Intel® Intelligent Power Node Manager 2.0

External Interface Specification using IPMI

August 2013

Order Number: 322999

Contents

Figures

Tables

1. Introduction

1.1 Purpose of this document

This document will help BMC vendors and external management software vendors support Intel® Intelligent Power Node Manager 2.0. It explains the IPMI commands that can be sent to the Intel® Management Engine (Intel® ME) in the Intel® 6 series chipset and Intel® C600 series chipsets. This document also describes IPMI sensors implemented to ensure the correct and reliable operation of the platform.

1.2 Scope

This document provides details on the IPMI commands used by Intel® Node Manager (Intel® NM) firmware components running on an Intel platform.

1.3 General conventions

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

1.4 System states and power management

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, but power is connected to the memory system.
S4	A system state where the host CPU and memory are not active
S5	A system state where all power to the host system is off, but the power cord is connected.
Sx	All S-states that are different from 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 in memory and resumed from the memory when the mouse or keyboard is clicked.
Shutdown	All power is off for the host machine, but the power cord is connected.

1.5 Reference documents

Ref	Document name	File/location
[Addr]	IPMB v1.0 Address Allocation, 1998.	*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*
[IPMB]	Intelligent Platform Management Bus Specification, version 1.0, 1999.	*http://www.intel.com/design/servers/ipmi/spec.htm*
[PET]	IPMI Platform Event Trap Format Specification, version 1.0, 1999.	*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*

§

2. Intel® Management Engine (Intel® ME) IPMI Interface

This section contains IPMI commands and sensor devices provided by Intel® ME. BMC shall use these commands and sensors to control Intel® Node Manager (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

Net function = Storage (0Ah)			
Code	**Command**	**Request, response data**	**Description**
48h	Get SEL Time	Request None	This is a standard IPMI 2.0 command. Intel® NM firmware responds to this command returning internal clock value.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in section 2.11) Bytes 2:5 - Present Timestamp value.	
43h	Get SEL Entry	Request Byte 1:2 = Reservation ID – Write as 00h 00h Byte 3:4 = SEL Record ID = 00h 00h – Get 1st Exception Log Entry from Intel® ME FW = 01h 00h – Get 2nd Exception Log Entry from Intel® ME FW … = 04h 00h – Get 5th Exception Log Entry from Intel® ME FW = 00h 01h – Get PSU Over Current statistics = 01h 01h – Get PSU Over Temperature statistics = 02h 01h – Get PSU Under Voltage statistics Byte 5 = Offset into record – Write as 00h Byte 6 = Bytes to read – Write as FFh	Allows for reading diagnostics information from Intel® ME firmware. The response from the command should be sent to Intel for analysis.
		Response Byte 1 – completion code =00h – Success (Remaining standard completion codes are shown in [IPMI]) =C9h –Parameter Out Of Range (in case of getting Exception Logs indicates that the log is empty) Byte 2 – Next SEL Record ID Byte 3:N – Record Data	

2.2 IPMI device "global" commands

Net function = App (06h)			
Code	**Command**	**Request, response data**	**Description**
02h	Cold Reset	Request None	This is a standard IPMI 2.0 command. Reboots Intel® ME without resetting host platform.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in section 2.11)	

Net function = App (06h)			
Code	**Command**	**Request, response data**	**Description**
01h	Get Device ID	Request None	This is a standard IPMI 2.0 command. **Note:** The patch number in the response is always set to 0.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in section 2.11) Byte 2 – Device ID =50h - Intel® Management Engine (Intel® ME) Byte 3 – Device Revision [7] = 1 - For DM and DNM firmware, the SKU device provides Device SDRs = 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 firmware Revision 1 [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 = 2 Byte 5 - Firmware Revision 2: Minor Firmware Revision. BCD encoded. Byte 6 - IPMI Version. Holds IPMI Command Specification Version. BCD encoded. 00h = reserved. Bits 7:4 hold the Least Significant digit of the revision, while Bits 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.	

Net function = App (06h)			
Code	**Command**	**Request, response data**	**Description**
		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 For DM and DNM, the SKU byte 7 is set to: [7] = 1 Chassis device Supported [6] = 0 Bridge not supported [5] = 0 IPMB event generator not supported [4] = 0 IPMB event receiver not supported [3] = 1 FRU inventory device supported [2] = 1 SEL device supported [1] = 1 SDR repository device supported [0] = 1 Sensor device supported 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 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® NM IPMI interface version =0 – Intel® NM not implemented/enabled =1 – Intel® NM Revision 1.5 =2 – Intel® NM Revision 2.0 Note: For Silicon Enabling, SKU byte 13 will be set to all zeros. Byte 14 –Firmware build number BCD encoded = A.B Byte 15 –The last digit of firmware build number and patch number BCD encoded = C.PATCH	

Net function = App (06h)			
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 *Note: Full version number is:* "Major Firmware Revision. Minor Firmware Revision.ABC.PATCH" where ABC is firmware build number.	
04h	Get Self-Test Results	Request None	This is a standard IPMI 2.0 command. If the platform supports configuration with both HSC and PSU devices, reporting of HSC failures have precedence over PSU errors.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in section 2.11.) =D5h – Returned if self-tests are 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 Monitoring Service continuously updates the error code in runtime in S0/S1 host power states. Additionally, the test runs 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 Monitoring Service continuously updates the error code in runtime in S0/S1 host power states. Additionally, the test runs in any host power state if Manufacturing Test On Command is issued. =FFh – reserved. 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] – Factory Presets checksum error.	

Net function = App (06h)			
Code	**Command**	**Request, response data**	**Description**
		[6] – SDR access error. [5] – FRU access error. [4] – SEL access error. [3] – PIA access error. [2] – SDR repository empty. [1] – Firmware boot error. [0] – Controller operational firmware corrupted. For byte 2 = 80h: PSU monitoring error bit field, where each bit corresponds to one of the PSUs 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" For byte 2 = 82h: HSC monitoring error bit field, where each bit corresponds to one of the HSCs 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.	

2.3 Sensor device commands

Net function = S/E (04h)			
Code	**Command**	**Request, response data**	**Description**
00h	Set Event Receiver	Request Byte 1 – Event Receiver Slave Address. =0FFh disables Event Message Generation, Otherwise: [7:1] – IPMB (I^2C) Slave Address [0] – always 0b when [7:1] hold I^2C 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 sent on any platform startup from G3 and on Global Platform Reset. (See definition in PCH EDS.)
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in section 2.11)	

Net function = S/E (04h)			
Code	**Command**	**Request, response data**	**Description**
01h	Get Event Receiver	Request None	
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in section 2.11) Byte 2 – Event Receiver Slave Address. 0FFh indicates Event Message Generation has been disabled. Otherwise: [7:1] – IPMB (I^2C) 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.	

Net function = S/E (04h)			
Code	**Command**	**Request, response data**	**Description**
26h	Set Sensor Thresholds	See [IPMI] for command description.	This is a standard IPMI 2.0 command.
27h	Get Sensor Thresholds	See [IPMI] for command description.	This is a standard IPMI 2.0 command.
28h	Set Sensor Event Enable	See [IPMI] for command description.	This is a standard IPMI 2.0 command.
29h	Get Sensor Event Enable	See [IPMI] for command description.	This is a standard IPMI 2.0 command.
2Ah	Rearm Sensor Events	See [IPMI] for command description.	This is a standard IPMI 2.0 command.
2Bh	Get Sensor Event Status	See [IPMI] for command description.	This is a standard IPMI 2.0 command.
2Dh	Get Sensor Reading	See [IPMI] for command description.	This is a 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 no reading - and bit [6] of byte 3 set to 1.

2.4 Intel® ME firmware debug event

After recovery from an Intel® ME system error Intel® ME firmware generates "Add SEL Entry" command with debug information about the error. The purpose of this message is solely in field debugging; it is not intended to replace health sensor events. We recommend the BMC support storing the SEL records passed in the "Add SEL Entry" command. By assumption, the debug event does not contain any sensitive information. Generation of this event may be disabled by setting Enable Debug SEL Entries parameter in Flash Image Tool. We recommend disabling debug event generation in the firmware loaded on the platforms shipped to the end customers.

Event is repeated every 5 sec until Intel® NM receives response with Success Completion Code.

2.4.1 Debug SEL entry definition

Debugging information in the message is intended for Intel Corporation engineering team usage only. Because the internal structure definition may be subject to changes, any support request should contain both message content and firmware revision. (This is not embedded in the message due to space constraints.)

