version Number) count. If there is no increment required this field shall be set to 0 by BTG ACM. Byte 8 [0:3] – Increment Boot Policy Manifest Security Version Number (INC_BPM_SVN). Contains new value of the Boot Policy Manifest SVN (Security Version Number) count. If there is no increment required this field shall be set to 0 by BTG ACM.	

Net Function = 2Eh-2Fh LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
		[4] – Start Enforcement. This bit indicates error during verification, ACM requests Enforcement. =0b – No action =1b – PCH starts the error enforcement logic [5] – Force Boot Guard ACM Boot Policy (F). This bit indicates that ACM must run. [6] – CPU Debug Disabled. This bit set indicates that the platform shall disable CPU debug mode. =0b – Platform CPU debug mode enabled =1b – Platform CPU debug mode disabled [7] – BSP Initialization Disabled. This bit set indicates that BSP will shut down when it receives CPU INIT signal. Byte 9 [0:4] – Error Status Code (ESC). Provides error condition passed from ACM to Intel® Intel® Server Platform Services firmware. =00h – Success =01h – Boot Guard initialization failed =02h – NEM (non-eviction mode) setup failed =03h – KM (key manifest) or BPM (boot policy manifest) verification failed =04h – IBB (initial boot block) measurement failed. [5] – Protect BIOS Environment Policy (PBE) Status. If PBE policy is enabled Boot Guard ACM is going to set up the protected environment for BIOS Initial Boot Block (IBB) execution. =0b – Take no action to control the environment during the execution of the BIOS component. =1b – Take action to control the environment during the execution of the BIOS component. [6:7] – Error Enforcement Policy (ENF). Reports enforcement configuration type that is used. The system will follow ENF policy when Boot Guard ACM encounters a fatal error. =00h – System will not shut down due to a Boot Guard ACM failure. =03h – System will shut down in 30minutes due to a Boot Guard ACM failure.	

Net Function = 2Eh-2Fh LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
		Byte 10 [0] – Measured Boot Policy (M). This bit set indicates that platform performs Measured boot. [1] – Verified Boot Policy (V). This bit set indicates that platform performs Verified boot. [2:5] – Boot Guard ACM Security Version Number (ACMSVN). Provides Boot Guard ACM revocation value. This value is Field Programmable Fuses (FPF) data. Once the platform has been configured with the updated Boot Guard ACM, then the platform will not accept a Boot Guard ACM with a lower Security Version Number. [6:7] – Reserved Byte 11 [0:3] – Key Manifest Security Version Number (KMSVN). Provides the key manifest revocation value. This value is Field Programmable Fuses (FPF) data. Once the platform has been configured with the updated KM, then the platform will not accept a KM with a lower Security Version Number (SVN). [4:7] – Boot Policy Manifest Security Version Number (BPMSVN). Provides the BIOS policy manifest revocation value. This value is Field Programmable Fuses (FPF) data. Once the platform has been configured with the updated BPM, then the platform will not accept a BPM with a lower Security Version Number (SVN). Byte 12 [0:3] – Key Manifest ID (KMID). Provides the hash of another public key used by the Boot Guard ACM to verify the Boot Policy Manifest. This value is Field Programmable Fuses (FPF) data. [4] – BSP Boot Policy Manifest Execution Status. This bit set indicates that Boot Policy Manifest execution is completed on BSP. [5] – PBE Error. This bit set indicates that CPU encountered an unexpected error and is asking Intel® Server Platform Services firmware to start the enforcement logic. Intel® Server Platform Services firmware will check this ERROR bit at the time of receiving the Startup Module (SM) completion on BSMES bit [26]. [6] – Boot Guard Disable. This bit set indicates that BTG is disabled in PCH fuse (not FPF). [7] – Reserved Byte 13 [0:7] – Reserved	

2.8 IPMI ME Storage Services Commands

The IPMI ME Storage Services commands enable saving crucial configuration or provisioning data needed to initialize and configure the platform or platform functionalities. After reset that configuration data can be retrieved to initialize the platform with the same settings as before a reset occurred.

Table 2-8 IPMI ME Storage Services commands

Net Function = 2Eh-2Fh LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
90h	ME Storage Services Read	Request Byte [1:3] – Intel Manufacturer ID – 000157h, LS byte first. Byte 4 – Storage Entry Identifier. =00h – TXT Provisioning Data. =01h-FFh – Reserved for future use. Byte [5:7] – Offset within Storage Entry of data to be read. Byte 8 – Length of data to be read.	This command can be used for retrieving data stored within Intel® ME non-volatile memory. For data size larger than 68 bytes the command should be issued multiple times with proper offset value. This command accesses the flash, so the response will be sent after completing the operation and may take longer than 250 ms. If length of data to be read (Byte 8) + Offset (Byte [5:7]) is longer than actual stored data the command will return only the existing number of bytes. Data will not be padded.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in Section 2.15). =80h – Operation refused. After this error code the operation should be repeated. =81h – Command Failed. Flash error. After this error code the operation should be repeated. =82h – An attempt to access an empty entry. =84h – Cannot access storage entry after End of POST (EOP). Repair action: verify storage entry Attributes. =C9h – Parameter out of range. Storage Entry Index or Offset Out Of Range. Byte [2:4] = Intel Manufacturer ID – 000157h, LS byte first. If Byte 1 Completion Code is Success (00h): Byte 5:N – Read data – where <N> should not exceed 72. Maximum supported number of data bytes is 68.	
91h	ME Storage Services Write	Request Byte [1:3] – Intel Manufacturer ID – 000157h, LS byte first. Byte 4 – Storage Entry Identifier =00h – TXT Provisioning Data. =01h-FFh – Reserved for future use. Byte 5:7 – Offset within Storage Entry in bytes	This command can be used for writing data into Intel® ME non-volatile memory. For data larger than 68 bytes the command should be issues multiple times with proper offset value. The last frame should have Message Complete bit set.

Net Function = 2Eh-2Fh LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
		Byte 8– Attributes [0] – Message Complete. This bit is used to indicate that this is the last message of a multi message transfer for message that is larger than the single IPMI frame (80 bytes). If this bit is 0, at least one other message is expected to complete the requestor message transfer. If this bit is 1, this message completes the transfer of a requestor message. For messages that can be transmitted in a single message, this bit will be set in that single message. [1:7] – Reserved. Write as 0000000b. Byte 9:N – Storage Entry Data – where <N> should not exceed 72. Maximum supported number of data bytes is 64. If Byte 8 – Attributes has Message Complete bit set to 1b Byte [9:12] – Total length of data written to given Storage Entry. Data start in this case from bytes 13. For such frame maximum payload is 60 bytes long.	This command accesses the flash, so the response will be sent after completing the operation and may take longer than 250 ms.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in Section 2.15). =80h – Operation refused. After this error code the operation should be repeated. =81h – Write Failed. Flash error. After this error code the operation should be repeated. =85h – Cannot write storage entry after End of POST (EOP). =C9h – Parameter out of range. Storage Entry Index or Offset Out Of Range. Byte [2:4] = Intel Manufacturer ID – 000157h, LS byte first. If this is a response for last frame i.e. if Byte 8 from the request – Attributes has Message Complete bit set to 1b. Byte 5:N – Hash of written data (SHA256).	

2.9 IPMI Device "Global" Sensors

The following IPMI sensors are always exposed by Intel® ME FW:

- Intel® ME Power State
- Intel® ME Firmware Health

Detailed description of these sensors is provided in the subsections and in B.3.

Reading Availability column specifies when the sensor reading is available:

- A – Always when Intel® ME is On
- H – When HOST CPU is On
- O – After reception of END_OF_POST notification
- E – No reading available (Event Only)

Defaults Configurable in FIT column defines whether the default configuration of the sensors can be set using Flash Image Tool. The default configuration includes:

- Thresholds
- Event Enable Mask
- Scanning Periods
- Scanning Enable Flag
- Per-sensor Event Enable Flag

Table 2-9 IPMI Device "Global" Sensors

Sensor #	Description	Firmware SKU Availability	Reading Availability	Defaults Configurable in Flash Image Tool
22	Intel® ME Power State	A	E	Yes
23	Intel® ME Firmware Health	A	E	No
8	PCH Thermal Sensor	A	H*	Yes
197	Host Partition Reset Warning Sensor	A	A	Yes

* Host CPU must configure this sensor to be functional

2.9.1 Intel® ME Power State sensor

Use the sensor to send Platform Event messages to BMC when Intel® ME power state is changing. The sensor uses Generic Event Reading code 0Ah. It supports only the offsets:

00h – Transition to Running – Intel® ME is started

02h – Transition to Power Off – Intel® ME is powered down

Note: Optionally, instead or in addition to the event, Intel® ME Firmware may send an IPMI command with power state change notification as defined in Section 4.4.2. Using Factory presets OEM may choose to use an event or OEM command or both.

2.9.2 Intel® ME Firmware Health Sensor

Use the sensor in Platform Event messages to BMC containing health information including but not limited to FW Upgrade and application errors.

2.9.3 PCH Thermal Sensor

This sensor provides the die temperature sensor value in Celsius degrees. This sensor is only available when the host is in S0. Retrieving PCH temperature while not in S0 will result in receiving the CBh completion code.

Note: BIOS must configure the PCH Thermal sensor in order to allow Intel® Server Platform Services Firmware collect temperature data.

2.9.4 Host Partition Reset Warning Sensor

This sensor sends an event to inform the BMC about an upcoming Platform Reset. The BMC can retrieve from the event the value of time by which ME will delay the reset. If Intel® ME does not receive the response for the event, it assumes that BMC is not functional and will not delay the reset, that is, it will proceed with the reset immediately after all retransmissions of the event completes. Depending on Intel® ME image configuration Intel® ME will send this event only when IERR is detected in one of the CPUs or always during Host Partition reset or Sx transition, this may increase the time needed for host to reset.

If the BMC is ready for the reset before the delay time expires, it can inform the Intel® ME about it by re-arming the Host Partition Reset Warning Sensor. In such a case Intel® ME will allow for the reset immediately.

Optionally BMC can read the Host Partition Reset Warning Sensor to learn what time Intel® ME is configured to delay the reset.

2.9.5 Event Messages Definition

Table 2-10 Event Messages Definition

Net Function = S/E (04h) LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
02h	Platform Event Message Intel® ME Power State	Request Byte 1 – EvMRev =04h (IPMI2.0 format) Byte 2 – Sensor Type =16h (microcontroller) Byte 3 – Sensor Number =22 – Intel® ME Power State Byte 4 – Event Dir | Event Type [7] – Event Dir =0 – Assertion Event. [6:0] – Event Type =0Ah – Availability Status. Byte 5 – Event Data 1 [7:6] = 00b – unspecified byte 2 [5:4] = 00b – unspecified byte 3 [3:0] – offset from event type code: =00h – Transition to Running =02h – Transition to Power Off	Note: OEM can select using Flash Image Tool that the notification is sent using OEM command instead of Platform Event Message. See Section OEM Management Engine Power State Change for definition of the OEM command.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in Section 2.15)	
02h	Platform Event Message	Request Byte 1 – EvMRev =04h (IPMI2.0 format)	

Net Function = S/E (04h) LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
	Intel® ME Firmware Health Event	Byte 2 – Sensor Type =DCh (OEM) Byte 3 – Sensor Number =23 – Intel® ME Firmware Health Sensor Byte 4 – Event Dir | Event Type [7] – Event Dir =00h – Assertion Event. [6:0] – Event Type =75h (OEM) Byte 5 – Event Data 1 [7:6] = 10b – OEM code in byte 2 [5:4] = 10b – OEM code in byte 3 [3:0] – Health Event Type =00h – Firmware Status. =01h - SMBus link failure For Event Data 1 Firmware Status (Health Event Type = 00h)	This platform event provides a run-time status of general Firmware status. Recovery from the errors may require Intel® ME reset or even FW upgrade or HW repair if the error is persistent. Note: This sensor cannot be disabled using Factory Image Tool.

		Byte 6 – Event Data 2 =00h – Recovery GPIO forced. Recovery Image loaded due to recovery MGPIO pin asserted. Pin number is configurable in factory presets, Default recovery pin is MGPIO1. Repair action: Deassert MGPIO1 and reset the Intel® ME =01h – Image execution failed. Recovery Image or backup operational image loaded because operational image is corrupted. This may be either caused by Flash device corruption or failed upgrade procedure. Repair action: Either the Flash device must be replaced (if error is persistent) or the upgrade procedure must be started again. =02h – Flash erase error. Error during Flash erasure procedure probably due to Flash part corruption. Repair action: The Flash device must be replaced. =03h – Flash state information. Repair action: Check extended info byte in Event Data 3 (byte 7) whether this is wear-out protection causing this event. If so just wait until wear-out protection expires, otherwise probably the flash device must be replaced (if error is persistent). =04h – Internal error. Error during firmware execution – FW Watchdog Timeout. Repair action: Firmware should automatically recover from error state. If error is persistent then operational image shall be updated or hardware board repair is needed. =05h – BMC did not respond correctly to Chassis Control - Power Down command triggered by Intel® Node Manager policy failure action and Intel® ME forced shutdown. Repair action: Verify the Intel® Node Manager policy configuration. =06h – Direct Flash update requested by the BIOS. Intel® ME Firmware will switch to recovery mode to perform full update from BIOS. Repair action: This is transient state. Intel® ME Firmware should return to operational mode after successful image update performed by the BIOS. =07h – Manufacturing error. Wrong manufacturing configuration detected by Intel® ME Firmware. Repair action: If error is persistent the Flash device must be replaced or FW configuration must be updated. =08h – Automatic Restore to Factory Presets Repair action: If error is persistent the Flash device must be replaced. =09h – Firmware Exception. Repair action: Restore factory presets using "Force ME Recovery" IPMI command or by doing AC power cycle with Recovery jumper asserted. If this does not clear the issue, reflash the SPI flash. If the issue persists, provide the content of Event Data 3 to Intel support team for interpretation. (Event Data 3 codes are not documented because they only provide clues that must be interpreted individually.) =0Ah – Flash Wear-Out Protection Warning. Warning threshold for number of flash operations has been exceeded. Repair action: No immediate repair action needed. This is just a warning event. =0Dh – PECI over DMI interface error. This is a notification that PECI over DMI interface failure was detected and it is not functional any more. It may indicate the situation when PECI over DMI was not configured by BIOS or a defect which may require a CPU Host reset to recover from. Repair action: Recovery via CPU Host reset or platform	Multiple SMBus link failure events are sent if the time between the failures exceeds the throttling timeout defined by spsFITc MS Configuration.

<table>
<tr><th colspan="4">Net Function = S/E (04h) LUN = 00b</th></tr>
<tr><th>Code</th><th>Command</th><th>Request, Response Data</th><th>Description</th></tr>
<tr><td></td><td></td><td>reset.
=0Eh – MCTP interface error. This is a notification that MCTP interface failure was detected and it is not functional any more. It may indicate the situation when MCTP was not configured by BIOS or a defect which may need a Host reset to recover from.
Repair action: Recovery via CPU Host reset or platform reset.
=0Fh – Auto-configuration finished. Operational image finished power source auto-configuration.
Repair action: Auto-configuration could be enforced by restore to factory defaults.
=10h - Unsupported Segment Defined Feature. Feature not supported in current segment detected by Intel® ME Firmware.
Repair action: Proper FW configuration must be updated or use the Flash device with proper FW configuration.
=11h – Reserved
=12h – CPU Debug Capability Disabled.
=13h – UMA operation error. This is a notification that UMA was not initialized correctly during POST or error occurred while copying page to/from UMA.
It may indicate situations when BIOS did not grant memory for UMA, granted memory size differs from requested, checksum of copied page differs from expected or timeout occurred during copying data to/from UMA.
Repair action: Platform reset when UMA not configured correctly, or when error occurred during normal operation on correctly configured UMA multiple times leading to ME entering Recovery or restricted operation mode.
=14h-15h – Reserved – Debug Message
=16h – PTT Health Event
=17h – Boot Guard Health Event
=18h – Restricted mode info. Firmware entered restricted mode due to error conditions met, or exited restricted mode due to ME reset or entering recovery mode.
=19h – multiPCH mode misconfiguration.
=1Ah – Flash Descriptor Region Verification Error. Repair action: Flash Descriptor Region must be created correctly.
=1Bh-FFh – Reserved
Byte 7 – Event Data 3
Extended error info. Should be used when reporting an error to the support. The following values can be interpreted directly:
For Event Data 2 (byte 6) equal to 03h
=00h – flash partition table, recovery image or factory presets image corrupted
=01h – flash erase limit has been reached
=02h – flash write limit has been reached, writing to flash has been disabled
=03h – writing to the flash has been enabled</td><td></td></tr>
</table>

Net Function = S/E (04h) LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
		For Event Data 2 (byte 6) equal to 07h =00h – Generic error =01h – Wrong or missing VSCC table =02h – Wrong sensor scanning period in PIA =03h – Wrong device definition in PIA =04h – Reserved (Wrong SMART/CLST configuration) =05h – Intel® ME FW configuration is inconsistent or out of range =06h – Reserved =07h – Intel® ME FW configuration is corrupted =08h – SMLink0/0B misconfiguration For Event Data 2 (byte 6) equal to 0Ah = percentage of flash write operations which have been conducted For Event Data 2 (byte 6) equal to 0Dh =01h – DRAM Init Done HECI message not received by ME before EOP =02h – System PCIe bus configuration not known or not valid on DID HECI message arrival to ME =03h – PECI over DMI run-time failure For Event Data 2 (byte 6) equal to 0Eh =01h – No DID HECI message received before EOP =02h – No MCTP_SET_BUS_OWNER HECI message received by ME on EOP arrival to ME, while MCTP stack is configured in Bus Owner Proxy mode For Event Data 2 (byte 6) equal to 0Fh [7] – Auto-configuration result =0b – Success =1b – Failure if bit 7 reports Success (0b) then the other bits are defined as follows; [6:5] – DC Power source =00b – BMC =01b – PSU =10b – On-board power sensor =11b – reserved [4:3] – Chassis Power input source =00b – BMC =01b – PSU =10b – On-board power sensor/ PSU efficiency =11b – not supported [2:1] – PSU efficiency source =00b – BMC =01b – PSU =10b – reserved =11b – not supported [0] – Unmanaged power source =0b – BMC =1b – estimated if bit [7] reports failure (1b) then the other bits are defined as follows; [6:5] – Failure =00b – BMC discovery failure =01b – Insufficient factory configuration =10b – Unknown sensor type =11b – Other error encountered [4:0] – Reserved	

Net Function = S/E (04h) LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
		For Event Data 2 (byte 6) equal to 10h =00h – Other Segment Defined Feature =01h – Fast NM limiting =02h – Volumetric Airflow and Outlet Temperature =03h – CUPS =04h – Thermal policies and Inlet Temperature =05h – Platform limiting with MICs =07h – Shared power supplies =08h – MIC Proxy =09h – Reset warning =0Ah – PMBus Proxy =0Bh – Always on =0Ch – IPMI ME FW update =0Dh – MCTP bus owner =0Eh – MCTP bus owner proxy =0Fh – Dual BIOS =10h – Battery less =11h – PCH Temperature =12h – Intel® ME Shutdown at End of POST =13h – USB Connect Status For Event Data 2 (byte 6) equal to 13h =00h – UMA Read integrity error. Checksum of data read from UMA differs from expected one. =01h – UMA Read/Write timeout. Timeout occurred during copying data from/to UMA. =02h – UMA not granted. BIOS did not grant any UMA or DRAM INIT done message was not received from BIOS before EOP. ME FW goes to recovery. =03h – UMA size granted by BIOS differs from requested. ME FW goes to recovery. For Event Data 2 (byte 6) equal to 16h =00h – PTT disabled (PTT region is not present). =01h – PTT downgrade (PTT data should be not available). =02h – PTT disabled (battery less configuration). For Event Data 2 (byte 6) equal to 17h = 00h – Boot Guard flow error (verification timeout, verification error or BIOS Protection error). For event data 2 (byte 6) equal to 18h =01h – Firmware entered restricted mode – UMA is not available, restricted features set. =02h – Firmware exited restricted mode. For Event Data 2 (byte 6) equal to 19h =01h - BIOS did not set reset synchronization in multiPCH mode =02h - PMC indicates different non/legacy mode for the PCH than BMC set on the GPIO =03h – Misconfiguration MPCH support enabled due to BTG support enabled For Event Data 2 (byte 6) equal to 1Ah =00h – OEM Public Key verification error =01h – Flash Descriptor Region Manifest verification error =02h – Soft Straps verification error For Event Data 1 SMBus link failure (Health Event Type = 01h) Byte 6 – Event Data 2 Sensor bus link on which the error was detected. =01h SMLINK0/SMLINK0B	

Net Function = S/E (04h) LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
		=02h SMLINK1 =03h SMLINK2 =04h SMLINK3 =05h SMLINK4 Byte 7 – Event Data 3 MUX address if there is a problem with a bus segment behind a MUX or FFh if there is a problem with the whole SMBus link. The address field format depends on the MUX type. For MGPIO MUX; [0:5] = Mux MGPIO index. [6:7] = Reserved, set to 00b. For SMBus MUX; [0] – Reserved, set to 0b. [7:1] – 7-bit SMBus MUX address.	
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in Section 2.15)	
02h	Platform Event Message Intel® Host Partition Reset Warning Sensor	Request Byte 1 – EvMRev =04h (IPMI2.0 format) Byte 2 – Sensor Type =DCh (OEM) Byte 3 – Sensor Number =197 – Host Partition Reset Warning Sensor Byte 4 – Event Dir | Event Type [7] – Event Dir =0 – Assertion Event [6:0] – Event Type =77h (OEM) Byte 5 – Event Data 1 [7:6] = 10b – OEM code in byte 2 [5:4] = 10b – OEM code in byte 3 [3:0] = 01h – Host Partition Reset triggered. Byte 6 – Event Data 2 If State is Asserted: Time for which Intel® ME will delay Platform Reset. =00h – FEh – time in unites specified in Event Data 3 Byte 7 – Event Data 3 If State is Asserted: Time Units for which Intel® ME will delay Platform Reset. =00h – reserved =01h – minutes =02h – FFh – reserved	This platform event provides information regarding upcoming Platform Reset. Rearming of this sensor means that BMC finished all the operations and allows Intel® ME for reset to be proceeded immediately.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in Section 2.15)	

2.10 Intel® ME Firmware Manufacturing Process Commands

The next command enables checking the status and reading configuration files stored in Field Programmable Fuses (FPF):

- Get File Attributes – makes it possible to check if the file exists, and what is the size of the file
- Read File – makes it possible to check the content of the configuration file

Note that the file names are dependent on the feature. Refer to [ME_BIOS_INT] for the list of configuration files.

Table 2-11 IPMI Manufacturing Process Commands

Net Function = 30h LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
EBh	Access FPF file	Request Byte 1:3 – Intel Manufacturer ID – 000157h, LS byte first Byte 4 – Command ID = 01h – GET_FILE_ATTRIBUTES = 02h – READ_FILE Byte 5:36 - File Name. Full path to the requested file. The file must reside in FPF (/fpf). File name length must be equal to 32 Bytes (if needed should be padded with 0x00 to 32 Bytes). For GET_FILE_ATTRIBUTES command Byte 37 – Flags [7:0] Reserved For READ_FILE command ID Byte 37:40 - Offset from the beginning of file. Byte 41:44 –Size. Number of bytes to read (max. size for one operation is 235B, limited additionally by the transport layer size). Byte 45 – Flags [7:2] Reserved [1] Get the hash value of the file Y/N [0] Reserved	
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in Section 2.15) =70h – File not found =71h – File is not an FPF file =72h – Requested read size is too long =73h – Requested offset is outside of the file =74h – Requested file cannot be accessed =75h – Invalid command ID =76h – FPF file has not been written =7Fh – Other FPF read error Byte 2:4 – Intel Manufacturer ID – 000157h, LS byte first. For GET_FILE_ATTRIBUTES command ID Byte 5:8 – Size. The size of the requested file. Byte 9 – Properties [7:1] Reserved [0] The file is encrypted Y/N For READ_FILE command ID	

Net Function = 30h LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
		Byte 5:8 – Size. The size of actually read data. Byte 9:N – Data. On successful operation the requested data is provided (up to 235B, limited by the maximum size of the transport layer).	

2.11 IPMI OEM Intel® ME Firmware Update backward compatible commands

Intel® Node Manager 5.0 does not support online firmware update. However the next IPMI commands backward compatible with online firmware update are supported.

Table 2-12 IPMI OEM Intel® ME Firmware Update backward compatible commands

Net Function = 2Eh-2Fh LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
A7h	Online Update Get Capabilities	Request Byte [1:3] = Intel Manufacturer ID – 000157h, LS byte first	
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in Section 2.15) Byte [2:4] = Intel Manufacturer ID – 000157h, LS byte first Byte 5 – Areas supported [0] – Reserved. Return as 0b. [1] – Operational code =1 – OpCode supported [2] – PIA =1 – PIA supported [3] – SDR =1 – SDR supported [4:7] – Reserved. Return as 0000b. Byte 6 – Special capabilities [0] – Rollback =1 – Rollback supported [1] – Recovery =1 – Recovery supported [2:7] – Reserved. Return as 000000b.	

Net Function = 2Eh-2Fh LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
A8h	Online Update Get Image Inventory	Request Byte [1:3] = Intel Manufacturer ID – 000157h, LS byte first Byte 4 = Image ID 0xh – Intel® ME Region =00h – Provide information for the Recovery image =01h – Provide information about 1st operational image =02h – Provide information about 2nd operational image =04h – Provide information about Flash Partition Table area =05h – Provide information about Shared Configuration Area =06h – Provide information about Factory presets area =07:0Fh – Reserved 1xh – BIOS Region =10h – Provide information about BIOS Region 1 =11h:1Fh - Reserved 2xh – Secondary BIOS Region =20h – Provide information about BIOS Region 2 = 21h:2Fh – Reserved 3xh – DER Region =30h – PTU Option ROM =31h: 3Fh – Reserved 4xh – Platform Data (not supported) =40h:4Fh – Reserved 5xh – Reserved 6xh - Reserved 7xh – Flash Descriptor Region =70h – Flash Descriptor 0 =71h:FFh – Reserved	This is the command to query image inventory. This command can be used to learn what images are present. BIOS, Platform Data and Flash Descriptor regions are not supported. **Note:** if for given Image ID there is no version defined, the response returns zeros.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in Section 2.15). =80h – Operation refused (too many requests). =81h – Flash error =82h – Operation in progress (flash erase) =83h – Operation not supported for specific Image ID =84h – Area not present =C9h – Parameter out of range. Image ID out of range Byte 2:4 = Intel Manufacturer ID – 000157h, LS byte first Byte 5 – Firmware Revision 1 [7] – Device available =0 – Normal operation. =1 – Device FW update or self-initialization in progress. [6:0] – Major Firmware Revision, binary encoded Byte 6 – Firmware Revision 2 Minor Firmware Revision - BCD encoded. Should contain the same value as the Get Device Id command's response byte 4 Byte 7:10 – Auxiliary Firmware Revision Information	

Net Function = 2Eh-2Fh LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
		Byte 7 – Implemented version of Intel® NM firmware IPMI command specification BCD encoded = 4.0 Byte 8 – Intel® NM firmware build number BCD encoded = A.B Byte 9 – Intel® NM firmware last digit of build number and patch number BCD encoded = C.PATCH Byte 10 – Image flags [7:3] – Reserved. Return as 00000b. [2] – Image is currently running =0 – Image not running. =1 – Image is currently a running image. [1:0] – Image type =00b – Recovery image. =01b – Operational image 1. =10b – Operational image 2. =11b – Unspecified: flash error indication. *Note: Full version number is:* "Major Firmware Revision. Minor Firmware Revision.ABC.PATCH" where ABC is firmware build number. PTU Option ROM version is only reported in Major Firmware Revision (Byte 5, bits [6..0]) and in the upper nibble of Firmware Revision (Byte6, bits [7..4]). All other fields are set to zero for PTU Option ROM	

2.12 IPMI Commands Supported by Recovery Boot Loader

The Intel® NM firmware supports only the commands for IPMI over IPMB interfaces next listed when running in recovery boot loader mode:

- **Get Device ID** (Net Function 06h, Command Code 01h)
- **Cold Reset** (Net Function 06h, Command Code 02h)
- **Get Self-Test Results** (Net Function 06h, Command Code 04h)
- **Online Update Get Image Inventory** (Net function 2Eh, Command Code A8h)
- **Unlock ME Region** (Net function 2Eh, Command Code E7h)
- **Force ME Recovery** (Net function 2Eh, Command code DFh)
- **Proxy Diagnostics Console** (Net function 30h, Command code 26h)

2.13 IPMI OEM PECI Proxy Commands

Starting with Haswell family of CPUs the PECI interface supports two controllers: PCU and VCU. It is important to note that a PECI interface failure may be caused by a failure of one of these two controllers not the PECI interface itself. In case of an IERR for example the PCU might not be responding while the VCU will still be providing valid and important data.

IPMI OEM PECI Proxy Completion Codes used in the commands response:

Table 2-13 IPMI OEM PECI Proxy Commands

Code	Definition
IPMI OEM PECI Proxy Completion Codes	
00h	Command Completed Normally.
A0h	Partial success (only few first responses are provided; the remaining responses did not fit in the IPMI response message as the response message would exceed maximum IPMI message size supported) – only for Aggregated Send RAW PECI.
A1h	Wrong CPU number.
A2h	Command response timeout, retry may be needed.
A3h	Inband PECI interface is down. (PECI) Response data invalid, if present.
A4h	Bad read FCS in the response (even after the retry).
A5h	Bad write FCS field in the response or Abort FCS in the response (even after the retry).
A6h	Wrong (unsupported) write length in IPMI request.
A7h	Wrong (unsupported) read length in IPMI request.
A8h	Selected PECI interface not available: Not configured by BIOS (for in-band PECI) or Not functional (for in-band PECI or serial PECI) or Not connected (for serial PECI, disabled using OEM configuration option)
ABh	Wrong (unknown/invalid/illegal) command code, request not understood by CPU.
ACh	CPU not present, this error code is returned if no response from PECI client (client device is not responding at all).
D5h	Command not supported in present state – Platform not in S0/S1 state.
FFh	Other error encountered (code returned for all other unexpected errors).

IPMI OEM PECI Proxy commands are product specific.

Table 2-14 IPMI OEM PECI Proxy Commands Definition

Net Function = 2Eh-2Fh LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
40h	Send Raw PECI	Request Byte [1:3] = Intel Manufacturer ID – 000157h, LS byte first. Byte 4 PECI Client Address and interface selection [7:6] – PECI Interface selection: =00b – Intel® ME will send the PECI request using PECI over DMI interface. If in-band PECI is not functional (not configured by BIOS or not working due to failures), Intel® ME will use serial PECI interface when connected directly to chipset. For nonlocal PCI Config Read requests, Intel® ME will only use serial PECI interface. =01b – Intel® ME will send the PECI request using PECI over DMI. Intel® ME will not try the serial PECI interface. This option is not supported for nonlocal PCI Config Read requests. =10b – Intel® ME will send the PECI request serial PECI interface, if the interface is connected directly to the chipset. =11b – Reserved – Not used [5:0] - PECI Client Address (values shall be in the range from 30h through 37h). Byte 5 – Write Length (part of PECI standard header); this field shall be set to the proper value for this PECI command as if there was AWFCS byte provided but Intel® ME FW does not verify if the length matches the PECI protocol specification. Byte 6 – Read Length (part of PECI standard header); this field shall be set to the proper value for this PECI command but Intel® ME FW does not verify if the length matches the PECI protocol specification. Byte 7:M – The remaining part of PECI command following the Read Length field (if any – this field does not exist for PECI Ping command); only write data bytes shall be put here, excluding AWFCS bytes (AWFCS will be added by Intel® ME FW); note that the retry bit shall normally be set to zero and the command code byte shall be one of the codes understood by Intel® ME FW (01h, F7h, A1h, A5h, B1h, B5h, 61h, 65h, E1h, E5h; note that only Domain 0 codes are supported).	The command initiates a single PECI transaction. Only PECI 3.0 command set is supported. Note: Processing of some Raw PECI requests by VCU may take even 250 ms. This is why the response for this command may be returned after a timeout specified by IPMB protocol. Note: In order to use this command OEM needs to sign “INTEL LICENSE AGREEMENT TO PLATFORM ENVIRONMENT CONTROL INTERFACE SPECIFICATION”.
		Response Byte 1 – Completion Code (Remaining standard Completion Codes are shown in Section 2.15) =00h – PECI response successfully retuned (see PECI response Completion Code for detailed response from PECI client, which may be not fully successful). =A2h – Command response timeout. =A3h - Inband PECI interface is down. PECI response data invalid. =A4h – Bad read FCS in the response. =A5h – Bad write FCS field in the response. =A8h - Selected PECI interface not available =ABh – Wrong command code. =ACh – CPU not present.	

		=D5h – Platform not in S0/S1 state. =FFh – Other error encountered. Byte [2:4] = Intel Manufacturer ID – 000157h, LS byte first. Bytes 5:N – PECI response data (if any – no data is returned for Ping command or for Completion Code in Byte#1 other than 00h); data following the Write FCS field are put here exactly as received from PECI client during Read transaction phase, excluding the Write FCS and Read FCS bytes. Retries: In case of PECI errors, before sending the IPMI response back to BMC, Intel® ME FW performs PECI retry attempts according to the regular PECI retry rules published for the specific processor (the processor specific External Design Specification document lists the retry rules and guidelines). This is true for situations when there is no response at all form the processor (all zeroes), checksum verification fails, including Abort FCS, as well as the PECI completion code indicates a PECI client error, such as a timeout. Also according to the PECI retry rules, ME FW sets the retry bit in the PECI request when necessary. An error response is returned to BMC only after all the retry attempts fail.	
41h	Aggregated Send Raw PECI	Request Byte [1:3] = Intel Manufacturer ID – 000157h, LS byte first. Bytes 4:M – Raw PECI command bytes formatted according to the same rules as bytes 4 to M in Send Raw PECI. Bytes M+1:N – Next RAW PECI command (if any). PECI Interface Selection field must be same for all PECI transactions listed in the request	The command initiates multiple PECI transactions. One IPMI request can contain multiple of PECI RAW transactions. The only PECI 3.0 command set is fully supported. The PECI 3.0 command suite retains only the Ping(), GetDIB() and GetTemp() PECI 2.0 commands. Other PECI 2.0 commands ARE NOT SUPPORTED. Note: Processing of some Raw PECI requests by VCU may take even 250 ms. That's why the response for this command may be returned after a timeout specified by IPMB protocol. This time depends on the number of Raw PECI Request encapsulated into single IPMI commands. Note: In order to use this command OEM needs to sign "INTEL LICENSE AGREEMENT TO PLATFORM ENVIRONMENT CONTROL INTERFACE SPECIFICATION".
		Response Byte 1 – Completion Code related to overall IPMI request (Remaining standard Completion Codes are shown in Section 2.15). =00h – Success; it means that all the PECI raw requests have been processed and the responses fit in the IPMI response message (not necessarily that all the responses have completed with success). The completion codes for the particular PECI raw transactions are included in the appropriate parts of this response frame. =A0h – Partial response (all the PECI commands have been executed but only few first responses are provided; the remaining responses did not fit in the IPMI response message as the response message would exceed maximum IPMI message size supported see 0). =A3h - Inband PECI interface is down. PECI response data invalid. =A8h – Selected PECI interface not available =ABh – Wrong command code. =D5h – Platform not in S0/S1 state. Byte [2:4] = Intel Manufacturer ID – 000157h, LS byte first. Byte 5 – Completion Code for this particular transaction – same as byte#1 in Send Raw PECI command response.	

		Byte 6:6+N – the first PECI response data received from PECI client during Read transaction phase (if any), formatted in the same way as response in Send Raw PECI. Byte N+1:M+1– Next PECI transaction: Completion Code byte + response (if any); individual responses are returned in the order they were in the IPMI request. Note – It is the BMC's responsibility to ensure the responses can fit in the IPMI response message (knowing the max IPMI response frame length supported by Intel® ME FW). If some responses do not fit into the IPMI response message, they are not returned (but execution of the corresponding PECI commands is attempted). Note – Intel® ME FW interprets each raw PECI request based on the Write Length field in each PECI request. Invalid Write Length in a malformed raw PECI request will likely cause all subsequent requests to be interpreted by Intel® ME FW starting from a wrong offset and will likely result in C7h error code for the whole IPMI request.	
42h	CPU Package Configuration Read	Request Byte [1:3] = Intel Manufacturer ID – 000157h, LS byte first. Byte 4 [7:3] – Reserved. [2:0] – CPU number (starting from 0). Intel® ME FW will support up to 8 sockets for the current platform generation. Byte 5 – PCS Index Byte [6:7] – Parameter Number (WORD) Byte 6 – Parameter [7:0] Byte 7 – Parameter [15:8] Byte 8 – Read Length – number of bytes to read [7:3] – Reserved. [2:0] – Read Length – number of bytes to read: = 0 – Reserved – shouldn't be used. = 1 – 1 byte. = 2 – 2 bytes (word). = 3 – 4 bytes (double word). = 4 – 8 bytes (64-bits) – only supported using inband PECI. = 5 to 7 – Reserved	This command provides read access to the "package Configuration Space" that is maintained by the CPU. Intel® ME sends the PECI command using in-band PECI interface if available. Otherwise, Intel® ME selects serial PECI connected directly to chipset.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in Section 2.15). =A1h – Wrong CPU number. =A2h – Command response timeout. =A3h - Inband PECI interface is down. Response data invalid. =A4h – Bad read FCS in the response. =A5h – Bad write FCS field in the response. =A7h – Wrong read length. =ABh – Wrong command code. =ACh – CPU not present. =D5h – Platform not in S0/S1 state. =FFh – Other error encountered. Byte [2:4] = Intel Manufacturer ID – 000157h, LS byte first.	