Net function = Storage(0Ah)			
Code	**Command**	**Request, response data**	**Description**
44h	Add SEL Entry	Request Byte 1:2 – Record ID =1..N Byte 3 – Record Type =EEh – OEM nontimestamped. Byte 4:16 – Debugging information.	This type of record contains extended information about errors detected by Intel® ME, solely for field debugging purposes. You may use the flash image configuration tool to disable generation of this event.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in section 2.11)	

2.5 IPMI OEM device commands

Net function = 2Eh-2Fh			
Code	**Command**	**Request, response data**	**Description**
DCh	Set ME Power State	Request Byte 1:3 = Intel Manufacturer ID – 000157h, LS byte first. Byte 4 – New power state: =00 – Turn off Intel® ME power. Note: After turning off power, Intel® ME will wake up on one of the following events: Automatically when Host CPU goes to S0 Write any message to any SMT device on the I2C address defined in softstraps. In that case, only Intel® ME wakes up.	
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in section 2.11) *Note: Intel® ME may be turned on and off by the BMC in Sx Host CPU states. In S0/S1 Host CPU state command will return D5h error code.* Byte 2:4 = Intel Manufacturer ID – 000157h, LS byte first.	

Net function = 2Eh-2Fh			
Code	**Command**	**Request, response data**	**Description**
DDh	Set 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 & Thermal (Intel® NM) Assist Module** = 00b – reserved = 01b – disable = 10b – reserved = 11b – enable	This command is only supported on Brickland platforms. This command can be used to disable Intel® Node Manager on nonlegacy PCH. 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 2.11) Byte 2:4 = Intel Manufacturer ID – 000157h, LS byte first. Byte 5:8 – Intel® ME FW version Byte 5 – Major FW revision Byte 6 – Minor FW revision Byte 7 – Build version Byte 8 – Patch revision 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 & State Control Version = 01h – v1.0 = 02h – v2.0 Other – reserved. Byte 13 – Proxies [7:6] – **MIC Discovery** = 00b – not supported = 01b – supported, not enabled = 10b – reserved = 11b – supported and enabled [4:5] – **IPMB Proxy**	

Net function = 2Eh-2Fh			
Code	**Command**	**Request, response data**	**Description**
		= 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 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 & Thermal (Intel® NM) Assist Module** = 00b – not supported = 01b – supported, not enabled = 10b – reserved = 11b – supported and enabled	
DEh	Get 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 = 01h – IPMI 1.0 = 02h – IPMI 2.0 Other – reserved. Byte 9:10 – IPMI Message Size Supported (bytes). Value includes encapsulation. Byte 11 – Proxies [7:6] – **MIC Discovery** = 00b – not supported = 01b – supported, not enabled = 10b – reserved = 11b – supported and enabled. [4:5] – **IPMB Proxy** = 00b – not supported = 01b – supported, not enabled = 10b – reserved = 11b – supported and enabled	This command is only supported on Brickland platforms. 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.

Net function = 2Eh-2Fh			
Code	**Command**	**Request, response data**	**Description**
		[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	
		Response Byte 1 – Completion Code = 00h – Success (Remaining standard Completion Codes are shown in section 2.11) Byte 2:4 = Intel Manufacturer ID – 000157h, LS byte first. Byte 5:8 – Intel® ME FW version Byte 5 – Major FW revision Byte 6 – Minor FW revision Byte 7 – Build version Byte 8 – Patch revision 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 & State Control Version = 01h – v1.0 = 02h – v2.0 Other – reserved. Byte 13 – Proxies [7:6] – **MIC Discovery** = 00b – not supported = 01b – supported, not enabled = 10b – reserved = 11b – supported and enabled [4:5] – **IPMB 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 Byte 14 – Assist Modules [7:6] **Performance Assist Module** = 00b – not supported	

Net function = 2Eh-2Fh			
Code	**Command**	**Request, response data**	**Description**
		= 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 & Thermal (Intel® NM) Assist Module** = 00b – not supported = 01b – supported, not enabled = 10b – reserved = 11b – supported and enabled	
DFh	Force 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. Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in section 2.11) 81h – Unsupported Command parameter value in the Byte 4 of the request. Byte 2:4 = Intel Manufacturer ID – 000157h, LS byte first.	With parameter Command = 01h Intel® ME FW resets and prevents loading the regular operational FW code – Intel® ME FW stops in Recovery Boot Loader (note: in DM/DNM FW, Intel® ME FW becomes inaccessible using DCMI over HECI or LAN protocols; the only access is from host – for example, to perform FW update from BIOS). 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 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.
E0h	**Get ME Factory Presets Signature**	Request Byte 1:3 = Intel Manufacturer ID – 000157h, LS byte first.	The signature length is Intel® ME firmware implementation specific and

Net function = 2Eh-2Fh			
Code	**Command**	**Request, response data**	**Description**
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in section 2.11) 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.	fixed for given Intel® ME firmware release, but it may be different in subsequent releases. Bytes 5:N exist in response frame only if Completion Code (Byte 1) = 00h

2.6 IPMI device “global” sensors

Intel® ME FW always exposes two IPMI sensors:

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

Detailed description of these sensors is provided in the subsections.

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

Sensor #	Description	Firmware SKU availability	Reading availability	Defaults configurable in flash image tool	Notes
22	Intel® ME Power State[1]	A	E	Yes	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, Rearm Sensor

1 *Optionally, instead of the event, Intel® NM Firmware may send an IPMI command with information as defined above. Using factory presets, the OEM may use an event and/or OEM command. The user can decide to disable the sensor and not disable the OEM command generation.*

Sensor #	Description	Firmware SKU availability	Reading availability	Defaults configurable in flash image tool	Notes
					Events, Set/Get Sensor Event Enable
23	Intel® ME Firmware Health	A	E	No	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, Rearm Sensor Events, Set/Get Sensor Event Enable
8	PCH Thermal Sensor	A	H	Yes	Temperature threshold sensor

2.6.1 Intel® ME power state sensor

The sensor is used 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 to be powered down

Note: Optionally, instead of or in addition to the event, Intel® ME firmware may send an IPMI command with power state change notification as defined in section 4.4.3. Using factory presets, the OEM may use an event and/or OEM command.

2.6.2 Intel® ME firmware health sensor

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

2.6.3 PCH thermal sensor

This sensor provides PCH die temperature sensor value in Celsius degrees.

2.6.4 Event messages definition

Net function = S/E (04h)			
Code	**Command**	**Request, response data**	**Description**
02h	Platform Event Message 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.	Note: OEM can select using Flash Image Tool that the notification is sent using OEM command instead of Platform Event Message. See section 4.4.3 for definition of the OEM command.

Net function = S/E (04h)			
Code	**Command**	**Request, response data**	**Description**
		[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	
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in section 2.11.)	
02h	Platform Event Message ME Firmware Health Event	Request Byte 1 – EvMRev =04h (IPMI2.0 format) Byte 2 – Sensor Type =DCh (OEM) Byte 3 – Sensor Number =23 – Intel® ME Firmware Health Byte 4 – Event Dir | Event Type [7] – Event Dir =0 – Assertion Event. =1 – Deassertion 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. Byte 6 – Event Data 2 =0 – 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* =1 – Image execution failed. Recovery Image or backup operational image loaded because operational image is corrupted. This may be 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.* =2 – Flash erase error. Error during flash erasure procedure probably due to flash part corruption. ***Repair action:*** *The flash device must be replaced.* =3 – Flash state information. **Repair action:** Check extended info byte in Event Data 3 (byte 7) whether wearout protection is causing this event. If so, wait until wearout protection expires; otherwise probably the flash device must be replaced (if error is persistent). =4 – Internal error. Error during firmware execution – if Event Data 3 (byte 7) contains 0 then FW Watchdog Timeout; otherwise please contact Intel representative for support. ***Repair action:*** *Operational image shall be updated to other version or hardware board repair is needed (if error is*	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: You cannot use FITc to disable this sensor.

Net function = S/E (04h)			
Code	**Command**	**Request, response data**	**Description**
		persistent). =5 – BMC did not respond to cold reset request and Intel® ME rebooted the platform. **Repair action:** Verify the Intel® NM configuration. =6 – 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. =7 – Manufacturing error. Wrong manufacturing configuration detected by Intel® ME firmware. **Repair action:** The flash device must be replaced (if error is persistent). =8 – Persistent storage integrity error. Flash file system error detected. **Repair action:** If error is persistent, restore factory presets using "Force ME Recovery" IPMI command or by doing AC power cycle with Recovery jumper asserted. =9 – 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.) =10..255 – Reserved Byte 7 – Event Data 3 Extended error info. Use when reporting an error to tech support. The following values can be interpreted directly: Event Data 2 (byte 6) equal to 3 = 00 - recovery bootloader image or factory presets image corrupted = 01 - flash erase limit has been reached = 02 - flash write limit has been reached; writing to flash has been disabled. = 03 - writing to the flash has been enabled Event Data 2 (byte 6) equal to 4 = 00h - FW Watchdog Timeout = 1Dh - Loader manifest validation failure = 37h - Unknown power management event = 45h - None graceful PMC reset event detected i.e. after Dynamic Fusing = 8Eh - Flash wearout protection (EFFS wearout violation) Event Data 2 (byte 6) equal to 7 = 04 - Intel® ME FW configuration is inconsistent or out of range.	
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in section 2.11)	

2.7 IPMI OEM Intel® ME firmware update commands

Server Intel® NM firmware supports online firmware update only in dual operational image with the recovery boot-loader image flash configuration. In this situation, the same set of commands for image upgrade is available from any operational image. During upgrade, all enabled functionalities will work. A rollback scenario is supported.

In single operational image mode, the online firmware update functionality is not supported.

To finish the operational image upgrade after a successful image registering, the "Cold reset" command should be sent, in order to boot the new operational image.

Online update of Operational image upgrades only the code. Settings, such as any configuration information including DCMI and Intel® NM configured Intel® NM policies, factory presets and recovery firmware remain unchanged after the operational code upgrade.

Runtime data sent by BIOS and/or BMC is preserved between cold resets of Intel® ME.

All the firmware update commands below are unavailable for 5 seconds after platform or CPU reset. In such case, Intel® NM firmware returns D5h (Command Not Supported in Present Stats) Completion Code.

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.

Net function = 2Eh-2Fh			
Code	**Command**	**Request, response data**	**Description**
A0h	Online Update Prepare For Update	Request Byte 1:3 = Intel Manufacturer ID – 000157h, LS byte first	This is the first command that should be sent to initiate FW upgrade. This command restarts the FW upgrade to the initial state. Upon success command **Online Update Get Status** returns "update requested" status. Upon failure, "update init failed" is returned.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in section 2.11) =80h – Operation refused (too many requests) =81h – Flash error =82h – Operation in progress (flash erase) Byte 2:4 = Intel Manufacturer ID – 000157h, LS byte first	
A1h	Online Update Open Area	Request Byte 1:3 = Intel Manufacturer ID – 000157h, LS byte first Byte 4 – Area type =00h – Reserved =01h – Operational code =02h – PIA =03h – SDR Byte 5 – Area flags [0:7] – Reserved. Write as 00000000b.	Only update of operational code will be supported *Note:* **Online Update Open Area** invalidates the rollback image partition. Only a successful image upload allows switching to a rollback image. Upon success, the command **Online Update Get Status** returns "update failed" status in case of flash error or "update in progress" status in case of success. This command performs access to the flash so the response will be sent after
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in section 2.11) 80h – Operation refused. After this error code, the FW upgrade should be reinitialized by sending **Online Update Prepare For Update** command. 81h – Flash error. After this error code, the FW upgrade should be reinitialized by sending **Online Update**	

Net function = 2Eh-2Fh			
Code	**Command**	**Request, response data**	**Description**
		Prepare For Update command. Byte 2:4 = Intel Manufacturer ID – 000157h, LS byte first	completing the operation and may take longer than 250 ms.
A2h	Online Update Write Area	Request Byte 1:3 = Intel Manufacturer ID – 000157h, LS byte first Byte 4 – Sequence number – Must start with 0 value Byte 5:N – Area data, where <N> should not exceed 73. Maximum supported number of data bytes is 68	This command writes data into opened area. Upon success command **Online Update Get Status** returns "update failed" status in case of flash error or "update in progress" status in case of success. This command performs access to 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 2.11.) =80h – Operation refused. Wrong CRC or image length mismatch. After this error code, the FW upgrade should be reinitialized by sending **Online Update Prepare For Update** command. =81h – Flash error. After this error code, the FW upgrade should be reinitialized by sending **Online Update Prepare For Update** command. Byte 2:4 = Intel Manufacturer ID – 000157h, LS byte first	
A3h	Online Update Close Area	Request Byte 1:3 = Intel Manufacturer ID – 000157h, LS byte first Byte 4:7 – Area size in bytes Byte 8:9 – Area checksum. Byte 9 should be set to 0. Byte 8 is a CRC8 ATM HEC (based on $x^8 + x^2 + x + 1$ polynomial) calculated over the operational binary image directly from the distribution package i.e. NMOperational.bin file.	This command verifies the image checksum and size. Upon success command **Online Update Get Status** returns "update failed" status in case of checksum verification failure or "update requested" status otherwise.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in section 2.11) =80h – Operation refused. Wrong CRC or image length mismatch. After this error code, the FW upgrade should be reinitialized by sending **Online Update Prepare For Update** command. =81h – Flash error. After this error code, the FW upgrade should be reinitialized by sending **Online Update Prepare For Update** command. =82h – In Progress. Not used on Nehalem-EP platform. Byte 2:4 = Intel Manufacturer ID – 000157h, LS byte first.	
A4h	Online Update Register Update	Request Byte 1:3 = Intel Manufacturer ID – 000157h, LS byte first Byte 4 – update type =0 – Reserved =1 – Normal update. Use the new image for the next boot. Use this update type to verify and to switch to a newly uploaded image – after a successful **Online Update Close Area** command. Upon success, command **Online Update Get Status** returns "update failed" status in case of image verification error or "update success" status otherwise.	Instructs the controller that all areas to be updated have been sent and schedules an update to occur on the next Intel® ME FW reset. End-user is expected to perform system power cycle to reset the Intel® ME FW and host. Upon success, command

Net function = 2Eh-2Fh			
Code	**Command**	**Request, response data**	**Description**
		=2 – Reserved =3 – Manual rollback. Use this update type to cancel the switch to a new image and to use the current code – when executed right after **Online Update Prepare For Update** command. Upon success command **Online Update Get Status** returns "update failed" status in case of image verification error or "update rolled back" status otherwise. =4 – Abort update. Exits upgrade. Upon success command **Online Update Get Status** returns "update aborted" status. Byte 5 – dependent flags [0:7] – Reserved. Write as 00000000b.	**Online Update Get Status** returns status as described in Figure 2-1. This command performs access to 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 2.11) =80h – Operation refused. =81h – Flash error. After this error code, the FW upgrade should be reinitialized by sending **Online Update Prepare For Update** command. =CCh – Invalid field Byte 2:4 = Intel Manufacturer ID – 000157h, LS byte first	
A6h	Online Update Get Status	Request Byte 1:3 = Intel Manufacturer ID – 000157h, LS byte first	Returns the current status of the FW Update.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in section 2.11) Byte 2:4 = Intel Manufacturer ID – 000157h, LS byte first Byte 5 – Image status [0] – Reserved. Return as 0b. [1] – Staging image (new) =1 – Image valid [2] – Rollback image =1 – Image valid [3:4] – Running image area =0 – CODE (Recovery mode) =1 – COD1 =2 – COD2 [5:7] – Reserved. Return as 000b. Byte 6 – Update state =0 – Idle (no update in progress) =1 – Update requested =2 – Update in progress =3 – Update success =4 – Update failed =5 – Update rolled back =6 – Update aborted =7 – Update initialization failed =8-255 – Reserved Byte 7 – Update Attempt Status =0-255 – Reserved. Return as 0. Byte 8 – Rollback Attempt Status =0-255 – Reserved. Return as 0.	

Net function = 2Eh-2Fh			
Code	**Command**	**Request, response data**	**Description**
		Byte 9 – Update Type =0-255 – Reserved. Return as 0. Byte 10 – Dependent Flags [0:7] – Reserved. Return as 0. Byte 11:14 – Free Area Size in bytes.	
A7h	Online Update Get Capabilities	Request Byte 1:3 = Intel Manufacturer ID – 000157h, LS byte first	PIA and SDR updates not supported
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in section 2.11) 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.	

2.7.1 Online update flow

The sequence of events that occurs during a full update with rollback is as follows.