		Byte 5:N – Configuration Data returned by CPU. Size of this field depends on Read Length parameter specified in the request.	
43h	CPU Package Configuration Write	Request Byte [1:3] = Intel Manufacturer ID – 000157h, LS byte first. Byte 4 [7:3] – Reserved. [2:0] – CPU number (starting from 0). Intel® ME FW will support up to 8 sockets for the current platform generation. Byte 5 – PCS Index. Byte [6:7] – Parameter Number (WORD) Byte 6 – Parameter [7:0]. Byte 7 – Parameter [15:8]. Byte 8 – Write Length – number of bytes to write [7:3] – Reserved. [2:0] – Write Length – number of bytes to write =0 – Reserved – shouldn't be used. =1 – 1 byte. =2 – 2 bytes (word). =3 – 4 bytes (double word). =4 – 8 bytes (64-bits) – only supported using inband PECI. =5 to 7 – Reserved Byte 9:N – Data to be written to CPU. Length of this data (1B, 2B, 4B) depends on Write Length value included in Byte 8 of this request.	This command provides write access to the "package Configuration Space" that is maintained by the CPU. Intel® ME sends the PECI command using in-band PECI interface if available. Otherwise, Intel® ME selects serial PECI connected directly to chipset.
		Response Byte 1 – Completion Code = 00h – Success (Remaining standard Completion Codes are shown in Section 2.15). =A1h – Wrong CPU number. =A2h – Command response timeout. =A3h - Inband PECI interface is down. =A4h – Bad read FCS in the response. =A5h – Bad write FCS field in the response. =A6h – Wrong write length. =ABh – Wrong command code. =ACh – CPU not present. =D5h – Platform not in S0/S1 state. =FFh – Other error encountered. Byte [2:4] = Intel Manufacturer ID – 000157h, LS byte first.	
44h	CPU PCI Configuration Read	Request Byte [1:3] = Intel Manufacturer ID – 000157h, LS byte first. Byte 4 [7] – Reserved. [6] = 1b – PCI local space; The RdPCIConfigLocal() command will be used that provides read access to the PCI configuration space that resides on the processor itself (named here - "local" PCI space). Accessing the local PCI space is possible before BIOS has enumerated the systems buses. [5:3] – Reserved. [2:0] – CPU number (starting from 0). Intel® ME FW will support up to 8 sockets for the current platform generation.	The command reads from PCI configuration space of selected CPU. This command allows BMC to read the configuration from the local PCI configuration space as well. Intel® ME sends the PECI command using in-band PECI interface if available. Otherwise, Intel® ME selects serial PECI connected directly to chipset. For nonlocal requests, Intel® ME will only use serial PECI.

		Byte 5:8 – PCI Address [31:28] – Reserved. [27:20] – Bus Number. [19:15] – Device Number. [14:12] – Function Number. [11:0] – Register Address. Byte 9 – Read Length – number of bytes to read [7:2] – Reserved. [1:0] – Read Length – number of bytes to read =0 – Reserved – shouldn't be used. =1 – 1 byte. =2 – 2 bytes (word). =3 – 4 bytes (double word).	
		Response Byte 1 – Completion Code = 00h – Success (Remaining standard Completion Codes are shown in Section 2.15). =A1h – Wrong CPU Number. =A2h – Command response timeout. =A3h - Inband PECI interface is down. Response data invalid. =A4h – Bad read FCS in the response. =A5h – Bad write FCS field in the response. =A7h – Wrong read length. =ABh – Wrong command code. =ACh – CPU not present. =D5h – Platform not in S0/S1 state. =FFh – Other error encountered. Byte [2:4] = Intel Manufacturer ID – 000157h, LS byte first. Byte 5:N – Register Value returned by CPU. Size of this field depends on Read Length parameter specified in the request.	
45h	CPU PCI Configuration Write	Request Byte [1:3] = Intel Manufacturer ID – 000157h, LS byte first. Byte 4 [7] – Reserved. [6] = 1b – PCI local space; The WrPCIConfigLocal() command will be used that provides write access to the PCI configuration space that resides on the processor itself (named here - "local" PCI space). Accessing The local PCI space is possible before BIOS has enumerated the systems buses. [5:3] – Reserved. [2:0] – CPU number (starting from 0). Intel® ME FW will support up to 8 sockets for the current platform generation. Byte 5:8 – PCI Address [31:28] – Reserved. [27:20] – Bus Number. [19:15] – Device Number.	The command writes a value to PCI configuration space of selected CPU. This command allows BMC to write the configuration to the local PCI configuration space as well. Intel® ME sends the PECI command using in-band PECI interface if available. Otherwise, Intel® ME selects serial PECI connected directly to chipset. For nonlocal requests, Intel® ME will only use serial PECI.

		[14:12] – Function Number. [11:0] – Register Address. Byte 9 – Write Length [7:2] – Reserved. [1:0] – Write Length – number of bytes to write =0 – Reserved – shouldn't be used. =1 – 1 byte. =2 – 2 bytes (word). =3 – 4 bytes (double word). Byte 10:N – Register Value to be written to CPU. Length of this data (1B, 2B, 4B) depends on Write Length value included in Byte 9 of this request.	
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in Section 2.15). =A1h – Wrong CPU Number. =A2h – Command response timeout. =A3h - Inband PECI interface is down. =A4h – Bad read FCS in the response. =A5h – Bad write FCS field in the response. =A6h – Wrong write length. =ABh – Wrong command code. =ACh – CPU not present. =D5h – Platform not in S0/S1 state. =FFh – Other error encountered. Byte 2:4 = Intel Manufacturer ID – 000157h, LS byte first.	
46h	CPU IA MSR Read	Request Byte 1:3 = Intel Manufacturer ID – 000157h, LS byte first. Byte 4 [7:3] – Reserved. [2:0] – CPU number (starting from 0). Intel® ME FW will support up to 8 sockets for the current platform generation. Byte 5 – Thread ID Bytes 6:7 – MSR Address Byte 6 – MSR Address [7..0] Byte 7 – MSR Address [15..8] Byte 8 – Read Length – number of bytes to read [7:3] – Reserved. [2:0] – Read Length – number of bytes to read =0 – Reserved – shouldn't be used. =1 – 1 byte. =2 – 2 bytes (word). =3 – 4 bytes (double word). =4 – 8 bytes (quad word). =5-7 – Reserved – illegal value in the current version.	This command provides access to the IA MSR space (core and uncore). Specific processors might limit accessibility to certain areas of the MSR space. Refer to the appropriate processor documentation for a description of the accessibility limitations. Intel® ME sends the PECI command using in-band PECI interface if available. Otherwise, Intel® ME selects serial PECI connected directly to chipset.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in Section 2.15). =A1h – Wrong CPU number. =A2h – Command response timeout. =A3h - Inband PECI interface is down. Response data invalid. =A4h – Bad read FCS in the response.	

		=A5h – Bad write FCS field in the response. =A7h – Wrong read length. =ABh – Wrong command code. =ACh – CPU not present. =D5h – Platform not in S0/S1 state. =FFh – Other error encountered. Byte 2:4 = Intel Manufacturer ID – 000157h, LS byte first. Byte 5:N – Data returned by CPU. Size of this field depends on Read Length parameter specified in the request.	
4Bh	Get CPU and Memory Temperature	Request Byte 1:3 = Intel Manufacturer ID – 000157h, LS byte first. Byte 4 – CPUs for which temperature readings are requested. [0] – Bit set indicates CPU#0 readings are requested; bit clear indicates that readings are not requested. [1] – Bit set indicates CPU#1 readings are requested; bit clear indicates that readings are not requested. [2] – Bit set indicates CPU#2 readings are requested; bit clear indicates that readings are not requested. [3] – Bit set indicates CPU#3 (PECI readings are requested; bit clear indicates that readings are not requested. [5:4] – CPU set – defines which CPU set should be used =0 – CPU0 to CPU3 =1 – CPU4 to CPU7 =2 – CPU8 to CPU11 (not supported by Intel® ME FW in the current generation – Intel® ME FW will return A1h completion code) =3 – CPU12 to CPU15 (not supported by Intel® ME FW in the current generation – Intel® ME FW will return A1h completion code) [6] – Memory channels set – defines which memory channel set should be used. =0 – Channel 0 to channel 3 =1 – Channel 4 to channel 7 [7] – Request format =0 – Standard frame format – up to 4 DIMM per CHANNEL =1 – Extended frame format – up to 8 DIMM per CHANNEL For standard frame format: Byte 5:6 – 16 bits for CPU#0 indicating memory channels and DIMMs for which temperature readings are requested (4x4 bitmask): Byte 5 [0] – CHANNEL#0(4), DIMM#0. Byte 5 [1] – CHANNEL#0(4), DIMM#1. Byte 5 [2] – CHANNEL#0(4), DIMM#2. Byte 5 [3] – CHANNEL#0(4), DIMM#3. Byte 5 [4] – CHANNEL#1(5), DIMM#0. Byte 5 [5] – CHANNEL#1(5), DIMM#1. Byte 5 [6] – CHANNEL#1(5), DIMM#2. Byte 5 [7] – CHANNEL#1(5), DIMM#3. Byte 6 [0] – CHANNEL#2(6), DIMM#0. Byte 6 [1] – CHANNEL#2(6), DIMM#1. Byte 6 [2] – CHANNEL#2(6), DIMM#2. Byte 6 [3] – CHANNEL#2(6), DIMM#3.	The command returns CPU and all Memory DIMMs temperature value for a selected CPUs and DIMMs. Intel® ME sends the PECI commands using in-band PECI interface if available. Otherwise, Intel® ME selects serial PECI connected directly to chipset. Compatibility Notes: On Mehlow platforms, Memory Temperature readings are not supported.

		Byte 6 [4] – CHANNEL#3(7), DIMM#0. Byte 6 [5] – CHANNEL#3(7), DIMM#1. Byte 6 [6] – CHANNEL#3(7), DIMM#2. Byte 6 [7] – CHANNEL#3(7), DIMM#3. Byte 7:8 – 16 bits for CPU#1 indicating memory channels and DIMMs for which temperature readings are requested (4x4 bitmask) – the format is the same as for CPU#0. Byte 9:10 – 16 bits for CPU#2 indicating memory channels and DIMMs for which temperature readings are requested (4x4 bitmask) – the format is the same as for CPU#0. Byte 11:12 – 16 bits for CPU#3 indicating memory channels and DIMMs for which temperature readings are requested (4x4 bitmask) – the format is the same as for CPU#0. For Extended frame format: Byte 5:8 – 32 bits for first CPU from set indicating memory channels and DIMMs for which temperature readings are requested (4x8 bitmask): Byte 5 [0] – CHANNEL#0(4), DIMM#0. Byte 5 [1] – CHANNEL#0(4), DIMM#1. Byte 5 [2] – CHANNEL#0(4), DIMM#2. Byte 5 [3] – CHANNEL#0(4), DIMM#3. Byte 5 [4] – CHANNEL#0(4), DIMM#4. Byte 5 [5] – CHANNEL#0(4), DIMM#5. Byte 5 [6] – CHANNEL#0(4), DIMM#6. Byte 5 [7] – CHANNEL#0(4), Memory Controller Temperature. Byte 6 [0] – CHANNEL#1(5), DIMM#0. Byte 6 [1] – CHANNEL#1(5), DIMM#1. Byte 6 [2] – CHANNEL#1(5), DIMM#2. Byte 6 [3] – CHANNEL#1(5), DIMM#3. Byte 6 [4] – CHANNEL#1(5), DIMM#4. Byte 6 [5] – CHANNEL#1(5), DIMM#5. Byte 6 [6] – CHANNEL#1(5), DIMM#6. Byte 6 [7] – CHANNEL#1(5), Memory Controller Temperature. Byte 7 [0] – CHANNEL#2(6), DIMM#0. Byte 7 [1] – CHANNEL#2(6), DIMM#1. Byte 7 [2] – CHANNEL#2(6), DIMM#2. Byte 7 [3] – CHANNEL#2(6), DIMM#3. Byte 7 [4] – CHANNEL#2(6), DIMM#4. Byte 7 [5] – CHANNEL#2(6), DIMM#5. Byte 7 [6] – CHANNEL#2(6), DIMM#6. Byte 7 [7] – CHANNEL#2(6), Memory Controller Temperature. Byte 8 [0] – CHANNEL#3(7), DIMM#0. Byte 8 [1] – CHANNEL#3(7), DIMM#1. Byte 8 [2] – CHANNEL#3(7), DIMM#2. Byte 8 [3] – CHANNEL#3(7), DIMM#3. Byte 8 [4] – CHANNEL#3(7), DIMM#4. Byte 8 [5] – CHANNEL#3(7), DIMM#5. Byte 8 [6] – CHANNEL#3(7), DIMM#6. Byte 8 [7] – CHANNEL#3(7), Memory Controller Temperature. Byte 9:12 – 32 bits for second CPU from set indicating memory channels and DIMMs for which temperature readings are requested (4x8 bitmask) – the format is the same as for CPU#0.	

		Byte 13:16 – 32 bits for third CPU from set indicating memory channels and DIMMs for which temperature readings are requested (4x8 bitmask) – the format is the same as for CPU#0. Byte 17:20 – 32 bits for fourth CPU from set indicating memory channels and DIMMs for which temperature readings are requested (4x8 bitmask) – the format is the same as for CPU#0.	
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in Section 2.15). =A1h – Wrong CPU number. =A3h - Inband PECI interface is down. Response data invalid. =D5h – Platform not in S0/S1 state. =ADh – Response cannot be delivered because its length is not supported for underlying transport. =FFh – Other error encountered. When byte 1 indicates success, the remaining bytes contain the thermal status information for requested CPUs and memory DIMMs. Information bytes are not returned for remaining CPUs or memory DIMMs (the length of the response depends on the number of requested CPUs or DIMMs). In other words, the order of bytes returning the readings is the same as listed in this specification in the request, skipping items that are not requested. Byte 2:4 = Intel Manufacturer ID – 000157h, LS byte first. Byte 5:N – CPU temperatures (up to 4 bytes – only bytes for the requested CPUs are returned); the data is returned as an unsigned integer; it is representing the number of degrees of Celsius below the Thermal Control Circuit Activation temperature, with some values are reserved to provide error indication, as specified next. Byte N+1:M – Memory DIMM temperatures (up to 64 bytes – only bytes for the requested DIMMs are returned); each byte shall be interpreted as an unsigned value containing the absolute temperature expressed in degrees of Celsius with the following values reserved to provide error indication: =FFh – Sensor or device not present. =FEh – Reserved. =FDh – Data unavailable due to sensor or interface failure.	

47h	Read Memory SMBus	Request Byte [1:3] = Intel Manufacturer ID – 000157h, LS byte first. Byte 4 [7:3] – Reserved. [2:0] – CPU number (starting from 0). ME FW will support up to 8 sockets for the current platform generation. Byte 5 – SMBus bus identifier - valid values are: zero and one – the SKX CPU supports two memory controllers and each of them is assigned one bus. Byte 6 – SMBus address (slave) – the target SMBus address on the bus [7:4] – 4-bit part of the SMBus address named DTI (Device Type Identifier) [3:1] – 3-bit part of the SMBus address named SA (Slave Address) [0] – Reserved, shall be zero Byte 7 – Command code – will be used when sending the transaction Byte 8 – Number of bytes to read minus one (at present, the only accepted value is zero = 1 byte to read; all other values are reserved); note that each byte is read as a separate SMBus transaction with the same command code.	This command executes a read byte transaction on the SMBus connected to a SKX memory controller. It can be used to read the DIMM SPD information by BMC that does not have direct access to the SMBus. This command works after ME FW gets End Of Post message from BIOS. ME FW will execute the command in such a way that it will not interfere with CPU TSOD polling mechanisms. For this reason, execution of this command may take more than 250 ms, defined on IPMB. Intel® ME sends the PECI command using in-band PECI interface. Serial PECI, even if connected directly to the PCH, is never used. **Compatibility Notes:** This command is not supported on Mehlow platform.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in [NM IPMI]). =A1h – Wrong CPU number or CPU is not available =A2h – PECI Command response timeout.\ =A3h - Inband PECI interface is down. Response data invalid. =A4h – PECI Bad read FCS in the response. =A5h – PECI Bad write FCS field in the response. =ACh – CPU not present. =ADh – SMBus error: illegal SMBus address =AAh – SMBus error: SMBus timeout. =D5h – Platform not in S0/S1 state. =D7h – Smart & CLST event happened and the transaction has been aborted – retry needed. =FFh – Other error encountered. Byte 2:4 = Intel Manufacturer ID – 000157h, LS byte first. Byte 5:5+N – Data byte(s) read from the device where N is the value of Byte 8 in the request.	
48h	Write Memory SMBus	Request Byte 1:3 = Intel Manufacturer ID – 000157h, LS byte first. Byte 4 [7:3] – Reserved. [2:0] – CPU number (starting from 0). ME FW will support up to 8 sockets for the current platform generation. Byte 5 – SMBus bus identifier - valid values are: zero and one – the SKX CPU supports two memory controllers and each of them is assigned one bus Byte 6 – SMBus address (slave) – the target SMBus address on the bus.	This command executes a write byte transaction on the SMBus connected to a SKX memory controller. It can be used to write to the DIMM SPD by BMC that does not have direct access to the SMBus. This command works after ME FW gets End Of Post message from BIOS. Intel® ME FW will execute the command in such a way that

		[7:4] – 4-bit part of the SMBus address named DTI (Device Type Identifier). [3:1] – 3-bit part of the SMBus address named SA (Slave Address). [0] – Reserved, shall be zero. Byte 7 – Command code – will be used when sending the transaction. Byte 8 – Number of bytes to write minus one (at present, the only accepted value is zero = 1 byte to write; all other values are reserved); note that each byte is written as a separate SMBus transaction with the same command code. Byte 9:9+N – data byte(s) to be written to the SMBus device (N is the value of Byte 8).	it will not interfere with CPU TSOD polling mechanisms. For this reason, execution of this command may take more than 250ms, defined on IPMB. Intel® ME sends the PECI command using in-band PECI interface. Serial PECI, even if connected directly to the PCH, is never used. **Compatibility Notes:** This command is not supported on Mehlow platform.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in section 2.11). =A1h – Wrong CPU number or CPU is not available. =A2h – PECI Command response timeout. =A3h - Inband PECI interface is down. =A4h – PECI Bad read FCS in the response. =A5h – PECI Bad write FCS field in the response. =ACh – CPU not present. =ADh – SMBus error: illegal SMBus address. =AAh – SMBus error: SMBus timeout. =D5h – Platform not in S0/S1 state. =D6h – write operations not permitted on this bus. =D7h – Smart & CLST event happened and the transaction has been aborted – retry needed. =FFh – Other error encountered. Byte 2:4 = Intel Manufacturer ID – 000157h, LS byte first. Byte 5 – Reserved (returned as zero).	

2.14 IPMI OEM PECI Proxy Sensors

The following table and subsection summarize the sensors exposed by PECI Proxy functionality in Intel® ME FW. Sensor readings are not available when the system is in low power state.

Note: The default configuration of the sensors can be set using Flash Image Tool.

The default configuration includes:

- Thresholds
- Event Enable Mask
- Scanning Periods
- Scanning Enable Flag
- Per-sensor Event Enable Flag

Table 2-15 IPMI OEM PECI Proxy Sensors

Sensor #	Description	Notes
28	CPU 0 Thermal Status	Discrete sensor
32	CPU 0 Thermal Control Circuit Activation	Threshold sensor
36	CPU 0 T-Control	OEM sensor that presents T-Control parameter of the CPU. The value of this sensor is constant and may only change upon HW configuration change.
48	CPU 0 T-JMAX	OEM sensor that presents TJmax parameter of the CPU. The value of this sensor is constant and may only change upon HW configuration change.
52	CPU 0 Memory Throttling	Threshold sensor

2.14.1 CPU Thermal Status Sensors

The sensors are discrete sensors presenting various states associated with CPU thermal status.

Table 2-16 CPU Thermal Status Sensors

Bit Offset	Name	Description
0	CPU Critical Temperature	Indicates whether CPU temperature is above critical temperature point.
1	PROCHOT# Assertions	Indicates whether PROCHOT# signal is asserted.
2	TCC Activation	Indicates whether CPU thermal throttling functionality is activated due to CPU temperature being above Thermal Circuit Control Activation point.
3	CPU Critical Temperature Log	Indicates whether CPU temperature has been above critical temperature point since a previous reset or the last time software cleared corresponding CPU thermal status bit.
4	PROCHOT# Assertions Log	Indicates whether PROCHOT# signal has been asserted since a previous reset or the last time software cleared corresponding CPU thermal status bit.
5	TCC Activation Log	Indicates whether CPU thermal throttling functionality has been activated due to CPU temperature being above Thermal Circuit Control Activation point since a previous reset or the last time software cleared corresponding CPU thermal status bit.

The value of this sensor is updated by Intel® ME FW every 250 ms.

2.14.2 CPU Thermal Control Circuit Activation Sensors

The sensors are threshold-based sensors presenting the percentage of time the processor has been operating at a lowered performance due to TCC activation. It does not include the TCC activation time as a result of an external assertion of PROCHOT# signal.

The value of the sensor is updated every 250 ms but the sensor returns the average over the last 6 seconds (24 samples).

2.14.3 CPU T-Control Sensors

These sensors are presenting T-Control parameter for the processors. T-Control value is fan speed control reference temperature. For detailed description of the value, see the [EDS] document.

Fan Temperature target offset (T-Control) indicates the relative offset from CPU Tjmax temperature at which fans should be engaged. For detailed description of the $T_{J\text{-}MAX}$ value, see the [EDS] document.

2.14.4 CPU $T_{J\text{-}MAX}$ Sensors

These sensors are presenting Tjmax parameter for the processors. CPU $T_{J\text{-}MAX}$ is the minimum temperature that the processor will start throttling due to TCC activation. For detailed description of the value, see the [EDS] document.

The value of this sensor is constant and may only change upon HW configuration change.

2.14.5 Memory Throttling Status Sensors

These sensors provide information on memory throttling as a percentage of memory cycles were throttled due to power limiting (valid range is 0..200, value 1 means 0.5%).

The value of the sensor is updated every 250 ms but the sensor returns the average over the last 6 seconds (24 samples).

2.15 IPMI Standard Completion Codes

Intel® Node Manager IPMI commands use standard Completion Codes from the next table and specific OEM commands codes if specified in command description. Unless specified otherwise, by a specific command description, fields following nonzero Completion Code are truncated[1].

Table 2-17 IPMI Standard Completion Codes

Code	Definition
Generic Completion Codes 00h, C0h-FFh	
00h	Command Completed Normally.

[1] Exception: for Net Function = 2Eh-2Fh for all Completion Codes up to 3 bytes containing IANA Enterprise Number are copied from the original request.

Code	Definition
C0h	Node Busy. Command could not be processed because command processing resources are temporarily unavailable. This Completion Code is returned when Intel® ME is erasing Flash and cannot handle IPMI requests – for example during firmware update procedure or when processing an Intel® NM configuration update request.
C1h	Invalid Command. Used to indicate an unrecognized or unsupported command. *Note:* For Net Function = 2Eh – 2Fh for all unrecognized IANA Enterprise Numbers C1h is returned in response followed by up to 3 bytes containing unrecognized IANA Enterprise Number are copied from the original request.
C2h	Command invalid for given LUN.
C3h	Timeout while processing command. Response unavailable.
C4h	Out of space. Command could not be completed because of a lack of storage space required to execute the given command operation.
C5h	Reservation Canceled or Invalid Reservation ID.
C6h	Request data truncated.
C7h	Request data length invalid.
C8h	Request data field length limit exceeded.
C9h	Parameter out of range. One or more parameters in the data field of the Request are out of range. This is different from "Invalid data field" (CCh) code in that it indicates that the erroneous fields has a contiguous range of possible values.
CAh	Cannot return number of requested data bytes.
CBh	Requested Sensor, data, or record not present.
CCh	Invalid data field in request.
CDh	Command illegal for specified sensor or record type.
CEh	Command response could not be provided.
CFh	Cannot execute duplicated request. This Completion Code is for devices which cannot return the response that was returned for the original instance of the request. Such devices should provide separate commands that allow the completion status of the original request to be determined. An Event Receiver does not use this Completion Code, but returns the 00h Completion Code in the response to (valid) duplicated requests.
D0h	Command response could not be provided. SDR Repository in update mode.
D1h	Command response could not be provided. Device in firmware update mode.
D2h	Command response could not be provided. BMC initialization or initialization agent in progress.
D3h	Destination unavailable. Cannot deliver request to selected destination. For example, this code can be returned if a request message is targeted to SMS, but receive message queue reception is disabled for the particular channel.
D4h	Cannot execute command due to insufficient privilege level or other security based restriction (e.g., disabled for "firmware firewall").
D5h	Cannot execute command. Command or request parameters not supported in present state.
D6h	Cannot execute command. Parameter is illegal because command sub-function has been disabled or is unavailable (e.g., disabled for "firmware firewall").
FFh	Unspecified error.
Device Specific (OEM) codes 01h-7Eh	

Code	Definition
01h-7Eh	Device specific (OEM) Completion Codes. This range is used for command specific codes that are also specific for a particular device and version. A prior knowledge of the device command set is required for interpretation of these codes.
Command Specific codes 80h-Beh	
80h-BEh	Standard command-specific codes. This range is reserved for command specific Completion Codes for commands specified in this document.

2.16 Generic Event/Reading Type Codes

Table 2-18 Generic Event/Reading Type Codes

Generic Event/ Reading Type Code	Event/Reading Class	Generic Offset	Description
01h	Threshold	00h 01h 02h 03h 04h 05h 06h 07h 08h 09h 0Ah 0Bh	Lower noncritical – going low Lower noncritical – going high Lower critical – going low Lower critical – going high Lower unrecoverable – going low Lower unrecoverable – going high Upper noncritical – going low Upper noncritical – going high Upper critical – going low Upper critical – going high Upper unrecoverable – going low Upper unrecoverable – going high

2.17 IPMI Diagnostics Commands

The next tables list diagnostics commands supported by Intel® ME FW.

Table 2-19 IPMI Diagnostics Commands

Net Function = Storage (0Ah) LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
43h	Get SEL Entry	Request Byte 1:2 = Reservation ID – Write as 0000h Byte 3:4 = SEL Record ID (should be put in least significant byte first order) = 0000h-00FFh Reserved = 0100h – Get PSU Over Current statistics = 0101h – Get PSU Over Temperature statistics = 0102h – Get PSU Under Voltage statistics Byte 5 = Offset into record – Write as 00h Byte 6 = Bytes to read – Write as FFh (entire record)	Allows for reading diagnostics information from Intel® ME Firmware. The response from the command should be sent to Intel for analysis. Note: This command works only in Intel® NM SKU.
		Response Byte 1 – completion code =00h – Success (Remaining standard completion codes are shown in [IPMI])	

Net Function = Storage (0Ah) LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
		=CBh – Record of given Record ID is not present. In addition next present Record ID is returned for this error code. Byte 2:3 – Next SEL Record ID Byte 4:N – Record Data (16 bytes for entire record) For records ID from 0100h to 0102h: Byte 4:5 – Record ID Byte 6 – Record type =EFh – OEM statistics Byte 7:10 – Last event timestamp (High Precision timer is used) Byte 11:14 – Initialization timestamp (High Precision timer is used) Byte 15 – SmaRT & CLST status =Events detected from last HOST S0 entry Byte 16:17 – Permanent event counter =Events number counted from last Intel® ME FW reset Byte 18:19 – Event counter =Events number counted from last HOST reset	
26h	Proxy Diagnostics Console	Request Byte 1:3 = Intel Manufacturer ID – 000157h, LS byte first. Byte 4:M – Diagnostics Console Frame Request	Allows for sending Diagnostics Console commands to Intel® ME Firmware using IPMI interface.
		Response Byte 1 – completion code =00h – Success (Remaining standard completion codes are shown in [IPMI]) Byte 2:4 = Intel Manufacturer ID – 000157h, LS byte first. Bytes 5:N – Diagnostics Console Frame Response	

2.18 IE Provisioning IPMI Commands

Intel® ME FW supports the following OEM commands to configure IE parameters such as Software Client and Event Receiver.

Table 2-20 IE Provisioning IPMI Commands

Net Function =2Eh LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
0Ch	Set IE Software Client ID	Request Byte 1:3 = Intel Manufacturer ID – 000157h, LS byte first. Byte 4: Software Client ID Byte 5: [7:2] – reserved. Write as 000000b. [1:0] – LUN. Use 00b	Allows IE FW to update IE Software Client to which ME FW to send commands.
		Response Byte 1 – completion code =00h – Success (Remaining standard completion codes are shown in [IPMI]) Byte 2:4 = Intel Manufacturer ID – 000157h, LS byte first.	
0Dh	Get IE Software Client ID	Request Byte 1:3 = Intel Manufacturer ID – 000157h, LS byte first.	Return IE Software Client ID
		Response Byte 1 – completion code =00h – Success (Remaining standard completion codes are shown in [IPMI]) Byte 2:4 = Intel Manufacturer ID – 000157h, LS byte first. Byte 5: IE Software Client ID Byte 6: [7:2] – reserved. Return as 000000b. [1:0] – IE Client LUN	
0Eh	Set IE Event Receiver	Request Byte 1:3 = Intel Manufacturer ID – 000157h, LS byte first. Byte 4: Software Client ID Byte 5: [7:2] – reserved. Write as 000000b. [1:0] – LUN. Use 00b	Allows IE FW to update IE Software Client as an Event Receiver.
		Response Byte 1 – completion code =00h – Success (Remaining standard completion codes are shown in [IPMI]) Byte 2:4 = Intel Manufacturer ID – 000157h, LS byte first.	

Net Function =2Eh LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
0Fh	Get IE Event Receiver	Request Byte 1:3 = Intel Manufacturer ID – 000157h, LS byte first.	
		Response Byte 1 – completion code =00h – Success (Remaining standard completion codes are shown in [IPMI]) Byte 2:4 = Intel Manufacturer ID – 000157h, LS byte first. Byte 5: Event Receiver Software Client ID Byte 6: [7:2] – reserved. Return as 000000b. [1:0] – IE Event Receiver LUN	

2.19 IPMI OEM Telemetry Hub Commands

Telemetry Hub exposes telemetry data to the external systems in a unified way via IPMI. In addition to the standard poll mechanism Telemetry Hub implements a push method which allows external entities to register to a set of readings and specify time intervals in which Telemetry Hub sends requested data to the recipient.

Table 2-21 IPMI OEM Telemetry Hub Commands

Net Function = 2Eh-2Fh, LUN = {00b, 01b, 10b, 11b}			
Code	**Command**	**Request, Response Data**	**Description**
50h	Update Reading Package	Request Byte 1:3 – Intel Manufacturer ID – 000157h, LS byte first Byte 4:7 – Reading Package GUID. Used when updating an existing package, otherwise it shall be set to 0 (when creating new package). Byte 8 – Reading Package Command [2:0] – Operation and parameters =0h – Append metrics or update existing metrics (based on metrics GUID). =1h – Delete this reading package. When this option is selected, then all other values are ignored. Other values reserved [3] – Persistency =0h – Package is volatile =1h – Package is stored on flash [7:4] – Reserved	This command adds or updates a set of metrics to be gathered by Telemetry Hub and reported in one package. The package is identified by an opaque Id. Adding the first reading creates a package when it does not exist. When reporting interval is configured, the requested data will be pushed to the recipient on configured intervals.

Net Function = 2Eh-2Fh, LUN = {00b, 01b, 10b, 11b}			
Code	**Command**	**Request, Response Data**	**Description**
		Byte 9 – Reporting Interval. Allows for setting poll delivery option or interval for pushing the data to BMC via “Receive Telemetry Reading”. The interval is set in 100 millisecond units. The following values have a special meaning: =00h – Disabled - use this value for polling mode or while deleting data. =F0h – FDh – Reserved. =FEh – On Change. The reading package is sent as soon as all new reading values are received and any of the readings is different from previous one. =FFh – On update. The reading package is sent as soon as all new reading values are received. Note: This field is ignored for delete operation. Byte 10:N – (Optional) Array of Readings to be added, updated or deleted. The array bytes shall not be provided when deleting all readings. Each element of the array consists of the following fields: Reading [Byte 1:4] – Metrics GUID Reading [Byte 5] – Grouping. Allows for averaging of metrics values obtained between the readouts. [3:0] – Aggregation Operation =00h – Return the latest value. =01h – Average =02h – Minimum =03h – Maximum Other values are reserved. [7:4] – Reserved, set as 0000b.	The reading packages are not persistent and shall be re-created after every ME reboot.
		Response Byte 1 – Completion Code =00h – Success. (Remaining standard Completion Codes are shown in section 2.11). =C4h – Out of space. There are no free records in existing package or cannot create any more packages. =90h – Invalid Package GUID. =91h – Invalid Metrics GUID. =93h – Invalid Reporting Interval. =94h – Invalid Grouping Operation. =95h – Invalid Reading Package Command. =96h – Invalid Persistency Parameter. Byte 2:4 – Intel Manufacturer ID – 000157h, LS byte first. Byte 5:8 – Reading Package Identifier.	
51h	Query Reading Package	Request Byte 1:3 - Intel Manufacturer ID – 000157h, LS byte first Byte 4:7 – Reading Package GUID Byte 8 – Start reading index (begins with 00h)	This command returns the configuration of reading package.
		Response Byte 1 – Completion Code =00h – Success. (Remaining standard Completion Codes are shown in section 2.11). =90h – Invalid Package GUID. =92h – Reading Index exceeds the number of elements.	

Net Function = 2Eh-2Fh, LUN = {00b, 01b, 10b, 11b}			
Code	**Command**	**Request, Response Data**	**Description**
		Byte 2:4 – Intel Manufacturer ID – 000157h, LS byte first. Byte 5 – Next Reading Index or 0 if all records returned. Byte 6 – Parameters [2:0] – Reserved. [3] – Persistency. =0h – Package is volatile. =1h – Package is stored on flash. [7:4] – Reserved. Byte 7 – Reporting Interval for pushing data in 0.1 second unit. The following values have a special meaning: =0h – Disabled, use for polling or while deleting data. =FFh – On update. The reading package is sent as soon as all new reading values are received. Byte 8:N – (Optional) Array of Readings. Reading [Byte 1:4] – Metrics GUID Reading [Byte 5] – Grouping. [3:0] – Operation =00h – Last readout =01h – Average =02h – Minimum =03h – Maximum Other values are reserved. [7:4] – Reserved, set as 0000b.	
52h	Get Telemetry Readings	Request Byte 1:3 – Intel Manufacturer ID – 000157h, LS byte first Byte 4:7 – Reading Package GUID Byte 8 – Start reading index (begins with 00h)	This command is implemented by Node Manager and used by an external entity to obtain metrics from previously defined reading package.
		Response Byte 1 – Completion Code =00h – Success. (Remaining standard Completion Codes are shown in section 2.11). =90h – Invalid Package GUID. =92h – Reading Index exceeds the number of elements. Byte 2:4 – Intel Manufacturer ID – 000157h, LS byte first. Byte 5 – Next Reading Index. The index of the next not returned record or 0 if all records returned. Byte 6:N – Array of Values (available when Success). Each element of the array consists of the following fields: [Byte 1:4] – Metrics Value, or 0xFFFFFFFF when metrics is not available.	
53h	Receive Telemetry Readings	Request Byte 1:3 – Intel Manufacturer ID – 000157h, LS byte first Byte 4:7 – Reading Package GUID. Byte 8 – First reported metrics index. In case there are too many metrics to fit in one command they will be sent in packages. The first package has always index 0. The other packages will report the index of the first metrics returned by given package.	This command shall be implemented by an external entity (e.g., BMC). When implemented then Node Manager uses this command to send metrics from the previously defined reading package.

Net Function = 2Eh-2Fh, LUN = {00b, 01b, 10b, 11b}			
Code	**Command**	**Request, Response Data**	**Description**
		Byte 9:N – Array of Readings (available when completion code = 00h). Each element of the array consists of the following fields: [Byte 1:4] – 32 bit Metric Value, or 0xFFFFFFFF when metric not available.	
		Response Byte 1 – Completion Code =00h – Success. (Remaining standard Completion Codes are shown in Section **Error! Reference source not ound.**). =C1h – BMC does not implement this command. =90h – Invalid Package GUID. When C1h or 90h completion code is received then Node Manager stops sending notification related to this Package Id. Byte 2:4 – Intel Manufacturer ID – 000157h, LS byte first.	

2.19.1 Telemetry Hub Average Value Field

This field provides average value computed since last Get Telemetry Readings and Receive Telemetry Readings command.

Average value is reset after each Get Telemetry Readings and Receive Telemetry Readings command.

If Get Telemetry Readings and Receive Telemetry Readings command is received / sent but no new reading is available since previous Get / Receive Telemetry Readings, Telemetry Hub returns previous average value. At least 1 reading needs to be available in order to provide new average value.

The average value provided by Telemetry Hub is cumulative average value.

2.19.2 Telemetry Hub Minimum Value Field

This field provides minimum value registered since last Get Telemetry Readings and Receive Telemetry Readings command.

Minimum value is reset after each Get Telemetry Readings and Receive Telemetry Readings command.

If Get Telemetry Readings and Receive Telemetry Readings command is received / sent but no new reading is available since previous Get / Receive Telemetry Readings, Telemetry Hub returns previous minimum value. At least 1 reading needs to be available in order to provide new minimum value.

2.19.3 Telemetry Hub Maximum Value Field

This field provides maximum value registered since last Get Telemetry Readings and Receive Telemetry Readings command.

Maximum value is reset after each Get Telemetry Readings and Receive Telemetry Readings command.

If Get Telemetry Readings and Receive Telemetry Readings command is received / sent but no new reading is available since previous Get / Receive Telemetry Readings, Telemetry Hub returns previous maximum value. At least 1 reading needs to be available in order to provide new maximum value.

3 Intel® ME Intel® NM IPMI Interface

This chapter describes the IPMI command interface used by the Intel® NM 5.0 implementation:

- External Intel® NM IPMI commands that should be used by the external management console on LUN 0. If BMC or chassis manager decide to use external Intel® NM IPMI commands it should use reservation mechanism by sending requests with responder LUN different than 0. For command details, see Section 3.1.
- Discovery mechanism for the Intel® NM functionality, see Section 3.1.
- Product specific Intel® NM IPMI commands used for low-level power management access and for debugging. For details, see Section 3.1.
- External IPMI sensors that should be monitored by external management console. Events from all sensors are generated from LUN 0. For details, see Section 0.
- External DCMI Power Management Commands that should be used by the external DCMI management console, see Section 3.3.

If Intel® NM SKU is disabled in firmware then firmware will respond with C1h Invalid Command Completion Code to the commands listed in this chapter. In addition, BMC should not expose the Intel® NM discovery SDR to the external console.

3.1 External Intel® NM Configuration and Control Commands

Intel® Intelligent Power Node Manager is a platform resident technology that enforces power and thermal policies for the platform. These policies are applied by exploiting subsystem knobs (such as processor P and T states) that can be used to control power consumption. Intel® Node Manager enables data center power and thermal management by exposing an external interface to management software through which platform policies can be specified. It also implements specific data center power management usage models such as power limiting.

The configuration and control commands are used by the external management software or BMC to configure and control the Intel® NM feature. Since Intel® NM firmware does not have any external interface, all these commands are first received by the BMC over LAN and then relayed to the Intel® Node Manager firmware over IPMB interface. The BMC merely acts as a relay and the transport conversion device for these commands using the standard IPMI bridging. In that case the privilege level to access to the ME SMLINK channel should be restricted to allow only the Admin level.