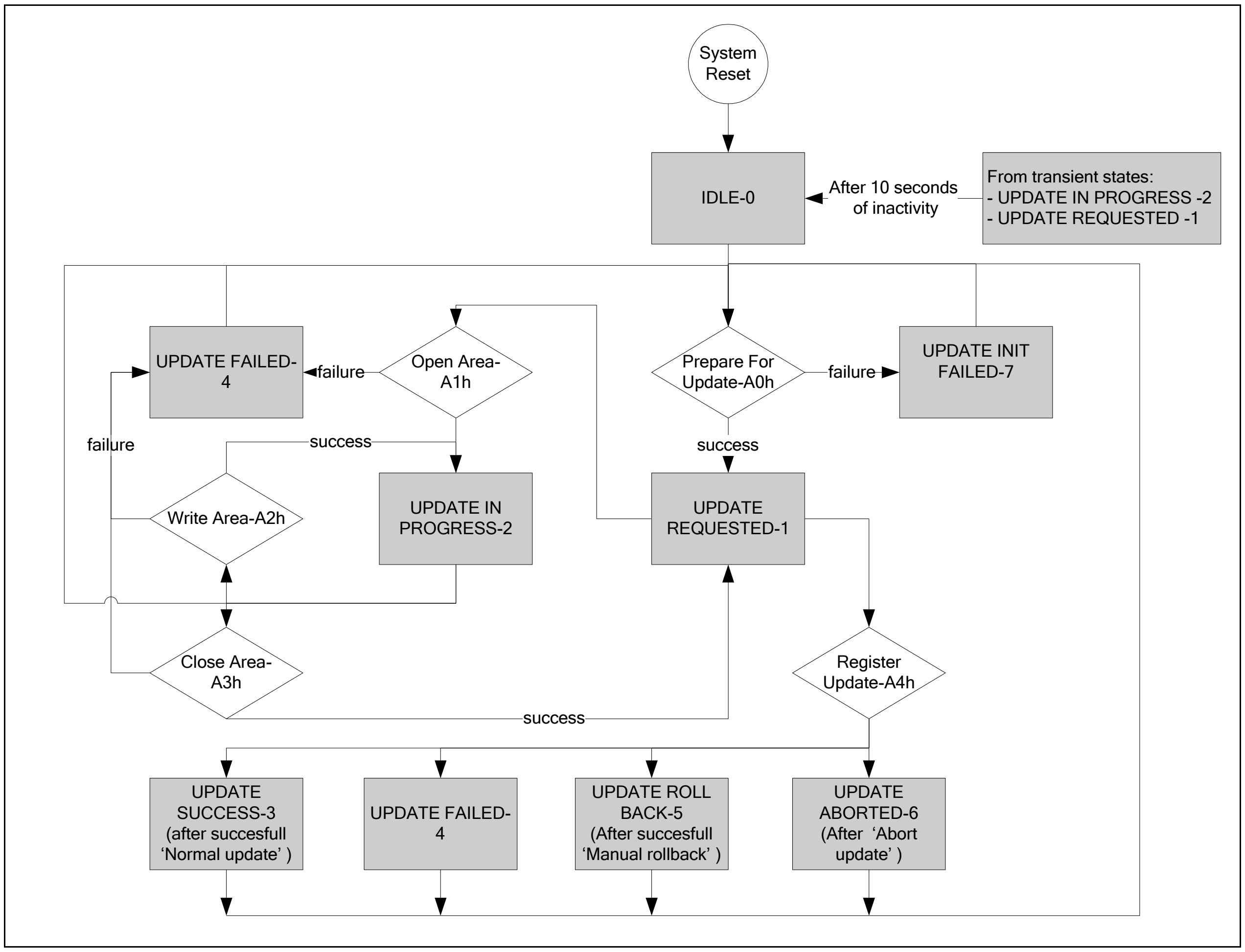
1. The online update application sends an *Online Update Prepare for Update* command, causing the Intel® ME to reinitialize the staging image and enter the update requested state. Update state returned by **Online Update Get Status** command should return value 1 – update requested or 7 – update init failed in a case of flash error.
2. The online update application sends an *Online Update Open Area* command, telling the Intel® ME to create an area in the staging image and prepare to receive download data of the specified type. Update state returned by **Online Update Get Status** should return value 2 – update in progress unless problem occurs with the area preparation. In that case, update will return value 4 – update failed.
3. The online update application sends multiple *Online Update Write Area* commands to fill the area and then sends an *Online Update Close Area* command to indicate the area is finished. The *Online Update Close Area* command contains length and checksum information that allows the Intel® ME to validate that all data was successfully received. Update state returned by **Online Update Get Status** should return value 2 – update in progress in a case of flash write success until *Online Update Close Area* command is sent. From that time, update state should return value 1 – update requested or 4 – update failed, depending on the CRC verification.

4. The online update application repeats the *Online Update Open Area*, *Online Update Write Area*, and *Online Update Close Area* sequence until all areas that are to be updated have been downloaded. Update state returned by **Online Update Get Status** should return value 2 – update in progress in the case of flash write success until *Online Update Close Area* command is sent. From that time, update state should return value 1 – update requested or 4 – update failed, depending on the checksum verification for details see **Online Update Close Area** command description.
5. The online update application sends an *Online Update Register Update* command to indicate the download is complete and is ready to be enacted. The Intel® ME verifies the validity of the staging area image and sets the status accordingly. When Intel® ME is running Intel® NM operational image during the update procedure, only basic image verification is done. The verification includes checking image CRC, but it does not include checking image signature (i.e. whether the image was correctly signed by Intel). The full image verification is performed when Intel® ME is loading the image. Update state returned by **Online Update Get Status** should return value 3 – update success (in a case of successful verification) or 4 – update failed (in a case of verification failure) unless *Online Update Register Update* command was sent without finalizing update with *Online Update Close Area* command. In that case, update state will return value 6 – update aborted. Value 5 – update rollback will be returned after successful manual rollback.
6. A system reset occurs (potentially much later).
7. The Intel® ME hard resets itself to allow the boot code to run.
8. The boot code verifies the status of the new downloaded operational image and transfers control to and executes the new image if verification succeeds.
9. If verification status of the update image is bad, the last-known-good operational code bank is automatically selected for execution, to prevent the server from being inoperable or degraded. Update state returned by **Online Update Get Status** should return value 0 – idle.

System power cycles are treated the same as system resets. If AC power is lost before the actual update copying process starts, the registered update is discarded. When AC power is reapplied, the Intel® ME comes up in the idle condition. If AC power is lost during the copy process or after it is started, the copy is resumed after AC power is restored.

There can be one and only one area of each type in a single update sequence. Partial updates are not permitted.

Figure 2-1 Intel® Management Engine online update flow



2.7.2 Backward-compatibility mode

The Intel® NM firmware supports backward compatibility mode with the Intel® NM 1.5 firmware for the online upgrade commands. So online upgrade commands are accessible also on the Net Function 0x30 with the commands codes A0h to A7h.

Up to 68 bytes of payload can be passed in one IPMI request. Update flow is the same as specified in section 2.7.1.

Net function = SDK general application (0x30)			
Code	**Command**	**Request, response data**	**Description**
A0h	Online Update Prepare For Update	**Request** None	This command, the first command that should be sent to initiate FW upgrade, restarts the FW upgrade to the initial state. Upon success, command **Online Update Get Status** returns "update requested" status. Upon failure, "update init failed" is returned.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in section 2.11) =80h – Operation refused (too many requests) =81h – Flash error =82h – Operation in progress (flash erase)	

Net function = SDK general application (0x30)			
Code	**Command**	**Request, response data**	**Description**
A1h	Online Update Open Area	**Request** Byte 1 – Area type =00h – Reserved =01h – Operational code =02h – PIA =03h – SDR Byte 2 – Area flags [0:7] – Reserved. Write as 00000000b.	Only update of operational code will be supported. *Note:* **Online Update Open Area** invalidates the rollback image partition. Only a successful image upload allows switching to a rollback image. Upon success, the command **Online Update Get Status** returns "update failed" status if there is a flash error or "update in progress" status if successful. 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 2.11) =80h – Operation refused. After this error code, the FW upgrade should be reinitialized by sending **Online Update Prepare For Update** command. =81h – Flash error. After this error code, the FW upgrade should be reinitialized by sending **Online Update Prepare For Update** command.	
A2h	Online Update Write Area	**Request** Byte 1 – Sequence number – Must start with 0 value Byte 2:N – Area data, where <N> should not exceed 70. Maximum supported number of data bytes is 68.	This command writes data into opened area. Upon success, the command Online Update Get Status returns "update failed" status if there is a flash error or "update in progress" status if successful. This command performs access to 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 2.11). =80h – Operation refused. After this error code, the FW upgrade should be reinitialized by sending **Online Update Prepare For Update** command. =81h – Flash error. After this error code, the FW upgrade should be reinitialized by sending **Online Update Prepare For Update** command.	
A3h	Online Update Close Area	**Request** Bytes 1:4 – Area size in bytes. Bytes 5:6 – Area checksum. Byte 6 should be set to 0. Byte 5 is a CRC8 ATM HEC (based on $x^8 + x^2 + x + 1$ polynomial) calculated over the operational binary image directly from the distribution package i.e. SPSOperational.bin file.	This command verifies the image checksum and size. Upon success, the command **Online Update Get Status** returns "update failed" status in the case of checksum verification failure or "update requested" status otherwise.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in section 2.11). =80h – Operation refused. Wrong CRC or image length mismatch. After this error code, the FW upgrade should be reinitialized by sending **Online Update Prepare For Update** command. =81h – Flash error. After this error code, the FW upgrade should be reinitialized by sending **Online Update Prepare For Update** command. =82h – In Progress. Not used on Turley.	