BMC provides the access point for remote commands from external management SW and generates alerts to them. In case Intel® NM is on the Intel® ME, which is an IPMI satellite controller, there have to be mechanisms to forward commands to Intel® ME and send response back to originator. Similarly, events from the Intel® ME have to be sent as alerts outside of BMC. It is the responsibility of BMC to implement these mechanisms for communication with Intel® Node Manager.

Note that all the next commands are not supported when Intel® NM Feature Enabled is set to ‘false’ using Flash Image Tool.

Table 3-1 Intel® NM Configuration and Control Commands

Net Function = 2Eh-2Fh LUN = {00b, 01b, 10b, 11b}			
Code	**Command**	**Request, Response Data**	**Description**
C0h	Enable/Disable Node Manager Policy Control	Request Byte 1:3 - Intel Manufacturer ID – 000157h, LS byte first Byte 4 – Flags [2:0] – Policy Enable/Disable =0h – Global Disable Intel® Node Manager policy control – disables policy control for all power domains regardless of the value set in Domain ID field (Byte 5). =1h – Global Enable Intel® Node Manager policy control – enables policy control for all power domains regardless of the value set in Domain ID field (Byte 5). =2h – Per Domain Disable Intel® Node Manager policies for the domain given by Byte 5. =3h – Per Domain Enable Intel® Node Manager policies for the domain given by Byte 5. =4h – Per Policy Disable Intel® Node Manager policy for the domain/policy given by Byte 5 and Byte 6. =5h – Per Policy Enable Intel® Node Manager policy for the domain/policy given by Byte 5 and Byte 6. [7:3] – Reserved. Write as 00000b. Byte 5 – Domain ID [3:0] – Domain ID Identifies the domain that this Intel® Node Manager policy applies to. This field is valid if Per Policy Enable/Disable is set or if Per Domain Policy Enable/Disable is set. =00h – Entire platform =01h – CPU subsystem =02h – Memory subsystem =03h – Reserved =04h – High Power I/O subsystem Other – Reserved [7:4] – Reserved. Write as 0000b Byte 6 – Policy ID This field is valid if Per Policy Enable/Disable is set.	Enable or Disable the Intel® Node Manager policy control feature. Global enable/disable affects all policies for all domains. Per Domain enable/disable affects all policies of the specified domain. Per Policy enable/disable affects only the policy for the specified domain/policy combination. After receiving the command it may take Intel® Node Manager up to 2s to stop limit power. This command doesn't affect HW Protection Policy state which would be always enabled regardless of Global Policy Control state. Only for policy disable/enable option responder LUN is validated. If responder LUN doesn't match to responder LUN from request when policy was created. In case of performing change on Global Policy Control or Domain Control requests for all responder LUNs would be accepted. This command accesses the flash, so the response will be sent after completing the operation and may take longer than 250 ms.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in Section 2.15). =80h – Policy ID Invalid. =81h – Domain ID Invalid. =D4h – Insufficient privilege level due wrong responder LUN Byte 2:4 – Intel Manufacturer ID – 000157h, LS byte first.	
C1h		Request Byte 1:3 - Intel Manufacturer ID – 000157h, LS byte first.	

Net Function = 2Eh-2Fh LUN = {00b, 01b, 10b, 11b}			
Code	**Command**	**Request, Response Data**	**Description**
	Set Node Manager Policy	Byte 4 – Domain ID [3:0] – Domain ID (Identifies the domain that this Intel® Node Manager policy applies to) =00h – Entire platform =01h – CPU subsystem =02h – Memory subsystem =03h – HW protection =04h – High Power I/O subsystem Others – Reserved [4] – Policy Enabled (set to 1 if policy should be enabled by default during policy creation/modification). Policy will be enforced (enabled and evaluated in runtime) if the corresponding Per Domain control as well as Global control is already enabled see C0h command. [7:5] – Reserved. Write as 000b. Byte 5 – Policy ID Byte 6 – Policy Type | Policy Trigger Type [3:0] – Policy Trigger Type =0 – No Policy Trigger (In that case Policy Trigger Limit should be ignored) =1 – Inlet Temperature Limit Policy Trigger in [Celsius]. =2 – Missing Power Reading Timeout in 1/10th of second. =3 – Time After Host Reset or Startup Trigger in 1/10th of second. If BMC does not send Set Event Receiver command within this time after Power Button Override or Host start up Intel® Node Manager will activate this policy. This policy could be defined in addition to the standard limiting policies. Once triggered, the policy shall remain active until next G3 entry or reception of Set Event Receiver. The value of the limit must be greater than 0. =4 – Boot time policy. This policy will apply the power policy at boot time. This type of policy can be applied only to the Domain 00h and will be applied on each platform restart. =5 – Reserved =6 – MGPIO Policy Trigger. This policy can be set only in the HW protection domain. If the policy already exists, this command will result in re-arming policy trigger. [4] – Policy Configuration Action =0 – Policy Pointed by Policy Id shall be removed (remaining bytes shall be ignored on read). Corresponding (with the same Policy ID) Alert Thresholds and Suspend Periods will be removed as well. =1 – Add Power Policy. This command creates/modifies policy of type that will maintain Power limit.	User can specify any valid Policy ID. If already existing, this command will overwrite/modify the parameters for the existing policy, otherwise a new policy will be created with this policy Id. Modification of some parameters is possible only if that policy for the specified Policy ID is disabled. For more details see Section 3.1.5. Note: The Policy ID is unique over all domains. Set done for existing Policy ID may move the policy to a different domain if different Domain ID is provided. The operator may define a special kind of policy called Minimum Power Consumption policy with the Power Limit set to 0. The policy does not have the power limit defined. When policy is triggered, the Intel® Server Platform Services firmware reduces the power consumption to minimum by requesting OSPM or SMM to set minimum P-state and T-state. The Minimum Power Consumption policy will not trigger the correction action to “System Shutdown”, but this setting could be used by the operator to specify whether the Intel® NM shall minimize power consumption by only reducing P-state or the firmware shall use both P-state and T-state.

Net Function = 2Eh-2Fh LUN = {00b, 01b, 10b, 11b}			
Code	**Command**	**Request, Response Data**	**Description**
		[6:5] – Aggressive CPU Power Correction For policies with Domain ID 0 (Entire platform) and 1 (CPU subsystem) the flag indicates whether Intel® NM can use CPU T-states to control CPU power consumption. This setting is ignored for some types of policies. I.e. a boot time policy is never aggressive and a policy with missing power reading timeout trigger is always aggressive. =00b – Automatic mode (default). Usage of T-states depends on Shutdown System bit in Policy Exception Actions field. When the bit is set to 1, Intel® NM shall use aggressive mode for power limiting (T-states and memory throttling). When the bit is cleared, Intel® NM does not use T-states and memory throttling. This behavior is backward compatible with Intel® NM 1.5. =01b – Force unaggressive mode e.g., Intel® NM is not allowed to use T-states and memory throttling. User should use this setting if the Intel® NM should use only performance-friendly controls. =10b – Force aggressive mode e.g., Intel® NM is allowed to use T-states and memory throttling. User should use this setting only if the target limit should be kept at any cost. =11b – Reserved. For policies with Domain ID 2 (Memory subsystem) the field shall be set to 0. [7] – Policy storage option =0b – persistent storage (default) to be used by external consoles =1b – volatile memory to be used by local management entities Byte 7 – Policy Exception Actions performed if policy cannot be maintained (if maintained policy power limit given by bytes 8-9 is exceeded over Correction Time Limit or 1 second in case of Predictive Power Limiting policy). [0] – Send alert [1] – Shutdown system (soft shutdown via BMC as described in [INT_GUIDE]). [6:2] – Reserved. Write as 00000b. [7] – The policy power domain. This setting influences the power limiting and reporting of this policy. This field is ignored for NM policies with Domain ID other than 0. =0h - The policy works in the primary side power domain. =1h - The policy works in the secondary (DC) power domain. Byte 8:9 – Policy Target Limit For the following Policy Trigger Type value 0, 1 or 3 (Byte 6 bits [3:0]) this field contains the power limit to be maintained in watts as unsigned integer value. Zero value is treated in a special way. If the limit is set to zero, Intel® NM sets highest throttling level regardless of the power consumption in the domain. In such case Intel® NM does not processed configured Policy Exception Actions, does not send Power Limit Exceeded event or execute system shutdown. The zero value can be set even if the minimum power set for the domain is not zero. For the following Policy Trigger Type value 2 – Throttling level of the platform in %, where 100% enables maximum throttling of the system.	The maximum number of policies supported for each domain can be obtained via Get Node Manager Capabilities command. Remove or modify operations would be performed only if responder LUN match with responder LUN from request when policy was created. Policy Exception Actions are not supported for Boot Time Policy and for Predictive Power Limiting Policy. Note: The Statistics Reporting Period used by Intel® ME FW may not match exactly the period set by this command. The exact period interval used is reported by Get Node Manager Statistics command while Get Node Manager Policy command returns the period requested by this command. When missing power readings policy is set for domain 04h NM will use PROCHOT# for platform power throttling. This command accesses the flash, so the response will be sent after completing the operation and may take longer than 250 ms. Compatibility note: Memory throttling is not used on Mehlow. Note: Zero watt policies throttle using control knobs available for given policy; e.g., they take into account Aggressive CPU Power Correction setting.

Net Function = 2Eh-2Fh LUN = {00b, 01b, 10b, 11b}			
Code	**Command**	**Request, Response Data**	**Description**
		For the following Policy Trigger Type value 4 – Power profile to be applied to the platform at boot time: [0] – Platform Booting mode =0 – Platform should boot (during BIOS POST) in power optimized mode. In this mode it's expected that BIOS will consume less power i.e. by running the CPU in LFM and using the least amount of threads. =1 – Platform should boot (during BIOS POST) in performance optimized mode. [7:1] – Cores Disabled – the number of physical CPU cores that should be disabled on each CPU socket. After disabling the cores BIOS POST should lock that value to the OS so that it cannot enable the cores. E.g. 1 passed on that field means that on each CPU package 1 core should be disabled by the BIOS. [15:8] – Reserved. Write as 00000000b. For the following Policy Trigger Type values 6 – Power limit to be maintained in Watts. Setting this value to zero will result in highest throttling level regardless of the power consumption in the domain. Byte 10:13 – Correction Time Limit – the maximum time in milliseconds, in which the Intel® Node Manager must take corrective actions in order to bring the platform back to the specified power limit before taking the action specified in the "Policy Exception Action" parameter. This is an unsigned integer value. Correction Time does not apply to Boot Time Policy. If Trigger Type defines Boot Time Policy (4) the Correction Time Limit parameter should be set to zero. Correction Time set to 0 with Policy Trigger Type set to No Policy Trigger indicates that Predictive Power Limiting policy should be created. This setting is applicable only for policies with both Domain ID=0 and power domain=0 (primary side). Byte 14:15 – Policy Trigger Limit For the following Policy Trigger Type value (Byte 6 bits [0:3]) this field contains the following data: =0 – Policy Trigger Value will be ignored. =1 – Policy Trigger Value should define the Inlet temperature in Celsius. The inlet temperature value will be compared against this limit and if exceeded, cause a trigger to start enforcing the Power Limit specified (Power limit will not be enforced until the trigger happens). =2 – Policy Trigger should define time in 1/10 of second to perform an action if Missing Power Reading Timeout is detected. =3 – Policy Trigger should define time in 1/10 of second after Host reset or startup. If BMC does not send Set Event Receiver command within this time after Host reset or start up Intel® Node Manager will activate this policy. This policy could be defined in addition to the standard limiting policies. The policy is automatically disabled after next Host reset. The value of the limit must be greater than 0. =4 – Policy Trigger is not applicable for boot time policy and should be set to 0. =6 – Policy Trigger is not applicable for MGPIO triggered policy and should be set to 0.	

Net Function = 2Eh-2Fh LUN = {00b, 01b, 10b, 11b}			
Code	**Command**	**Request, Response Data**	**Description**
		Byte 16:17 – Statistics Reporting Period in seconds. The number of seconds that the measured power will be averaged over for the purpose of reporting statistics to external management SW. This is a moving window length. Note that this value is different from the period that Intel® NM uses for maintaining an average for the purpose of power control. This is unsigned integer value.	
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in Section 2.15). =80h – Policy ID Invalid. =81h – Domain ID Invalid. =82h – Unknown or unsupported Policy Trigger Type. =84h – Power Limit out of range. =85h – Correction Time out of range (use Get Node Manager Capabilities command to get the minimal and maximal correction time for given domain and policy type). =86h – Policy Trigger value out of range. =89h – Statistics Reporting Period out of range. =8Bh – Invalid value of Aggressive CPU Power Correction field or Exception Action invalid for the given policy type. =D4h – Insufficient privilege level due wrong responder LUN =D5h – Policy could not be updated since Policy ID already exists and one of it parameters, which is changing, isn't modifiable when policy is enabled. =D6h – Policies in given power domain cannot be created in the current configuration (e.g. attempt to create predictive power limiting policy in DC power domain). For more details, see Section 3.1.5. Byte 2:4 – Intel Manufacturer ID – 000157h, LS byte first.	
C2h	Get Node Manager Policy	Request Byte 1:3 – Intel Manufacturer ID – 000157h, LS byte first. Byte 4 – Domain ID [3:0] - Domain ID (Identifies the domain that this Intel® Node Manager policy applies to). =00h – Entire platform. =01h – CPU subsystem. =02h – Memory subsystem. =03h – HW Protection*. =04h – High Power I/O subsystem. Others – Reserved. [7:4] – Reserved. Write as 0000b. Byte 5 – Policy ID	Gets the Intel® NM policy parameters. To allow for faster enumeration of all defined policies the error code 80h returns extended error information. Request would be executed even if responder LUN does not match with responder LUN from request when policy was created. *Standard policy in the domain 3 is created automatically and cannot be set by user, but it can be queried. There is MGPIO triggered policy possible to set by user in domain 3, which can also be queried.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in Section 2.15). =80h – Policy ID Invalid. In addition to bytes 2 to 4 extended error information is returned for this error code. =81h – Domain ID Invalid. In addition to bytes 2 to 4 extended error information is returned for this error code.	

Net Function = 2Eh-2Fh LUN = {00b, 01b, 10b, 11b}			
Code	**Command**	**Request, Response Data**	**Description**
		For Completion Code 00h (Success) response bytes 2 to 17 are defined as follows: Byte 2:4 - Intel Manufacturer ID – 000157h, LS byte first. Byte 5 – Domain ID [3:0] - Domain ID (Identifies the domain that this Intel® Node Manager policy applies to). =00h – Entire platform. =01h – CPU subsystem. =02h – Memory subsystem. =03h – HW Protection.* =04h – High Power I/O subsystem. Others – Reserved. [4] – Policy enabled. [5] – Per Domain Intel® Node Manager policy control enabled. [6] – Global Intel® Node Manager policy control enabled. [7] – Set to 1 if policy is created and managed by other management client e.g. DCMI management API, OSPM or responder LUN doesn't match. If policy is managed by external agent it could not be modified by Intel® NM IPMI commands. Byte 6 – Policy Type | Policy Trigger Type [3:0] – Policy Trigger Type =0 – No Policy Trigger, Policy will maintain Power limit (in that case Policy Trigger Value will be equal to the Power Limit). =1 – Inlet Temperature Limit Policy Trigger in [Celsius]. =2 – Missing Power Reading Timeout in 1/10th of second. =3 – Time After Host Reset Trigger in 1/10th of second. =4 – Boot time policy. =5 – Reserved. =6 – MGPIO Policy Trigger. [4] – Policy Type =1 – Power Control Policy. Policy will maintain Power limit. [6:5] – Aggressive CPU Power Correction For policies with Domain ID 0 (Entire platform) and 1 (CPU subsystem) the flag indicates whether Intel® Node Manager can use CPU T-states to control CPU power consumption. =00b – Usage of T-states depends on Shutdown System bit in Policy Exception Actions field. When the bit is set to 1, Intel® NM shall use T-states. When the bit is cleared, Intel® NM does not use T-states. This behavior is backward compatible with Intel® NM 1.5. =01b – Intel® NM is not allowed to use T-states. =10b – Intel® NM is allowed to use T-states. For policies with Domain ID 2 (Memory subsystem) the field shall be set to 0. [7] – Policy storage option =0b – persistent storage (Policy has been saved to the nonvolatile memory). =1b – volatile memory has been used for the policy storing.	

Net Function = 2Eh-2Fh LUN = {00b, 01b, 10b, 11b}			
Code	**Command**	**Request, Response Data**	**Description**
		Byte 7 – Policy Exception Actions (if maintained policy power limit given by bytes 8-9 is exceeded over Correction Time Limit). [0] – Send alert. [1] – Shutdown system. [6:2] – Reserved. Write as 00000b. [7] – The policy power domain. This setting influences the power limiting and reporting of this policy. This field is ignored for NM policies with Domain Id other than 0. =0h - The policy works in the primary side power domain. =1h - The policy works in the secondary (DC) power domain. Byte 8:9 – Power Limit. Byte 10:13 – Correction Time Limit - the maximum time in milliseconds, in which Intel® NM must take corrective actions in order to bring the platform back within the specified power limit before taking the action specified in the "Policy Exception Action" parameter. This is unsigned integer value. The time is counted from the moment when the average power consumption exceeds the power limit. The average power is calculated as arithmetic moving average with the time period equal to the half of Correction Time Limit. It means that Intel® NM may take the exception action after the time period equal to 1.5 of Correction Time Limit parameter starting from the moment when instantaneous power crossed the power limit. Bytes 14:15 – Policy Trigger Limit For the following returned Policy Trigger Type value (Byte 6 bits [3:0]) this field contains the following data: =0 – The same value as Power Limit (i.e. it does not contain the trigger value passed to Intel® NM using Set Node Manager Policy). This is unsigned integer value. =1 – Inlet temperature in Celsius. The inlet temperature value will be compared against this limit and if exceeded, cause a trigger to start enforcing the Power Limit specified (Power limit will not be enforced until the trigger happens). =2 – Time After Host Reset Trigger in 1/10th of second. =3 – Policy Trigger defines time in 1/10 of second after Host reset or startup. If BMC does not send Set Event Receiver command within this time after Host reset or start up Intel® Node Manager will activate this policy. This policy could be defined in addition to the standard limiting policies. The policy is automatically disabled after next Host reset. =4 – Boot time policy. This field is set to 0. =6 – The same value as Power Limit. Byte 16:17 – Statistics Reporting Period The number of seconds that the measured power will be averaged over for the purpose of reporting statistics to external management SW. This is a moving window length. Note that this value is different from the period that Intel® NM uses for maintaining an average for the purpose of power control. This is unsigned integer value. For Completion Code 80h (Policy ID Invalid) response bytes 2 to 6 are defined as follows:	

Net Function = 2Eh-2Fh LUN = {00b, 01b, 10b, 11b}			
Code	**Command**	**Request, Response Data**	**Description**
		Byte 2:4 - Intel Manufacturer ID – 000157h, LS byte first. Byte 5 – Next valid Policy ID. The field contains lowest valid Policy ID that is higher than Policy ID specified in the request for the Domain ID specified in the request. If no such Policy ID exists, zero value is returned. Byte 6 – Number of defined policies for the specified in request Domain ID. Note – This information can be used to query all existing policies within specified domain. Start with Domain ID and Policy ID set to 0. Increment Policy ID treated as an unsigned integer value by one on success and set Policy ID to Byte 5 on reception of Completion Code 80h. For Completion Code 81h (Domain ID Invalid) response bytes 2 to 6 are defined as follows: Byte 2:4 – Intel Manufacturer ID – 000157h, LS byte first. Byte 5 – Next valid Domain ID. [3:0] – Next valid Domain ID. The field contains lowest valid Domain ID that is higher than Domain ID specified in the request. If no such Domain ID exists, zero value is returned. [7:4] – Reserved. Write as 0000b. Byte 6 – Number of available domains. Note – This information can be used to query all available domains. Start with Domain ID and Policy ID both set to 0. Increment Domain ID treated as an unsigned numerical value by one on success and set Domain ID to Byte 5 bits [3:0] on reception of Completion Code 81h.	
C3h	Set Node Manager Policy Alert Thresholds	Request Byte 1:3 – Intel Manufacturer ID – 000157h, LS byte first. Byte 4 – Domain ID [3:0] – Domain ID (Identifies the domain that this Intel® Node Manager policy applies to.) =00h – Entire platform. =01h – CPU subsystem. =02h – Memory subsystem. =03h – HW Protection. =04h – High Power I/O subsystem. Others – Reserved. [7:4] – Reserved. Write as 0000b. Byte 5 – Policy ID Byte 6 – Number of alert thresholds as unsigned integer value.	Sets, updates or deletes the Intel® Node Manager Policy alert thresholds. This is part of the Intel® Node Manager Policy described earlier and applies to the same policy as specified by Policy ID. Update of thresholds once set for given policy is performed by sending a new alert threshold array for this policy.

Net Function = 2Eh-2Fh LUN = {00b, 01b, 10b, 11b}			
Code	**Command**	**Request, Response Data**	**Description**
		Byte 7:N – Alert threshold array (the array length is based on the number of thresholds given in the byte 6). The interpretation of the array's content depends on the Trigger Type of the corresponding policy. For a policy without trigger, the thresholds array contains average power consumption in watts. For a policy with Inlet Temperature Trigger, the array contains temperature in degrees Celsius. For Missing Power Reading Timeout and Time After Host Reset triggers, the array contains time in 1/10 of second. For Boot Time policy, the thresholds cannot be defined. Intel® NM will generate an event if the trigger value or the average power (computed over an averaging period derived from correction time limit) exceeds any of the configured alert thresholds. Assertion is generated for exceeding the threshold (going high) and de-assertion for going low. The hysteresis value for avoiding jitters around the threshold will be OEM configurable using factory-preset values set in spsFITC. Refer to [INT_GUIDE] for details on how to configure threshold hysteresis. Note – Max 3 alert thresholds are supported per policy. Each alert threshold is an unsigned integer 2 bytes in length (LSB first). If number of alert thresholds is 0 then the previously set alert thresholds (if present) are removed from the policy.	Sending of a shorter threshold array deletes the thresholds with indexes outside the size of the new array. Modification of policy alert thresholds would be performed only if responder LUN match with responder LUN from request when policy was created. Note – This command isn't supported for policy with Policy Trigger Type set to Boot time. For Missing Power Reading Timeout deassertion events are sent (in a random order) after the first correct power reading is received. For Time After Host Reset triggers deassertion events are sent (in a random order) after Set Event Receiver command is received from BMC. This command accepts the following threshold ranges: 1..32767 Watts for power policies 0..100 degrees for temperature triggers 1..32767 for time triggers This command accesses the flash, so the response will be sent after completing the operation and may take longer than 250 ms.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in Section 2.15). =80h – Policy ID Invalid. =81h – Domain ID Invalid. =82h – Unknown or unsupported Policy Trigger Type. =84h – Alert Thresholds is out of range. =87h – Number of thresholds is too large or power limits are invalid. =D4h – Insufficient privilege level due wrong responder LUN Byte 2:4 – Intel Manufacturer ID – 000157h, LS byte first.	
C4h	Get Node Manager Policy Alert Thresholds	Request Byte 1:3 – Intel Manufacturer ID – 000157h, LS byte first. Byte 4 – Domain ID [3:0] – Domain ID (Identifies the domain that this Intel® Node Manager policy applies to). =00h – Entire platform. =01h – CPU subsystem. =02h – Memory subsystem. =03h – HW Protection. =04h – High Power I/O subsystem. Others – Reserved. [7:4] – Reserved. Write as 0000b. Byte 5 – Policy ID	Gets the Intel® NM Policy alert thresholds. Request would be executed even if responder LUN does not match with responder LUN from request when policy was created.
		Response	

Net Function = 2Eh-2Fh LUN = {00b, 01b, 10b, 11b}			
Code	**Command**	**Request, Response Data**	**Description**
		Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in Section 2.15). =80h – Policy ID Invalid. =81h – Domain ID Invalid. Byte 2:4 – Intel Manufacturer ID – 000157h, LS byte first. Byte 5 – Number of alert thresholds as unsigned integer value. Byte 6:N – Alert threshold array (the array length is based on the number of threshold given in the byte 5). If number of alert thresholds is 0 then the array length is 0 bytes. The interpretation of the content of the array depends on the Trigger Type of the corresponding policy. For a policy without trigger the thresholds array contains average power consumption in Watts. For a policy with Inlet Temperature Trigger the array contains temperature in Celsius degrees. For Missing Power Reading Timeout and Time After Host Reset triggers the array contains time 1/10th of second. For Boot Time policy the thresholds cannot be defined. Note – Max 3 alert thresholds are supported per policy. Each alert threshold is 2 bytes in length (LSB first). All alert thresholds are unsigned integer values.	
C5h	Set Node Manager Policy Suspend Periods	Request Byte 1:3 – Intel Manufacturer ID – 000157h, LS byte first. Byte 4 – Domain ID [3:0] - Domain ID (Identifies the domain that this Intel® Node Manager policy applies to). =00h – Entire platform. =01h – CPU subsystem. =02h – Memory subsystem. =03h – HW Protection.* =04h – High Power I/O subsystem. Others – Reserved. [7:4] – Reserved. Write as 0000b. Byte 5 – Policy ID Byte 6 – Number of policy suspend periods. This value should be specified as 0 if all the suspend periods are to be removed (if previously set). This is an unsigned integer value. Byte 7:N – array of policy suspend periods (following information is repeated for each suspend period). Each suspend period is defined by 3 bytes: 1st byte – Policy suspend start time. It is 0 – 239 number of minutes from midnight divided by 6. Values 240 – 255 are reserved. 2nd byte – Policy suspend stop time. It is 1 – 240 number of minutes from midnight divided by 6. Values 0 and 241 – 255 are reserved.	Sets the Intel® Node Manager Policy suspend period (during which no platform power policy control will be enforced). The suspend periods are applied only if Intel® ME has valid RTC time. If RTC synchronization existing periods are ignored. Note that RTC synchronization failure situation is signaled with Intel® NM Health Event (see Section 3.5.7) Modification of policy suspend periods would be performed only if responder LUN match with responder LUN from request when policy was created. This command accesses the flash, so the response will be sent after completing the operation and may take longer than 250 ms.

Net Function = 2Eh-2Fh LUN = {00b, 01b, 10b, 11b}			
Code	**Command**	**Request, Response Data**	**Description**
		3rd byte – Suspend period recurrence pattern: [0] – Repeat suspend period every Monday. [1] – Repeat suspend period every Tuesday. [2] – Repeat suspend period every Wednesday. [3] – Repeat suspend period every Thursday. [4] – Repeat suspend period every Friday. [5] – Repeat suspend period every Saturday. [6] – Repeat suspend period every Sunday. [7] – Reserved. Write as 0b. Note – Policy suspend start and stop time is 1 byte in length each. Max 5 suspend periods can be specified per policy. If the number of policy suspend period (i.e. byte 6) is 0, the rest of the bytes in the request message are not required and previously configured suspend periods are removed from the system for the specified policy ID. The suspend periods are specified as an array. For e.g. if policy suspend start time is in byte 7 then byte 8 will contain the policy suspend stop time and byte 9 will contain suspend period recurrence pattern. Similarly, if the second set of suspend periods are to be specified, then they will be present in bytes 10:12. The suspend times are encoded on one byte each as number of minutes from midnight divided by 6 to fit into one byte. If there is a need to specify an end-time that is beyond midnight, use two suspend periods, one ending at midnight (suspend stop time byte set to 240) and one from midnight (suspend start time set to 0) until the necessary end-time of the next day.	Note – This command is not supported for policies with trigger type set to Boot time, Time After Host Reset or Missing Power Readings. *Suspend periods are only available for the MGPIO triggered policy (Policy Trigger Type 6).
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in Section 2.15). =80h – Policy ID Invalid. =81h – Domain ID Invalid. =82h – Unknown or unsupported Policy Trigger Type. =85h – One of periods in the table is inconsistent. Start time is greater than or equal to stop time or stop time sets time beyond 1 day. =87h – Number of policy suspend periods invalid. =D4h – Insufficient privilege level due wrong responder LUN Byte 2:4 – Intel Manufacturer ID – 000157h, LS byte first.	
C6h	Get Node Manager Policy Suspend Periods	Request Byte 1:3 – Intel Manufacturer ID – 000157h, LS byte first. Byte 4 – Domain ID [3:0] – Domain ID (Identifies the domain that this Intel® Node Manager policy applies to). =00h – Entire platform. =01h – CPU subsystem. =02h – Memory subsystem. =03h – HW Protection.* =04h – High Power I/O subsystem. Others – Reserved. [7:4] – Reserved. Write as 0000b. Byte 5 – Policy ID	Get the Intel® NM Policy suspend periods Request would be executed even if responder LUN doesn't match with responder LUN from request when policy was created. *Suspend periods are only available for the MGPIO triggered policy (Policy Trigger Type 6).

Net Function = 2Eh-2Fh LUN = {00b, 01b, 10b, 11b}			
Code	**Command**	**Request, Response Data**	**Description**
		Response Byte 1 – Completion Code = 00h – Success (Remaining standard Completion Codes are shown in Section 2.15). = 80h – Policy ID Invalid. = 81h – Domain ID Invalid. = 82h – Unknown or unsupported Policy Trigger Type. Byte 2:4 – Intel Manufacturer ID – 000157h, LS byte first. Byte 5 – Number of policy suspend periods. Byte 6:N – array of suspend periods. Each suspend period is defined by 3 bytes: 1st byte – Policy suspend start time encoded as a number of minutes from midnight divided by 6. 2nd byte – Policy suspend stop time encoded as a number of minutes from midnight divided by 6. 3rd byte – Suspend period recurrence pattern: [7] – Reserved. Write as 0b. [6] – Repeat suspend period every Sunday. [5] – Repeat suspend period every Saturday. [4] – Repeat suspend period every Friday. [3] – Repeat suspend period every Thursday. [2] – Repeat suspend period every Wednesday. [1] – Repeat suspend period every Tuesday. [0] – Repeat suspend period every Monday. Note – If byte 5 is 00h then no subsequent bytes will be present in the response. This means that there are no suspend periods configured for the specified policy Id. Note – The suspend periods are specified as an array. For example, if first suspend period is in bytes 6:8 then bytes 9:11 will contain the second policy suspend period.	
C7h	Reset Node Manager Statistics	Request Byte 1:3 – Intel Manufacturer ID – 000157h, LS byte first. Byte 4 – Mode [4:0] – Mode =00h – Reset global statistics including power statistics, throttling statistics, inlet temperature statistics, volumetric airflow statistics, outlet temperature statistics, chassis power statistics and energy accumulators. =01h – Reset per policy statistics including power, throttling statistics and trigger statistics. =02h – 1Ah – Reserved. =1Bh – Reset global Host Unhandled Requests statistics. =1Ch – Reset global Host Response Time statistics. =1Dh – Reset global CPU throttling statistics. =1Eh – Reset global memory throttling statistics. =1Fh – Reset global Host Communication Failure statistics. [7:5] – Reserved. Write as 000b.	This command clears Intel® NM statistics of given type. Note that all statistics get cleared at Intel® ME restart. Thus in a system where Intel® ME works in S0/S1 Only power mode the statistics get cleared every time the system enters any sleeping states.

Net Function = 2Eh-2Fh LUN = {00b, 01b, 10b, 11b}			
Code	**Command**	**Request, Response Data**	**Description**
		Byte 5 – Domain ID [3:0] – Domain ID (Identifies the domain that this Intel® Node Manager policy applies to) =00h – Entire platform. =01h – CPU subsystem. =02h – Memory subsystem. =03h – HW Protection. =04h – High Power I/O subsystem. Others – Reserved. Note – For Mode (byte 4) in a range 1Bh – 1Fh Domain ID must be set to 00h. [7:4] – Reserved. Write as 0000b. Byte 6 – Policy ID (ignored if field Mode isn’t set to 01h).	Per policy reset statistics (mode 01h) requests would be executed only if responder LUN matches with responder LUN from request when policy was created. Responder LUN value doesn’t affect execution of requests for other modes.
		Response Byte 1 – Completion code =00h – Success (Remaining standard Completion Codes are shown in Section 2.15). =80h – Policy ID Invalid =81h – Domain ID Invalid =88h – Invalid Mode =D4h – Insufficient privilege level due wrong LUN. Byte 2:4 – Intel Manufacturer ID – 000157h, LS byte first.	
C8h	Get Node Manager Statistics	Request Byte 1:3 – Intel Manufacturer ID – 000157h, LS byte first. Byte 4 – Mode [4:0] – Mode =01h – Global power statistics in [Watts] =02h – Global inlet temperature statistics in [Celsius]. =03h – Global throttling statistics [%] =04h – Global volumetric airflow statistics [1/10th of CFM] =05h – Global outlet airflow temperature statistics [Celsius] =06h – Global chassis power statistics [Watts] =07h – Reserved =08h – Global energy accumulator [mJ] =09h – 10h – Reserved =11h – Per policy power statistics in [Watts] =12h – Per policy trigger statistics in [Celsius] =13h – Per policy throttling statistics in [%] =14h – 1Ah – Reserved =1Bh – Global Host Unhandled Requests statistics. =1Ch – Global Host Response Time statistics =1Dh – Global CPU throttling statistics (deprecated, Mode=03h, Domain ID=01h shall be used instead) =1Eh – Global memory throttling statistics (deprecated, Mode=03h, Domain ID=02h shall be used instead) =1Fh – Global Host Communication Failure statistics [7:5] – Reserved. Write as 000b.	This command provides statistics of requested type. The statistics are collected since last Intel® ME restart or since it was cleared with Reset Intel® NM Statistics command. Note that the average values provided here may be different from the averaged values used by Intel® Node Manager for taking corrective action or triggering alerts based a ‘Set Node Manager Alert Threshold’ because the averaging periods could be different.

Net Function = 2Eh-2Fh LUN = {00b, 01b, 10b, 11b}			
Code	**Command**	**Request, Response Data**	**Description**
		Byte 5 – Domain ID [3:0] – Domain ID (Identifies the domain that this Intel® Node Manager policy applies to) =00h – Entire platform =01h – CPU subsystem =02h – Memory subsystem =03h – HW Protection =04h – High Power I/O subsystem Others – Reserved. Note: For Mode in a range 1Bh – 1Fh Domain ID must be set to 00h. [4] – Power domain. This field is ignored for Domain ID other than 00h. =0h – Statistics shall be reported for the primary side power domain (AC). =1h – Statistics shall be reported for the secondary side power domain (DC). [6:5] – Reserved. Write as 000b. [7] – Per-component control – allows for getting data for chosen domain component =0 – Accumulated data for whole domain will be returned =1 – Data from single component in domain will be returned and Byte 6 contains component identifier Byte 6 – Policy ID / Component Identifier For Modes 01h and 08h if bit 7 in Byte 5 is set to 1 then this field indicates component identifier for which data will be returned. Refer to Section 3.1.6 for description of this field. For modes 11h-13h this field indicates the Policy ID for which the statistics are requested. For all other modes this field is ignored.	Modes for per policy statistics are correlated to policy trigger type. A request for mode 11h can be only issued for policies with trigger type 0 (no trigger defined) and 3 (Time After Host Reset). Mode 12h is available only for policies with trigger type set to 1 (Inlet Temperature). Mode 13h can be issued for policies with trigger type 2 (Missing Power Readings). Power Domain and Per-Component Control parameters are valid for Mode 01h and 08h. For other values of Mode, the parameters should be set to 0. Note that Global CPU throttling statistics (1Dh) and Global memory throttling statistics (1Eh) provide information about actual available performance of the platform that is adjusted to the defined power cap limit and the load that runs on the platform. Current Value of those statistics will always be greater than 0 if the policy is triggered and is actively limiting to the defined power limit (restore Byte 21 [7]- Policy activation state bit set to 1). Because Intel® NM is continuously adjusting available platform performance this will be true even if the power consumption will be below the policy power limit. In order to collect reliable statistics it is recommended to issue the “Reset Intel® NM Statistics” (C7h) before issuing “Get Intel® NM Statistics” first time after a DC cycle. Note that for Mode (byte 4) in a range 02h, 04h-06h and 1Bh – 1Fh Domain ID must be set to 00h.
		Response Byte 1 – Completion code =00h – Success (Remaining standard Completion Codes are shown in Section 2.15). =80h – Policy ID / Component Identifier Invalid. =81h – Domain ID Invalid. =88h – Invalid Mode. =CCh – Power domain not supported for given Domain ID and Mode combination. Byte 2:4 – Intel Manufacturer ID – 000157h, LS byte first. Byte 5:12 Statistics data For Mode set to 08h (Global energy accumulator); Byte 5:12 Accumulated energy value (unsigned integer) [mJ] For other Modes; Byte 5:6 – Current Value* see 3.1.1 Byte 7:8 – Minimum Value* see 3.1.2 Byte 9:10 – Maximum Value* see 3.1.3 Byte 11:12 – Average Value* see 3.1.4 For Mode 08h (Global energy accumulator); Byte 13:16 – Timestamp of the last collected sample in milliseconds. For other Modes;	

Net Function = 2Eh-2Fh LUN = {00b, 01b, 10b, 11b}			
Code	**Command**	**Request, Response Data**	**Description**
		Byte 13:16 – Timestamp as defined by the IPMI v2.0 specification indicating when the response message was sent. If Intel® NM cannot obtain valid time, the timestamp is set to FFFFFFFFh as defined in the IPMI v2.0 specification. Byte 17:20 – Statistics Reporting Period (the timeframe in seconds, over which the firmware collects statistics). This is unsigned integer value. For all global statistics this field contains the time after the last statistics reset. Byte 21 – Domain ID | Policy State [3:0] – Domain ID (Identifies the domain that this Intel® Node Manager policy applies to) =00h – Entire platform. =01h – CPU subsystem. =02h – Memory subsystem. =03h – HW Protection. =04h – High Power I/O subsystem. =Others – Reserved. [4] – Policy/Global Administrative state. If request Byte 4 is in range 11h–13h state =1 – If policy is enabled by user and Intel® Node Manager Policy Control is Globally Enabled (see C0h command) and Intel® Node Manager Domain control is also Enabled (see C0h command). =0 – Otherwise. If request Byte 4 is in range 01h–08h state = 1 – if Intel® Node Manager Policy Control is Globally Enabled (see C0h command). =0 – Otherwise. If request Byte 4 is in range 1Bh–1Fh state = Reserved [5] – Policy Operational state If request Byte 4 is in range 01h–08h or 1Bh–1Fh state =Reserved. If request Byte 4 is in range 11h–13h state =1 – Policy is actively monitoring defined trigger (power or thermal) and will start enforcing the power limit if defined trigger is exceeded. =0 – Policy is suspended so it cannot actively limit to defined power limit. It may happen if one of the following defined events happens: – Suspend period is enforced. – There is a problem with trigger readings. – There is a host communication problem. – Host is in Sx state. – Host did not send End Of POST notification. – Policy limit is below minimal power draw range. – Policy is administratively disabled. [6] – Measurements state If request Byte 4 is in range 01h–08h or 11h–13h state =1 – Measurements in progress =0 – Measurements stopped or readings problems detected	Per policy get statistics (modes 11h-13h) requests would be executed even if responder LUN doesn't match with responder LUN from request when policy was created. LUN value does not affect execution of requests for other modes. *encoded as 16-bit 2s-complement signed integer

Net Function = 2Eh-2Fh LUN = {00b, 01b, 10b, 11b}			
Code	**Command**	**Request, Response Data**	**Description**
		Note: Measurements state depends on readings source availability in given power state. For some devices readings in Sx states aren't available (e.g., CPU) thus for measurements related to these devices this bit could be cleared as a result of entering not supported power state or reading problem detection in supported state. For devices that are available in all power states this bit would remain set as long as the devices provide proper readings. If request Byte 4 is in range 1Bh–1Fh state =Reserved. [7] – Policy activation state If request Byte 4 is in range 01h–08h or 1Bh–1Fh state =Reserved. If request Byte 4 is in range 11h–13h state =1 – Policy is triggered and is actively limiting to the defined power limit. =0 – Policy is not triggered. Note: For Host Response Time, Host Unhandled Requests, Host Communication Failure, CPU, and memory throttling statistics Byte 21 is always set to 0.	
C9h	Get Node Manager Capabilities	Request Byte 1:3 – Intel Manufacturer ID – 000157h, LS byte first. Byte 4 – Domain ID [3:0] – Domain ID (Identifies the domain that this Intel® Node Manager policy applies to) =00h – Entire platform. =01h – CPU subsystem. =02h – Memory subsystem. =03h – HW Protection. =04h – High Power I/O subsystem. Others – Reserved. [7:4] – Reserved. Write as 0000b. Byte 5 – Policy Type | Policy Trigger Type [3:0] – Policy Trigger Type =0 – No Policy Trigger. =1 – Inlet Temperature Policy Trigger value in [Celsius]. =2 – Missing Power Reading Timeout in 1/10th of second. =3 – Time After Host Reset Trigger in 1/10th of second. =4 – Boot time policy. =5 – Reserved. =6 – MGPIO Policy Trigger. Others – Reserved. [6:4] – Policy Type =1 – Power Control Policy. Others – Reserved. [7] – The policy power domain. This field is ignored for Domain Id other than 0. =0h - For policies working in the primary side power domain. =1h - For policies working in the secondary (DC) power domain.	Get Intel® Node Manager capabilities. This command would be executed regardless of responder LUN value. Common setting is returned for all LUNs.
		Response	

Net Function = 2Eh-2Fh LUN = {00b, 01b, 10b, 11b}			
Code	**Command**	**Request, Response Data**	**Description**
		Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in Section 2.15). =81h – Domain ID invalid or not supported in the current configuration. =82h – Unknown or unsupported Policy Trigger Type. =83h – Unknown Policy Type. Byte 2:4 – Intel Manufacturer ID – 000157h, LS byte first. Byte 5 – Max Concurrent Settings – number of policies supported for the given policy trigger type, policy type and Domain ID. Byte 6:7 Max Power/Thermal/Time After Reset If Policy Trigger Type in the request equals 0 or 6 (No Policy Trigger or MGPIO Policy Trigger) then this field contains maximum power Limit to be maintained. The power value is expressed in [Watts] as unsigned integer value. It can be either received from BIOS (for total power limit in domain 0) or read by Intel® NM from CPU (for CPU and memory power limit in domains 1 and 2). It can also be set manually using Set Power Draw Range command. If Intel® NM does not impose any limit or hardware capabilities are currently not known this field is equal to 0x7FFF. If Policy Trigger Type in the request equals 1 (Inlet Temperature Policy Trigger) then this field contains maximum temperature value to be settable as trigger. The temperature is expressed in [Celsius] as unsigned integer value. If Policy Trigger Type in the request equals 2 (Missing Power Reading Timeout) or 3 (Time After Host Reset) then this field contains maximum time to be settable as trigger. The time is expressed in 1/10 seconds as unsigned integer value. If Policy Trigger Type in the request equals 4 (Boot time policy) then this field is not applicable and is set to 0. Byte 8:9 – Min Power/Thermal/Time After Reset If Policy Trigger Type in the request equals 0 or 6 (No Policy Trigger or MGPIO Policy Trigger) then this field contains minimum power Limit to be maintained. The power value is expressed in [Watts] as unsigned integer value. It can be either received from BIOS (for total power in domain 0) or read by Intel® NM from CPU (for CPU and memory power limit in domains 1 and 2). It can also be set manually using Set Power Draw Range command. If Intel® NM does not impose any limit or hardware capabilities are currently not known this field is equal to 0x01. If Policy Trigger Type in the request equals 1 (Inlet Temperature Policy Trigger) then this field contains minimum temperature value to be settable as trigger. The temperature is expressed in [Celsius] as unsigned integer value.	

Net Function = 2Eh-2Fh LUN = {00b, 01b, 10b, 11b}			
Code	**Command**	**Request, Response Data**	**Description**
		If Policy Trigger Type in the request equals 2 (Missing Power Reading Timeout) or 3 (Time After Host Reset) then this field contains minimum time to be settable as trigger. The time is expressed in 1/10 seconds as unsigned integer value. If Policy Trigger Type in the request equals 4 (Boot time policy) then this field is not applicable and is set to 0. Byte 10:13 – Min Correction Time settable in milliseconds as unsigned integer value. Byte 14:17 – Max Correction Time settable in milliseconds as unsigned integer value. Byte 18:19 – Min Statistics Reporting Period in seconds as unsigned integer value. Byte 20:21 – Max Statistics Reporting Period in seconds as unsigned integer value. Byte 22 – Domain limiting scope [0:3] – Domain ID (Identifies the domain that this Intel® Node Manager policy applies to) =00h – Entire platform. =01h – CPU subsystem. =02h – Memory subsystem. =03h – HW Protection. =04h – High Power I/O subsystem Others – Reserved. [4:6] – Reserved. Write as 000b. [7] – Domain limiting based on* =0 – Primary side (Wall input) power – Intel® NM has access to PSU input power. =1 – DC power only – PSU output power or bladed system or direct DC reading from the CPU. * This setting applies also to global statistics reporting and power draw range of given policy Domain ID and Trigger Type. Policies with Domain ID 00h may be configured to work in different power domain on per policies basis.	
CAh	Get Intel® Node Manager Version	Request Byte 1:3 – Intel Manufacturer ID – 000157h, LS byte first.	Get Intel® Node Manager firmware version. Major Firmware revision and Minor Firmware revision unambiguously identify firmware release. For every release, at least one of these numbers changes. This command would be executed regardless of responder LUN value. Common setting is returned for all LUNs.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in Section 2.15). Byte 2:4 – Intel Manufacturer ID – 000157h, LS byte first. Byte 5 – Intel® NM version =01h – Supported Intel® NM 1.0 – one power policy. =02h – Supported Intel® NM 1.5 – multiple policies and thermal triggers for power policy. =03h – Supported Intel® NM 2.0 – multiple policies and thermal triggers for power policy. =04h – Supported Intel® NM 2.5 =05h – Supported Intel® NM 3.0 =06h – Supported Intel® NM 4.0 =07h – Supported Intel® NM 5.0 =08h – FFh – Reserved for future use.	

Net Function = 2Eh-2Fh LUN = {00b, 01b, 10b, 11b}			
Code	**Command**	**Request, Response Data**	**Description**
		Byte 6 – IPMI interface version =01h – Intel® NM IPMI version 1.0. =02h – Intel® NM IPMI version 2.0 =03h – Intel® NM IPMI version 3.0 =04h – Intel® NM IPMI version 4.0 =05h – Intel® NM IPMI version 5.0 (version defined in this document). Byte 7 – Patch version (binary encoded). Note – Change on this byte does not impact IPMI interface (Byte 6) nor Intel® NM version (Byte 5). Should be set to 0h if patch version is not used by the firmware. Byte 8 – Major Firmware revision (binary encoded) – identifies current build of the code –and should contain the same value as the Get Device Id command's response byte 4 [6:0] – Major firmware revision. Note – Change on this byte does not impact IPMI interface (Byte 6) nor Intel® NM version (Byte 5). Byte 9 – Minor Firmware revision (BCD encoded) – identifies current build of the code and should contain the same value as the "Get Device Id" command response byte 5 Minor firmware revision. Note – Change on this byte does not impact IPMI interface (Byte 6) nor Intel® NM version (Byte 5).	
CBh	Set Node Manager Power Draw Range	Request Byte 1:3 – Intel Manufacturer ID – 000157h, LS byte first. Byte 4 – Domain ID [3:0] – Domain ID (Identifies the domain that this setting applies to) =00h – Entire platform. =01h – CPU subsystem. =02h – Memory subsystem. =03h – HW Protection. =04h – High Power I/O subsystem. Others – Reserved. [7:4] – Reserved. Write as 0000b. Byte 5:6 – Minimum Power Draw in [Watts]. For domain 00h, if set to 0 the minimum power draw value will be invalidated and no validation of policy parameters against minimum power consumption will be performed. For domain 01h and 02h, if set to zero, minimum power draw will be obtained from CPU via PECI. This is an unsigned integer value in a range <0:7FFFh>. Byte 7:8 – Maximum Power Draw in [Watts]. For domain 00h, if set to 0 the maximum power draw value will be invalidated and no validation of policy parameters against maximum power consumption will be performed. For domain 01h and 02h, if set to zero, maximum power draw will be obtained from CPU via PECI. For domain 03h on modular systems defines the actual limit for HW Protection Policy. This is an unsigned integer value in a range <0:7FFFh>.	Set the Min/Max power consumption ranges. For all domains except HW Protection Domain this information is preserved in the persistent storage. After receiving the request Intel® NM validates whether there are any policies with limit below the new minimum power draw range. If Intel® NM detects such policies, it sends Intel® NM Health Event with Policy Misconfiguration flag set and suspends the policies (sets their operational state to 0) until the minimal power draw or policy limit changes so that the policy becomes valid again. The same action is taken, when Intel® NM receives a new power draw range from CPU via PECI at POST.

Net Function = 2Eh-2Fh LUN = {00b, 01b, 10b, 11b}			
Code	**Command**	**Request, Response Data**	**Description**
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in Section 2.15). =C9h – Parameter out of range. Returned when Minimum Power Draw Range exceeds the Maximum one. =81h – Domain ID invalid or not supported in the current configuration. Returned for domain ID 03, when PSU is configured as the power reading source. Byte 2:4 – Intel Manufacturer ID – 000157h, LS byte first.	In case Maximum Power Draw is set to 0W for HW Protection domain, HW Protection policy limit is cleared to default value of 32767W. This command would be executed regardless of responder LUN value however there is only one global setting that would be updated. New value would modify settings for other responder LUNs. Note: Power Draw Range for Domain ID 03 is configured automatically when a compatible power supply is configured as the power reading source. This command accesses the flash, so the response will be sent after completing the operation and may take longer than 250 ms.
CEh	Set Node Manager Alert Destination	Request Byte 1:3 – Intel Manufacturer ID – 000157h, LS byte first. Byte 4 – Channel number [3:0] – BMC channel number over which to send the alert from BMC to management console. Alerts can be sent to only one console. [4:6] – Reserved. Write as 000b. [7] – Alert receiver deactivation =0 – Register alert receiver. If there is an active receiver already, it becomes invalid and stops receiving alerts. =1 – Unregister alert receiver. Use this bit to invalidate the current destination configuration. Alerts will be blocked. Byte 5 – Destination Information [0:3] – Destination selector. Selects which alert destination should go to =0h – Use volatile destination info. =1h – Fh – Use nonvolatile destination info. [4:7] – Reserved. Write as 0000b. Byte 6 – Alert String Selector. Selects which Alert String, if any, to use with the alert. [6:0] – String selector. =00h – Use volatile Alert String. =01h – 7Fh – Use nonvolatile string selector. Alert String Selector definition is the same as in the "Set/Get PEF Configuration Parameters" command. [7] =0b – Do not send Alert String. =1b – Send Alert String identified by following string selector.	Provide alert destination information for Intel® Node Manager to send direct alerts that bypass the BMC SEL. Destination Selector/Operation and Alert String Selector fields correspond to the associated configuration parameters applicable to the BMC channel number over which to send the alert. This command would be executed regardless of responder LUN value however there is only one global setting that would be updated. New value would modify settings for other responder LUNs.

Net Function = 2Eh-2Fh LUN = {00b, 01b, 10b, 11b}			
Code	**Command**	**Request, Response Data**	**Description**
		Response Byte 1 – Completion Code = 00h – Success (Remaining standard Completion Codes are shown in Section 2.15). Byte 2:4 – Intel Manufacturer ID – 000157h, LS byte first.	
CFh	Get Node Manager Alert Destination	Request Byte 1:3 – Intel Manufacturer ID – 000157h, LS byte first.	Provides alert destination information that is used to send alerts from for Intel® Node Manager. This command would be executed regardless of responder LUN value. Common setting is returned for all responder LUNs.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in Section 2.15). Byte 2:4 – Intel Manufacturer ID – 000157h, LS byte first. Byte 5 – Channel number [3:3] – BMC channel number over which alert from BMC to management console will be sent. [6:4] – Reserved. Write as 000b. [7] – Alert receiver deactivation =0 – Configuration valid. Alert receiver registered. =1 – Configuration invalid. Alert receiver not registered. Alerts are blocked. Byte 6 – Destination Selector/Operation [3:0] – Destination selector. Selects which alert destination should go to. =0h – Use volatile destination info. =1h – Fh – Use nonvolatile destination info. [7:4] – reserved. Write as 00000b. Byte 7 – Alert String Selector. Selects which Alert String, if any, to use with the alert. [6:0] – String selector =00h – Use volatile Alert String. =01h – 7Fh – Use nonvolatile string selector. [7] =0b – Do not send Alert String. =1b – Send Alert String identified by following string selector.	

3.1.1 Get Intel® NM Statistics Current Value Field

For global power statistics, per policy power statistics and chassis power statistics this field contains the recently measured power consumption. It is calculated as an arithmetical average of samples collected in a one second time period. The value is expressed in Watts.

For global and per policy throttling level statistics, this field contains the recently measured throttling level applied in given domain. It is calculated as arithmetical average of samples collected in a one second time period. The value is expressed as percentages. A value of 100% means the system is throttled to the maximum level that Intel® NM can enforce. A value of 0% means the system is not throttled at all (i.e., it is running on maximum performance level).

For Host Communication Failure statistics if Host Communication Failure is detected, the field contains the time in 1/100 of second between the moment Intel® NM switched to Host Communication Failure and the moment of sending the response. If OSPM responds to Intel® NM requests the field contains 0.

For Host Response Time statistics, the field contains the recently measured time in 1/100 of second between Intel® NM requesting P/T-state change and ASL code acknowledging the change.

For CPU throttling and memory throttling statistics the field contains the CPU or memory throttling level applied by Intel® NM at the moment of receiving the command. The throttling level is expressed as a percentage. A value of 100% means the CPU/memory is throttled to the maximum level that Intel® Node Manager can enforce. A value of 0% means that the CPU/memory is not throttled at all (i.e., they are running on maximum performance level).

For Host Unhandled Requests statistics the field contains the number of P/T/PUR-state change requests not handled properly by the OSPM.

3.1.2 Get Intel® NM Statistics Minimum Value Field

For global power statistics, per policy power statistics and chassis power statistics this field contains the minimum measured power consumption after last reset of statistics. The value is expressed in Watts.

For global and per policy throttling level statistics this field contains the minimum measured throttling level applied in given domain after last reset of statistics. The value is expressed a percentage. A value of 100% means the system is throttled to the maximum level that Intel® NM can enforce. A value of 0% means that the system is not throttled at all (i.e. it is running on maximum performance level).

For Host Communication Failure statistics, the field contains the shortest time during which Intel® NM stayed in Host Communication Failure mode. It does not include the time indicated in Current field. The time is expressed in 1/100 of second.

For Host Response Time statistics, the field contains the minimum time measured between Intel® Node Manager requesting P/T-state change and ASL code acknowledging the change. The time is expressed in 1/100 of second.

For CPU throttling and memory throttling statistics the field contains the minimum CPU or memory throttling level applied by Intel® NM after the last reset of statistics. The throttling level is expressed as a percentage. A value of 100% means that the CPU/memory is throttled to the maximum level the Intel® Node Manager can enforce. A value of 0% means that the CPU/memory is not throttled at all (i.e. they are running on maximum performance level).

For Host Unhandled Requests statistics, the field always contains zero.

3.1.3 Get Intel® NM Statistics Maximum Value Field

For global power statistics, per policy power statistics and chassis power statistics this field contains the maximum measured power consumption after last reset of statistics. The value is expressed in Watts.

For global and per policy throttling level statistics, this field contains the maximum measured throttling level applied in given domain after last reset of statistics. The value is expressed as a percentage. A value of 100% means the system is throttled to the maximum level the Intel® NM can enforce. A value of 0% means that the system is not throttled at all (i.e., it is running on maximum performance level).

For Host Communication Failure statistics, the field contains the longest time during which Intel® NM stayed in Host Communication Failure mode. It does not include the time indicated in Current field. The time is expressed in 1/100 of second.

For Host Response Time statistics, the field contains the maximum time measured between Intel® Node Manager requesting P/T-state change and ASL code acknowledging the change. The time is expressed in 1/100 of second.

For CPU throttling and memory throttling statistics, the field contains the maximum CPU or memory throttling level applied by Intel® NM after the last reset of statistics. The throttling level is expressed as a percentage. A value of 100% means the CPU/memory is throttled to the maximum level the Intel® NM can enforce. A value of 0% means that the CPU/memory is not throttled at all (i.e., they are running on maximum performance level).

For Host Unhandled Requests statistics, the field contains the number of P/T/PUR-state change requests not handled properly by the OSPM.

3.1.4 Get Intel® NM Statistics Average Value Field

For global power statistics, per policy power statistics and chassis power statistics, this field contains the average measured power consumption. The average is calculated as arithmetic average after last statistics reset. The value is expressed in Watts.

For global and per policy throttling level statistics, this field contains the minimum measured throttling level applied in given domain. The average is calculated as arithmetic average after last statistics reset. The value is expressed as a percentage. A value of 100% means the system is throttled to the maximum level that Intel® Node Manager can enforce. A value of 0% means that the system is not throttled at all (i.e., it is running on maximum performance level).

For Host Communication Failure statistics, the field contains the percentage of time Intel® NM worked in Host Communication Failure mode from the recent statistics reset.

For Host Response Time statistics, the field contains the arithmetic average of all the Host response time measurements. The time is expressed in 1/100 of second.

For CPU throttling and memory throttling statistics the field contains the average CPU or memory throttling level applied by Intel® NM after the last reset of statistics. The average is calculated as arithmetic average after last statistics reset. The throttling level is expressed as a percentage. A value of 100% means the CPU/memory is throttled to the maximum level that Intel® NM can enforce. A value of 0% means that the CPU/memory is not throttled at all (i.e. they are running on maximum performance level).

For Host Unhandled Requests statistics, the field always contains zero.

3.1.5 Policy Modification

Intel® NM provides the possibility to modify some of the policies parameters during policy working. Modification of different parameters affects policy behavior in different ways.

All policy parameters from Set Intel® NM Policy command are listed with impact on policy behavior if changed:

Domain ID (Byte 4 [3:0]) – not possible to modify if policy is enabled. If policy is disabled it would be moved to new domain if storage for it would be present (in each domain number of policies of different type could be created is limited).

Policy Enable (Byte 4 [4]) – changing of this parameter is similar to execution of Enable/Disable Node Manager Policy Control with per policy parameters.

Policy ID (Byte 5) – not applicable as Policy ID is unique.

Policy Trigger Type (Byte 6 [3:0]) – not possible to modify if policy is enabled. If policy is disabled new policy would be created if storage for it would be present (in each domain number of policies of different type could be created is limited).

Policy Configuration Action (Byte 6 [4]) – if changed to 0 policy would be removed. If changed to 1 policy would be modified or created.

Aggressive CPU Power Correction (Byte 6 [6:5]) – new policy aggressiveness would be applied in Correction Time period.

Policy Storage Option (Byte 6 [7]) – if changed to 1 policy would be removed from persistent storage. If change to 0 policy would be storage in persistent memory.

Policy Exception Action (Byte 7 [1:0]) – if changed exception action timeout is reset and new exception action, if needed would be taken after next Correction Time.

Policy Target Limit (Byte 9:8) – new policy limit would be applied in next Correction Time period.

Correction Time Limit (Byte 13:10) – all measurements based on Correction Time would be reset (e.g. limiting algorithm, exception action timeout).

Policy Trigger Limit (Byte 15:14) – new policy trigger limit would be applied in next Correction Time period.

Statistics Reporting Period (Byte 17:16) – if changed all policy statistics would be reset.

During execution of Set Intel® NM Policy Alert Thresholds command for enabled policy updated thresholds state is checked;

- If a new policy Alert Threshold is added then its current assertion state is always set to de-asserted so an assertion event will be generated during the Threshold creation only if the new threshold is already crossed.
- If a threshold value is updated for an existing policy Alert Threshold then the threshold's assertion state is not modified, thus an event will be generated only if the new threshold value changes the assertion state from de-asserted to asserted or vice versa (e.g., if before modification it was asserted and after update it is still asserted, then no event will be generated).
- In case of removing Alert Threshold, its current assertion state is lost, thus in case of recreation, the policy Alert Threshold will be treated as a newly created threshold (with the assertion state set to de-asserted).

If Set Node Manager Policy Suspend Periods is executed on enabled policy new configuration would be applied in 1 second time period.

3.1.6 Component Identifier

Intel® Intelligent Power Node Manager does not only distinguish domains but also allows execution of some operations on per domain component basis. This is enabled via Component Identifier field. Component identifier depends on its domain.

- For Domain ID 01h (CPU subsystem) component identifier is the CPU socket number (0–7) where 0 refers to lowest numbered address on the PECI bus.
- For Domain ID 02h (memory subsystem) component identifier is the CPU socket number (0–7) where 0 refers to lowest numbered address on the PECI bus associated with memory channels.

3.2 Local Platform Intel® NM Configuration and Control Commands

The following commands should not be exposed to the external software. Only BMC may use the following commands.

Note: None of the following commands are supported when Intel® NM Feature Enabled is set to false using Flash Image Tool.

Table 3-2 Local Platform Intel® NM Configuration and Control Commands

Net Function = 2Eh-2Fh LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
D0h	Set Total Power Budget	Request Byte 1:3 – Intel Manufacturer ID – 000157h, LS byte first. Byte 4 – Domain ID [3:0] – Domain ID (Identifies the domain that this setting applies to) =00h – Entire platform. =01h – CPU subsystem. =02h – Memory subsystem. =03h – Reserved. =04h – High Power I/O subsystem. Others – Reserved. [6:4] – Reserved. Write as 000b. [7] – Per-component control – allows for setting power budget for chosen domain component. =0 – Power budget shall be applied to whole domain (persistent setting). =1 – Power budget shall be applied to given component in domain (volatile setting). Byte 5:6 – Target power budget in [Watts] that should be maintained by the Power Budget Control Service. Value 0 removes the existing power budget limit for given domain. Byte 7 - (optional*) Component Identifier If bit 7 in Byte 4 is set to 0 this field is optional and would be ignored if present. If bit 7 in Byte 4 is set to 1 this field contains component identifier for which power budget should be set. See Component Identifier for description.	Sets total power budget for given domain or its components. This command is optional and may be unavailable on certain implementations. This command controls the domain's power limit using the most aggressive settings. Note – Domain power budget configured via this command is treated evenly to Policy Target Limit configured by "Set Node Manager Policy". The budget for the whole domain is stored in the persistent storage while the per-component budget is volatile. The allowed power range for the whole CPU subsystem is determined by the power draw range, while the budget for sub-component is not restricted (0-32767W) This command accesses the flash, so the response will be sent after completing the operation and may take longer than 250 ms. *This parameter is not supported in Mehlow
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in Section 2.15). =80h – Invalid Component Identifier. =81h – Invalid Domain ID. =84h – Power Budget out of range. Byte 2:4 – Intel Manufacturer ID – 000157h, LS byte first.	
D1h	Get Total Power Budget	Request Byte 1:3 – Intel Manufacturer ID – 000157h, LS byte first.	

Net Function = 2Eh-2Fh LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
		Byte 4 – Domain ID [3:0] – Domain ID (Identifies the domain that this setting applies to) =00h – Entire platform. =01h – CPU subsystem. =02h – Memory subsystem. =03h – Reserved. =04h – High Power I/O subsystem. Others – Reserved. [6:4] – Reserved. Write as 000b. [7] – Per-component control – allows for getting power budget for chosen domain component. =0 – Power budget applied to whole domain should be returned. =1 – Power budget applied to given component in domain should be returned. Byte 5 - (optional*) Component Identifier If bit 7 in Byte 4 is set to 0 this field is optional and would be ignored if present. If bit 7 in Byte 4 is set to 1 this field contains component identifier for which power budget should be returned. See Component Identifier for description.	Returns the power budget for given domain or 0 if the budget is not set. This command is optional and may be unavailable on certain implementations. *This parameter is not supported in Mehlow
		Response Byte 1 – Completion Code = 00h – Success (Remaining standard Completion Codes are shown in Section 2.15). =80h – Invalid Component Identifier. =81h – Invalid Domain ID. Byte 2:4 – Intel Manufacturer ID – 000157h, LS byte first. Byte 5:6 – Target power budget in [Watts] that should be maintained by the Power Budget Control Service.	
D2h	Set Max Allowed CPU P-state/T-state	Request Byte 1:3 – Intel Manufacturer ID – 000157h, LS byte first. Byte 4 – Domain ID [3:0] – Domain ID – Identifies the domain that this Intel® NM setting applies to =00h – Entire platform – for compatibility with previous Intel® NM versions P/T state settings are applied to CPU subsystem. Others – Reserved. [5:4] – Control Knob =00b – Set max allowed CPU P-state/T-state. =01b – Set max allowed logical processors. =10b – Reserved. =11b – Reserved. [7:6] – Reserved. Write as 00b. For Control Knob set to 00b: Byte 5 – P-state number to be set. Byte 6 – T-state number to be set. For Control Knob set to 01b: Byte 5:6 – Set max allowed logical processors. Note – If any of the fields is set to FFh, it should be omitted when setting the value. Note – This setting is volatile. It is cleared after reset of either host or Intel® ME side.	This command is optional and may be unavailable on certain implementations. This command imposes additional limit on the host side apart of the limit that may be applied by Intel® NM Policy Control, or Total Power Budget. If the limit applied by Policy Control or Power Budget is lower than the effect of P-state/T-state limit then that power limit is used. If P-state/T-state limit imposes lower power limit than the P-state/T-state limit is kept.

Net Function = 2Eh-2Fh LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in Section 2.15). =81h – Invalid Domain ID. =8Ah – P-state or T-state out of range. =D5h – Command does not work since Hardware-controlled Performance States (HWP) are enabled in BIOS and legacy mode is not supported. Byte 2:4 – Intel Manufacturer ID – 000157h, LS byte first.	Additional power limit imposed by this command does not impact NM throttling statistics. Completion code 00h does not indicate that the requested P/T-state or number of logical processors has been set. It indicates that the request has been sent to OSPM.
D3h	Get Max Allowed CPU P-state/T-state	Request Byte 1:3 – Intel Manufacturer ID – 000157h, LS byte first. Byte 4 – Domain ID [3:0] – Domain ID – Identifies the domain that this Intel® NM setting applies to =00h – Entire platform – for compatibility with previous Intel® NM versions P/T state settings are applied to CPU subsystem. Others – Reserved. [5:4] – Control Knob =00b – get max allowed CPU P-state/T-state. =01b – get max allowed logical processors. =10b – Reserved. =11b – Reserved. [7:6] – Reserved. Write as 00b.	Returns the maximal CPU P/T state or number of logical CPUs cores. In case of multi socket system this command returns the maximum P/T state of all available CPU sockets. The number of available cores is counted as a sum from all CPU cores. This command returns 0 for P/T state when power is throttled due to SmarRT&CLST (both max throttling and ramp). This command is optional and may be unavailable on certain implementations.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in Section 2.15). =81h – Invalid Domain ID. =D5h – Command does not work since Hardware-controlled Performance States (HWP) are enabled in BIOS and legacy mode is not supported. Byte 2:4 – Intel Manufacturer ID – 000157h, LS byte first. For Control Knob set to 00b: Byte 5 – Current maximum P-state. Byte 6 – Current maximum T-state. For Control Knob set to 01b: Byte 5:6 – Total requested by Intel® ME number of allowed logical processors on a system. This is a number confirmed by OSPM and does not have to be equal to the one requested by Intel® ME since OSPM is not obliged to fulfill a core idling request. Note – If any of the fields is set to FFh, it means that value is unavailable. Note – For Control Knob set to 01b this command returns the value recently confirmed by the host side. For a short while, it may be differ from the value set with D2h. If the difference persists, it indicates an issue on the host side.	
D4h		Request Byte 1:3 – Intel Manufacturer ID – 000157h, LS byte first.	

Net Function = 2Eh-2Fh LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
	Get Number Of P-states/T-states	Byte 4 – Domain ID [3:0] – Domain ID – Identifies the domain that this setting applies to =00h – Entire platform – for compatibility with previous Intel® NM versions state settings are applied to CPU subsystem. Others – Reserved. [5:4] – Control Knob =00b – Max allowed processor P-states/T-states. =01b – Max allowed logical processors. =10b – Reserved. =11b – Reserved. [7:6] – Reserved. Write as 00b.	This command is optional and may be unavailable on certain implementations.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in Section 2.15). =81h – Invalid Domain ID. Byte 2:4 – Intel Manufacturer ID – 000157h, LS byte first. For Control Knob set to 00b: Byte 5 – Number of P-states available on the platform. This number is provided by BIOS. Byte 6 – Current Number of T-states available on the platform. This number is provided by BIOS. For Control Knob set to 01b: Byte 5:6 – Number of logical processors on the platform.	
D7h	Set PSU Configuration	Request Byte 1:3 – Intel manufacturer ID – 000157h, LS byte first. Byte 4 – Domain ID [3:0] – Domain ID - Identifies Domain which uses the defined PSU set =00h – Entire platform. Others – Reserved. [4:7] – Reserved. Write as 0000b. Byte 5 – PSU 1 address and monitoring mode. [0] – PSU monitoring mode =1 – PSU is installed and lack of power readings from this PSU will be reported to Management Console via Intel® NM Health Event. =0 (default) – PSU is not installed or it is installed but will not generate power reading failure events for a single PSU failure. [7:1] – 7-bit PSU SMBus address. Shall be set to 00h if address is not used (only monitoring mode shall be updated). Byte 6:12 – PSU 2 to PSU 8 address and monitoring mode encoded as in the Byte 5.	This command may override address and monitoring mode for supported PMBus PSU. The result of this command is stored in the persistent storage. This command should be send to Intel® Node Manager by BMC with monitoring mode bit set to 1 if the lack of reading from a single PSU should be reported to the Management Console using Intel® NM Health Event. Otherwise, Intel® Node Manager will send a notification only if all PSUs disappear. This command accesses the flash, so the response will be sent after completing the operation and may take longer than 250 ms.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in Section 2.15). =81h – Invalid Domain ID. Byte 2:4 – Intel manufacturer ID – 000157h, LS byte first.	

Net Function = 2Eh-2Fh LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
			Note: This command cannot change the mux address of PSUs (mux devices are considered to be not hot-swappable).
D8h	Get PSU Configuration	Request Byte 1:3 – Intel manufacturer ID – 000157h, LS byte first. Byte 4 – Domain ID [3:0] – Domain ID - Identifies Domain which uses the defined PSU set =00h – Entire platform. Others – Reserved. . [7:4] – Reserved. Write as 0000b.	Note: This command could change due requirements for per-rail readings and mux support.
		Response Byte 1 – Completion Code = 00h – Success (Remaining standard Completion Codes are shown in Section 2.15). = 81h – Invalid Domain ID. Byte 2:4 – Intel manufacturer ID – 000157h, LS byte first. Byte 5 – Domain ID [3:0] – Domain Id (Identifies Domain which uses the defined PSU set). Currently, FW supports only one domain – Domain 0). [7:4] – Reserved. Return as 0000b. Byte 6 – PMBUS PSU address 1. Intel® Node Manager will monitor the presence of the defined PSU [0] – PSU mode =1 – PSU is installed and lack of power readings should be reported to Management Console. =0 (default) – PSU is installed or may be attached in the future. [7:1] – 7-bit PSU SMBUS address. Set to 00h if address is not used. Bytes 7:13 – PMBUS PSU address 2 to PMBUS PSU address 8 encoded as in the Byte 6.	
D9h	Send Raw PMBUS command	Request Byte 1:3 – Intel Manufacturer ID – 000157h, LS byte first. Byte 4 – Flags [0] – Reserved. Write as 0b.	

Net Function = 2Eh-2Fh LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
		[3:1] – SMBUS message transaction type =0h – SEND_BYTE. =1h – READ_BYTE. =2h – WRITE_BYTE. =3h – READ_WORD. =4h – WRITE_WORD. =5h – BLOCK_READ. =6h – BLOCK_WRITE. =7h – BLOCK_WRITE_READ_PROC_CALL. [5:4] – Device address format. =0h – Standard device address =1h – Extended device address Other – reserved. [6] = 1b – Do not report PEC errors in Completion Code. If the bit is set, Intel® NM firmware does not return "bad PEC" Completion Codes. In this case BMC can learn that the transaction failed by checking PEC in the PMBUS response message. The flag does not disable PMBUS command retries. [7] = 1b – Enable PEC. For Standard device address (Byte 4 bits [5:4] equal 0) Byte 5 – Target PSU Address [0] – reserved should be set to 0 [7:1] – 7-bit PSU SMBUS address Byte 6 – MGPIO MUX configuration [5:0] = Mux MGPIO index or 0 if mux is not used. [7:6] = Reserved. Write as 00b. Byte 7 – Transmission Protocol parameter [4:0] = Reserved. Write as 00000b. [5] = Transmission Protocol =0b PMBus =1b I2C (Raw I2C transaction without the command Field) [7:6] = Reserved. Write as 00b. Byte 8 – Write Length Byte 9 – Read Length. This field is used to validate if the slave returns proper number of bytes Byte 10:M – PMBUS command For Extended device address (Byte 4 bits [5:4] equal 1) Byte 5:9 Extended device address Byte 5 – Sensor Bus =00h SMBUS =01h SMLINK0/SMLINK0B =02h SMLINK1 =03h SMLINK2 =04h SMLINK3 =05h SMLINK4 Other - reserved Byte 6 – Target PSU Address [0] – Reserved. Write as 0b. [7:1] – 7-bit SMBUS address. Byte 7 – MUX Address [0] – Reserved. Write as 0b. [7:1] – 7-bit SMBUS address for SMBUS MUX or 0 for MGPIO controlled. Byte 8 – MUX channel selection This field indicates which line of MUX should be enabled	This command sends one PMBUS command to the specified address. Address is validated against factory presets. Note – This command should be implemented by BMC in case when reverse PMBUS proxy functionality should be supported. Note – This command could change due requirements for per-rail readings. This command and functionality is supported and enabled regardless of Intel® NM state. It could be supported and enabled/disabled separately from Intel® NM (aka. Power & Thermal Assist Module). For more details see Set/Get Intel® ME FW Capabilities command.

Net Function = 2Eh-2Fh LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
		Byte 9 – MUX configuration state [0] – MUX support =0 – ignore MUX configuration (MUX not present) =1 – use MUX configuration [7:1] – Reserved. Write as 0000000b. Byte 10 – Transmission Protocol parameter [4:0] = Reserved. Write as 00000b. [5] = Transmission Protocol =0b PMBus =1b I2C (Only the following SMBus transactions are translated to I2C; send byte, read/write word, read/write block.) [7:6] = Reserved. Write as 00b. Byte 11 – Write Length Byte 12 – Read Length. This field is used to validate if the slave returns proper number of bytes Byte 13:M – PMBUS command	
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in Section 2.15). =80h – Command response timeout. SMBUS device was not present. =81h – Command not serviced. Not able to allocate the resources for serving this command at this time. Retry needed. =82h – Command not executed due to conflict with PSU Optimization feature. =A1h – Illegal SMBUS PSU Slave Target Address. =A2h – PEC error. =A3h – Number of bytes returned by the Slave different from Read Length see byte 9 of request. =A5h – Unsupported Write Length. =A6h – Unsupported Read Length. =A9h – MUX communication problem. =AAh – SMBUS timeout. Byte 2:4 – Intel Manufacturer ID – 000157h, LS byte first. Byte 5:N – PMBUS response data received from PSU during Read transaction phase. Response from the slave is returned for Completion Codes: 00h, A3h.	
EAh	Get Host CPU data	Request Byte 1:3 – Intel Manufacturer ID – 000157h, LS byte first. Byte 4 – Domain ID [3:0] – Domain ID - Identifies the set of processors supported by the domain =00h – Entire platform. Others – Reserved. [7:4] – Reserved. Write as 0000b.	The command returns the CPU configuration data passed from BIOS to Intel® ME using HECI messages.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in Section 2.15). =81h – Invalid Domain ID. Byte 2:4 – Intel Manufacturer ID – 000157h, LS byte first.	