Net function = SDK general application (0x30)			
Code	**Command**	**Request, response data**	**Description**
A4h	Online Update Register Update	**Request** Byte 1 – Update type =0 – Reserved =1 – Normal update. Use the new image for the next boot. Use this update type to verify and to switch to a newly uploaded image – after a successful **Online Update Close Area** command. Upon success, command **Online Update Get Status** returns "update failed" status in case of image verification error or "update success" status otherwise. =2 – Reserved =3 – Manual rollback. Use this update type to cancel the switch to a new image and to use the current code – when executed right after **Online Update Prepare For Update** command. Upon success command **Online Update Get Status** returns "update failed" status in case of image verification error or "update rolled back" status otherwise. =4 – Abort update. Exits upgrade. Upon success command **Online Update Get Status** returns "update aborted" status. Byte 2 – Dependent flags [0:7] – Reserved. Write as 00000000b.	Instructs the controller that all areas to be updated have been sent and schedules an update to occur on the next system reset or system power cycle. Upon success, the command **Online Update Get Status** returns status as described in Figure 2-1. This command performs access to 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 2.11). =80h – Operation refused. =81h – Flash error. After this error code, the FW upgrade should be reinitialized by sending **Online Update Prepare For Update** command. =CCh – Invalid field.	
A6h	Online Update Get Status	**Request** None	Returns the current status of the FW Update.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in section 2.11). Byte 2 – Image status [0] – Reserved. Return as 0b. [1] – Staging image (new) =1 – Image valid. [2] – rollback image =1 – Image valid. [3:4] – Running image area =0 – CODE (Recovery mode) =1 – COD1 =2 – COD2 [5:7] – Reserved. Return as 000b. Byte 3 – Update state =0 – Idle (no update in progress). =1 – Update requested. =2 – Update in progress. =3 – Update success. =4 – Update failed.	

Net function = SDK general application (0x30)			
Code	**Command**	**Request, response data**	**Description**
		=5 – Update rolled back. =6 – Update aborted. =7 – Update initialization failed. =8-255 – Reserved. Byte 4 – UpdateAttemptStatus =0-255 – Reserved. Return as 0. Byte 5 – RollbackAttemptStatus =0-255 – Reserved. Return as 0. Byte 6 – Update Type =0-255 – Reserved. Return as 0. Byte 7 – DependentFlags [0:7] – Reserved. Return as 0. Byte 8:11 – Free Area Size in bytes.	
A7h	Online Update Get Capabilities	Request None	PIA and SDR updates not supported.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in section 2.11). Byte 2 – 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 3 – Special capabilities [0] – Rollback = 1 – Rollback supported. [1] – Recovery = 1 – Recovery supported. [2:7] – Reserved. Return as 000000b.	

2.7.3 IPMI OEM image inventory command

Net function = 2Eh-2Fh			
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 =0 – Provide information for the Recovery image. =1 – Provide information about 1st operational image. =2 – Provide information about 2nd operational image. =3-255 – Reserved for future use.	This is the command to query image inventory. This command can be used to learn what images are present.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in section 2.11).	

Net function = 2Eh-2Fh			
Code	**Command**	**Request, response data**	**Description**
		=80h – Operation refused (too many requests). =81h – Flash error. =82h – Operation in progress (flash erase). =C9h – 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 =2 Byte 6 – Firmware Revision 2: Minor Firmware Revision. BCD encoded. Byte 7:10 – Auxiliary Firmware Revision Information Byte 7 – Implemented version of Intel® NM firmware IPMI command specification BCD encoded = 2.0 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.	

2.7.4 Optimizing online upgrade performance

An Intel® NM image can take up to 2 MB of space on the flash, so, in order to speed up the upgrade process, the application responsible for upgrading should implement the following requirements:

To limit the number of IPMI requests the application should send maximum payload size of 64 to 68 bytes in each IPMI request.

Intel® NM firmware upgrade is able to maintain a receive window of 16 IPMI requests, so the application should send up to 16 **Online Update Write Area** requests without waiting for IPMI message response.

Intel® NM firmware supports out-of-order IMPI messages in a window size up to 16. It switches to that mode right after verifying that the initial part of the image (~16 KB) is valid Intel® ME firmware load. This allows remote console to automatically resend only the missing messages.

Intel® NM firmware will try to send responses to **Online Update Write Area** requests in order. If a lost message is detected, Intel® NM firmware will send acknowledgment to the next successfully received message, so the application may choose to resend just the missing request (optimal) or the whole window.

Flash erase requests are made asynchronously to the transfer, but flash erase may cause delays in upgrade processing. To avoid timeouts and retransmissions on the SMBUS line if the buffer for the IPMI upgrade messages is exhausted, Intel® NM firmware will respond with Node Busy Completion Code.

2.8 IPMI commands supported by Recovery Boot Loader

The Intel® NM firmware supports only the commands for IPMI over IPMB interface listed below 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)
- **Force ME Recovery** (Net function 2Eh, Command code DFh)

In recovery mode, only local update from the BIOS is possible.

2.9 IPMI OEM PECI Proxy commands

Note: On Brickland platforms, when running SiEnabling Intel® ME FW in Operational mode on the non-legacy PCH, the BMC is not allowed to use PECI Proxy commands.

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

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.
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.
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).
ADh	Response cannot be delivered because its length is not supported for underlying transport.
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.

Net function = 2Eh-2Fh			
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 (part of PECI standard header – values shall be in the range from 0x30 through 0x37). 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 (0x01, 0xF7, 0xA1, 0xA5, 0xB1, 0xB5, 0x61, 0x65, 0xE1, 0xE5; note that only Domain 0 codes are supported).	The command initiates a single PECI transaction on PECI bus. **Only PECI 3.0 command set is supported.** Note: In order to use this command, the OEM must sign "INTEL LICENSE AGREEMENT TO PLATFORM ENVIRONMENT CONTROL INTERFACE SPECIFICATION".
		Response Byte 1 – Completion Code (Remaining standard Completion Codes are shown in section 2.11) =00h – PECI response successfully retuned (see PECI response Completion Code for detailed response from PECI client, which may be not fully successful). =A4h – Bad read FSC in the response. =A5h – Bad write FCS field in the response. =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 FSC field are put here exactly as received from PECI client during Read transaction phase, excluding the Write FCS and Read FCS bytes. Retries: For no response (all zeros), Intel® ME FW performs one retry attempt on the PECI bus before the IPMI response is sent back to BMC For commands other than Ping(), GetTemp(0x01) and GetDIB(0xF7): Intel® ME FW verifies the checksums in the PECI response from client, and it performs one retry on the PECI bus for Bad FSC (equivalent to Abort FCS, or no	

Net function = 2Eh-2Fh			
Code	**Command**	**Request, response data**	**Description**
		response). If checksum is valid, Intel® ME FW interprets the PECI Completion Code byte and performs one retry on PECI bus for Completion Codes of type 0x8X – in this case, the retry bit is set in the PECI request and the AW FCS field is updated for the following commands: WrPkgConfig (0xA5), WrIAMSR (0xB5), WrPCIConfig (0x65), and WrPCIConfigLocal(0xE5). After 3 retries on PECI bus, subsequent retries of the same command are not attempted. The response is returned to BMC even in case of an error (no further retries).	
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).	The command initiates multiple PECI transactions on PECI bus. One IPMI request can contain several 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: In order to use this command, the OEM must 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 2.11). =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 section 4.7). =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 appeared in the IPMI request. **Note** – It is the responsibility of the BMC to ensure that the responses can fit into 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 the execution of the corresponding PECI commands is	

Net function = 2Eh-2Fh			
Code	**Command**	**Request, response data**	**Description**
		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 – CPU Number [7:3] – Reserved. [2:0] – CPU number (starting from 0). 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 [7:2] – 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).	This command provides read access to the "package Configuration Space" that is maintained by the CPU.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in section 2.11). =A1h – Wrong CPU number. =A2h – Command response timeout. =A4h – Bad read FSC 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 – CPU Number [7:3] – Reserved. [2:0] – CPU number (starting from 0). 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	This command provides write access to the "package Configuration Space" that is maintained by the CPU.

Net function = 2Eh-2Fh			
Code	**Command**	**Request, response data**	**Description**
		[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 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.	
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in section 2.11). =A1h – Wrong CPU number. =A2h – Command response timeout. =A4h – Bad read FSC 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 – CPU Number [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). 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).	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.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in section 2.11). =A1h – Wrong CPU Number.	

Net function = 2Eh-2Fh			
Code	**Command**	**Request, response data**	**Description**
		=A2h – Command response timeout. =A4h – Bad read FSC 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 – PCI type & CPU Number [7] – Reserved. [6] = 1b – PCI local space; The RdPCIConfigLocal() 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 system buses. [5:3] – Reserved. [2:0] – CPU number (starting from 0). 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 – 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.	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.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in section 2.11). =A1h – Wrong CPU Number. =A2h – Command response timeout. =A4h – Bad read FSC 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.	