Net Function = 2Eh-2Fh LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
		Byte 5 – Host CPU data [2:0] – Reserved. Write as 000b. [3] – Set to 1 if Intel® NM already activated regular power limiting policies after Host startup. [4] – Set to 1 if Host CPU discovery data provided with that command is valid. [6:5] – Reserved. Write as 00b. [7] – Set to 1 if End of POST notification was received. **Note:** Bytes 6:25 are ignored if Byte 5 bit [4] is set to 0. If Byte 5 bit [4] is set to 1 Bytes 6:25 should describe the actual Host CPU data of the platform. Additionally bytes 6:24 should be set to 0 if the CPU discovery data is passed to Intel® NM directly by the BIOS. Per processor, discovery data will be provided only for the lowest number processor that is installed. In the multiprocessor environment, all other processors installed on board should match the number of performance states and each processor performance state must have identical performance and power consumption parameters. Byte 6 – Number of P-states supported by the current platform CPU configuration =0 – If P-states are disabled by the user. =1 – If CPU does not support more P-states or in the multiprocessor environment some processors installed on board do not match the lowest number processor power consumption parameters. =2 – 255 – Actual number of supported P-states by the lowest number processor. Note – Other processors should match the number of performance states of lowest number processor. Byte 7 – Number of T-states supported by the current platform CPU configuration =0 – If T-states are disabled by the user. =1 – 255 – Actual number of supported T-states by the lowest number processor. Note – Other processors should match the number of throttling states of lowest number processor. Byte 8 – Number of installed processor packages. This value is calculated as a number of all sockets with CPU package present. Bytes 9:12 – HWP Capabilities bits[7:0] - Highest bits[15:8] - Guaranteed Performance bits[23:16] - Most Efficient Performance bits[31:24] - Lowest Performance Bytes 13:25 – Reserved.	
ECh	Aggregated Send Raw PMBus commands	Request Byte 1:3 – Intel Manufacturer ID – 000157h, LS byte first. Bytes 4:M - Definition of the first PMBus request block formatted according to the same rules as bytes 4 to M in Send Raw PMBUS Command. Bytes M+1:N – Next PMBus request block (if any).	This command is based on syntax of "Send Raw PMBus" command. It sends multiply "Raw PMBus" frames in one IPMI request.
		Response	

Net Function = 2Eh-2Fh LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
		Byte 1 – Completion Code related to overall IPMI request (Remaining standard Completion Codes are shown in Section 2.15). =00h – Success; it means that all the PMBus raw requests have been processed and the responses fit in the IPMI response message (not necessarily that all the responses have completed with success). The completion codes for the particular PMBus raw transactions are included in the appropriate parts of this response frame. =A0h - Partial Filling. Response to one or more of requests behind did not fit into IPMI response because of IPMI message length limit. Only full (non-truncated) PMBus responses are included. =A1h - Address field of one of PMBus requests has an incorrect value =A5h - Write Length of one of PMBus requests is too big to be received by PMBus =A6h - Read Length of one of PMBus requests is too big to be sent by PMBus =83h - Particular requests are targeted to different physical buses Byte 2:4 – Intel Manufacturer ID – 000157h, LS byte first. The following bytes are present only in the case when general Completion Code is equal to 0x00 (Success) or 0xA0 (Partial Filling). Byte 5 – Completion Code for the first PMBUS transaction – same as byte#1 in Send Raw PMBUS command response. Byte 6:N – PMBUS response data received from PSU during Read transaction phase (if any). N+1:M - Response to subsequent request (if any).	Addresses are validated against factory presets. For more details see description of "Send Raw PMBus command".
F0h	Set HW Protection Coefficient	Request Byte 1:3 = Intel manufacturers ID – 000157h, LS byte first Byte 4 – K Coefficient [%]. When set to 0 HW Protection Policy would be disabled.	This command configures K-coefficient value for HW Protection Policy – units: percentage. The allowed value range is from 0 to 255 inclusive. Value 0 disables the HW Protection policy. This command accesses the flash, so the response will be sent after completing the operation and may take longer than 250 ms.
		Response Byte 1 – completion code =00h – Success (Remaining standard completion codes are shown in Section 2.15) =81h – Domain ID invalid or not supported in the current configuration. Returned for domain ID = 03h, when PSU is configured as the power reading source. =A1h – Incorrect K Coefficient value Byte 2:4 = Intel manufacturers ID – 000157h, LS byte first	
F1h	Get HW Protection Coefficient	Request Byte 1:3 = Intel manufacturers ID – 000157h, LS byte first	This command retrieves K-coefficient value for HW Protection Policy – units: percentage.
		Response Byte 1 – completion code =00h – Success (Remaining standard completion codes are shown in Section 2.15) =81h – Domain ID invalid or not supported in the current configuration. Returned for domain ID = 03h, when PSU is configured as the power reading source.	

Net Function = 2Eh-2Fh LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
		Byte 2:4 = Intel manufacturers ID – 000157h, LS byte first Byte 5 – K Coefficient [%]	
F2h	Get Limiting Policy ID	Request Byte 1:3 = Intel manufacturers ID – 000157h, LS byte first Byte 4 – Domain ID [3:0] = Domain ID (Identifies the domain for which the response is to be provided) =00h – Entire platform =01h – CPU subsystem =02h – Memory subsystem =03h – HW Protection =04h – High Power I/O subsystem. =Others – Reserved [7:4] = Reserved. Write as 0000b.	This command fetches policy which is currently limiting.
		Response Byte 1 – completion code =00h – Success (Remaining standard completion codes are shown in Section 2.15) =81h – Invalid Domain ID =A1h – No policy is currently limiting for the specified Domain ID Byte 2:4 = Intel manufacturers ID – 000157h, LS byte first For completion code 00h (Success) response byte 5 is defined as follows: Byte 5 – The ID of the Intel® NM Policy which is currently limiting	
F3h	Set PMBUS Device Configuration	Request Byte 1:3 = Intel manufacturers ID – 000157h, LS byte first Byte 4 – Device Index [4:0] = PMBUS-enabled Device Index [5] – Reserved. Write as 0. [7:6] – Device address format. =0h – Standard device address =1h – Extended device address =3h – Data format common configuration Other – reserved. **For Standard device address** (Byte 4 bits [7:6] equal 0) Byte 5 – SMBUS address. [0] – Reserved. Write as 0b. [7:1] – 7 bit PSU SMBUS address. Set to 00h if the device is not present.	Allows reconfiguring a PMBUS device. For Device Index description see paragraph 3.2.1. Setting SMBUS address to 00h puts the device into not-configured state. In this state the device is not polled by Intel® NM and is not accessible via PMBUS Proxy.

Net Function = 2Eh-2Fh LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
		Byte 6 – MGPIO MUX configuration [5:0] = Mux Address [6] – Disabled State =0 – PMBUS device is enabled and may be polled for readings by Intel® NM =1 – PMBUS device is disabled and should not be poled for readings, but access to the device may be available using PMBUS Proxy (on condition that SMBUS address <> 00h) [7] – Device Mode: =1 – the device is installed and lack of power readings should be reported to Management Console using Intel® NM Health Event =0 (default) – the device is installed or may be attached in the future. [7:6] = Reserved. Write as 000b. **For Extended device address** (Byte 4 bits [6:7] equal 1) Byte 5:9 Extended device address Byte 5 – Sensor Bus =00h SMBUS =01h SMLINK0/SMLINK0B =02h SMLINK1 =03h SMLINK2 =04h SMLINK3 =05h SMLINK4 Other - reserved Byte 6 – Target PSU Address [0] – Reserved. Write as 0b. [7:1] – 7-bit SMBUS address. Byte 7 – MUX Address [0] – Reserved. Write as 0b. [7:1] – 7-bit SMBUS address for SMBUS MUX or 0 for MGPIO controlled. Byte 8 – MUX channel selection This field indicates which lines of MUX should be enabled Byte 9 – MUX configuration state [0] – MUX support =0 – ignore MUX configuration (MUX not present) =1 – use MUX configuration [7:1] – Reserved. Write as 0000000b. Byte 10 – Device configuration [0] – Disabled State =0 – PMBUS device is enabled and may be polled for readings by Intel® NM =1 – PMBUS device is disabled and should not be poled for readings, but access to the device may be available using PMBUS Proxy (on condition that SMBUS address <> 00h) [1] – Device Mode: =0 (default) – the device is installed or may be attached in the future. =1 – the device is installed and lack of power readings should be reported to Management Console using Intel® NM Health Event (not applicable to telemetry only devices) [7:2] = Reserved. Write as 000000b. **For data format common configuration** (Request Byte 4 bits [6:7] equal 3)	The Disabled State bit allows enabling / disabling polling by Intel® NM any device that is accessed using PMBUS protocol. If disabled, the device can still be accessed using Send Raw PMBUS command but it is not polled by Intel® NM for readings. The Device Mode parameter has the same meaning as the 'PSU mode' parameter for the 'Set PSU Configuration' defined in [NM_IPMI]. This command accesses the flash, so the response will be sent after completing the operation and may take longer than 250 ms.

Net Function = 2Eh-2Fh LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
		Byte 5 - first data byte [0] - PSU redundancy mode =0b - Full N+1 redundancy if this PSU is present (SmaRT functionality will be automatically disabled if at least 2 PSUs are on) =1b - non-redundant PSU [1:2] Iout OC Warning event masking [4:3] Vin UV Fault event masking [6:5] OT Warning event masking All event masking values have the same meaning: = 0 Automatic - Intel® NM may mask the relevant event in PSU if needed. = 1 Event Enabled - Intel® NM will never mask the related event. If the event is already masked, the mask will be removed. = 2 Event Disabled - This event will be masked in PSU as a result of this command. The masking values set by this command are persistent. They will be re-applied to PSUs after Intel® ME G3 exit during the first PSU discovery. [7] = 0, Reserved. Byte 6 = Reserved. Write as 00h.	
		Response Byte 1 – completion code =00h – Success (Remaining standard completion codes are shown in Section 2.15) =A1h – Conflicting Address. A PMBUS device with the same SMBUS and MUX address already exists Byte 2:4 = Intel manufacturers ID – 000157h, LS byte first	
F4h	Get PMBUS Device Configuration	Request Byte 1:3 = Intel manufacturers ID – 000157h, LS byte first Byte 4 – Device Index [4:0] = PMBUS-enabled Device Index [5] – Reserved. Write as 0. [6:7] – Device address format. =0h – Standard device address =1h – Extended device address =3h – Common configuration Other – reserved.	Allows reading PMBUS-enabled Device Configuration. For Device Index description see paragraph 3.2.1. To allow for faster enumeration of all defined PMBUS-enabled devices the error code A1h returns extended error information.
		Response Byte 1 – completion code =00h – Success (Remaining standard completion codes are shown in Section 2.15) =80h – Invalid Device Index (device not configured). In addition byte 5 extended error information is returned for this error code. Byte 2:4 = Intel manufacturers ID – 000157h, LS byte first For completion code 00h (Success) response bytes 5 -7 are defined as follows: For Standard device address (Request Byte 4 bits [7:6] equal 0) Byte 5 – SMBUS address. [0] – Reserved. Write as 0b. [7:1] – 7 bit PSU SMBUS address. Set to 00h if address is not used.	

Net Function = 2Eh-2Fh LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
		Byte 6 – MGPIO MUX configuration [5:0] = Mux address [6] – Disabled state =0 – PMBUS device is enabled and may be polled for readings by Intel® NM =1 – PMBUS device is disabled and should not be poled for readings, but access to the device may be available using PMBUS Proxy (on condition that SMBUS address <> 00h) [7] – Device mode: =1 – the Device is installed and lack of power readings should be reported to Management Console =0 (default) – the Device is installed or may be attached in the future [7:6] = Reserved. Write as 000b. Byte 7 – Device Type [3:0] Device Type Index selected via ftool (PSU, VR, Current Monitor) [7:4] – Reserved. Write as 0000b For Extended device address (Request Byte 4 bits [6:7] equal 1) Byte 5:9 – Extended device address Byte 5 – Sensor Bus =00h SMBUS =01h SMLINK0/SMLINK0B =02h SMLINK1 =03h SMLINK2 =04h SMLINK3 =05h SMLINK4 Other - reserved Byte 6 – Target PSU Address [0] – Reserved. Write as 0b. [7:1] – 7-bit SMBUS address. Byte 7 – MUX Address [0] – Reserved. Write as 0b. [7:1] – 7-bit SMBUS address for SMBUS MUX or 0 for MGPIO controlled. Byte 8 – MUX channel selection This field indicates which lines of MUX should be enabled Byte 9 – MUX configuration state [0] – MUX support =0 – ignore MUX configuration (MUX not present) =1 – use MUX configuration [7:1] – Reserved. Write as 0000000b. Byte 10 – Device configuration [0] – Disabled state =0 – PMBUS device is enabled and may be polled for readings by Intel® NM =1 – PMBUS device is disabled and should not be poled for readings, but access to the device may be available using PMBUS Proxy (on condition that SMBUS address <> 00h) [1] – Device mode: =1 – the Device is installed and lack of power readings should be reported to Management Console =0 (default) – the Device is installed or may be attached in the future [7:2] = Reserved. Write as 000000b.	

Net Function = 2Eh-2Fh LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
		Byte 11 – Device Type [3:0] Device Type Index defined via ftool (PSU, VR, Current Monitor) [7:4] – Reserved. Write as 0000b For completion code 80h (Invalid Device Index) response bytes 5 – 6 are defined as follows: Byte 5 – Next valid Device Index. The field contains lowest valid Device Index that is higher than Device specified in the request. If no such Device Index exists, zero value is returned. Byte 6 – Number of defined Devices. For data format Common configuration (Request Byte 4 bits [7:6] equal 3) Byte 5 - first data byte [0] - PSU redundancy mode =0b - Full N+1 redundancy if this PSU is present (SmaRT functionality will be automatically disabled if at least 2 PSUs are on) =1b - non-redundant PSU [6:1] – PSU event masking. All event masking values have the same meaning: = 0 Automatic (default) - Intel® NM may mask the relevant event in PSU if needed. = 1 Event Enabled - Intel® NM will never mask the related event. If the event is already masked, the mask will be removed. = 2 Event Disabled - This event will be masked in PSU as a result of this command. The masking bits are split in the following fields: [2:1] Iout OC Warning event masking [4:3] Vin UV Fault event masking [6:5] OT Warning event masking [7] = 0 – Reserved Byte 6 – Reserved. Note: This information can be used to query all existing PMBUS devices. Start with Device Index set to 0. Increment Device Index treated as an unsigned integer value by one on success and set Device Index to Byte 5 on reception of completion code 80h.	
F5h	Get PMBUS Readings	Request Byte 1:3 = Intel manufacturers ID – 000157h, LS byte first Byte 4 – Device Index [4:0] = PMBUS-enabled Device Index [7:5] = Reserved. Write as 000b. Byte 5 – History index [3:0] = History index. Supported value: 0Fh to retrieve current samples. Others – Reserved. [7:4] – Page number – used only for devices which support pages. For others Reserved. Byte 6 – First Register Offset [3:0] – First Register Offset [7:4] – Reserved. Write as 00000b.	This command retrieves values of group of monitored registers retrieved from single PSU device. Maximum 8 values can be retrieved For Device Index description see paragraph 3.2.1.
		Response	

Net Function = 2Eh-2Fh LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
		Byte 1 – completion code =00h – Success (Remaining standard completion codes are shown in Section 2.15) =A1h – Illegal Device Index =A3h – Illegal First Register Offset =A4h – History snapshot not available =A5h – Page number not supported =A6h – Reading not available Byte 2:4 = Intel manufacturers ID – 000157h, LS byte first Bytes 5:8 – Timestamp as defined by the IPMI v2.0 specification indicating when the samples collection was stopped. Bytes 9:10 – Register Value of Monitored Register [First Register Offset] (for READ_EIN and READ_EOUT this field contains value converted to Watts) Length of the response depends on number of monitored registers. Bytes 12:32 are used only if the PMBUS-enabled device is monitored for the sensors. Bytes 11:12 - Monitored Register Value of register [First Register Offset + 1] Bytes 13:14 - Monitored Register Value of register [First Register Offset + 2] Bytes 15:16 - Monitored Register Value of register [First Register Offset + 3] Bytes 17:18 - Monitored Register Value of register [First Register Offset + 4] Bytes 19:20 - Monitored Register Value of register [First Register Offset + 5] Bytes 21:22 - Monitored Register Value of register [First Register Offset + 6] Bytes 23:24 - Monitored Register Value of register [First Register Offset + 7]	The First Register Offset field allows obtaining registers values if more than 8 registers are monitored for specified PMBUS-enabled Device Type. If First Register Offset = 0, Monitored Registers 0 – 7 will be retrieved. If First Register Offset = 8, Monitored Registers 8 – MAX will be retrieved. The A6h Completion Code is returned when none of the monitored values can be provided because of error in communication with PMBUS device. If some of the registers are disabled, their values will be returned as 0xFFFF if there are more registers to be reported correctly, or will be skipped if there are no more registers with valid values. If some of the registers cannot be read from PMBus device, their values will be returned as 0xFFFF if there are more registers to be reported correctly. The unavailable registers do not influence the timestamp value.
F6h	Aggregated Get PMBus Readings	Request Byte 1:3 = Intel manufacturers ID – 000157h, LS byte first Byte 4 – Register Offset [3:0] = Offset of the register [7:4] = Page number – used only for devices which support pages. For others Reserved. Bytes 5:N – each byte should contain Device Index (byte 5 – Device 0, byte65 – Device 1, …)	This command reads the same register value from a group of PMBus-enabled devices. Up to 8 values can be read. The Register which should be read is defined in the Register Offset field. The list of devices from which the values are to be obtained is specified on Bytes 5:N.
		Response	

Net Function = 2Eh-2Fh LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
		Byte 1 – completion code =00h – Success (Remaining standard completion codes are shown in Section 2.15) =A1h – Illegal Device Index =A2h – Incompatible device types =A3h – Illegal Register Offset= A5h – Page number not supported =A6h - Reading not available Byte 2:4 = Intel manufacturers ID – 000157h, LS byte first Length of the response depends on number of requested registers. Bytes 5:6 = Monitored Register Value for Device 0 … Bytes 19:20 = Monitored Register Value for Device 7	For Device Index description see paragraph 3.2.1. The A6h Completion Code is returned when reading value cannot be provided because of error in communication with PMBUS device.
F7h	Get Energy Counter Unit	Request Bytes 1:3 - Intel Manufacturer ID – 000157h, LS byte first.	This command retrieves CPLD Energy Counter Unit – units: uJ.
		Response Byte 1 – completion code =00h – Success (Remaining standard completion codes are shown in [NM IPMI]) Byte 2:4 = Intel manufacturers ID – 000157h, LS byte first Byte 5:8 = Energy Counter Unit in uJ, LS byte first	
F8h	Set Energy Counter Unit	Request Byte 1:3 - Intel Manufacturer ID – 000157h, LS byte first. Byte 4:7 = Energy Counter Unit in uJ, LS byte first	This command configures CPLD Energy Counter Unit – units: uJ. This command accesses the flash, so the response will be sent after completing the operation and may take longer than 250 ms.
		Response Byte 1 – completion code =00h – Success (Remaining standard completion codes are shown in [NM IPMI]) =A1h – Incorrect Energy Counter Unit value Byte 2:4 = Intel manufacturers ID – 000157h, LS byte first	
F9h	Set Intel® NM Parameter	Request Bytes 1:3 - Intel Manufacturer ID – 000157h, LS byte first. Byte 4 – Parameter ID =00h – 01h – Reserved =02h – Slope calibration coefficient for device reading reporting =03h – Offset calibration coefficient for device reading reporting =04h – Reserved =05h – Scanning Period =06h – Enforce auto-configuration =07h – Enable / Disable Fast NM Limiting =08h – Reserved =09h – Reset Calibration coefficients (both Slope and Offset) =0Ah – PROCHOT# assertion level Others – Reserved Bytes 5:8 – Sub ID For Parameter ID = 02h, 03h or 09h Byte 5 – User defined device index configured by Intel® Server Platform Services FITc	This command is used for setting various configuration parameters of Intel® Node Manager. Calibration coefficients are stored in dedicated flash files. Only one slot may be assigned to a register. For the calibration coefficients to take effect, Intel® NM is required to be restarted.

Net Function = 2Eh-2Fh LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
		Byte 6 – Calibration coefficients slot index (Index in the “Device Calibration” array configured by Intel® Server Platform Services FITc). For Parameter ID = 09h value FF indicates all slots. Bytes 7:8 – Reserved, write as 00h For Parameter ID = 05h: Byte 5 – Sensor Number (Intel® ME sensor ID, as defined by Intel® Server Platform Services FITc) Bytes 6:8 – Reserved, write as 000000h For other Parameter ID values: Reserved – write as 00000000h Bytes 9:12 – Parameter value For Parameter ID = 02h or 03h: Byte 9 = Register index (compensated register offset in PDT table) Byte 10 – Device Power State and Page [3:0] Device power state (VRs only, returns 0 otherwise) [7:4] Page number (starting from 0) Byte 11:12 = New parameter value, LS byte first For Parameter ID = 05h: Byte 9 – Scanning Period = 1 – 100 ms (Auto/Fast - When used with PSU sensors, NM adjusts its internal parameters optimally for given PSU type) = 2 – 200 ms = 3 – 250 ms = 4 – 500 ms = 5 – 1000 ms Others – Reserved Bytes 10:12 – Reserved, write as 000000h For Parameter ID = 07h: Byte 9 – Enable / Disable Fast NM Limiting [7:1] Reserved, write as 0000000b [0] Fast NM Limiting = 0 – Disable Fast NM Limiting = 1 – Enable Fast NM Limiting Bytes 10:12 – Reserved, write as 000000h For Parameter ID = 0Ah: Byte 9 = CPU ratio that will set CPU frequency when PROCHOT# line is asserted. Bytes 10:12 – Reserved, write as 000000h Note: Note that requested PROCHOT# assertion level would not be checked against the valid range and the hardware supported level will be silently used by CPU. PROCHOT# assertion level set to 0 will restore CPU default PROCHOT# assertion level. PROCHOT# assertion level functionality will not work if BIOS sets PROCHOT_LOCK bit (MSR 0x1FC, bit 27). For other Parameter ID values: Reserved – write as 00000000h	Issuing this command with parameter ID 6 will cause Intel® ME Intel® NM to perform the auto-configuration routine. This functionality is supported only if auto-configuration is enabled in XML. For Device Index description see paragraph 3.2.1. This command accesses the flash, so the response will be sent after completing the operation and may take longer than 250 ms. **Scanning Period parameter notes**: Scanning period is common for all sensor of the same type (e.g. PSU AC Power Input), thus update of scanning period for one sensor results in update of scanning period for all sensors of given type. The scanning period value is updated at runtime immediately after the command execution and it is persistent.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in Section 2.11). =A1h – Incorrect Parameter value	

Net Function = 2Eh-2Fh LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
		=A2h – Incorrect Parameter ID =A3h – Incorrect Sub ID =C4h – Out Of Space – error occurred during write of PROCHOT ratio to flash for request Parameter ID = 0Ah. =D3h – Cannot Execute Command – error occurred during communication with CPU for request Parameter ID = 0Ah. =D5h – The parameter cannot be set when platform not in S0/S1 state. =D6h – Cannot Execute Command – register locked by BIOS for request Parameter ID = 0Ah. Bytes 2:4 – Intel Manufacturer ID – 000157h, LS byte first.	
FAh	Get Intel® NM Parameter	Request Byte 1:3 – Intel Manufacturer ID – 000157h, LS byte first. Byte 4 – Parameter ID =00h – 01h – Reserved =02h – Slope Comp parameter for VR reading reporting =03h – Offset parameter for VR readings reporting =04h – PMBus register configuration =05h – Scanning Period =06h – Auto-configuration results =0Ah – PROCHOT# assertion level =10h – NM Throttling Status Others – Reserved Bytes 5:8 – Sub ID For Parameter ID = 02h or 03h Byte 5 - PMBus Device Index Byte 6 - Compensation coefficients slot (offset in compensation coefficients array) Byte 7:8 - Reserved For Parameter ID = 04h Byte 5 – PDT index (offset of the device type in the PDTs list) Byte 6 – register index (offset of the register in the PDT table) Bytes 7:8 – Reserved For Parameter ID = 05h Bytes 5 – Sensor Number Bytes 6:8 – Reserved For Parameter ID = 10h Byte 5 – Throttling Data Type =00h – Global NM Throttling Status =01h – Policy Throttling status =02h – Throttling Details Data Bytes 6:8 – Throttling Parameter For Throttling Data Type = 00h: Byte 6 – Flags [0] – Clear throttling sources sticky bits. Clears all the bits successfully returned by this command, leaving the policy individual bits and detailed throttling data untouched. [1] – Reset Throttling Data. When set clears all throttling data which means global and policy sticky bits and throttling details data. [2:7] – Reserved Bytes 7:8 – Reserved	This command is used for reading various configuration parameters of Intel® Node Manager. Can be used to read all Power Device Template (PDT) monitored register configuration. For Device Index description see paragraph 3.2.1. Using this command with parameter ID 4 retrieves the configuration of a specified monitored register of given device type from the PDTs list. **Scanning Period parameter notes**: The returned value is the device configured frequency for this sensor type.

Net Function = 2Eh-2Fh LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
		For Throttling Data Type = 01h: Byte 6 – Flags [0] – Clear returned bits. Clears the throttling bits for policies successfully queried by this command. [1:7] – Reserved Byte 7 – Queried Polices ID Range Index =0 – Policy IDs 0..31 =1 – Policy IDs 32..63 =2 – Policy IDs 64..95 =3 – Policy IDs 96..127 =4 – Policy IDs 128..159 =5 – Policy IDs 160..191 =6 – Policy IDs 192..223 =7 – Policy IDs 224..255 Others – Reserved Byte 8 – Reserved For Throttling Data Type = 02h: Byte 6 – Flags [0] – Clear on read. Clears the throttling data the throttling source successfully queried by this command. [1:7] – Reserved Byte 7 – Throttling Source Selection. Selects for which throttling source the detailed information shall be returned. =0 – Entire platform Domain =1 – CPU subsystem Domain =2 – Memory subsystem Domain =3 – Fast NM Limiting / HW Protection Domain =4 – High Power I/O subsystem Domain =5 – SMART & CLST =6 – Failure action Others – Reserved Byte 8 – Reserved For other Parameter ID and reserved values – write as 00000000h	
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in Section 2.15). =A2h – Incorrect Parameter ID value =A3h – Incorrect Sub ID value Bytes 2:4 – Intel Manufacturer ID – 000157h, LS byte first. Bytes 5:12 – Parameter value: For Parameter ID = 02h or 03h: Byte 5 = Register index (compensated register offset in PDT table) Byte 6 = Device power state (only for VRs) Bytes 7:8 = New parameter value, LS byte first For Parameter ID = 04h: Byte 5 = Monitored register of given device type and register indices Byte 6 – Register configuration: [2:0] Reading frequency: = 000b – Unused entry (do not monitor the register) = 001b – 10 Hz = 010b – 5 Hz = 011b – 2 Hz = 100b – 1 Hz	

Net Function = 2Eh-2Fh LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
		Others – Reserved [3] Collect history samples: = 0 – History is not collected = 1 – History is collected [4] Power state: = 0 – Monitor in S0/1 = 1 – Monitor in all S-states [5] Use Page Plus Command = 0 – Register is read directly = 1 – Register is read using PAGE_PLUS_READ [6] Use Averaging = 0 –Averaging is disabled = 1 – Averaging is enabled [7] Compute energy: = 0 – Energy counter not enabled for this register = 1 – Energy counter enabled for this register Bytes 7:12 – Reserved For Parameter ID = 05h: Byte 5 - Scanning Period = 0 – Scanning disabled = 1 – 100 ms = 2 – 200 ms = 3 – 250 ms = 4 – 500 ms = 5 – 1000 ms Others – Reserved Bytes 6:8 – Reserved For Parameter ID = 06h: Byte 5 – Auto-configuration result 1 [0] – Unmanaged power source =0b – BMC =1b – estimated [2:1] – PSU efficiency source =00b – BMC =01b – PSU =10b – reserved =11b – not supported [4:3] – Chassis Power input source =00b – BMC =01b – PSU =10b – On-board power sensor/ PSU efficiency =11b – not supported [6:5] – DC Power source =00b – BMC =01b – PSU =10b – On-board power sensor =11b – reserved [7] – Auto-configuration result =0b – Success =1b – Failure / in-progress / not configured Byte 6 – Auto-configuration result 2 [1:0] – HW protection source =00b – BMC =01b – PSU =10b – On-board power sensor =11b – reserved [7:2] – Reserved, write as 000000b Bytes 7:8 – Reserved	

Net Function = 2Eh-2Fh LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
		For Parameter ID = 0Ah: Byte 5 – Current PROCHOT# assertion ratio. This is the min value across all CPU sockets. Bytes 6:12 – Reserved For Parameter ID = 10h: For Throttling Data Type = 00h: Bytes 5:6 – Throttling sources bits. When bit is set this indicates that the corresponding throttling source was active at least once (force throttling of the platform power). [0] – Entire platform Domain [1] – CPU subsystem Domain [2] – Memory subsystem Domain [3] – Fast NM Limiting / HW Protection Domain [4] – High Power I/O subsystem Domain [5] – SMART & CLST [6] – Failure action Others – Reserved Bytes 7:12 - Reserved For Throttling Data Type = 01h: Bytes 5:8 - Policy throttling bits [0:31] – Bits representing throttling status of the associated policy. The least significant bit corresponds to the policy ID from the beginning of the range provided in the request Range Index parameter (e.g. for Range Index = 1, it corresponds to policy with Policy ID = 32). The bit is set if the corresponding policy was ever actively throttling. Bytes 9:12 – Reserved For Throttling Data Type = 02h: Bytes 5:8 – Timestamp for the throttling source selected in the request. This is the start timestamp of the first throttling activation as host RTC time. Byte 9 – Maximum throttling level during first activation in percent - as it would be returned by Get NM Statistics command. Byte 10 – Throttling Source. For NM Domains: = Policy ID of the first throttling policy For Fast NM Limiting =0 – Go-to mode =1 – Interrupt mode For SMART & CLST: =0 – OC Warning =1 – OT Warning =2 – UV Fault For Failure Action: =0 – PECI over DMI failure =1 – Missing reading policy trigger =2 – Policy Shutdown Failure Action Bytes 11:12 – Throttling Duration. Approximate duration of the first throttling with 0.1 second resolution. FFFFh if the throttling lasted longer than 1h:49min:13.6s For other valid Parameter ID values: Byte 5:8 – Parameter value. LSB first.	
FBh	Get PMBus Device Energy	Request Byte 1:3 – Intel Manufacturer ID – 000157h, LS byte first.	

Net Function = 2Eh-2Fh LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
		Byte 4 – Device Index [4:0] = PMBus-enabled Device Index [7:5] = Reserved. Write as 000b. Byte 5 – Register Offset (offset of the register from which of the registers of the device the command should provide the energy) Byte 6 (optional) – Device Page (applicable to paged devices only, zero-based, default 0) Byte 7 – Reserved. Write as 00h.	This command allows getting energy counter from a monitoring device. The Register Offset corresponds to the offset number of the register with the Compute Energy bit set in the PDT configuration. Supported numbers are 0 to 15. The Timestamp value provided in the response is the internal Intel® ME Firmware time counter value converted into ms. The value provided indicates the time for the recent reading from the PMBus device performed by the Intel® Node Manager. This command has two versions a standard one 5 bytes long and an extended one 7 bytes long. The extended command must be sent with both optional bytes present. For Device Index description see paragraph 3.2.1.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in Section 2.15). = A1h – Illegal Device Index or Illegal Offset = A6h - Reading not available = A7h – Device not configured to provide Energy Readings or energy computation was not enabled for given Register Offset in PDT. Byte 2:4 - Intel Manufacturer ID – 000157h, LS byte first. Byte 5:8 – Reading Value as a 32-bit unsigned integer Energy Counter [mJ]. Byte 9:12 – Timestamp [ms].	

3.2.1 Device Enumeration

Intel® NM firmware supports up to 32 PMBus-enabled devices. Each device had unique index associated with assigned sensor number in configuration (range 80h to 9Eh). The Device Index is configurable via FITc as "User Device ID" paUpdate rrameter of the device configuration. The Device Index value range is [0…31].

3.3 External DCMI Power Management Commands

The standard DCMI Power Management commands defined in [DCMI] offer functionality that is a subset of the Intel® Intelligent Power Technology Node Manager (see 3.1). In Intel® NM firmware the following commands are supported:

Table 3-3 External DCMI Power Management Commands

Net Function = DCGRP (2Ch) LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
01h	Get DCMI Capability Info	Request Byte 1 – Group Extension Identification = DCh. Byte 2 – Parameter Selector =5 – Enhanced Power Statistics attributes. Other parameter selector values are not supported.	This command is intended for BMC or Remote Console to provide information about DCMI Power Manager capabilities.
		Response Byte 1 – Completion Code = C1h – Returned if DCMI mode is not present. Byte 2 – Group Extension Identification = DCh. Byte 3:4 – DCMI Specification Conformance Byte 3 – Major Version = 1. Byte 4 – Minor Version = 1. Byte 5 – Parameter Revision = 02h – Enhanced Power Statistics attributes. Byte 6 – The number of supported rolling average time periods =09h. Byte 7:15 – Rolling Average Time periods [5:0] – Time duration The following periods are supported =05h – 5 seconds. =0Fh – 15 seconds. =1Eh – 30 seconds. =41h – 1 minute. =43h – 3 minutes. =47h – 7 minutes. =4Fh – 15 minutes. =5Eh – 30 minutes. =81h – 1 hour. [7:6] – Time duration units =00b – Seconds. =01b – Minutes. =10b – Hours. =11b – Days.	

Net Function = DCGRP (2Ch) LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
02h	Get Power Reading	Request Byte 1 – Group Extension Identification = DCh. Byte 2 – Mode =1 – System Power Statistics. =2 – Enhanced System Power Statistics. Byte 3 – Rolling Average Time periods For Mode = 1 =00h – statistics collected from up-time. Other values are reserved For Mode = 2 – One of periods reported in bytes 7:14 of the response to the Get DCMI Capability Info command. Byte 4 – Reserved.	
		Response Byte 1 – Completion Code =C1h – Returned if DCMI mode is not present. Byte 2 – Group Extension Identification = DCh. Byte 3:4 – Current Power in [Watts]. Byte 5:6 – Minimum Power over sampling duration in [Watts]. Byte 7:8 – Maximum Power over sampling duration in [Watts]. Byte 9:10 – Average Power over sampling duration in [Watts]. Byte 11:14 – IPMI Specification based Time Stamp based on SEL. For Mode = 2 – The time stamp specifies the end of the averaging window. Byte 15:18 – Statistics reporting time period. For Mode = 1 – Timeframe in milliseconds, over which the controller collects statistics. For Mode = 2 – Timeframe reflects the Averaging Time period in units. Byte 19 – Power Reading State [5:0] – Reserved. [6] =1b – Power Measurement active. =0b – No Power Measurement is available. [7] – Reserved.	
03h	Get Power Limit	Request Byte 1 – Group Extension Identification = DCh. Byte 2:3 – Reserved for future use. Write 0000h.	Note that this command returns Active Power Limit, which means that if the limit has not been activated with Activate/Deactivate Power Limit command this command will not return the limit set by the Set Power Limit command.
		Response Byte 1 – Completion Code =00h – Power Limit Active. =80h – No Set Power Limit. =C1h – Returned if DCMI mode is not present. Byte 2 – Group Extension Identification = DCh. Byte 3:4 – Reserved for future use. Byte 5 – Exception actions. Actions taken if the power limit is exceeded and cannot be controlled within the correction time limit. =00h – No action =01h – Hard Power Off system.	

Net Function = DCGRP (2Ch) LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
		Byte 6:7 – Power Limit Requested in [Watts]. Byte 8:11 – Correction time limit in milliseconds. Maximum time taken to limit the power, otherwise exception action will be taken as configured. Byte 12:13 – Reserved for future use. Byte 14:15 – Management application Statistics Sampling period in seconds.	
04h	Set Power Limit	Request Byte 1 – Group Extension Identification = DCh. Byte 2:4 – Reserved for future use. Byte 5 – Exception actions. Actions taken if the power limit exceeded and cannot be controlled within the correction time limit. =00h – No Action. =01h – Hard Power Off system. Byte 6:7 – Power Limit Requested in [Watts]. Byte 8:11 – Correction time limit in milliseconds. Maximum time taken to limit the power, otherwise exception action will be taken as configured. The Exception Action shall be taken if the system power usage constantly exceeds the specified power limit for more than the Correction Time Limit interval. The Correction Time Limit timeout automatically restarts if the system power meets or drops below the Power Limit. Byte 12:13 – Reserved for future use. Byte 14:15 – Management application Statistics Sampling period in seconds.	The following defaults are used by Intel® Node Manager during policy creation (see command "Set Node Manager Policy"): Domain= 0 Is Dcmi=TRUE Policy Trigger Type= 0 Aggressive CPU Power Correction= 0 Trigger Limit= 0
		Response Byte 1 – Completion Code =84h – Power Limit out of range. =85h – Correction Time out of range. =89h – Statistics Reporting Period out of range. =C1h – Returned if DCMI mode is not present. Byte 2 – Group Extension Identification = DCh.	
05h	Activate/Deactivate Power Limit	Request Byte 1 – Group Extension Identification = DCh. Byte 2 – Power Limit Activation =00h – Deactivate Power Limit. =01h – Activate Power Limit. Byte 3:4 – Reserved.	
		Response Byte 1 – Completion Code =C1h – Returned if DCMI mode is not present. Byte 2 – Group Extension Identification = DCh.	

3.4 External Intel® NM PTU Configuration and Control Commands

Intel® NM PTU is a platform resident technology that aides in the Platform Power Characterization.

The configuration and control commands are used by the external management software or BMC to configure and control and launch the Platform Power characterization.

Note: All the following commands are not supported when Intel® NM Feature Enabled is set to 'false' using spsFITC.

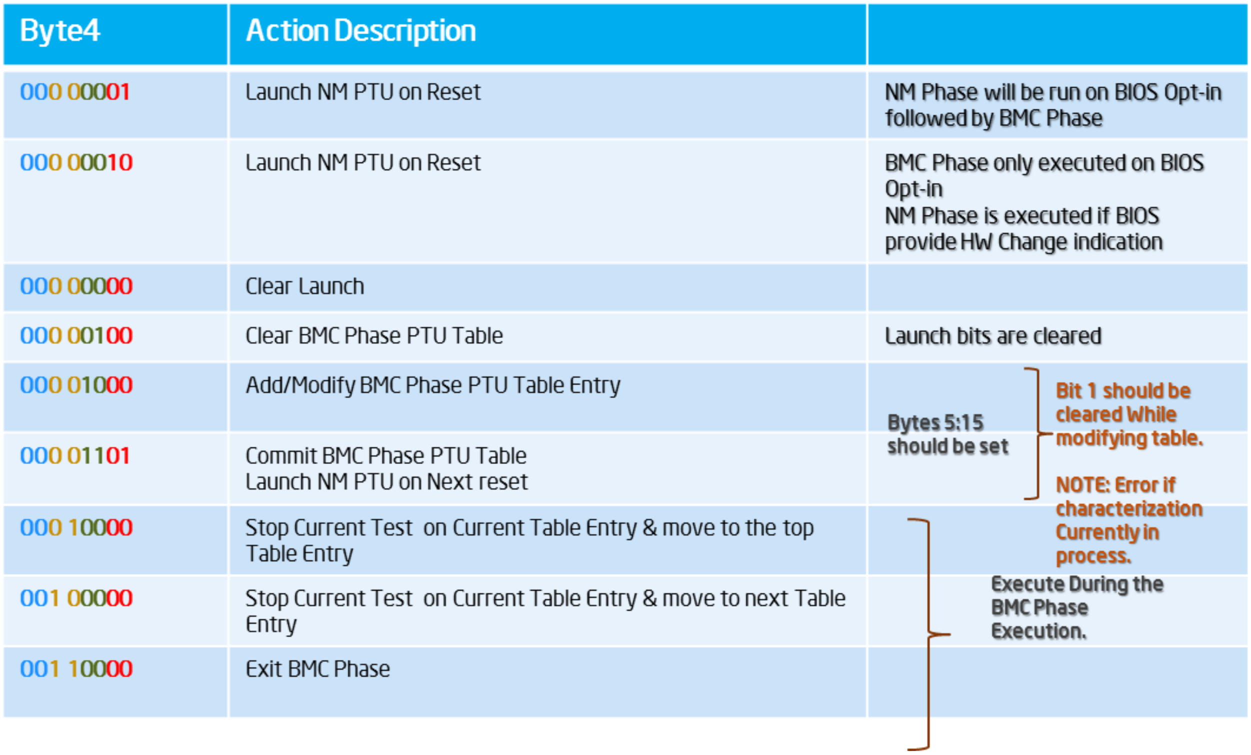
Table 3-4 External Intel® NM PTU Control Commands

Net Function = 2Eh LUN = {00b, 01b, 10b, 11b}			
Code	**Command**	**Request, Response Data**	**Description**
60h	Platform Characterization Launch Request	Request Byte 1:3 - Intel Manufacturer ID – 000157h, LS byte first Byte 4 – Command Request [1:0] – Launch Power Characterization =00b – Don't launch or Cancel Previous Launch. =01b – Launch Node Manager and BMC Characterization (Cleared to 00b after platform Reset) =10b – Launch BMC Characterization only =11b – Reserved [3:2] – BMC Table Configuration Phase Action =00b – No Action =01b – Clear All Table Entries =10b – Write Table Entry =11b – Reserved [5:4] – BMC Phase State Machine Action =00b – No Action =01b – Restart BMC Phase =10b – Skip to the next table entry =11b – Exit BMC Phase [7:6] – Reserved (Bytes 5:14 will be ignored when Byte4.Bits [3:2] != 10b) Byte 5 - Power Domain Id =00h – Platform =01h – CPU subsystem =02h – Memory subsystem =03h – 255h – Reserved Byte 6 – Power Draw Characterization Point =00h – Max =01h – Min =02h – Efficient =03h - 255h - Reserved Byte 7:10 – Delay [31:0] – Time in milliseconds to delay before executing next table entry (NOTE: 5 seconds is max for this field) Byte 11:14 – Time To Run [31:0] - Time in milliseconds to run current table Entry	Intel® NM PTU maintains an internal BMC phase table of the characterizations requested by the BMC which this command allows for the configuration and clearing of the characterization request data stored in it prior to NM PTU running during the next reboot cycle. This command can also be used while a characterization is in progress to alter the state machine flow to restart the BMC phase or to skip a BMC phase table entry. Launch NM Characterization, Byte 4 bit [1:0] =01b, shall launch NM characterization independent of BIOS HW Change. BMC phase if enabled will launch after Intel® NM characterization. Launch NM Characterization, Byte 4 bit [1:0] =10b, shall launch characterization with BMC phase only. BMC Table Configuration Phase Action, Byte 4 bit [3:2], is not allowed when the launch bits are set and when the characterization is in progress. All BMC table configuration Action commands should be done with "Clear Launch", Byte 4 bits [1:0] =00b, setting. Intel® NM FW shall send error when the BMC table Configuration is not allowed.
		Response	

Net Function = 2Eh LUN = {00b, 01b, 10b, 11b}			
Code	**Command**	**Request, Response Data**	**Description**
		Byte 1 – Completion Code Generic IPMI as defined in Section 2.15 plus the following command specific codes: =80h – BMC table configuration cannot be performed; clear launch bits to perform BMC Table Configuration. =81h – BMC table configuration cannot be performed; Wait until the current characterization completes. =82h – BMC phase state machine action only allowed during the BMC phase. Byte 2:4 – Intel Manufacturer ID – 000157h, LS byte first.	By default, the BMC table is empty and contains 8 configurable slots. BMC phase state machine action, Byte 4 bit [5:4], is only allowed during the BMC phase. Further restrictions are applicable on these actions during the BMC phase. Intel® NM FW sends error when the action is not allowed. Only one type of command request, Byte4, can be performed at a time. Except, clear launch with BMC Table configurable. All the changes from the 'Clear launch' to the 'Commit BMC Table' will be committed. Time To Run bytes 11:14 are limited by the following: • Minimum value of 0 will result in a default time to run of 7 seconds • Maximum value is FFFFh (65 seconds) Any other values outside of the range 0 – FFFFh will result in an error completion code being returned.

This summarizes the use of the bits contained in Command Request byte 4 of command 60h.

Table 3-5 Bits Contained in Command Request Byte 4



Byte4	Action Description	
000 00001	Launch NM PTU on Reset	NM Phase will be run on BIOS Opt-in followed by BMC Phase
000 00010	Launch NM PTU on Reset	BMC Phase only executed on BIOS Opt-in NM Phase is executed if BIOS provide HW Change indication
000 00000	Clear Launch	
000 00100	Clear BMC Phase PTU Table	Launch bits are cleared
000 01000	Add/Modify BMC Phase PTU Table Entry	Bytes 5:15 should be set Bit 1 should be cleared While modifying table. NOTE: Error if characterization Currently in process.
000 01101	Commit BMC Phase PTU Table Launch NM PTU on Next reset	
000 10000	Stop Current Test on Current Table Entry & move to the top Table Entry	Execute During the BMC Phase Execution.
001 00000	Stop Current Test on Current Table Entry & move to next Table Entry	
001 10000	Exit BMC Phase	

Table 3-6 summarizes the PTU launch action based on the status of BIOS activate and Command Request byte 4[1:0].

Table 3-6 NM PTU Launch actions

Bios Activate	IPMI command 60h Launch byte 4, bits [1:0]	PTU Execution Action after Reset
0	00b	No Action
0	01b	Intel® NM then BMC
0	10b	BMC
1	00b	Intel® NM
1	01b	Intel® NM then BMC
1	10b	Intel® NM then BMC

Table 3-7 External Intel® NM PTU Configuration Commands

Net Function = 2Eh LUN = {00b, 01b, 10b, 11b}			
Code	**Command**	**Request, Response Data**	**Description**
61h	Get Node Manager Power Characterization Range	Request Byte 1:3 - Intel Manufacturer ID – 000157h, LS byte first. Byte 4 – Domain ID [3:0] – Domain ID (Identifies the Intel® NM PTU domain to obtain characterization data from) =00h – Entire platform =01h – CPU subsystem =02h – Memory subsystem =03h-0Fh – Reserved [6:4] – Reserved [7] – Power domain. This field is ignored for Domain ID other than 0. =0b – Primary (AC) side power domain (default) =1b – Secondary (DC) side power domain.	External power manager/BMC can collect the power draw data determined during previous Intel® NM PTU characterizations which are stored persistently using this command. Intel® Node Manager responds with the characterization data from the requested domain ID. The data returned is from the latest characterization.
		Response Byte 1 – Completion Code Generic IPMI as defined in Section 2.15 plus the following command specific codes: =81h – Invalid Domain ID. This code is also returned when for domain 03h on monolithic systems. =82h – Invalid Characterization. Indicates an error during the characterization process. Byte 2:4 – Intel Manufacturer ID – 000157h, LS byte first. Byte 5:8 – IPMI Specification based Time Stamp which identifies when the Calibration data collected in Intel® ME. Byte 9:10 – Maximum Power Draw in [Watts].This is an unsigned integer value. Byte 11:12 – Minimum Power Draw in [Watts]. This is an unsigned integer value Byte 13:14 – Efficient Power Draw in [Watts]. This is an unsigned integer value Byte 15 – Intel® NM PTU version ID number BCD encoded =A.B where A = Major ID and B = Minor ID Byte 16 - Inflection Point Power Value Determination [0] – Maximum =0b – Consecutive data within ±3% variability =1b – Average of data within 1 standard deviation from entire sample set [1] – Minimum =0b – Consecutive data within ±3% variability =1b – Average of data within 1 standard deviation from entire sample set [2] – Efficient (not applicable for domain ID equal to 02h) =0b – Consecutive data within ±3% variability =1b – Average of data within 1 standard deviation from entire sample set [7:3] – Reserved	

Table 3-8 provides special encoding of the power draw data returned with Get Node Manager Power Characterization Range command response in bytes 9:14.

Table 3-8 Special encodings of the power draw data

Value	Comment
0x0000	Characterization has not been executed for the requested domain.
0xFFF8	Received no data for sensor domain and sensor registration failed (incorrect XML configuration).
0xFFF9	Received no data for sensor domain but sensor registration passed.
0xFFFA	Monitoring service flagged power data as outside of expected min (0) and max (32767) range.
0xFFFB	Sensor device not ready or unavailable (PECI, IPMB or BMC issue).
0xFFFC	Reserved for future use.
0xFFFD	Sensor device returning 0 for power data during characterization.
0xFFFE	Reserved for future use.
0xFFFF	Thermal event occurred during the execution of a particular characterization point such as PROCHOT# assertion or the processors thermal control circuit was triggered.

Table 3-9 OEM Intel® NM PTU Notification Command

Net Function = 30h LUN = {00b, 01b, 10b, 11b}			
Code	**Command**	**Request, Response Data**	**Description**
E9h	OEM Platform Power Characterization Notification	Request Byte 1 – Notification [1:0] – Notification Information When bits [7:2] = {00h or 01h} then bits [1:0] = Domain ID =00b – Platform =01b – CPU subsystem =10b – Memory subsystem =11b – Reserved When bits [7:2] = {02h or 03h} then bits [1:0] = Phase Status =00b – Phase Completed successfully =01b – Phase Interrupted All others – Reserved [7:2] – Notification Type =00h – Intel® NM Phase Characterization Progression =01h – BMC Phase Characterization Progression =02h – Intel® NM Phase Status =03h – BMC Phase Status All others – Reserved Byte 2 – Power Characterization Point [1:0] – Power Characterization Type If byte 1 bits[7:2] = {00h or 01h} then bits [1:0] = Power Type =00h – Maximum =01h – Minimum =02h – Efficient =03h – Reserved If byte 1 bits[7:2] = {02h or 03h} then bits [1:0] = 00b [7:2] – Reserved	Notification sent by Intel® ME to BMC indicating current phase status of characterization. BMC in its response shall supply a time delay in milliseconds to accommodate the time needed to ramp up or ramp down fans or IO allowing the BMC to delay the next Intel® NM PTU state machine action. 5 seconds is the maximum allowed delay permitted for the BMC to return in its response. Intel® ME FW will cap the delay at 5 seconds when the BMC responds with a value greater than 5 seconds.

Net Function = 30h LUN = {00b, 01b, 10b, 11b}			
Code	**Command**	**Request, Response Data**	**Description**
		Byte 3 - Intel® ME Power Characterization Stage [1:0] – Stage (Identifies the internal ME FW state) If byte 1 bits[7:2] = {00h or 01h} then bits [1:0] = Characterization Stage =00h – Reserved =01h – Initialize (Initializing the Characterization) =02h – Monitor (Monitoring Power) =03h – Done (Characterization Complete) If byte 1 bits[7:2] = {02h or 03h} then bits [1:0] = 00b [7:2] - Reserved	
		Response Byte 1 – Completion Code Generic IPMI as defined in Section 2.15 Byte 2:5 – Delay time in milliseconds	

Table 3-10 summarizes the action of the delay provided in the BMC response to the E9h command sent by the Intel® ME firmware.

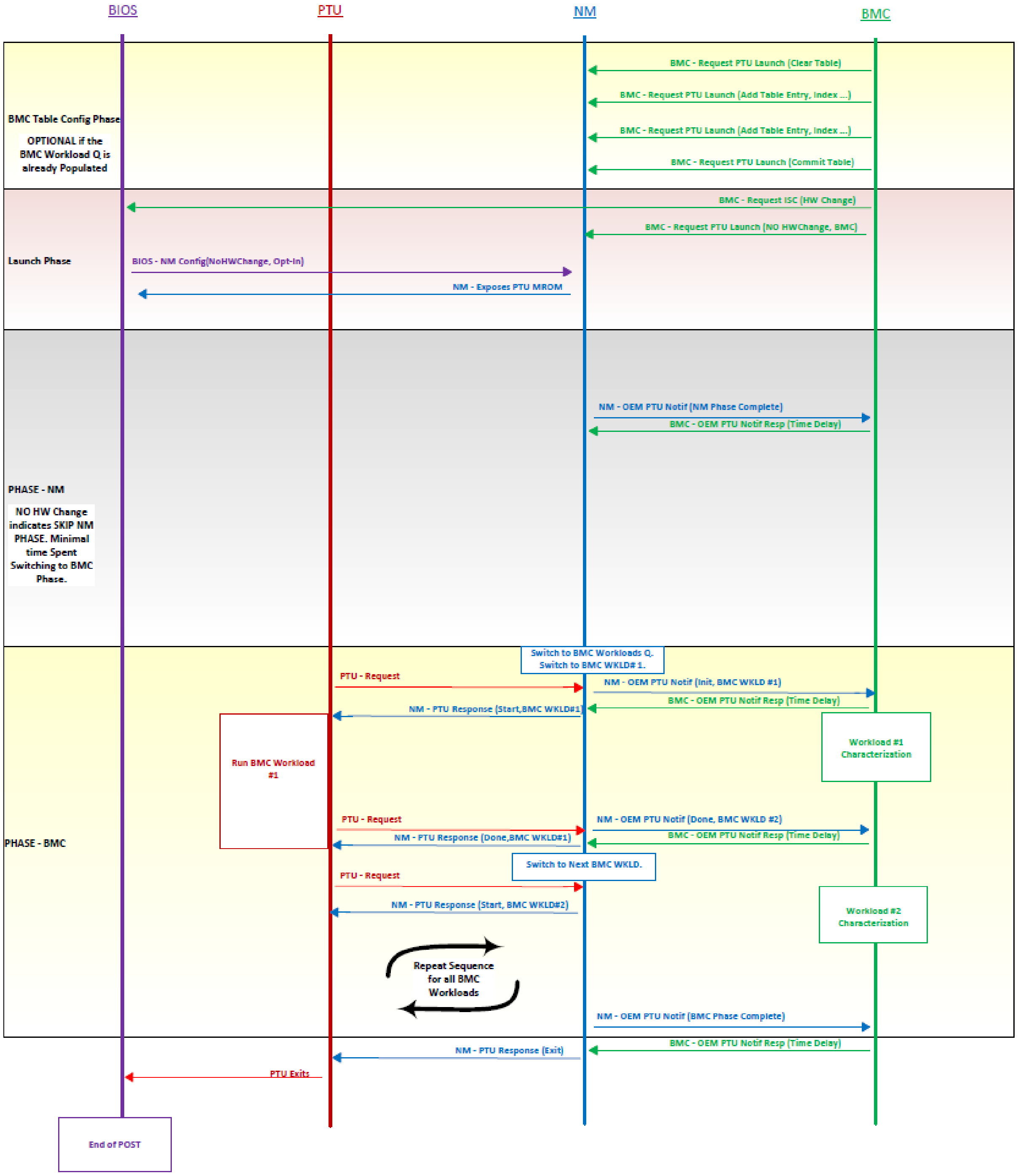
Table 3-10 BMC response to the E9h command

Notification Type (Request Byte4[7:2])	Notification Information	Point	Stage	Delay Action
00h = Intel® NM Phase Characterization Progression 01h = BMC Phase Characterization Progression	All applicable domains	All applicable points	Initialize	Delay enforced before execution of the characterization.
	All applicable domains	All applicable points	Monitor	Delay enforced before collecting the power consumption readings.
	All applicable domains	All applicable points	Done	Ignored.
02h = Intel® NM Phase Status 03h = BMC Phase Status	Phase Completed successfully	00h	00h	Delay before executing the next phase or next stage.
	Phase Interrupted	00h	00h	Ignored due to Intel® ME having exited the characterization in progress.

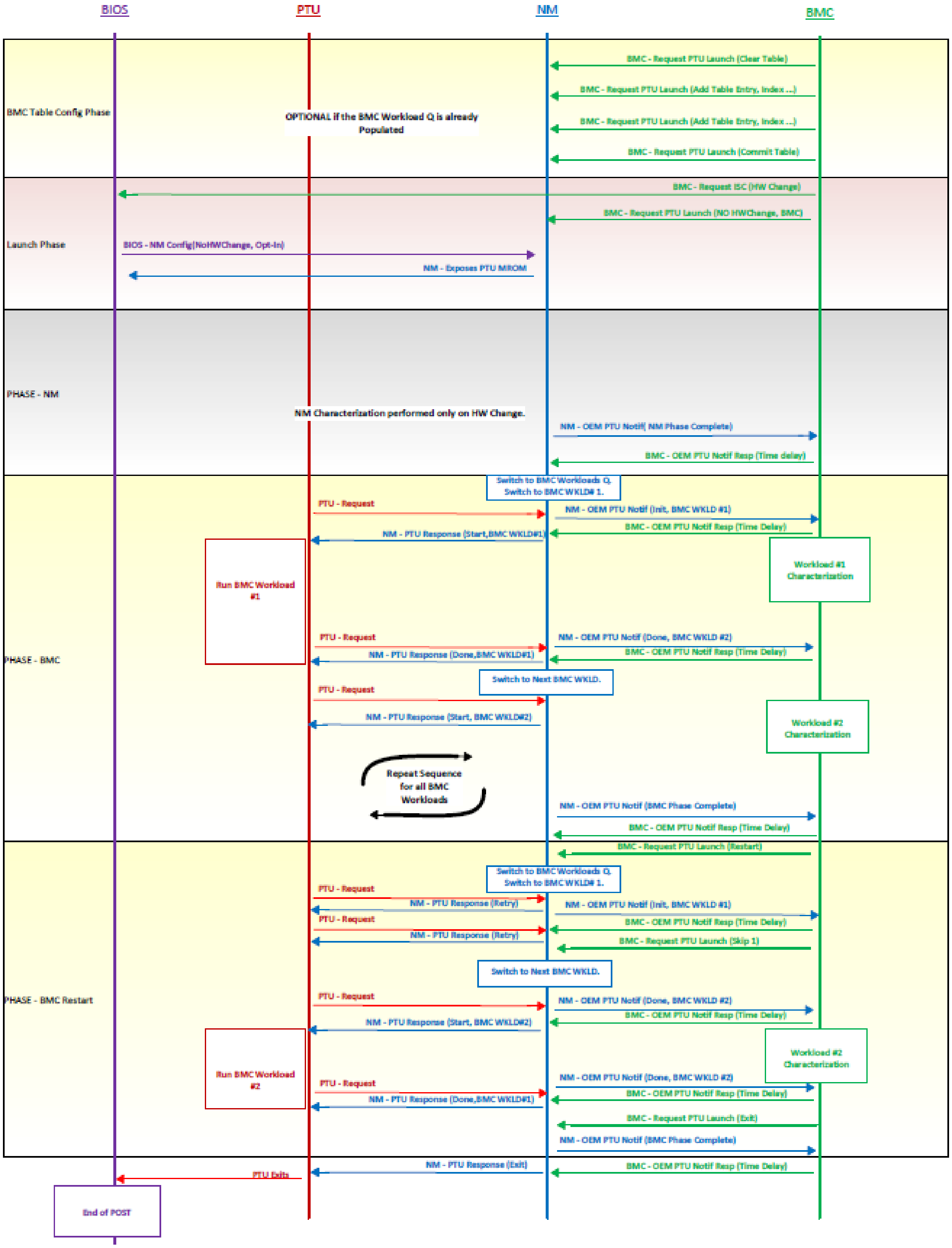
Figure 3-1 depicts three commonly used scenarios of launching the Intel® NM PTU characterization process.

Single BMC phase only with no hardware change detected (BIOS activate de-asserted).

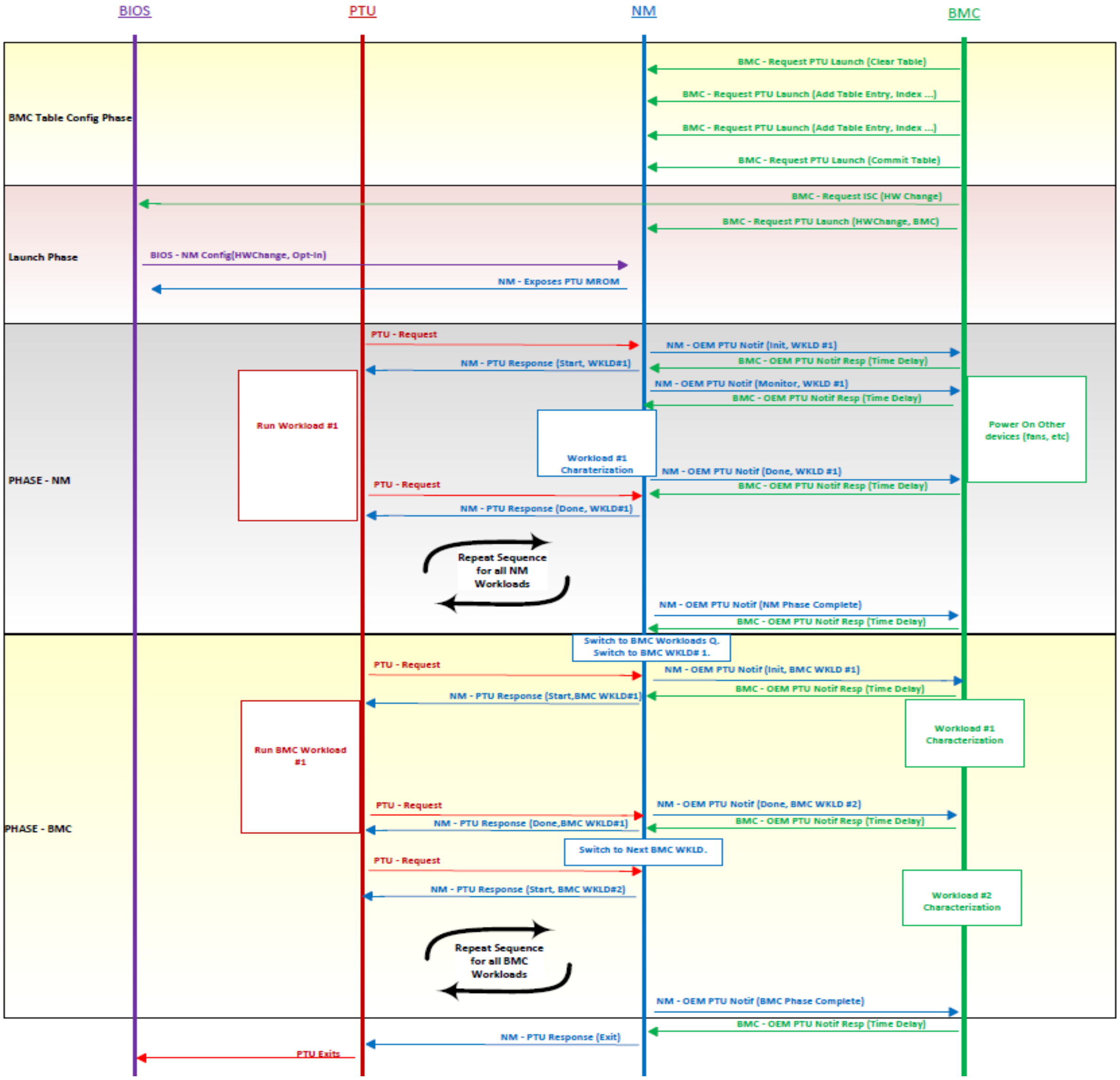
Figure 3-1 Commonly Used Scenarios of Launching the Intel® NM PTU


BIOS
PTU
NM
BMC
BMC Table Config Phase
OPTIONAL if the BMC Workload Q is already Populated
BMC - Request PTU Launch (Clear Table)
BMC - Request PTU Launch (Add Table Entry, Index ...)
BMC - Request PTU Launch (Add Table Entry, Index ...)
BMC - Request PTU Launch (Commit Table)
Launch Phase
BMC - Request ISC (HW Change)
BMC - Request PTU Launch (NO HWChange, BMC)
BIOS - NM Config(NoHWChange, Opt-In)
NM - Exposes PTU MROM
PHASE - NM
NO HW Change indicates SKIP NM PHASE. Minimal time Spent Switching to BMC Phase.
NM - OEM PTU Notif (NM Phase Complete)
BMC - OEM PTU Notif Resp (Time Delay)
PHASE - BMC
Switch to BMC Workloads Q. Switch to BMC WKLD# 1.
PTU - Request
NM - OEM PTU Notif (Init, BMC WKLD #1)
BMC - OEM PTU Notif Resp (Time Delay)
NM - PTU Response (Start,BMC WKLD#1)
Run BMC Workload #1
Workload #1 Characterization
PTU - Request
NM - OEM PTU Notif (Done, BMC WKLD #2)
NM - PTU Response (Done,BMC WKLD#1)
BMC - OEM PTU Notif Resp (Time Delay)
Switch to Next BMC WKLD.
PTU - Request
NM - PTU Response (Start, BMC WKLD#2)
Workload #2 Characterization
Repeat Sequence for all BMC Workloads
NM - OEM PTU Notif (BMC Phase Complete)
BMC - OEM PTU Notif Resp (Time Delay)
NM - PTU Response (Exit)
PTU Exits
End of POST

Multiple BMC phases (state machine restarts) with no hardware change detected (BIOS activate de-asserted).


BIOS
PTU
NM
BMC
BMC Table Config Phase
OPTIONAL if the BMC Workload Q is already Populated
BMC - Request PTU Launch (Clear Table)
BMC - Request PTU Launch (Add Table Entry, Index ...)
BMC - Request PTU Launch (Add Table Entry, Index ...)
BMC - Request PTU Launch (Commit Table)
Launch Phase
BMC - Request ISC (HW Change)
BMC - Request PTU Launch (NO HWChange, BMC)
BIOS - NM Config(NoHWChange, Opt-In)
NM - Exposes PTU MROM
PHASE - NM
NM Characterization performed only on HW Change.
NM - OEM PTU Notif(NM Phase Complete)
BMC - OEM PTU Notif Resp (Time delay)
PHASE - BMC
Switch to BMC Workloads Q. Switch to BMC WKLD# 1.
PTU - Request
NM - OEM PTU Notif (Init, BMC WKLD #1)
BMC - OEM PTU Notif Resp (Time Delay)
NM - PTU Response (Start,BMC WKLD#1)
Run BMC Workload #1
Workload #1 Characterization
PTU - Request
NM - OEM PTU Notif (Done, BMC WKLD #2)
NM - PTU Response (Done,BMC WKLD#1)
BMC - OEM PTU Notif Resp (Time Delay)
Switch to Next BMC WKLD.
PTU - Request
NM - PTU Response (Start, BMC WKLD#2)
Workload #2 Characterization
Repeat Sequence for all BMC Workloads
NM - OEM PTU Notif (BMC Phase Complete)
BMC - OEM PTU Notif Resp (Time Delay)
PHASE - BMC Restart
BMC - Request PTU Launch (Restart)
Switch to BMC Workloads Q. Switch to BMC WKLD# 1.
PTU - Request
NM - PTU Response (Retry)
NM - OEM PTU Notif (Init, BMC WKLD #1)
PTU - Request
BMC - OEM PTU Notif Resp (Time Delay)
NM - PTU Response (Retry)
BMC - Request PTU Launch (Skip 1)
Switch to Next BMC WKLD.
PTU - Request
NM - OEM PTU Notif (Done, BMC WKLD #2)
BMC - OEM PTU Notif Resp (Time Delay)
NM - PTU Response (Start, BMC WKLD#2)
Workload #2 Characterization
Run BMC Workload #2
NM - OEM PTU Notif (Done, BMC WKLD #2)
PTU - Request
BMC - OEM PTU Notif Resp (Time Delay)
NM - PTU Response (Done,BMC WKLD#1)
BMC - Request PTU Launch (Exit)
NM - OEM PTU Notif (BMC Phase Complete)
NM - PTU Response (Exit)
BMC - OEM PTU Notif Resp (Time Delay)
PTU Exits
End of POST

Hardware change detected (BIOS activate asserted) which runs both the Intel® NM and BMC phases.


BIOS
PTU
NM
BMC
BMC Table Config Phase
BMC - Request PTU Launch (Clear Table)
BMC - Request PTU Launch (Add Table Entry, Index ...)
BMC - Request PTU Launch (Add Table Entry, Index ...)
BMC - Request PTU Launch (Commit Table)
Launch Phase
BMC - Request ISC (HW Change)
BMC - Request PTU Launch (HWChange, BMC)
BIOS - NM Config(HWChange, Opt-In)
NM - Exposes PTU MROM
PHASE - NM
PTU - Request
NM - OEM PTU Notif (Init, WKLD #1)
BMC - OEM PTU Notif Resp (Time Delay)
NM - PTU Response (Start, WKLD#1)
NM - OEM PTU Notif (Monitor, WKLD #1)
BMC - OEM PTU Notif Resp (Time Delay)
Run Workload #1
Power On Other devices (fans, etc)
Workload #1 Charaterization
NM - OEM PTU Notif (Done, WKLD #1)
BMC - OEM PTU Notif Resp (Time Delay)
PTU - Request
NM - PTU Response (Done, WKLD#1)
Repeat Sequence for all NM Workloads
NM - OEM PTU Notif (NM Phase Complete)
BMC - OEM PTU Notif Resp (Time Delay)
Switch to BMC Workloads Q. Switch to BMC WKLD# 1.
PHASE - BMC
PTU - Request
NM - OEM PTU Notif (Init, BMC WKLD #1)
BMC - OEM PTU Notif Resp (Time Delay)
NM - PTU Response (Start,BMC WKLD#1)
Run BMC Workload #1
Workload #1 Characterization
PTU - Request
NM - OEM PTU Notif (Done, BMC WKLD #2)
BMC - OEM PTU Notif Resp (Time Delay)
NM - PTU Response (Done,BMC WKLD#1)
Switch to Next BMC WKLD.
PTU - Request
NM - PTU Response (Start, BMC WKLD#2)
Workload #2 Characterization
Repeat Sequence for all BMC Workloads
NM - OEM PTU Notif (BMC Phase Complete)
BMC - OEM PTU Notif Resp (Time Delay)
NM - PTU Response (Exit)
PTU Exits

3.5 IPMI Sensors Implemented by Intel® Node Manager FW

Table 3-11 summarizes the sensors exposed by Intel® Node Manager FW.

Table 3-11 IPMI Sensors Implemented by Intel® NM

Event	Sensor Type	Event Dir	Event Type	Alert Immediate (See Section 0)
Intel® NM Exception Event	DCh – OEM	0 – assertion	72h – OEM	No
Intel® NM Health Event	DCh – OEM	0 – assertion	73h – OEM	Yes
Intel® NM Operational Capabilities Change Event	DCh – OEM	0 – assertion 1 – deassertion	74h – OEM	Yes
Intel® NM Alert Threshold Exceeded	DCh – OEM	0 – assertion 1 – deassertion	72h – OEM	Yes
Typical Power Consumption in Sx	DCh – OEM	N/A	N/A	N/A
PSU Status	08h – Power Supply	0 – assertion 1 – deassertion	6Fh – Sensor specific	No
Power Threshold Event	09h – Power Unit	0 – assertion 1 – deassertion	05h – Generic – limit status	No
Power Off Event	12h – System Event	0 – assertion	0Ah – Generic - system state	No
Chassis Power	0Bh – Other units	0 – assertion 1 – deassertion	01h – Threshold	No
HSC Input Power	0Bh – Other Units	0 – assertion 1 – deassertion	01h – Threshold	No
HSC Voltage	02h - Voltage	0 – assertion 1 – deassertion	01h – Threshold	No
HSC Status Byte Low	0Bh – Other Units	0 – assertion 1 – deassertion	6Fh – Sensor Specific	No
HSC Status Byte High	0Bh – Other Units	0 – assertion 1 – deassertion	6Fh – Sensor Specific	No
HSC Status MFR Specific	0Bh – Other Units	0 – assertion 1 – deassertion	6Fh – Sensor Specific	No
HSC Status Input	0Bh – Other Units	0 – assertion 1 – deassertion	6Fh – Sensor Specific	No
PSU Temperature	01h – Temperature	0 – assertion 1 – deassertion	01h - Threshold	No
PMBUS OEM device	08h – Power Supply	0 – assertion 1 – deassertion	01h – Threshold	No
PSU DC Power	0Bh - Other Units	0 – assertion 1 – deassertion	01h - Threshold	No

Firmware SKU Availability column specifies whether the sensor is supported only in some SKU:

- A – sensor available in all FW SKUs,
- Intel® NM – sensor available only when Intel® Node Manager Feature Enabled parameter is set to 'true' using Flash Image Tool,
- PECI – sensor available only when PECI is attached to the PCH.

Reading Availability column specifies when the sensor reading is available:

- A – always when Intel® ME is On,
- H – when HOST CPU is On,
- O – after reception of END_OF_POST notification,
- E – No reading available (Event Only).

Defaults Configurable in FIT column defines whether the default configuration of the sensors can be set using Flash Image Tool. The default configuration includes:

- Thresholds
- Event Enable Mask
- Scanning Periods
- Scanning Enable Flag
- Per-sensor Event Enable Flag

The default configuration is applied by Intel® NM Firmware at first management engine startup after G3 condition on Global Platform Reset (see definition in PCH EDS).

Table 3-12 Intel® NM Sensors

Sensor #	Description	Firmware SKU Availability	Reading Availability	Defaults Configurable in Flash Image Tool	Notes
21	Typical Power Consumption in Sx	Intel® NM	A	Yes	OEM sensor that presents Typical Power consumption in Sx state. The value of this sensor is constant and may only change upon HW configuration change. Intel® Node Manager uses this data to estimate the power consumed in Sx state. If this sensor is configured, power readings from PSUs in Sx are ignored.
24	Intel® NM Exception Event Sensor	Intel® NM	E	No	OEM Event only sensor. Event will be sent each time when maintained policy power limit is exceeded over Correction Time Limit. Events are sent no faster than every 300 milliseconds. Events are not queued internally - it means that some events may be lost if they are generated more often than 300ms.

Sensor #	Description	Firmware SKU Availability	Reading Availability	Defaults Configurable in Flash Image Tool	Notes
					First occurrence of not acknowledged event will be retransmitted according to IPMB specification after ~230 milliseconds. "Command illegal for specified sensor or record type (CDh)" error code is returned in response to the following commands: Get Sensor Reading, Set/Get Sensor Thresholds, Re-Arm Sensor Events, Set/Get Sensor Event Enable.
25	Intel® NM Health Event Sensor	Intel® NM	E	No	OEM Event only sensor used to send events about integrity of Intel® Node Manager policy or necessary sensor readings. Events are sent no faster than every 300 milliseconds. Events are not queued internally - it means that some events may be lost if they are generated more often than 300ms. First occurrence of not acknowledged event will be retransmitted according to IPMB specification after ~230 milliseconds. "Command illegal for specified sensor or record type (CDh)" error code is returned in response to the following commands: Get Sensor Reading, Set/Get Sensor Thresholds, Re-Arm Sensor Events, Set/Get Sensor Event Enable.
26	Intel® NM Operational Capabilities sensor	Intel® NM	A	No	OEM sensor, that value will indicate the operational capabilities of the sensor. Whenever the sensor value changes, an immediate alert is also sent. See the event description for the description of the values of the sensor. "Command illegal for specified sensor or record type (CDh)" error code is returned in response to the following commands: Set/Get Sensor Thresholds.

Sensor #	Description	Firmware SKU Availability	Reading Availability	Defaults Configurable in Flash Image Tool	Notes
					Events are sent no faster than every 300 milliseconds. Events are not queued internally - it means that some events may be lost if they are generated more often than 300ms. Current value of not acknowledged capabilities sensor will be retransmitted according to IPMB specification after ~230 milliseconds.
27	Intel® NM Alert Threshold Exceeded sensor	Intel® NM	E	No	OEM Event only sensor used to send events when Node Manager detects that a specified alert threshold for one of the policies is exceeded. Events are sent no faster than every 300 milliseconds. Events are not queued internally - it means that some events may be lost if they are generated more often than 300ms. First occurrence of Threshold exceeded event assertion/deassertion will be retransmitted according to IPMB specification after ~230 milliseconds. "Command illegal for specified sensor or record type (CDh)" error code is returned in response to the following commands: Get Sensor Reading, Set/Get Sensor Thresholds, Re-Arm Sensor Events, Set/Get Sensor Event Enable.
58	Power Off Sensor	Intel® NM	A	Yes	Sensor is defined in [DCMI].
59	Power Threshold Sensor	Intel® NM	A	Yes	Sensor is defined in [DCMI].
102-109	Power Supply <n> Status n = 0-7	Intel® NM	A	Yes	Discrete sensor. This sensor is used for getting the Power Supply status.
173	Chassis power	Intel® NM	A	Yes	Sensor that represents chassis power consumption.
41, 45, 224, 230	HSC Input Power	Intel® NM	A	Yes	Sensor represent power reported by Hot Swap Controller.
42, 46, 225, 231	HSC Voltage	Intel® NM	A	Yes	Sensor represent voltage reported by Hot Swap Controller.

Sensor #	Description	Firmware SKU Availability	Reading Availability	Defaults Configurable in Flash Image Tool	Notes
40, 44, 220, 226	HSC Status Byte Low	Intel® NM	A	Yes	Sensor represent lower byte of Hot Swap Controller's STATUS_WORD.
66, 68, 221, 227	HSC Status Byte High	Intel® NM	A	Yes	Sensor represent higher byte of Hot Swap Controller's STATUS_WORD.
94, 111, 222, 228	HSC Status MFR Specific	Intel® NM	A	Yes	Sensor represent Hot Swap Controller's STATUS_MFR_SPECIFIC.
159, 161, 223, 229	HSC Status Input	Intel® NM	A	Yes	Sensor represent Hot Swap Controller's STATUS_INPUT.
86 - 93	PSU Temperature	Intel® NM	A	Yes	Sensor represents PSU's READ_TEMPERATURE_1.
178	NM SmaRT&CLST Sensor	Intel® NM	A	No	Sensor represents state of system throttling due SmaRT & CLST functionality.
128 - 158	PMBUS OEM device	Intel® NM	A	Yes	Sensor represents user defined device reading.
164 - 171	PSU DC Power	Intel® NM	A	Yes	Sensor represents PSU's output power.
190	Core CUPS Sensor	Intel® NM/ SiEn	H	No	Sensor that represents CPU Utilization on all the available CPUs.
191	IO CUPS Sensor	Intel® NM/ SiEn	H	No	Sensor that represents IO Utilization.
192	Memory CUPS Sensor	Intel® NM/ SiEn	H	No	Sensor that represents Memory Utilization on all the available Memory channels.
193	CUPS Event Sensor	Intel® NM/ SiEn	E	No	OEM Event only sensor used to send events when PTAS-CUPS detects that a specified alert threshold for one of the policies is trigger.