Net function = 2Eh-2Fh			
Code	**Command**	**Request, response data**	**Description**
		=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 – CPU Number [7:3] – Reserved. [2:0] – CPU number (starting from 0). Byte 5 – Thread ID. Byte 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. Please refer to the appropriate processor documentation for a description of the accessibility limitations.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in section 2.11). =A1h – Wrong CPU number. =A2h – Command response timeout. =A4h – Bad read FSC 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	The command returns CPU and all Memory DIMMs temperature value for selected CPUs and DIMMs. The frame format is supported for the Brickland platform as it supports up to 16 CPUs, 4 channels per CPU, 6 DIMMs, and 1 memory controller per channel. Note:

Net function = 2Eh-2Fh			
Code	**Command**	**Request, response data**	**Description**
		requested. [5:4] – CPU set – defines which CPU set should be used = 0 – CPU0 to CPU3 = 1 – CPU4 to CPU7 = 2 – CPU8 to CPU11 = 3 – CPU12 to CPU15 [6] – Reserved. Should be set to 0. [7] – Request format 0 – Standard frame format - up to 4 DIMMs per DDR3 CHANNEL 1 – Extended frame format – up to 6 DIMMs per Intel® SMI2 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, DIMM#0. Byte 5 [1] – CHANNEL#0, DIMM#1. Byte 5 [2] – CHANNEL#0, DIMM#2. Byte 5 [3] – CHANNEL#0, DIMM#3. Byte 5 [4] – CHANNEL#1, DIMM#0. Byte 5 [5] – CHANNEL#1, DIMM#1. Byte 5 [6] – CHANNEL#1, DIMM#2. Byte 5 [7] – CHANNEL#1, DIMM#3. Byte 6 [0] – CHANNEL#2, DIMM#0. Byte 6 [1] – CHANNEL#2, DIMM#1. Byte 6 [2] – CHANNEL#2, DIMM#2. Byte 6 [3] – CHANNEL#2, DIMM#3. Byte 6 [4] – CHANNEL#3, DIMM#0. Byte 6 [5] – CHANNEL#3, DIMM#1. Byte 6 [6] – CHANNEL#3, DIMM#2. Byte 6 [7] – CHANNEL#3, 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] – MC#0, SMI2#0, DDR3#0, DIMM#0. Byte 5 [1] – MC#0, SMI2#0, DDR3#0, DIMM#1. Byte 5 [2] – MC#0, SMI2#0, DDR3#0, DIMM#2. Byte 5 [3] – Reserved Byte 5 [4] – MC#0, SMI2#0, DDR3#1, DIMM#0. Byte 5 [5] – MC#0, SMI2#0, DDR3#1, DIMM#1. Byte 5 [6] – MC#0, SMI2#0, DDR3#1, DIMM#2. Byte 5 [7] – MC#0, SMI2#0, Memory Controller	MC: Memory Controller, SMI2: Intel® SMI2 Channel, DDR3: DDR3 subchannel. Note: Due to CPU limitations, it is not possible to read memory temperature on Denlow platforms

Net function = 2Eh-2Fh			
Code	**Command**	**Request, response data**	**Description**
		Temperature. Byte 6 [0] – MC#0, SMI2#1, DDR3#0, DIMM#0. Byte 6 [1] – MC#0, SMI2#1, DDR3#0, DIMM#1. Byte 6 [2] – MC#0, SMI2#1, DDR3#0, DIMM#2. Byte 6 [3] – Reserved. Byte 6 [4] – MC#0, SMI2#1, DDR3#1, DIMM#0. Byte 6 [5] – MC#0, SMI2#1, DDR3#1, DIMM#1. Byte 6 [6] – MC#0, SMI2#1, DDR3#1, DIMM#2. Byte 6 [7] – MC#0, SMI2#1, Memory Controller Temperature. Byte 7 [0] – MC#1, SMI2#2, DDR3#0, DIMM#0. Byte 7 [1] – MC#1, SMI2#2, DDR3#0, DIMM#1. Byte 7 [2] – MC#1, SMI2#2, DDR3#0, DIMM#2. Byte 7 [3] – Reserved Byte 7 [4] – MC#1, SMI2#2, DDR3#1, DIMM#0. Byte 7 [5] – MC#1, SMI2#2, DDR3#1, DIMM#1. Byte 7 [6] – MC#1, SMI2#2, DDR3#1, DIMM#2. Byte 7 [7] – MC#1, SMI2#2, Memory Controller Temperature. Byte 8 [0] – MC#1, SMI2#3, DDR3#0, DIMM#0. Byte 8 [1] – MC#1, SMI2#3, DDR3#0, DIMM#1. Byte 8 [2] – MC#1, SMI2#3, DDR3#0, DIMM#2. Byte 8 [3] – Reserved Byte 8 [4] – MC#1, SMI2#3, DDR3#1, DIMM#0. Byte 8 [5] – MC#1, SMI2#3, DDR3#1, DIMM#1. Byte 8 [6] – MC#1, SMI2#3, DDR3#1, DIMM#2. Byte 8 [7] – MC#1, SMI2#3, 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 2.11). =A1h – Wrong CPU number. =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	

Net function = 2Eh-2Fh			
Code	**Command**	**Request, response data**	**Description**
		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 Celsius below the Thermal Control Circuit Activation temperature, with some values reserved to provide error indication, as specified below. 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 Celsius with the following values reserved to provide error indication: =0xFF – Sensor or device not present. =0xFE – Reserved. =0xFD – Data unavailable due to sensor or interface failure.	

2.10 IPMI OEM PECI Proxy sensors

Table 2-1 and this section summarize the sensors exposed by PECI Proxy functionality in Intel® ME FW. Sensor readings are not available when system is in low power state.

Note that 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 2-1 PECI Proxy sensors

Sensor #	Description	Notes
28	CPU#0 Thermal Status	Discrete sensor.
29	CPU#1 Thermal Status	Discrete sensor.
30	CPU#2 Thermal Status	Discrete sensor.
31	CPU#3 Thermal Status	Discrete sensor.
192	CPU#4 Thermal Status	Discrete sensor.
193	CPU#5 Thermal Status	Discrete sensor.
194	CPU#6 Thermal Status	Discrete sensor.
195	CPU#7 Thermal Status	Discrete sensor.
32	CPU#0 Thermal Control Circuit Activation Sensor	Threshold sensor.
33	CPU#1 Thermal Control Circuit Activation Sensor	Threshold sensor.
34	CPU#2 Thermal Control Circuit Activation Sensor	Threshold sensor.

Sensor #	Description	Notes
35	CPU#3 Thermal Control Circuit Activation Sensor	Threshold sensor.
196	CPU#4 Thermal Control Circuit Activation Sensor	Threshold sensor.
197	CPU#5 Thermal Control Circuit Activation Sensor	Threshold sensor.
198	CPU#6 Thermal Control Circuit Activation Sensor	Threshold sensor.
199	CPU#7 Thermal Control Circuit Activation Sensor	Threshold sensor.
36	CPU#0 Tcontrol Sensor	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.
37	CPU#1 Tcontrol Sensor	
38	CPU#2 Tcontrol Sensor	
39	CPU#3 Tcontrol Sensor	
200	CPU#4 Tcontrol Sensor	
201	CPU#5 Tcontrol Sensor	
202	CPU#6 Tcontrol Sensor	
203	CPU#7 Tcontrol Sensor	
48	CPU#0 Tjmax Sensor	OEM sensor that presents Tjmax parameter of the CPU. The value of this sensor is constant and may only change upon HW configuration change.
49	CPU#1 Tjmax Sensor	
50	CPU#2 Tjmax Sensor	
51	CPU#3 Tjmax Sensor	
204	CPU#4 Tjmax Sensor	
205	CPU#5 Tjmax Sensor	
206	CPU#6 Tjmax Sensor	
207	CPU#7 Tjmax Sensor	
52	Memory Throttling for CPU#0	Threshold sensor.
53	Memory Throttling for CPU#1	Threshold sensor.
54	Memory Throttling for CPU#2	Threshold sensor.
55	Memory Throttling for CPU#3	Threshold sensor.
208	Memory Throttling for CPU#4	Threshold sensor.
209	Memory Throttling for CPU#5	Threshold sensor.
210	Memory Throttling for CPU#6	Threshold sensor.
211	Memory Throttling for CPU#7	Threshold sensor.

2.10.1 CPU thermal status sensors

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

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.

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

2.10.2 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).

2.10.3 CPU T-control sensors

These sensors present T-Control parameters for the processors. T-Control value is fan speed control reference temperature. For detailed description of the value, refer to [EMTS] document.

Fan Temperature target offset (T-Control) indicates the relative offset from CPU Tjmax temperature at which fans should engage. For detailed description of the Tjmax value, refer to [EMTS] document.

2.10.4 CPU Tjmax sensors

These sensors are presenting Tjmax parameter for the processors. CPU Tjmax is the minimum temperature that the processor will start throttling due to TCC activation. For detailed description of the value, refer to [EMTS] document.

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

2.10.5 Memory throttling status sensors

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

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

2.11 IPMI standard completion codes

Intel® NM firmware IPMI commands use standard Completion Codes from table below 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.[2]

Code	Definition
Generic completion codes 00h, C0h-FFh	
00h	Command Completed Normally.
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.

2 *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
D4h	Cannot execute command due to insufficient privilege level or other security based restriction (for example, 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 subfunction has been disabled or is unavailable (for example, disabled for 'firmware firewall').
FFh	Unspecified error.
Device-specific (OEM) codes 01h-7Eh	
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-priori 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.