3.5.1 Typical Platform Power Consumption in Sx state

This is a sensor that does not generate any events and supports only Get Sensor Reading command. The sensor is used in Platform Event messages to BMC when policy limit is exceeded. This Sensor presents Typical Power consumption in Sx state in Watts. The value of this sensor is constant and may only change upon HW configuration change. Intel® NM uses this data to estimate the power consumed by the platform in Sx state.

3.5.2 Intel® NM Exception Event Sensor

This is an Event-Only sensor that does not support Get Sensor Reading command nor re-arm IPMI command. The sensor is used in Platform Event messages to BMC when policy limit is exceeded.

3.5.3 Intel® NM Threshold Exceeded Event Sensor

This is an Event-Only sensor that does not support Get Sensor Reading command nor rearm IPMI command. The sensor is used in Immediate Alert Event messages to remote console when policy threshold or policy limit is exceeded.

3.5.4 Intel® NM Health Sensor

OEM Event only sensor used to send events about integrity of Intel® Node Manager Policy or necessary sensor readings.

3.5.5 Intel® NM Operational Capabilities Change Sensor

OEM sensor indicating the operational capabilities of the Intel® NM service. Whenever the sensor value changes, an immediate alert is also sent. The Operational Capabilities sensor supports both assertion and deassertion events. The assertion event means that Intel® NM has gained some capabilities while the deassertion one indicates that one or more capabilities have been lost. The event always delivers the current state of the Intel® NM Operational Capabilities in Event Data 1 despite this is an assertion or deassertion event. See the event description for the description of the values of the sensor.

Table 3-13 Intel® NM Operational Capabilities

Intel® NM Operational Capability	Description	Enable Condition
Policy Interface	Intel® NM accepts policy configuration change requests.	1. Intel® NM operational image is running. 2. Flash wear-out protection mechanism did not lock access to SPI Flash.
Monitoring Capability	Intel® NM monitors total system power consumption, CPU power and Memory power.	1. There is at least one of power sensor configured for the current platform power state. 2. Intel® NM did not detect power reading failure conditions. **Note** - The monitoring capability reports the status of the whole domain, not a single device in the domain. A reading failure condition for the whole domain is detected when expected readings from a single device are missing for 20 seconds.
Power Limiting Capability	Intel® NM is able to limit power consumption.	1. Intel® NM has CPU information from BIOS. 2. Total Power Monitoring Capability is enabled or Intel® Node Manager enforces nonzero CPU or memory throttling level. 3. OSPM driver is responding to requests from Intel® Node Manager. 4. PECI over DMI interface is working properly. **Note** – When Intel® NM doesn't actively limit power, either because no power limit is configured or power consumption is below all the configured limits, Intel® NM may not detect a change in this capability. When platform/CPU power limit enforced is below capabilities and

		communication with host OS stops (last P-state is enforced and all cores are parked) Intel® NM may not detect a change in this capability.

3.5.6 PSU Status Sensors

This sensor is used for getting the Power Supply status. There is a separate sensor per PSU. Sensors are numbered from 102 to 109. Each sensor supports the following bits:

Table 3-14 PSU Status Sensors

Sensor-specific Offset	PSU Status Assertion Bit	Description
00h	Presence detected	Asserted if power supply module is present. Events are only logged for power supply presence upon changes in the presence status after AC power is applied (no events logged for initial state).
01h	Power supply failure detected	Asserted if power supply module has failed. This is a logical OR of all the faults such as IOUT OC FAULT, POUT OP FAULT (STATUS_IOUT), OT FAULT (STATUS_TEMPERATURE), FAN 1 FAULT, FAN 2 FAULT (STATUS_FANS_1_2).
02h	Predictive failure	Asserted if a condition that is likely to lead to a power supply module failure has been detected, such as a failing fan. This is a logical OR of all the PSU status bits such as IOUT OC WARNING, POUT OP WARNING (STATUS_IOUT), IIN OC WARNING, PIN OP WARNING (STATUS_INPUT), OT WARNING (STATUS_TEMPERATURE), FAN 1 WARNING, FAN 2 WARNING (STATUS_FANS_1_2).
03h	Power Supply input lost (AC/DC) aka. AC lost	Asserted if there is no AC power input to a power supply module. It would be reported when Unit Off for Low Input Voltage (STATUS_INPUT) would be asserted.
04h	Power Supply lost or out-of-range	Assertion event would be send if VIN_UV_FAULT (STATUS_INPUT) would be asserted in PSU.
05h	Power Supply input out-of-range, but present	Assertion event would be send if VIN_UV_WARNING (STATUS_INPUT) would be asserted in PSU.
06h	Configuration Error	Asserted for not supported PSU, when other information cannot be read. Configuration error is also generated if SmaRT & CLST needs to be disabled due to not compatible PSU attached.

3.5.7 HSC Status Byte Low, Status Byte High and Status Input Sensors

These sensors are used for getting the Hot Swap Controllers status. Each sensor supports bit offset described in the HSC datasheet for every HSC status sensor/register.

3.5.8 HSC Status MFR Specific Sensor

This sensor is used for getting the Hot Swap Controller status. Each sensor supports HSC STATUS_MFR register value described in the HSC datasheet.

3.5.9 Intel SmaRT & CLST Sensor

This sensor is used for indicating that the SmaRT & CLST functionality was triggered to start protecting the platform from failures caused by PSU's working conditions. When triggered an Intel® Node Manager SmaRT & CLST Event is send with a PSU Status sensor number associated with the first PSU for which the event condition was detected. The event contains a severity code in byte 2 which translates to the PSU status as follows:

Table 3-15 Intel SmaRT & CLST Sensor Severity Codes

Severity Code	PSU Condition
00h - transition to OK	All present PSU faults disappeared.
01h - transition to noncritical from OK	SMBAlert# has been asserted by PSU but it should be ignored (e.g. PSU goes to off state due insufficient input voltage).
02h - transition to critical from less severe	SMBAlert# was asserted due to one of the following PSU events: - UV_Fault or - OT_Warning or - OC_Warning
04h - transition to noncritical from more severe	There has been a critical condition and one of the following events happened: - UV Fault lasted more than preconfigured time (default 500ms) or - PSU causing disappeared or - PSU unit become off or - OT warning lasted more than 500ms for one PSU and there is another PSU present with no OT warning.

3.5.10 Compute Usage per Second (CUPS)

The CUPS IPMI API shall follow the IPMI Standard Specification.

Table 3-16 CUPS IPMI Commands Definition

Net Function = OEM (2Eh) LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
64h	Get CUPS Capabilities	Request Byte 1:3 = Intel manufacturers ID – 000157h, LS byte first	This command allows BMC or Remote Console to discover the PTAS-CUPS Capabilities. The capabilities depend on the features implemented by the PTAS-CUPS Intel® ME FW Module, HW support.
		Response Byte 1 – Completion Code Generic IPMI as defined in Section 2.15 Byte 2:4 – Intel manufacturers ID – 000157h, LS byte first Byte 5 – CUPS Capabilities [0] – CUPS feature availability =0b – CUPS feature is disabled =1b – CUPS feature is enabled [1:7] – Reserved should be 0000000b	

Net Function = OEM (2Eh) LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
		Byte 6 – CUPS Policy [0] – CUPS policy configuration =0b – CUPS policies configuration not available =1b – CUPS policies configuration available [1:7] – Reserved should be 0000000b Byte 7 – CUPS version =1 – version used with 4th generation Intel® Core™ processors Byte 8 – Reserved for future use	
65h	Get CUPS Data	Request Byte 1:3 = Intel manufacturers ID – 000157h, LS byte first Byte 4 = Parameter Selector =1 – CUPS Index =2 – Dynamic load factors =3-255 - Reserved	This command allows BMC or Remote Console to retrieve the CUPS data. Request byte 4 is the parameter selector which is provided in the next table. The number of response bytes varies based on the parameter selector requested.
		Response Byte 1 – Completion Code Generic IPMI as defined in Section 2.15 plus the following command specific codes: =80h – Internal error =81h – Hardware resources not available =82h – Not initialized Byte 2:4 – Intel manufacturers ID – 000157h, LS byte first Byte 5:N – Requested parameter data For request byte 4 = 1 (CUPS Index) Data bytes 1:2 – CUPS Index (MS-byte first) For request byte 4 = 2 (Dynamic Load Factors) Data bytes 1:6 – Dynamic load factors (MS-byte first) [1:2] – CPU CUPS dynamic Load factor [3:4] – Memory CUPS dynamic Load factor [5:6] – IO CUPS dynamic Load factor	
66h	Set CUPS Configuration	Request Byte 1:3 = Intel manufacturers ID – 000157h, LS byte first Byte 4 – Enable CUPS Feature [0] – Enable/Disable CUPS =0b – Disable CUPS =1b – Enable CUPS [1:7] – Reserved should be 0000000b Byte 5 – CUPS Load Factor Valid Domain Mask [0] – Core Load Factor =0b – Ignore Core Load Factor =1b – Set Core Load Factor [1] – Memory Load Factor =0b – Ignore Memory Load Factor =1b – Set Memory Load Factor [2] – IO Load Factor =0b – Ignore IO Load Factor =1b – Set IO Load Factor [3:7] – Reserved should be 0000b	This command allows BMC or Remote Console to set PTAS-CUPS configuration. The command settings are persistent and overwrite previous Nonvolatile settings using this command or set at manufacturing using FITC. Sample count - determines the number of CUPS readings to sample over when determining the utilization average for each domain. Can range from 5 to 20 and defaults to 10.

Net Function = OEM (2Eh) LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
		Byte 6 – Set Load Factors [0] – Toggle Switch =0b – No change =1b – Toggle between dynamic and static [1:7] – Reserved should be 0000000b Byte 7:8 – Static Core Load Factor Byte 9:10 – Static Memory Load Factor Byte 11:12 – Static IO Load Factor Byte 13 – Sample count	
		Response Byte 1 – Completion Code Generic IPMI as defined in Section 2.15 plus the following command specific codes: =80h – Internal Error =81h – Hardware resources not available =82h – Not Initialized Byte 2:4 – Intel manufacturers ID – 000157h, LS byte first	
67h	Get CUPS Configuration	Request Byte 1:3 = Intel manufacturers ID – 000157h, LS byte first	This command allows BMC or Remote Console to get the previously PTAS-CUPS configuration.
		Response Byte 1 – Completion Code Generic IPMI as defined in Section 2.15 plus the following command specific codes: =80h – Internal Error =81h – Hardware resources not available =82h – Not Initialized Byte 2:4 – Intel manufacturers ID – 000157h, LS byte first Byte 5 – CUPS Feature Enabled Status [0] – Enabled Status =0b – CUPS feature is disabled =1b – CUPS feature is enabled [1:7] – Reserved should be 0000000b Byte 6 – Load Factor Configuration [0] – Load Factor Type =0b – Dynamic =1b – Static [1:7] – Reserved should be 0000000b Byte 7:8 – Static Core Load Factor Byte 9:10 – Static Memory Load Factor Byte 11:12 – Static IO Load Factor Byte 13 – Sample count	
68h	Set CUPS Policies	Request Byte 1:3 = Intel manufacturers ID – 000157h, LS byte first Byte 4 – Reserved. Write 00h	

Net Function = OEM (2Eh) LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
		Byte 5 – CUPS Policy ID [0:3] – Domain Identifier =01h – Core Domain =02h – Memory Domain =04h – IO Domain all other – Reserved [4:7] – Target identifier =00h – BMC =01h – Remote Console all other – Reserved Byte 6 – Policy Status [0] – Disable/Enable Policy =0b – Disable Policy =1b – Enable Policy [1:7] – Reserved should be 0000000b Byte 7 – Policy Type [0:6] – Reserved should be 0000000b [7] – Policy Storage Option =0b – persistent storage (Policy has been saved to the nonvolatile memory). =1b – volatile memory has been used for the policy storing. Byte 8 – Policy Excursion Actions [0] – Alerting To Target =0b – Disable alerting =1b – Enable sending of alert [1:7] – Reserved should be 0000000b Byte 9 – CUPS Threshold Byte 10:11 – Averaging Window (in Seconds)	This command allows BMC or Remote Console to set PTAS-CUPS configuration. The command settings are persistent and overwrite previous nonvolatile settings using this command or set at manufacturing using spsFITC. Policy excursion alert message sent follows standard IPMI event message formatting with the following exceptions: Sensor Number = 190 – CUPS Event Sensor Event Data 1 [0:1] – Threshold Number = 0 – Lower Threshold [2] – Reserved. [3] – CUPS Policy Event = 0 – Threshold exceeded Event Data 2 [0:3] – Domain ID (Identifies the CUPS domain) = 01h – Core = 02h – Memory = 04h – IO Event Data 3 – <Policy ID>.
		Response Byte 1 – Completion Code Generic IPMI as defined in Section 2.15 plus the following command specific codes: =80h – In Policy ID Invalid Byte 2:4 – Intel manufacturers ID – 000157h, LS byte first	
69h	Get CUPS Policies	Request Byte 1:3 = Intel manufacturers ID – 000157h, LS byte first Byte 4 – Reserved write 0 Byte 5 – CUPS Policy ID [0:3] – Domain Identifier =01h – Core Domain =02h – Memory Domain =04h – IO Domain all other – Reserved [4:7] – Target identifier =00h – BMC =01h – Remote Console all other – Reserved	This command allows BMC or Remote Console to read PTAS-CUPS configuration. For Completion Code 80h (Policy ID Invalid) response bytes 5 to 6 are defined as follows: Byte 5 – Next valid Policy ID. The field contains lowest valid Policy ID that is higher than Policy ID specified in the request. Byte 6 – 0h
		Response Byte 1 – Completion Code Generic IPMI as defined in Section 2.15 plus the following command specific codes:	

Net Function = OEM (2Eh) LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
		=80h – In Policy ID Invalid. In addition to bytes 2 to 4 extended error information is returned for this error code Byte 2:4 – Intel manufacturers ID – 000157h, LS byte first Byte 5 – Policy Status [0] – Policy Disabled/Enabled =0b – Policy Disabled =1b – Policy Enabled [1:7] – Reserved should be 0000000b Byte 6 – Policy Type [0:6] – Reserved should be 0000000b [7] – Policy Storage Option =0b – Persistent storage (policy has been saved to the nonvolatile memory) =1b – Volatile memory has been used for the policy storing Byte 7 – Policy Excursion Actions [0] – Alerting To Target =0b – Alerting disabled =1b – Sending of alert enabled [1:7] – Reserved should be 0000000b Byte 8 – CUPS Threshold Byte 9:10 – Averaging Window in seconds	

3.5.10.1 IPMI CUPS Sensors

Table 3-17 and Table 3-18 summarize the sensors exposed with the CUPS feature.

Table 3-17 IPMI CUPS Sensors Overview

Sensor Name	Sensor Type	Event Dir	Event Type	Alert Immediate (See Section 4.3)
Core CUPS Sensor	0Bh – Other Units	0 – assertion 1 – deassertion	01h – Threshold	No
IO CUPS Sensor	0Bh – Other Units	0 – assertion 1 – deassertion	01h – Threshold	No
Memory CUPS Sensor	0Bh – Other Units	0 – assertion 1 – deassertion	01h – Threshold	No
CUPS Event Sensor	DCh – OEM	0 – assertion 1 – deassertion	72h – OEM	Yes

Table 3-18 IPMI CUPS Sensors Definition

Sensor Number	Description	Firmware SKU Availability	Reading Availability	Defaults Configurable in spsFITC	Notes
190	Core CUPS Sensor	NM	A	Yes	Sensor that represents CPU Utilization on all the available CPUs.
191	IO CUPS Sensor	NM	A	Yes	Sensor that represents IO Utilization.

Sensor Number	Description	Firmware SKU Availability	Reading Availability	Defaults Configurable in spsFITC	Notes
192	Memory CUPS Sensor	NM	A	Yes	Sensor that represents Memory Utilization on all the available Memory channels.
193	CUPS Event Sensor	NM	E	No	OEM Event only sensor used to send events when PTAS-CUPS detects that a specified alert threshold for one of the policies is trigger. "Command illegal for specified sensor or record type (CDh)" error code is returned in response to: Get Sensor Reading, Set/Get Sensor Thresholds, Re-Arm Sensor Events, Set/Get Sensor Event Enable.

3.5.11 Chassis power

This sensor is used to get overall System Chassis power consumption. Sensor reading is in watts.

3.5.12 Event Messages Definition

Table 3-19 Event Messages Definition

Net Function = S/E (04h) LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
02h	Platform Event Message Intel® NM Exception Event	Request Byte 1 – EvMRev =04h – IPMI2.0 format. Byte 2 – Sensor Type =DCh – OEM. Byte 3 – Sensor Number =24 – Intel® Node Manager Event Sensor. Byte 4 – Event Dir | Event Type [0:6] – Event Type =72h – OEM. [7] – Event Dir =0 – Assertion Event. =1 – Deassertion Event. Byte 5 – Event Data 1 [0:2] – Reserved.	Event will be sent each time when maintained policy power limit is exceeded over Correction Time Limit. First occurrence of not acknowledged event will be retransmitted no faster than every 300 milliseconds. Event Policy Correction Time Exceeded is send with requester LUN set to 0 regardless of LUN whit which policy was created.

Net Function = S/E (04h) LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
		[3] – Intel® Node Manager Policy event =0 – Reserved. =1 – Policy Correction Time Exceeded – policy did not meet the contract for the defined policy. The policy will continue to limit the power or shutdown the platform based on the defined policy action. [4:5] = 10b – OEM code in byte 3. [6:7] = 10b – OEM code in byte 2. Byte 6 – Event Data 2 [0:3] – Domain ID (Identifies the domain that this Intel® Node Manager policy applies to) =00h – Entire platform. =01h – CPU subsystem. =02h – Memory subsystem. =03h – HW Protection. =04h – High Power I/O subsystem. Others – Reserved. [4:7] – Reserved. Byte 7 – Event Data 3 =<Policy ID>	
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in Section 2.15). Others – Error.	
16h	Alert Immediate Message Intel® NM Health Event	Request Bytes 1:3 – As set in the "Set Node Manager Alert Destination" IPMI command. Byte 4 – Generator ID [7:1] – 7-bit IPMB address of the Firmware. [0] = 0b – Generator ID is IPMB address. Byte 5 – EvMRev =04h – IPMI2.0 format. Byte 6 – Sensor Type =DCh – OEM. Byte 7 – Sensor Number =25 – Intel® Node Manager Health sensor. Byte 8 – Event Dir | Event Type [0:6] – Event Type =73h – OEM. [7] – Event Dir =0 – Assertion Event. =1 – Reserved Byte 9 – Event Data 1 [0:3] – Health Event Type =02h – Sensor Intel® Node Manager. [4:5] = 10b – OEM code in byte 3. [6:7] = 10b – OEM code in byte 2. Byte 10 – Event Data 2 [0:3] – Domain ID.	This message provides a run-time error indication about Intel® Node Manager's health. Note – Misconfigured policy error can happen when the max/min power consumption set using Set Power Draw Range exceeds the values in any of the configured policy. Real-time clock synchronization failure alert is sent when Intel® NM is enabled and capable of limiting power, but within 10 minutes the firmware cannot obtain valid calendar time from system RTC.

Net Function = S/E (04h) LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
		=00h – Entire platform =01h – CPU subsystem =02h – Memory subsystem =03h – HW Protection =04h – High Power I/O subsystem Others – Reserved. [4:7] – Error Type 0 – 6 – Reserved. =7 – Extended Telemetry Device Reading Failure. =8 – Outlet Temperature Reading Failure =9 – Volumetric Airflow Reading Failure. =10 – Policy Misconfiguration. =11 – Power Sensor Reading Failure. =12 – Inlet Temperature Reading Failure. =13 – Host Communication Error. =14 – Real-time clock synchronization failure. This is sent 10 minutes after Intel® NM reads invalid time from system RTC. =15 – Platform shutdown initiated by Intel® NM policy due to execution of action defined by Policy Exception Action see "Set NM Policy" command Byte 7 bit [1]. Byte 11 – Event Data 3 For Error Type = 7 <User Defined Device Index>. For Error Type = 8 <Outlet Sensor Address>. For Error Type = 9 <Volumetric Sensor Address>. For Error Type = 10 or 15 – <Policy ID>. For Error Type = 11 SMBus slave address of faulty device or 0 indicating reading failure of the whole power domain. For Error Type = 12 <Inlet Sensor Address>.	Host Communication Error is only detected when in Intel® NM there is a policy actively limiting power or when policy-based power limiting is disabled and Intel® NM enforces limit configured using Set Total Power Budget for power domain 0 or 1. Power Sensor Reading Failure may indicate single sensor failure, and sensor address is provided in Event Data 3 byte, or it may indicate lack of reading from all sensors. In this case Event Data 3 byte is set to zero. First occurrence of not acknowledged event will be retransmitted no faster than every 300 milliseconds. All events are sent with LUN set to 0 especially Event Misconfiguration is send with requester LUN set to 0 regardless of LUN whit which policy was created.
		Response Byte 1 – Completion Code = 00h – Success (Remaining standard Completion Codes are shown in Section 2.15). Others – Error.	
16h	Alert Immediate Message Intel® NM Operational Capabilities Change	Request Bytes 1:3 – As set in the "Set Node Manager Alert Destination" IPMI command. Byte 4 – Generator ID [0] = 0b – Generator ID is IPMB address. [7:1] – 7-bit IPMB address of the Firmware. Byte 5 – EvMRev =04h – IPMI2.0 format. Byte 6 – Sensor Type =DCh – OEM. Byte 7 – Sensor Number =26 – Intel® NM Operational Capabilities Sensor Byte 8 – Event Dir | Event Type [6:0] – Event Type = 74h – OEM. [7] – Event Dir =0 – Assertion Event. =1 – Deassertion Event.	This message provides a run-time error indication about Intel® NM's operational capabilities. This applies to all domains. Assertion and deassertion of these events are supported. Refer to Section 3.5.5 for detailed description of the conditions triggering changes of the capabilities bitmask. Current value of not acknowledged capabilities sensor will be retransmitted no faster every 300 milliseconds.

Net Function = S/E (04h) LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
		Byte 9 – Event Data 1 [2:0] – Current state of Operational Capabilities. The same value is also returned by the Get Sensor Reading command invoked for Intel® NM Operational Capabilities Sensor. [0] – Policy interface capability =0 – Not Available. =1 – Available. [1] – Monitoring capability =0 – Not Available. =1 – Available. [2] – Power limiting capability =0 – Not Available. =1 – Available. [7:3] – Reserved. Byte 10 – Event Data 2 =FFh – Not present Byte 11 – Event Data 3 =FFh – Not present	Event is only send with requester LUN set to 0.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in Section 2.15). Others – Error.	
16h	Alert Immediate Message Intel® NM Alert Threshold Exceeded	Request Bytes 1:3 – As set in the "Set Node Manager Alert Destination" IPMI command. Byte 4 – Generator ID [0] = 0b – Generator ID is IPMB address. [7:1] – 7-bit IPMB address of the Firmware. Byte 5 – EvMRev =04h – IPMI2.0 format. Byte 6 – Sensor Type =DCh – OEM. Byte 7 – Sensor Number =27 – Intel® Node Manager Event Sensor. Byte 8 – Event Dir | Event Type [6:0] – Event Type =72h – OEM. [7] – Event Dir =0 – Assertion Event. =1 – Deassertion Event. Byte 9 – Event Data 1 [1:0] – Threshold Number =0 – 2 – Threshold index. [2] – Reserved. [3] – Intel® Node Manager Policy Event =0 – Threshold exceeded. =1 – Policy Correction Time Exceeded – policy did not meet the contract for the defined policy. The policy will continue to limit the power or shutdown the platform based on the defined policy action. [5:4] = 10b – OEM code in byte 3. [7:6] = 10b – OEM code in byte 2.	Policy Correction Time Exceeded Event will be sent each time when maintained policy power limit is exceeded over Correction Time Limit. First occurrence of not acknowledged event will be retransmitted no faster than every 300 milliseconds. First occurrence of Threshold exceeded event assertion/ deassertion will be retransmitted no faster than every 300 milliseconds. Events Policy Correction Time Exceeded and Threshold exceeded are sent with requester LUN set to 0 regardless of LUN whit which policy was created.

Net Function = S/E (04h) LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
		Byte 10 – Event Data 2 [3:0] – Domain ID (Identifies the domain that this Intel® Node Manager policy applies to) =00h – Entire platform. =01h – CPU subsystem. =02h – Memory subsystem. =03h – HW Protection. =04h – High Power I/O subsystem. Others – Reserved. [7:4] – Reserved. Byte 11 – Event Data 3 – <Policy ID>.	
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in Section 2.15). Others – Error.	
16h	Alert Immediate Message CUPS Event Sensor	Request Bytes 1:3 – As set in the “Set Node Manager Alert Destination” IPMI command. Byte 4 – Generator ID [0] = 0b – Generator ID is IPMB address. [7:1] – 7-bit IPMB address of the Firmware. Byte 5 – EvMRev =04h – IPMI2.0 format. Byte 6 – Sensor Type =DCh – OEM. Byte 7 – Sensor Number =193 – CUPS Event Sensor. Byte 8 – Event Dir | Event Type [6:0] – Event Type =72h – OEM. [7] – Event Dir =0 – Assertion Event. =1 – Deassertion Event. Byte 9 – Event Data 1 [1:0] – Threshold Number =0 – 2 – Threshold index. [2] – Reserved. [3] – CUPS Policy Event =0 – Threshold triggered. [5:4] = 10b – OEM code in byte 3. [7:6] = 10b – OEM code in byte 2. Byte 10 – Event Data 2 [3:0] – Domain ID (Identifies the domain that this Intel® Node Manager policy applies to) =01h – Core. =02h – Memory. =04h – IO. Others – Reserved. [7:2] – Reserved. Byte 11 – Event Data 3 – <Policy ID>.	
		Response	

Net Function = S/E (04h) LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
		Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in Section 2.15). Others – Error.	
02h	Platform Event Message Intel® NM PSU Status Event	Request Byte 1 – EvMRev =04h – IPMI2.0 format. Byte 2 – Sensor Type =08h – Power Supply. Byte 3 – Sensor Number =102 – 109 – PSU Status Sensor. Byte 4 – Event Dir | Event Type [0:6] – Event Type =6Fh – Sensor Specific. [7] – Event Dir =0 – Assertion Event. =1 – Deassertion Event. Byte 5 – Event Data 1 [3:0] – Offset from Event/Reading Code for discrete event state. (Assertion and Deassertion refer to Byte 4, bit [7] - Event Dir). The detailed meaning of the bits is described in Section 3.5.6. =0h – Presence detected =1h – Power supply failure =2h – Predictive failure =3h – Power Supply input lost (AC/DC) =4h – Power Supply lost or out-of-range =5h – Power Supply input out-of-range, but present =6h – Configuration Error (generated if SmaRT & CLST needs to be disabled due to not compatible PSU attached) = 7h--Fh – Reserved. [5:4] = 00b – Unspecified byte 3. [7:6] = 00b – Unspecified byte 2. Byte 6 – Event Data 2 [3:0] = Fh – unspecified [7:4] = Fh - unspecified Byte 7 – Event Data 3 =FFh - not present	Event will be sent each time when monitored PSU Status changes. "Re-arm Sensor Events" IPMI command on bit [5] will reset CLST events (over-current and over-temperature) in the PSU. This allows in the manual CLST mode to reset the platform throttling by the BMC when the corrective action on the platform was performed. Note – In Intel® NM SKU this event is logged to SEL. For detailed description of the Severity Event Codes see Section 3.5.6.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in Section 2.15). Others – Error.	
02h	Platform Event Message Intel® NM SmaRT&CLST Event	Request Byte 1 – EvMRev =04h – IPMI2.0 format. Byte 2 – Sensor Type =DCh (OEM) Byte 3 – Sensor Number =178 - NM SmaRT & CLST Sensor	Event will be sent each time when SmaRT & CLST status changes for corresponding PSU.

Net Function = S/E (04h) LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
		Byte 4 – Event Dir | Event Type [0:6] – Event Type =03h - Digital Discrete. [7] – Event Dir =0 – Assertion Event. =1 – Reserved. Byte 5 – Event Data 1 [3:0] – Offset from Event/Reading Code (SmaRT & CLST throttling status) =0h – State Deasserted (throttling released) =1h – State Asserted (throttling enforced) [5:4] = 10b – OEM code in byte 3. [7:6] = 01b – Previous state and/or severity in byte 2. Byte 6 – Event Data 2 [3:0] – Fh (Optional offset from Event/Reading Type Code for previous discrete event unspecified.) [7:4] – Optional offset from 'Severity' Event/Reading Code. =00h - transition to OK =01h - transition to noncritical from OK =02h - transition to critical from less severe =03h - transition to unrecoverable from less severe =04h - transition to noncritical from more severe =05h - transition to critical from unrecoverable =06h - transition to unrecoverable =07h - monitor =08h - informational Byte 7 – Event Data 3 Corresponding Power Supply Status sensor number or 0 if the source of SmaRT & CLST assertion is external (for example BMC).	When sensor gets asserted, the corresponding PSU Status Sensor indicates the fault source. Note – In Intel® NM SKU this event is logged to SEL. For detailed description of the optional Severity Event Codes see Section 3.5.9.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in Section 2.15). Others – Error.	

3.6 Error Conditions

There may be situations when the Intel® NM is not available to respond to IPMI interface. This may happen when the management controller is not powered by stand-by power and the system is powered down or when there is a firmware update in progress. In addition, when using bridged commands, the BMC may not respond to the IPMI commands when it is unavailable for reasons such as flash update.

The external management SW needs to be aware of the fact there may be situations like these resulting in scenarios when responses to bridged command may not arrive or alerts may not be generated. The external management software needs to be designed appropriately to account for these.

§

4 BMC IPMI Interface

This section contains IPMI commands and sensor devices which shall be provided by BMC in order to enable Intel® NM firmware.

According to [Addr] specification by default SPS FW expects BMC at address 10h in 7-bit format (20h in 8-bit). On SPS FW side BMC address may be configured to different value with spsFITc tool.

To support initialization of Intel® ME owned sensors based on associated SDRs, OEM BMC must be able to use both the slave address and BMC channel number fields of the type1 and type2 SDRs for the purpose of directing the IPMI sensor commands to Intel® ME.

For proper Intel® ME initialization, BMC should send "Set Event Receiver" IPMI command. If BMC does not send "Set Event Receiver" command within this time after Host reset or startup, Intel® NM will activate "Time after Host Reset" policy.

4.1 IPMI Device "Global" Commands

Table 4-1 IPMI Device "Global" Command

Net Function = Chassis (00h) LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
02h	Chassis Control Command	Request Byte 1 [7:4] – Reserved. Write as 0000b. [3:0] – Control Command =0 – Power Down. =5 – Soft Shutdown (via ACPI) (optional).	This is standard IPMI 2.0 command. Note – Intel® NM Firmware will use Soft Shutdown (optional) and Power Down.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in Section 2.15).	

4.2 Sensor Device Commands

Table 4-2 Sensor Device Command

Net Function = S/E (04h) LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
02h	Platform Event Message	For command description see [IPMI]	This is general format of Platform Event Message. Detailed description of all messages generated by Intel® NM firmware can be found in Sections 2.9.5 and 3.5.7

4.3 Alert Immediate Command

BMC should support the Alert Immediate Command to properly forward the alerts from the Intel® NM to the remote management software.

Table 4-3 Alert Immediate Command

Net Function = S/E (04h) LUN = 00b			
Code	**Command**	**Request, Response Data**	**Description**
16h	Alert Immediate	For command description see [IPMI]	This is standard IPMI 2.0 command. Note – Alert Immediate command used by Intel® ME and supported by BMC must be compliant with IPMI Errata E358. It is expected that BMC will send to the remote console at least the contents of the bytes 5 to 11 of the IPMI message. If Event Data 1 and/or Event Data 2 are not defined in the event definition Intel® ME physically sends these bytes in the message with FFh value.

4.4 OEM Commands Implemented by BMC

Depending on the firmware factory settings SHOULD be supplied by the BMC for the associated Intel® ME services to work correctly. If the referenced services are not activated, the BMC does not need to provide the sensors and/or OEM commands.

4.4.1 Power Consumption Readings

If the firmware is configured to use BMC for power readings the sensors return (depends on configuration):

- Total Platform Power consumption
- Supplementary Power consumption (in addition to other power source readings, e.g. PSU)
- Single power reading, subtotal of per-rail readings from one PSU as well as total of single run of readings across all attached PSU’s cannot exceed 32767 Watt.

Such a power reading will be treated as a reading failure. That rule applies to any power reading source.

Depending on the configuration Firmware may use one of the following sources for platform power consumption readings:

1. PMBUS compliant PSU or voltage regulators attached directly to PCH SMLINK1 (recommended) or PCH SMLINK0. In that case there is no need to implement any support in the BMC.
2. BMC sensor read by management engine Firmware using OEM command implemented by the BMC. In that case BMC should implement OEM command to return the sensor value on the query from management engine Firmware. BMC should average the power over 1 second to allow management engine Firmware to read the power two to ten times per second[2]. This type of power reading allow for non-PMBUS compliant PSU or voltage regulator support. Additionally, the OEM command code may be configured using Factory presets:

Table 4-4 Power Consumption Readin Command

OEM command	Description	Encoding
E2h	OEM Get Reading with type "Platform Power Consumption"	The value of the reading is encoded on 16-bit encoding 2s-complement signed integer. Values below 0 are ignored and treated as a power reading failure.

Note: Command code for the "OEM Get Reading" command is configurable via the spsFITC tool.

4.4.2 OEM Management Engine Power State Change

Optionally, instead of the event Intel® Server Platform Services Firmware may send an IPMI command with information as defined previously. Using Factory presets OEM may choose to use an event or OEM command.

Table 4-5 OEM Management Engine Power State Change Command

OEM command	Description	Encoding
E3h	OEM ME Power State Change	The values returned by the command: 00h – Transition to Running – Intel® ME is started. 02h – Transition to Power Off – Intel® ME is to be powered down.

Note: Command code for the `OEM Power State Change' command is configurable via the spsFITC tool.

4.4.3 OEM Intel® NM PTU Notification

Intel® Server Platform Services Firmware shall send an IPMI command with information with the current inflection point being characterization.

[2] The frequency of power readings of 10, 5 or 2 per second should be defined by the OEM in the factory presets. For the fastest correction time of Intel® Node Manager Policy that is below one second the frequency of 10 readings per second should be supported by BMC.

Table 4-6 OEM NM PTU Notification Command

OEM command	Description	Encoding
E9h	OEM Platform Power Characterization Notification	The values returned by the command: Time period in milliseconds.

Note: Command code for the OEM Platform Power Characterization Notification command is configurable via the spsFITC tool.

Table 4-7 presents possible indications from BMC asking Intel® ME FW to delay its next action.

Table 4-7 NM PTU Delay Actions

Notification Type	Notification Information	Point	Stage	Delay Action
{ NM Phase Characterization Progression, BMC Phase Characterization Progression }	All applicable domains	All applicable points	Initialize	Delay enforced before execution Power Virus.
	All applicable domains	All applicable points	Monitor	Delay enforced before collecting the Power consumption readings.
	All applicable domains	All applicable points	Done	Ignored.
{ NM Phase Status, BMC Phase Status }	Phase Completed successfully	00h	00h	Delay before executing the next phase or next stage.
	Phase Interrupted	00h	00h	Ignored. (Intel® ME exited Characterization).

4.4.4 OEM Command Definition

Table 4-8 OEM Commands

Net Function = SDK General Application (30h) LUN = 00b			
Default Code	**Command**	**Request, Response Data**	**Description**
E2h	OEM Get Reading	Request Byte 1:N – Array of request, each of them consisting of the following bytes;	This command is optional and may be implemented by the BMC.

Net Function = SDK General Application (30h) LUN = 00b			
Default Code	**Command**	**Request, Response Data**	**Description**
		1st byte – Domain ID/Reading Type [0:3] – Domain ID. Currently only domain 0 will be queried. Other domains are queried internally by ME using. [4:7] – Reading Type =00h – Platform Power Consumption on the DC side. If the Power Source Efficiency is not supplied, NM treats this reading as a reading on the primary side (assuming Power Source Efficiency to be 100%). For platform power consumption the Domain ID will be set to 0. =01h – Inlet Air Temperature. For Inlet Air Temperature the Domain ID will be set to 0. =02h – Reserved. =03h – Zone Fan Speed. For this reading type the Domain ID will be set to 0. =04h – Outlet Air Temperature. For Outlet Air Temperature the Domain ID will be set to 0. =05h – Supplementary Power consumption on DC side. This is power consumed by the platform provided by a power source which is not accessible to Intel® NM for reading. The reported value is included in the total power consumption of the platform. For this reading type the Domain ID will be set to 0. =06h – Chassis power consumption on the primary side. For this reading type the Domain ID will be set to 0. =07h – Power source efficiency. This value is used to convert the readings on the primary side to the DC side and vice versa. =08h – Volumetric Airflow. For this reading type the Domain ID will be set to 0. Other values are reserved 2nd byte – Optional Parameter: For Reading Type = 03h it contains the index of fan zone [0-5]. For other reading types this value is ignored. 3rd byte – Reserved. Write as 00h.	The command can be used to send an array of request of different types. Each request consists of 3 bytes. The number of requests that can be packed in one command is restricted by capabilities of the underlying physical interface. When the array of request is sent, an array of responses is returned. Each response element consists of 4 bytes and corresponds to the request element under the same array index. Each of the response bytes has a completion code at the first byte. The completion code in a given response applies only to the request related to this response, thus an error completion code in one of the responses invalidates only the request related to this response. Intel® NM supports the following platform power reading source configurations: 1. BMC does not provide any power readings. 2. BMC reports only platform power consumption. Primary side=DC, efficiency = 100%. 3. BMC reports platform and chassis power consumption for single chassis configuration. In this case the chassis power represents the setup power consumption on the primary side. Intel® NM limiting correction time depends on the configured total platform power scanning rate. 4. BMC reports platform, chassis and efficiency. This configuration is reserved for blade systems.
		Response If an array of request was sent an array of responses is returned. Each response corresponds to a request with the same index. Byte 1:N – Array of responses, each of them consisting of the following bytes**; 1st Byte – Completion Code for this request =00h – Success =C9h - Parameter out of range for Reading Type = 03h if the provided fan zone is outside the supported range or does not exist on the given platform. =D6h – BMC does not support requested reading type. (Remaining standard Completion Codes are shown in Section 2.15).	