2.12 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 Nonrecoverable – going low Lower Nonrecoverable – going high Upper Noncritical – going low Upper Noncritical – going high Upper Critical – going low Upper Critical – going high Upper Nonrecoverable – going low Upper Nonrecoverable – going high

§

3. Intel® Intelligent Power Node Manager IPMI Interface

This chapter describes IPMI command interface used by the Intel® Intelligent Power Node Manager 2.0 implementation:

- For external Intel® NM IPMI commands that should be used by the external management console, see section 3.1.
- For the discovery mechanism for Intel® NM functionality, see section 4.5.
- For product-specific Intel® NM IPMI commands used for low-level power management access and for debugging, see section 4.5.
- For external IPMI sensors that should be monitored by external management console, see section 3.4.
- For 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, the 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 Technology 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® NM 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 external management software uses the configuration and control commands 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® NM firmware over IPMB channel. 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 Intel® 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® NM.

Note: The commands listed in this section are not supported when Intel® NM Feature Enabled is set to *False* using the Flash Image Tool.

Net function = 2Eh-2Fh			
Code	**Command**	**Request, response data**	**Description**
C0h	Enable/Disable Intel® NM Policy Control	Request Byte 1:3 - Intel Manufacturer ID – 000157h, LS byte first Byte 4 – Flags [0:2] – Policy Enable/Disable =0h – Global Disable Intel® NM policy control – disables policy control for all power domains regardless of the value set in Domain ID field (Byte 5). =1h – Global Enable Intel® NM 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® NM policies for the domain given by Byte 5. =3h – Per Domain Enable Intel® NM policies for the domain given by Byte 5. =4h – Per Policy Disable Intel® NM policy for the domain/policy given by Byte 5 and Byte 6. =5h – Per Policy Enable Intel® NM policy for the domain/policy given by Byte 5 and Byte 6. [3:7] – Reserved. Write as 00000b. Byte 5 – Domain ID [0:3] – Domain ID Identifies the domain that this Intel® NM 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. [4:7] – Reserved. Write as 0000b. Byte 6 – Policy ID. This field is valid if Per Policy Enable/Disable is set. Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in section 2.11). =80h – Policy ID Invalid. =81h – Domain ID Invalid. Byte 2:4 – Intel Manufacturer ID – 000157h, LS byte first.	Enable or Disable the Intel® NM 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® NM up to 2 sec to stop limit power. Responder LUN is validated only for policy disable/enable. This means the responder LUN must match the responder LUN from the policy creation request. For changes on Global Policy Control or Domain Control, requests with any responder LUN would be accepted.
C1h	Set Intel® NM Policy	Request Byte 1:3 - Intel Manufacturer ID – 000157h, LS byte first. Byte 4 – Domain ID [0:3] – Domain ID (Identifies the domain that this Intel® NM policy applies to) =00h – Entire platform. =01h – CPU subsystem. =02h – Memory subsystem. =03h – Reserved. =04h – High Power I/O subsystem. Others – Reserved. [4] – Policy Enabled (set to 1 if policy should be enabled by	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

Net function = 2Eh-2Fh			
Code	**Command**	**Request, response data**	**Description**
		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. [5:7] – Reserved. Write as 000b. Byte 5 – Policy ID. Byte 6 – Policy Type | Policy Trigger Type [0:3] – 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 1/10 of second. =3 – Time After Platform Reset Trigger in 1/10 of second. =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. [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. [5:6] – 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. For example, 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 nonaggressive 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). Policy shall persist across an Intel® ME reset. = 1b – volatile memory. Policy shall not persist across an Intel® ME reset. Byte 7 – Policy Exception Actions performed if policy cannot be maintained (if maintained policy power limit given by bytes 8-9	possible only if that policy for the specified Policy ID is disabled. For 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 Inlet Air Temperature policy called Minimum Power Consumption policy with the Power Limit set to 0. The policy does not have the power limit defined. When the inlet air temperature rises above the trigger value defined in the policy, the SPS 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 does not allow setting the correction action to "System Shutdown", but the operator can 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. It is possible to specify whether the

Net function = 2Eh-2Fh			
Code	**Command**	**Request, response data**	**Description**
		is exceeded over Correction Time Limit). [0] – Send alert. [1] – Shutdown system (hard shutdown via BMC). [2:7] – reserved. Write as 000000b. Byte 8:9 – Policy Target Limit For the following Policy Trigger Type value (Byte 6 bits [0:3]) this field contains the following data: 0, 1, 3 – 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. Intel® NM does not send Power Limit Exceeded event. The zero value can be set even if the minimum power set for the domain is not zero. 2 – Throttling level of the platform in %, where 100% enables maximum throttling of the system. 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 (by running the CPU in LFM and using the fewest amount of threads). =1 – Platform should boot (during BIOS POST) in performance optimized mode. [1:7] – 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. For example, 1 passed on that field means than on each CPU package 1 core should be disabled by the BIOS. [8:15] – Reserved. Write as 00000000b. Byte 10:13 – Correction Time Limit – the maximum time in ms, in which the Intel® NM 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. 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 platform reset or startup. If BMC does not send Set Event Receiver command within this time after platform reset or startup, Intel® NM 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.	Policy Exception Actions Send Alert will be sent once or after every Correction Time in which the policy power limit given by bytes 8-9 is exceeded. This can be specified in the FITc by toggling the "Repeat Alert Exception Action" option. Remove or modify operations would be performed only if the responder LUN matches the responder LUN from the request when the policy was created. **Note:** You can remove an Intel® NM Policy without first disabling the policy because you can change Intel® NM parameters during runtime.

Net function = 2Eh-2Fh			
Code	**Command**	**Request, response data**	**Description**
		4 – Policy Trigger is not applicable for boot time policy and should be set to 0. 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 2.11). =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. =86h – Policy Trigger value out of range. =89h – Statistics Reporting Period out of range. =8Bh – Invalid value of Aggressive CPU Power Correction field. =D5h – Policy could not be updated since Policy ID already exists and is enabled. Byte 2:4 – Intel Manufacturer ID – 000157h, LS byte first.	
C2h	Get Intel® NM Policy	Request Byte 1:3 – Intel Manufacturer ID – 000157h, LS byte first. Byte 4 – Domain ID [0:3] - Domain ID (Identifies the domain that this Intel® NM policy applies to). =00h – Entire platform. =01h – CPU subsystem. =02h – Memory subsystem. =03h – Reserved. =04h – High Power I/O subsystem. Others – Reserved. [4:7] – 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 doesn't match with responder LUN from request when policy was created. Get Intel® NM Policy checks the administrative status of enabled policies and will respond that Intel® NM policy is enabled even if it is in a nonoperational mode caused by setting Intel® NM Power Draw Range above the policy limit.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in section 2.11). =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. 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 [0:3] - Domain ID (Identifies the domain that this Intel® NM policy applies to).	

Net function = 2Eh-2Fh			
Code	**Command**	**Request, response data**	**Description**
		=00h – Entire platform. =01h – CPU subsystem. =02h – Memory subsystem. =03h – Reserved. =04h – High Power I/O subsystem. Others – Reserved. [4] – Policy enabled. [5] – Per Domain Intel® NM policy control enabled. [6] – Global Intel® NM policy control enabled. [7] – Set to 1 if policy is created and managed by other management client e.g. DCMI management API. Byte 6 – Policy Type | Policy Trigger Type [0:3] – 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 Platform Reset Trigger in 1/10th of second. =4 – Boot time policy. [4] – Policy Type =1 – Power Control Policy. Policy will maintain Power limit. [5:6] – 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. =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 = 1b – volatile memory. 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. [2:7] – Reserved. Write as 000000b. Byte 8:9 – Power Limit. Byte 10:13 – Correction Time Limit - the maximum time in ms, 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	

Net function = 2Eh-2Fh			
Code	Command	Request, response data	Description
		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 [0:3]) 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 Intel® NM 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 Platform Reset Trigger in 1/10th of second. 3 – Policy Trigger defines time in 1/10 of second after platform reset or startup. If BMC does not send Set Event Receiver command within this time after platform reset or startup, 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 filed is set to 0. 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: 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. [0:3] – 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. [4:7] – 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 [0:3] on reception of Completion Code 81h.	

Net function = 2Eh-2Fh			
Code	**Command**	**Request, response data**	**Description**
C3h	Set Intel® NM Policy Alert Thresholds	Request Byte 1:3 – Intel Manufacturer ID – 000157h, LS byte first. Byte 4 – Domain ID. [0:3] – Domain ID (Identifies the domain that this Intel® NM policy applies to.) =00h – Entire platform. =01h – CPU subsystem. =02h – Memory subsystem. =03h – Reserved. =04h – High Power I/O subsystem. Others – Reserved. [4:7] – Reserved. Write as 0000b. Byte 5 – Policy ID. Byte 6 – Number of alert thresholds as unsigned integer value. 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 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 Platform Reset triggers the array contains time 1/10th 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 based on correction time limit) exceeds any of the configured alert thresholds. Assertion is generated for exceeding the threshold (going high) and deassertion for going low. The hysteresis value for avoiding jitters around the threshold will be OEM configurable using factory-preset values. **Note** – Max 3 alert thresholds are supported per policy. Each alert threshold is 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. All the thresholds are unsigned integer values.	Sets the Intel® NM Policy alert thresholds. This is part of the Intel® NM Policy described earlier and applies to the same policy as specified by Policy ID. Modification of policy alert thresholds would be performed only if responder LUN match with responder LUN from request when policy was created
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in section 2.11). =80h – Policy ID Invalid. =81h – Domain ID Invalid. =84h – Limit in one of thresholds is invalid. =87h – Number of thresholds is too large or power limits are invalid. =D5h – Alert thresholds cannot be changed for enabled policy; disable it first. Byte 2:4 – Intel Manufacturer ID – 000157h, LS byte first.	