Net Function = SDK General Application (30h) LUN = 00b			
Default Code	**Command**	**Request, Response Data**	**Description**
		2nd byte – Reading Type [0:3] – Domain ID copied from request (depending on the Byte 2 identifies the processor which should be queried for power rail). [4:7] – Reading Type =00h – Platform Power Consumption in [Watts]. Values below 0 are ignored and treated as a power reading failure. =01h – Inlet Air Temperature in degrees centigrade. Values below -128 degrees centigrade and above +127 degrees centigrade will be ignored and treated as a temperature reading failure. =02h – Reserved. =03h – Zone Fan Speed in RPM. Values below 0 will be ignored and treated as reading failure. =04h – Outlet Air Temperature in degrees centigrade. Values below -128 degrees centigrade and above +127 degrees centigrade will be ignored and treated as a temperature reading failure. =05h – Supplementary Power consumption in Watts. =06h – Chassis power consumption in Watts. =07h – Power source efficiency in percent. =08h – Volumetric Airflow in Cubic Feet per Minute (CFM) Other values are reserved. 3rd and 4th byte – Reading value 16-bit encoding 2s-complement signed integer.	Power source efficiency calculation may be performed using energy, power or current (whichever unit provides better accuracy for given power source). For example for PMBus 1.2 compliant PSUs this value shall be calculated as EOUT/EIN*100. Intel® Node Manager accepts the following ranges for values returned by BMC; - power consumption [0:32767] Watt(*) - efficiency [0:100] % - temperature [-128:127] degrees Centigrade - airflow [0:1000] CFM - fan speed [0:32767] RPM * Platform Power Consumption must be greater than 0, otherwise Intel® NM triggers Missing Power Reading procedure. ** In case an array of responses is returned, each response shall consist of exactly 4 bytes. In case an unsuccessful completion code is returned for the response the remaining bytes of this response will be ignored.
E3h	OEM ME Power State Change	Request Byte 1 – Power State =00h – Transition to Running – Intel® ME is started. =02h – Transition to Power Off – Intel® ME is to be powered down.	This command is optional and may be implemented by the BMC.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in Section 2.15).	
E9h	OEM Platform Power Characterization Notification	Request	

Net Function = SDK General Application (30h) LUN = 00b			
Default Code	**Command**	**Request, Response Data**	**Description**
		Byte 1 – Power characterization Notification Type [7:2] – Notification Type =00h – Intel® NM Phase Characterization Progression =01h – BMC Phase Characterization Progression =02h – Intel® NM Phase Status =03h – BMC Phase Status All others – Reserved [1:0] – Notification Information For Notification Type = {00h, 01h} Notification Domain ID (The domain being characterized) =00h – Platform =01h – CPU subsystem =02h – Memory subsystem =03h – Reserved For Notification Type = {02h, 03h} =00h – Phase Completed successfully. =01h – Phase Interrupted. All others Reserved Byte 2 – Power Draw characterization Point [7:2] – Reserved [1:0] – Point (The Calibration point being characterized) For Notification Type = {00h, 01h} =00h – Max =01h – Min =02h – Efficient =03h – Reserved For Notification Type = {02h, 03h}, set to 00h. Byte 3 – Intel® ME Power Characterization Stage [7:2] – Reserved [1:0] – Stage (Identifies the internal ME FW state) For Notification Type = {00h, 01h} =00h – Reserved =01h – Initialize (Initializing the Characterization) =02h – Monitor (Monitoring Power) =03h – Done (Characterization Complete) For Notification Type = {02h, 03h} , set to 00h.	The max allowed setting for this parameter is 5 seconds. Intel® ME FW will cap the delay at 5 seconds if BMC responds with a value greater than 5 seconds. Table 4-7 shows possible indications from BMC asking Intel® ME FW to delay its next action.
		Response Byte 1 – Completion Code related to overall IPMI request (Remaining standard Completion Codes are shown in Section 2.15). Bytes 2:5 – Delay Time in milliseconds	

Note: Command codes for PECI Proxy commands are configurable via the spsFITC tool.

4.4.5 Summary of Options

Table 4-9 Intel Diagnostics Agent Summary of Options

Summary of Implementation Options		
Functionality	**Type**	**Description**
4.4.1 Power Consumption Readings	Factory presets	Power readings via “OEM Get Reading” command are only needed if no PMBUS PSU is directly connected to the SMLINK. Default in the Factory Presets is to use PMBUS PSU.
4.4.2 Intel® ME Power State Change	Factory presets	Intel® ME Firmware is able to send notification about the power state change using standard IPMI sensor or via OEM command. Default is to use OEM command.

4.5 BMC requirements for Intel® NM Discovery

The following discovery mechanism should be implemented by the BMC in order to allow external management software to properly configure the communication channel between Intel® NM and the external management software.

For command routing purposes, the external SW needs to know which microcontroller implements the Intel® NM functionality. Additionally, the external SW needs to know the IPMI sensor numbers associated with each Intel® NM sensor of interest. This information is provided via an Intel® NM OEM SDR[3].

The first step in the Intel® NM's discovery process is for the software to search the SDR repository for this OEM. If the Device Slave Address found in this SDR matches that of the BMC (20h), then all of the Intel® NM-related IPMI commands are sent directly to the BMC. Otherwise, standard IPMI bridging is used to send these commands to the satellite Intel® NM controller[4]. The SW application uses the sensor information in this SDR to comprehend the mapping of the sensor numbers to the Intel® NM sensors of interest. Additional sensor information can be retrieved by searching for associated type1, type2, or type3 SDRs for the specific sensors.

OEM SDR records are of type C0h. They contain a manufacturer ID and OEM data in the record body. Intel OEM SDR records also have a sub-type field in them as the first byte of the OEM data indicates the type of record following.

Table 4-10 Intel® NM Discovery

Byte (beginning after SDR record header)		Name	Description
0:2		OEM ID	Intel Manufacturer ID = 000157h
3		Record Subtype	Intel® NM Discovery = 0Dh
Intel® NM Record	4	Version number of this record subtype	=01h – for the version specified in this document.
	5	Intel® NM Device Slave Address	[7:1] – 7-bit I2C Slave Address of Intel® NM controller on channel. [0] – Reserved.
	6	Channel Number / Sensor Owner LUN	[7:4] – Channel number for the channel that the Intel® NM management controller is on. Use 0h if the primary BMC is the Intel® NM controller. [3:2] – Reserved. Write as 00b. [1:0] – Sensor owner LUN used for accessing all Intel® NM sensors enumerated in this record.
	7	Intel® NM Health Event sensor	=25
	8	Intel® NM Exception Event sensor	=24
	9	Intel® NM Operational Capabilities sensor	=26

[3] BMC should not expose the Intel® NM discovery SDR to the external console if Intel® NM SKU is disabled in the firmware.

[4] The privilege level to access to the Intel® ME SMLINK channel should be restricted to allow only the Admin level.

Byte (beginning after SDR record header)		Name	Description
	10	Intel® NM Threshold Exceeded Event Sensor	=27

4.6 Alerts

Events mentioned in Section 0 that are not marked as 'Alert Immediate' are sent as IPMI alerts to external software using PEF/PET. The BMC will perform the filtering based on filters set up by the external software.

In order to avoid excessive logging into the SEL due to Intel® NM "threshold exceeded" and "Intel® NM Health" events, a mechanism is provided to send PET alerts without use the PEF mechanism. These use the 'Alert Immediate' mechanism. This requires that the external SW application provide the Intel® NM with the alert destination and alert string information needed to properly form and send the alert. The external SW must first properly configure the alert destination and string in the BMC LAN configuration using standard IPMI commands, then provide the associated selectors to the BMC using the "Set Intel® Node Manager Alert Destination" OEM command. Section 3.1 contains the description of this OEM command.

Setting alert destination using "Set Intel® Node Manager Alert Destination" will cause all events marked "Immediate Alert" in the table of events to be routed to that destination as PET alerts. It is not possible to have some types of events sent to one destination and others to another.

No provision will be accommodated at this time for an in-band agent to receive alerts. The identification of an in-band alerting method if required will be defined at a later time. When Intel® NM needs to generate multiple events at the same time, Intel® NM will not generate a new Platform Event and Alert Immediate command until BMC does not send positive acknowledgment for the previous one (i.e. the Completion Code must be zero).

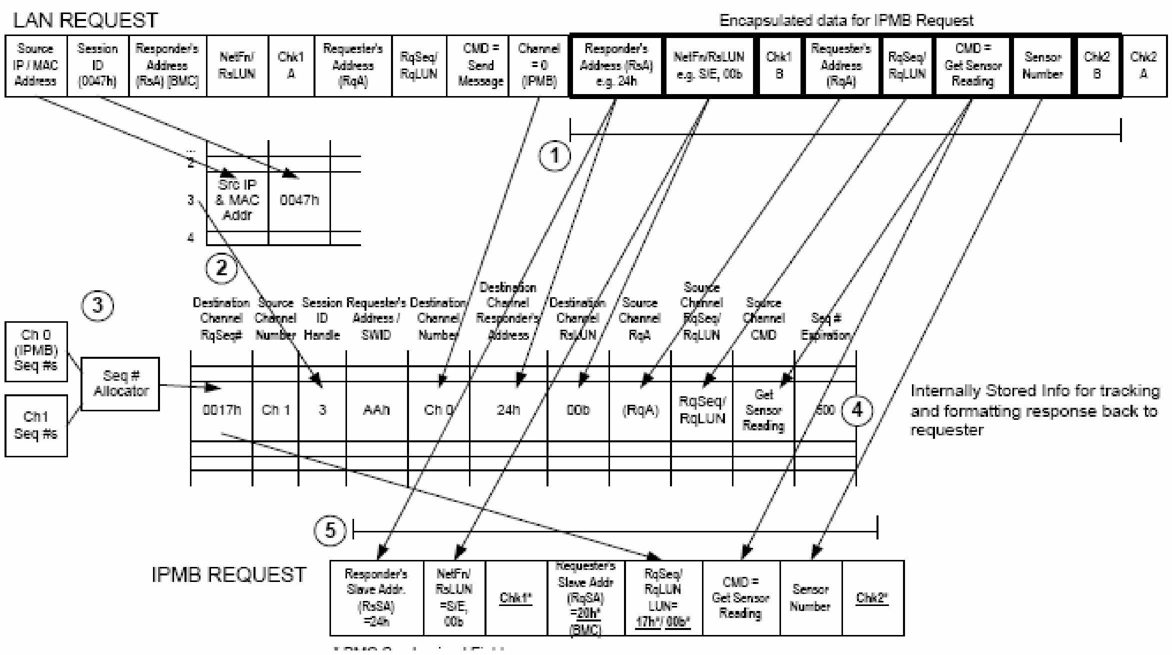
4.7 Command Passing via BMC

If Intel® NM is implemented on a controller other than BMC, then external SW will send 'bridged' IPMI commands to BMC. Encapsulated bridged IPMI commands must follow the format for the channel to which it is being bridged.

This will be in the form of IPMI packet encapsulated in another packet. BMC will need to examine the payload of the packet addressed to it, construct the proper IPMB packet, forward it to Intel® NM, and return the response from Intel® NM to the sender. Refer to IPMI 2.0 specification for details.

Because of the potentially performance-limiting and system shut-down effects of some of the commands that can be bridged, the BMC shall restrict the IPMI command bridging to the Administrator privilege level.

Figure 4-1 Example IPMI Command Bridging from LAN


LAN REQUEST
Encapsulated data for IPMB Request
Source IP / MAC Address
Session ID (0047h)
Responder's Address (RsA) (BMC)
NetFn/ RsLUN
Chk1 A
Requester's Address (RqA)
RqSeq/ RqLUN
CMD = Send Message
Channel = 0 (IPMB)
Responder's Address (RsA) e.g. 24h
NetFn/RsLUN e.g. S/E, 00b
Chk1 B
Requester's Address (RqA)
RqSeq/ RqLUN
CMD = Get Sensor Reading
Sensor Number
Chk2 B
Chk2 A
1
...
2
3
4
Src IP & MAC Addr
0047h
2
3
Ch 0 (IPMB) Seq #s
Ch1 Seq #s
Seq # Allocator
Destination Channel RqSeq#
Source Channel Number
Session ID Handle
Requester's Address / SWID
Destination Channel Number
Destination Channel Responder's Address
Destination Channel RsLUN
Source Channel RqA
Source Channel RqSeq/ RqLUN
Source Channel CMD
Seq # Expiration
0017h
Ch 1
3
AAh
Ch 0
24h
00b
(RqA)
RqSeq/ RqLUN
Get Sensor Reading
500
4
Internally Stored Info for tracking and formatting response back to requester
5
IPMB REQUEST
Responder's Slave Addr. (RsSA) =24h
NetFn/ RsLUN =S/E, 00b
Chk1*
Requester's Slave Addr (RqSA) =20h* (BMC)
RqSeq/ RqLUN LUN= 17h*/ 00b*
CMD = Get Sensor Reading
Sensor Number
Chk2*

For the purpose of constructing a bridged IPMI command, external software would need to know the following parameters to construct the proper IPMI packet:

- 'Responder's address'
- Destination channel #
- Network Function/LUN
- Channel protocol

It is the responsibility of the BMC to let external software know of the 'Responder's address'. This will be done as part of Intel® NM discovery procedure. The responder in this case will be the actual management controller implementing Intel® NM (Intel® ME, BMC).

In addition, the actual medium to communicate the bridged packet on (IPMB, MCTP in future) is also needed. This is done as part of its initial discovery by query of the BMC for channel protocol using the IPMI command "Get Channel Info". For example, if Intel® Node Manager is implemented in the management engine (satellite controller), BMC needs to forward the packet over IPMB. This information is used by external software to construct the encapsulated packet appropriate to the medium.

The length of the supported IPMI frames is extended to 80 bytes so up to 68 bytes of payload can be passed in one IPMI request.

4.7.1 BMC IPMI Commands Bridging Security Impact Considerations

Some IPMI OEM commands defined in this document enable unrestricted access to platform devices (e.g. SMBus devices) which may potentially have serious security consequences up to including permanent platform level Denial of Service. To avoid such security threats it is highly recommended to disable BMC bridging of those IPMI OEM commands at all privilege levels (including Administrator level). In particular it is recommended for BMC to completely disable bridging of the following PECI/PMBus/MIC Proxy services IPMI commands:

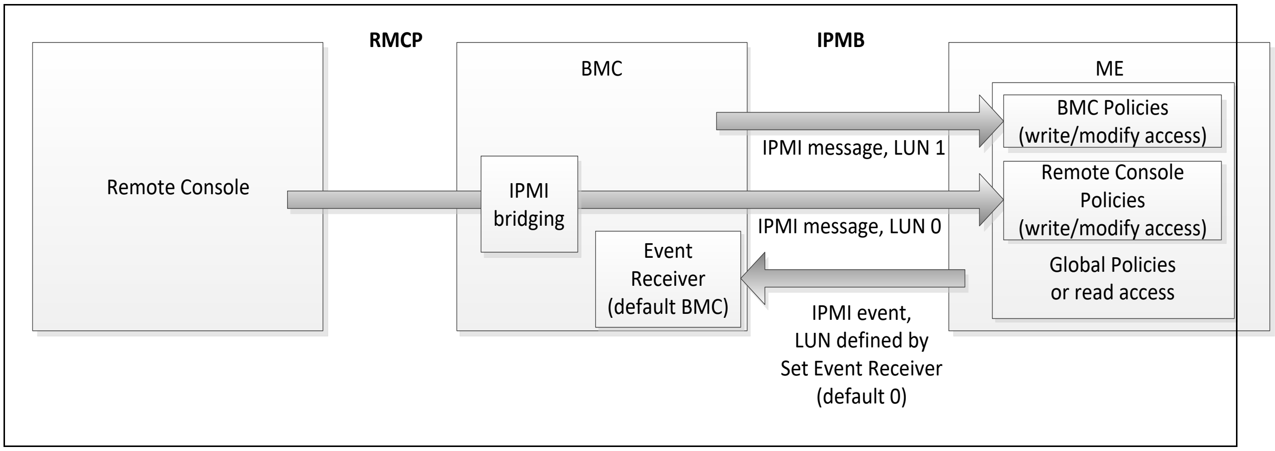
Table 4-11 IPMI OEM Commands to be disabled for BMC bridging

NET Function	Command Code	Command Name
2Eh + 000157h	40h	Send Raw PECI
2Eh + 000157h	41h	Aggregated Send Raw PECI
2Eh + 000157h	43h	CPU Package Configuration Write
2Eh + 000157h	45h	CPU PCI Configuration Write
2Eh + 000157h	47h	Read Memory SMBus
2Eh + 000157h	48h	Write Memory SMBus
2Eh + 000157h	D9h	Send Raw PMBus
2Eh + 000157h	ECh	Aggregated Send Raw PMBus
2Eh + 000157h	E7h	Unlock ME Region

4.8 Policies Reservation Mechanism

There are some use cases when Intel® NM policies could be removed unintentionally. For example, after BMC sets its policy, a user can disable it by accident via the IPMITool. To avoid such situations a Responder's LUN support will be added to these IPMI commands that modify policies so that only the request with appropriate LUN will be allowed to perform modifications.

Figure 4-2 Intel® NM Policy Reservation Based on LUN


RMCP
IPMB
Remote Console
BMC
IPMI bridging
Event Receiver (default BMC)
ME
BMC Policies (write/modify access)
Remote Console Policies (write/modify access)
Global Policies or read access
IPMI message, LUN 1
IPMI message, LUN 0
IPMI event, LUN defined by Set Event Receiver (default 0)

The commands that enforce LUN checking are:

- Set/Get Node Manager Policy
- Enable/Disable Node Manager Policy Control*
- Set Node Manager Policy Alert Thresholds
- Set Node Manager Policy Suspend Periods
- Reset Intel® Node Manager Statistics - the LUN number is used only to distinguish from each other the per-policy parameters. The global and per-domain parameters apply to the entire platform regardless of the LUN number.
- All the other Intel® Node Manager commands work the same way regardless of the provided LUN.

The LUN number is handled as follows:

- When the policy is set via the Set **Node Manager Policy** command, the Responder's LUN number is stored together with the policy information.
- When a request comes that modifies the policy the LUN of the incoming message is compared with the stored one. When they match the command is processed normally. If not, the command will not get executed and an error code D4h – Insufficient Privilege Level is returned.
- When a request doesn't modify the policy it is executed as if LUN from request match with the one stored with the policy. Only exception is that the **Get Node Manager Policy** command would return "managed by the external policy" bit set in the Domain ID field when it is called with a LUN number that does not much the one stored with the policy. This bit would be cleared in case when LUNs match.

The LUN numbers are used only for verification of the policy owner. Their existence does not have any influence on the number of available policies or triggers neither it influences the way they work. For example when there is a policy set for a given LUN, the system will not allow creating of a policy with the same Policy Id for another LUN.

The LUN numbers used for this functionality are the ones that come with the NetFn of the command that sets the policy, since this LUN is common for IPMB, and IPMI over RMCP+.

4.9 IPMB Reset Scenarios

BMC and management engine Firmware share the same SMBUS link. The management engine Firmware will perform SMBUS hang recovery only when the hang happens during a transaction initiated by the management engine. The recovery method is by “bit banging” 9 clock pulses. The Intel® ME FW does not reset SMLINK on startup.

§

5 IPMI OEM Command Summary

The following table summarizes the OEM commands used by Intel® NM firmware for Intel® ME-BMC communication. All the commands use LUN 00b with Intel IANA Number 000157h.

Table 5-1 IPMI OEM Commands

NET Function	Command Code	Command Name	Implemented by	SKU	NM 5.0 Platform
2Eh + 000157h	40h-4Fh	PECI Proxy and PECI related commands in Intel® NM 5.0	Intel® ME BMC	SiEn Intel® NM Base	Mehlow
2Eh + 000157h	50h-53h	Telemetry Hub commands	Intel® ME BMC	SiEn Intel® NM Base	Mehlow
2Eh + 000157h	60h	Intel® NM Power Characterization launch Request	Intel® ME	Intel® NM Base	Mehlow
2Eh + 000157h	61h	Get Intel® Node Manager Power Characterization Range	Intel® ME	Intel® NM Base	Mehlow
2Eh + 000157h	64h	Get CUPS Capabilities	Intel® ME	SiEn Intel® NM SDF	
2Eh + 000157h	65h	Get CUPS Data	Intel® ME	SiEn Intel® NM SDF	
2Eh + 000157h	66h	Set CUPS Configuration	Intel® ME	SiEn Intel® NM SDF	
2Eh + 000157h	67h	Get CUPS Configuration	Intel® ME	SiEn Intel® NM SDF	
2Eh + 000157h	68h	Set CUPS Policy	Intel® ME	SiEn Intel® NM SDF	
2Eh + 000157h	69h	Get CUPS Policy	Intel® ME	SiEn Intel® NM SDF	
2Eh + 000157h	70h	Get Intel® Platform Trust Technology Version	Intel® ME	SiEn Intel® NM SDF	Mehlow
2Eh + 000157h	71h	Get Intel® Platform Trust Technology Capabilities	Intel® ME	SiEn Intel® NM SDF	Mehlow
2Eh + 000157h	72h	Get Intel® Platform Trust Technology Health	Intel® ME	SiEn Intel® NM SDF	Mehlow
2Eh + 000157h	80h	Get Intel® Boot Guard Version	Intel® ME	SiEn Intel® NM SDF	Mehlow

NET Function	Command Code	Command Name	Implemented by	SKU	NM 5.0 Platform
2Eh + 000157h	81h	Get Intel® Boot Guard Capabilities	Intel® ME	SiEn Intel® NM SDF	Mehlow
2Eh + 000157h	82h	Get Intel® Boot Guard Health	Intel® ME	SiEn Intel® NM SDF	Mehlow
2Eh + 000157h	0C-0Fh	IE Configuration Commands	Intel® ME	SiEn	
2Eh + 000157h	90h	ME Storage Services Read	Intel® ME	SiEn	
2Eh + 000157h	91h	ME Storage Services Write	Intel® ME	SiEn	
2Eh + 000157h	92-9Fh	Reserved for SGX	Intel® ME	SiEn	
2Eh + 000157h	A7-A8h	Commands backward compatible with Online Firmware Update	Intel® ME	SiEn Intel® NM Base	Mehlow
2Eh + 000157h	C0-CBh	Intel® Node Manager Commands	Intel® ME	Intel® NM Base* Intel® NM SDF	Mehlow
2Eh + 000157h	CEh	Set Node Manager Alert Destination	Intel® ME	SiEn Intel® NM Base	Mehlow
2Eh + 000157h	CFh	Get Node Manager Alert Destination	Intel® ME	SiEn Intel® NM Base	Mehlow
2Eh + 000157h	D0h	Set Total Power Budget	Intel® ME	Intel® NM Base* Intel® NM SDF	Mehlow
2Eh + 000157h	D1h	Get Total Power Budget	Intel® ME	Intel® NM Base* Intel® NM SDF	Mehlow
2Eh + 000157h	D2h	Set Max Allowed CPU P-state/T-state	Intel® ME	Intel® NM Base	Mehlow
2Eh + 000157h	D3h	Get Max Allowed CPU P-state/T-state	Intel® ME	Intel® NM Base	Mehlow
2Eh + 000157h	D4h	Get Number Of P-states/T-states	Intel® ME	Intel® NM Base	Mehlow
2Eh + 000157h	D7h	Set PSU Configuration	Intel® ME	SiEn Intel® NM Base	Mehlow
2Eh + 000157h	D8h	Get PSU Configuration	Intel® ME	SiEn Intel® NM Base	Mehlow
2Eh + 000157h	D9h	Send Raw PMBus	Intel® ME	SiEn Intel® NM Base	Mehlow
2Eh + 000157h	DDh	Set Intel® ME FW Capabilities	Intel® ME	SiEn Intel® NM Base	Mehlow
2Eh + 000157h	DEh	Get Intel® ME FW Capabilities	Intel® ME	SiEn Intel® NM Base	Mehlow
2Eh + 000157h	DFh	Force Intel® ME Recovery	Intel® ME	SiEn Intel® NM Base	Mehlow

NET Function	Command Code	Command Name	Implemented by	SKU	NM 5.0 Platform
2Eh + 000157h	E0h	Get Intel® ME Factory Presets Signature	Intel® ME	SiEn Intel® NM Base	Mehlow
2Eh + 000157h	E1h	Set Intel® ME FW USB Port Override	Intel® ME	SiEn	Mehlow
2Eh + 000157h	E6h	Get Exception Data	Intel® ME	SiEn	Mehlow
2Eh + 000157h	E7h	Unlock ME Region	Intel® ME	SiEn	Mehlow
2Eh + 000157h	E8h	Restart MCTP Discovery	Intel® ME	SiEn	Mehlow
2Eh + 000157h	EAh	Get Host CPU Data	Intel® ME	Intel® NM Base	Mehlow
2Eh + 000157h	ECh	Aggregated Send Raw PMBus	Intel® ME	SiEn Intel® NM Base	Mehlow
2Eh + 000157h	F0h	Set HW Protection Coefficient	Intel® ME	Intel® NM Base	Mehlow
2Eh + 000157h	F1h	Get HW Protection Coefficient	Intel® ME	Intel® NM Base	Mehlow
2Eh + 000157h	F2h	Get Limiting Policy ID	Intel® ME	Intel® NM Base* Intel® NM SDF	Mehlow
2Eh + 000157h	F3h	Set PMBUS Device Configuration	Intel® ME	SiEn Intel® NM Base	Mehlow
2Eh + 000157h	F4h	Get PMBUS Device Configuration	Intel® ME	SiEn Intel® NM Base	Mehlow
2Eh + 000157h	F5h	Get PMBUS Readings	Intel® ME	SiEn Intel® NM Base	Mehlow
2Eh + 000157h	F6h	Aggregated Get PMBus Readings	Intel® ME	SiEn Intel® NM Base	Mehlow
2Eh + 000157h	F7h	Get Energy Counter Unit	Intel® ME	SiEn Intel® NM Base	Mehlow
2Eh + 000157h	F8h	Set Energy Counter Unit	Intel® ME	SiEn Intel® NM Base	Mehlow
2Eh + 000157h	F9h	Set Intel® NM Parameter	Intel® ME	SiEn Intel® NM Base* Intel® NM SDF	Mehlow
2Eh + 000157h	FAh	Get Intel® NM Parameter	Intel® ME	SiEn Intel® NM Base* Intel® NM SDF	Mehlow
2Eh + 000157h	FBh	Get PMBus Device Energy	Intel® ME	Intel® NM Base	Mehlow
2Ch (DCGRP) + DCh	01-05h	DCMI Power management Commands	Intel® ME	Intel® NM Base	Mehlow
30h (SDK General Application)	E9h	OEM Platform Power Characterization Notification	Intel® ME	Intel® NM Base	Mehlow

NET Function	Command Code	Command Name	Implemented by	SKU	NM 5.0 Platform
30h (SDK General Application)	E2-E3h	OEM Commands	Intel® ME	SiEn Intel® NM Base	Mehlow
30h (SDK General Application)	EBh	Manufacturing Commands	Intel® ME	SiEn Intel® NM Base Intel® NM SDF	Mehlow

(*) For Intel® NM Base this command has reduced functionality (see command description for details).

For a list of commands supported by recovery boot loader, see Section 2.11.

§

A Intel® NM 5.0 Platform Changes

This appendix describes changes between Intel® Intelligent Power Node Manager External Interface Specification using IPMI for Intel® NM 4.0 and Intel® NM 5.0.

A.1 IPMI Commands Not Supported on Intel® NM 5.0 Platform

The following IPMI commands are not supported by Intel® Node Manager 5.0 firmware in comparison with Intel® NM 4.0.

Table A-1 Not Supported IPMI OEM Commands

NET Function	Command Code	Command Name	Implemented by	SKU	NM 5.0 Platform
30h (SDK General Application)	A0-A6h	Online Firmware Update Commands	Intel® ME	SiEn Intel® NM Base	Not Supported

B IPMI Sensors Implemented by Intel® NM Firmware

B.1 Event Generation Control

The following table summarizes how event generation can be enabled/disabled for each sensor implemented by Intel® NM Firmware.

Table B-1 Event Generation Control

Sensor Types	IPMI command	Event Generation Control Methods
PECI Proxy Sensors	Platform Event	Event generation can be enabled/disabled using the IPMI commands: Set Event Receiver. Set Event Enable – both per sensor and per threshold control support. Default event enable flags can be set using Flash Image Tool for each threshold.
Intel® ME Power State Sensor	Platform Event or OEM command (settable using FIT)	When the notification is sent using Platform Event message, event generation can be enabled/disabled using the IPMI command: Set Event Receiver. If the event is sent using OEM message, the notification is sent always to address 20h regardless of the Event Receiver settings.
Intel® NM Exception Event Sensor	Platform Event	Event generation can be enabled/disabled using the IPMI command: Set Event Receiver
Intel® NM Operational Capabilities Sensor	Alert Immediate	Event generation can be enabled/disabled using the IPMI commands: Set Intel® Node Manager Alert Destination – unregistering alert destination disables generation of events sent using Alert Immediate IPMI command. Set Event Enable – Scanning Enabled bit disables/enables the whole sensor. Changing per sensor Event Enable flag is supported.
Intel® NM Health Event Sensor Intel® NM Threshold Exceeded Event Sensor	Alert Immediate	Event generation can be enabled/disabled using the IPMI command: Set Intel® Node Manager Alert Destination – unregistering alert destination disables generation of events sent using Alert Immediate IPMI command.
Intel® NM FW Health Sensor	Platform Event	Not possible to enable/disable event generation from this sensor.
PSU Status Sensors	Platform Event	Event generation can be enabled/disabled using the IPMI commands: 1. Set Event Receiver. 2. Set Event Enable – both per sensor and per threshold control support.

Sensor Types	IPMI command	Event Generation Control Methods
		Default event enable flags can be set using Flash Image Tool for each threshold.

B.2 IPMI Platform Event Messages Generated by Intel® NM FW

Intel® ME Firmware will try to deliver the all the events by resending them if event reception is not acknowledged by the BMC. Reception of the following Completion Codes in event response from BMC will stop resending events from Intel® ME Firmware:

Table B-2 IPMI Completion Codes that stop event resending

Code	Definition
00h	Command Completed Normally.
C1h	Invalid Command.
C2h	Command invalid for given LUN.
C4h	Out of space.
CDh	Command illegal for specified sensor or record type.
FFh	Unspecified error.

Table B-3 Messages Generated by Intel® NM FW

Event	Sensor Type	Event Dir	Event Type	Event command
Memory Throttling Status	0Ch – Memory	0 – Assertion 1 – Deassertion	01h – Threshold	Platform Event
Intel® ME Power State	16h – Microcontroller	0 – Assertion	0Ah – Availability	Platform Event
Intel® NM Exception Event	DCh – OEM	0 – Assertion	72h – OEM	Platform Event
Intel® NM Threshold Exceeded Event Sensor	DCh – OEM	0 – Assertion 1 – Deassertion	72h – OEM	Alert Immediate
Intel® NM Health Event	DCh – OEM	0 – Assertion	73h – OEM	Alert Immediate
Intel® NM Operational Capabilities Change Event	DCh – OEM	0 – Assertion 1 – Deassertion	74h – OEM	Alert Immediate
Intel® ME Firmware Health Event	DCh – OEM	0 – Assertion	75h – OEM	Platform Event
CPU Thermal Status	07h – CPU	0 – Assertion 1 – Deassertion	76h – OEM	Platform Event
CPU Thermal Control Circuit Activation	07h – CPU	0 – Assertion 1 – Deassertion	01h – Threshold	Platform Event
Intel® NM PSU Status Event	08h – PSU	0 – Assertion 1 – Deassertion	6Fh – Sensor Specific	Platform Event

Event	Sensor Type	Event Dir	Event Type	Event command
Intel® NM SmaRT&CLST Event	DCh – OEM	0 – Assertion	03h - Digital Discrete	Platform Event

B.3 IPMI Sensors Provided by Intel® ME FW

Table B-4 IPMI Sensors Provided by Intel® ME FW

Sensor #	Description	Firmware SKU Availability	Notes	NM 5.0 Platform
8	PCH Thermal Threshold	SiEn/ NM		Mehlow
21	Typical Power Consumption in Sx	NM		Mehlow
22	ME Power State Event	Recovery/SiEn/ NM	OEM Event only sensor “Command illegal for specified sensor or record type (CDh)” error code is returned in response to the following commands: Get Sensor Reading, Set/Get Sensor Thresholds, Re-Arm Sensor Events, Set/Get Sensor Event Enable	Mehlow
23	ME FW Health Event	Recovery/SiEn/ NM	OEM Event only sensor “Command illegal for specified sensor or record type (CDh)” error code is returned in response to the following commands: Get Sensor Reading, Set/Get Sensor Thresholds, Re-Arm Sensor Events, Set/Get Sensor Event Enable	Mehlow
24	NM Exception Event	NM		Mehlow
25	NM Health Event	NM		Mehlow
26	NM Operational Capabilities	NM		Mehlow
27	NM Threshold Exceeded Event	NM		Mehlow
28	CPU 0 Thermal Status	SiEn/NM		Mehlow
32	CPU 0 Thermal Control Circuit Activation	SiEn/NM		Mehlow
36	CPU 0 T-Control	SiEn/NM		Mehlow
40	HSC 0 Status Low	NM		Mehlow
41	HSC 0 Input Power	NM		Mehlow
42	HSC 0 Input Voltage	NM		Mehlow
44	HSC 1 Status Low	NM		Mehlow
45	HSC 1 Input Power	NM		Mehlow
46	HSC 1 Input Voltage	NM		Mehlow
48	CPU 0 T-JMAX	SiEn/NM		Mehlow

Sensor #	Description	Firmware SKU Availability	Notes	NM 5.0 Platform
52	CPU 0 Memory Throttling	SiEn/NM		Mehlow
58	Power Off Event	NM		Mehlow
59	Power Threshold	NM		Mehlow
66	HSC 0 Status Byte High	NM		Mehlow
68	HSC 1 Status Byte High	NM		Mehlow
78	PSU 0 AC Power Input	SiEn/NM		Mehlow
79	PSU 1 AC Power Input	SiEn/NM		Mehlow
80	PSU 2 AC Power Input	SiEn/NM		Mehlow
81	PSU 3 AC Power Input	SiEn/NM		Mehlow
82	PSU 4 AC Power Input	SiEn/NM		Mehlow
83	PSU 5 AC Power Input	SiEn/NM		Mehlow
84	PSU 6 AC Power Input	SiEn/NM		Mehlow
85	PSU 7 AC Power Input	SiEn/NM		Mehlow
86	PSU 0 Temperature	SiEn/NM		Mehlow
87	PSU 1 Temperature	SiEn/NM		Mehlow
88	PSU 2 Temperature	SiEn/NM		Mehlow
89	PSU 3 Temperature	SiEn/NM		Mehlow
90	PSU 4 Temperature	SiEn/NM		Mehlow
91	PSU 5 Temperature	SiEn/NM		Mehlow
92	PSU 6 Temperature	SiEn/NM		Mehlow
93	PSU 7 Temperature	SiEn/NM		Mehlow
94	HSC 0 Status MFR Specific	NM		Mehlow
102	PSU 0 Status	SiEn/NM		Mehlow
103	PSU 1 Status	SiEn/NM		Mehlow
104	PSU 2 Status	SiEn/NM		Mehlow
105	PSU 3 Status	SiEn/NM		Mehlow
106	PSU 4 Status	SiEn/NM		Mehlow
107	PSU 5 Status	SiEn/NM		Mehlow
108	PSU 6 Status	SiEn/NM		Mehlow
109	PSU 7 Status	SiEn/NM		Mehlow
111	HSC 1 Status MFR Specific	NM		Mehlow
159	HSC 0 Status Input	NM		Mehlow
161	HSC 1 Status Input	NM		Mehlow
164	PSU 0 DC Power Output	SiEn/NM		Mehlow
165	PSU 1 DC Power Output	SiEn/NM		Mehlow
166	PSU 2 DC Power Output	SiEn/NM		Mehlow

Sensor #	Description	Firmware SKU Availability	Notes	NM 5.0 Platform
167	PSU 3 DC Power Output	SiEn/NM		Mehlow
168	PSU 4 DC Power Output	SiEn/NM		Mehlow
169	PSU 5 DC Power Output	SiEn/NM		Mehlow
170	PSU 6 DC Power Output	SiEn/NM		Mehlow
171	PSU 7 DC Power Output	SiEn/NM		Mehlow
173	Total Chassis power	SiEn/NM		
178	NM SmaRT&CLST	NM		Mehlow
190	Core CUPS Event	NM	Sensor that represents CPU Utilization on all the available CPUs.	
191	IO CUPS mi	NM	Sensor that represents I/O Utilization.	
192	Memory CUPS	NM	Sensor that represents Memory Utilization on all the available Memory channels.	
193	CUPS Event	NM	OEM Event only sensor used to send events when PTAS-CUPS detects that a specified alert threshold for one of the policies is triggered. “Command illegal for specified sensor or record type (CDh)” error code is returned in response to the following commands: Get Sensor Reading, Set/Get Sensor Thresholds, Rearm Sensor Events, Set/Get Sensor Event Enable.	
197	Host Partition Reset Warning Event	SiEn/NM	This sensor sends an event to inform the BMC about an upcoming Platform Reset.	Mehlow
220	HSC 2 Status Low	NM		Mehlow
221	HSC 2 Status High	NM		Mehlow
222	HSC 2 Status MFR Specific	NM		Mehlow
223	HSC 2 Status Input	NM		Mehlow
224	HSC 2 Input Power	NM		Mehlow
225	HSC 2 Input Voltage	NM		Mehlow
226	HSC 3 Status Low	NM		Mehlow
227	HSC 3 Status High	NM		Mehlow
228	HSC 3 Status MFR Specific	NM		Mehlow
229	HSC 3 Status Input	NM		Mehlow
230	HSC 3 Input Power	NM		Mehlow
231	HSC 3 Input Voltage	NM		Mehlow
232	IMON Input Power	NM		Mehlow

§