Net function = 2Eh-2Fh			
Code	**Command**	**Request, response data**	**Description**
C4h	Get Intel® NM Policy Alert Thresholds	Request Byte 1:3 – Intel Manufacturer ID – 000157h, LS byte first. Byte 4 – Domain ID [0:3] – Domain ID (Identifies the domain that this Intel® NM policy applies to). =00h – Entire platform. =01h – CPU subsystem. =02h – Memory subsystem. =03h – Reserved. =04h – High Power I/O subsystem. Others – Reserved. [4:7] – Reserved. Write as 0000b. Byte 5 – Policy ID.	Gets the Intel® NM Policy alert thresholds. Request would be executed even if responder LUN doesn't match with responder LUN from request when policy was created.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in section 2.11). =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 Platform 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 Intel® NM Policy Suspend Periods	Request Byte 1:3 – Intel Manufacturer ID – 000157h, LS byte first. Byte 4 – Domain ID [0:3] - Domain ID (Identifies the domain that this Intel® NM policy applies to). =00h – Entire platform. =01h – CPU subsystem. =02h – Memory subsystem. =03h – Reserved. =04h – High Power I/O subsystem. Others – Reserved. [4:7] – 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.	Sets the Intel® NM 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 fails new suspend periods cannot be configured and existing periods are

Net function = 2Eh-2Fh			
Code	**Command**	**Request, response data**	**Description**
		Byte 7:N – Array of policy suspend periods (following information is repeated for each suspend period). Each suspend period is defined by 3 bytes: First byte – Policy suspend start time. It is 0 – 239 number of minutes from mid-night divided by 6. Values 240 – 255 are reserved. Second byte – Policy suspend stop time. It is 1 – 240 number of minutes from midnight divided by 6. Values 0 and 241 – 255 are reserved. Third byte – Suspend period recurrence pattern: [0] – Repeat the suspend period every Monday. [1] – Repeat the suspend period every Tuesday. [2] – Repeat the suspend period every Wednesday. [3] – Repeat the suspend period every Thursday. [4] – Repeat the suspend period every Friday. [5] – Repeat the suspend period every Saturday. [6] – Repeat the 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 then 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 example, 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, 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 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.	ignored. Note that RTC synchronization failure situation is signaled with Intel® NM Health Event (see 3.4.8) Modification of policy suspend periods would be performed only if responder LUN match with responder LUN from request when policy was created.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in section 2.11). =80h – Policy ID Invalid. =81h – Domain ID Invalid. =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. =D5h – Suspend periods cannot be changed for enabled policy, disable it first. =D6h – Command disabled/unavailable due to lack of RTC synchronization. Byte 2:4 – Intel Manufacturer ID – 000157h, LS byte first.	

Net function = 2Eh-2Fh			
Code	**Command**	**Request, response data**	**Description**
C6h	Get Intel® NM Policy Suspend Periods	Request Byte 1:3 – Intel Manufacturer ID – 000157h, LS byte first. Byte 4 – Domain ID [0:3] – Domain ID (Identifies the domain that this Intel® NM policy applies to). =00h – Entire platform. =01h – CPU subsystem. =02h – Memory subsystem. =03h – Reserved. =04h – High Power I/O subsystem. Others – Reserved. [4:7] – 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.
		Response Byte 1 – Completion Code **=**00h – Success (Remaining standard Completion Codes are shown in section 2.11). =80h – Policy ID Invalid. =81h – Domain ID Invalid. 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 mid-night divided by 6. 2nd byte – Policy suspend stop time encoded as a number of minutes from mid-night divided by 6. 3rd byte – Suspend period recurrence pattern: [7] – Reserved. Write as 0b. [6] – Repeat the suspend period every Sunday. [5] – Repeat the suspend period every Saturday. [4] – Repeat the suspend period every Friday. [3] – Repeat the suspend period every Thursday. [2] – Repeat the suspend period every Wednesday. [1] – Repeat the suspend period every Tuesday. [0] – Repeat the 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 e.g. if 1st suspend period is in bytes 6:8 then bytes 9:11 will contain the 2nd policy suspend period.	

Net function = 2Eh-2Fh			
Code	**Command**	**Request, response data**	**Description**
C7h	Reset Intel® NM Statistics	Request Byte 1:3 – Intel Manufacturer ID – 000157h, LS byte first. Byte 4 – Mode [0:4] – Mode =00h – Reset global statistics including power statistics, throttling statistics and inlet temperature statistics. =01h – Reset per policy statistics including power, throttling statistics and trigger statistics. =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. [5-7] – Reserved. Write as 000b. Byte 5 – Domain ID [0:3] – Domain ID (Identifies the domain that this Intel® NM policy applies to) =00h – Entire platform. =01h – CPU subsystem. =02h – Memory subsystem. =03h – Reserved. =04h – High Power I/O subsystem. Others – Reserved. [4:7] – Reserved. Write as 0000b. For Global Host Communication Failure, CPU throttling and memory throttling statistics Domain ID must be set to 00h. Byte 6 – Policy ID (ignored if field Mode is set to 00h).	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. 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 2.11). =80h – Policy ID Invalid. =81h – Domain ID Invalid. =88h – Invalid Mode. Byte 2:4 – Intel Manufacturer ID – 000157h, LS byte first.	
C8h	Get Intel® NM Statistics	Request Byte 1:3 – Intel Manufacturer ID – 000157h, LS byte first. Byte 4 – Mode [0:4] – Mode =01h – Global power statistics in [Watts]. =02h – Global inlet temperature statistics in [Celsius]. =03h – 10h – Reserved. =11h – Per policy power statistics in [Watts]. =12h – Per policy trigger statistics in [Celsius]. =13h – Per policy throttling statistics in [%]. =1Ah – Reserved. =1Bh – Global Host Unhandled Requests statistics. =1Ch – Global Host Response Time statistics. =1Dh – Global CPU throttling statistics. =1Eh – Global memory throttling statistics. =1Fh – Global Host Communication Failure statistics. [5:7] – 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® NM for

Net function = 2Eh-2Fh			
Code	**Command**	**Request, response data**	**Description**
		Byte 5 – Domain ID [0:3] – Domain ID (Identifies the domain that this Intel® NM policy applies to) =00h – Entire platform. =01h – CPU subsystem. =02h – Memory subsystem. =03h – Reserved. =04h – High Power I/O subsystem. Others – Reserved. [4:7] – Reserved. Write as 0000b. **Note** – For Mode (byte 4) in a range 1Bh – 1Fh Domain ID must be set to 00h. Byte 6 – Policy ID **Note** – If Mode field equals 11h or 12h, this field indicates Policy ID for which the statistics are requested. Otherwise the field shall be set to 0.	taking corrective action or triggering alerts based a 'Set Intel® NM Alert Threshold' because the averaging periods could be different. Modes for per policy statistics are correlated to policy trigger type. A request for mode 0x11 can be only issued for policies with trigger type 0 (no trigger defined) and 3 (Time After Platform Reset) and type 4 (Boot time policy). Mode 0x12 is available only for policies with trigger type set to 1 (Inlet Temperature). Mode 0x13 can be issued for policies with trigger type 2 (Missing Power Readings). 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
		Response Byte 1 – Completion code =00h – Success (Remaining standard Completion Codes are shown in section 2.11). =80h – Policy ID Invalid. =81h – Domain ID Invalid. =88h – Invalid Mode. Byte 2:4 – Intel Manufacturer ID – 000157h, LS byte first. Byte 5:6 – Current Value (integer) see 3.1.1. Byte 7:8 – Minimum Value (unsigned integer) see 3.1.2. Byte 9:10 – Maximum Value (unsigned integer) see 3.1.3. Byte 11:12 – Average Value (unsigned integer) see 3.1.4. 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 [0:3] – Domain ID (Identifies the domain that this Intel® NM policy applies to) =00h – Entire platform. =01h – CPU subsystem. =02h – Memory subsystem. =03h – Reserved. =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® NM Policy Control is Globally Enabled (see C0h command) and Intel® NM Domain control is also Enabled (see C0h command). =0 – Otherwise. If request Byte 4 equals 01h or 02h state =1 – if Intel® NM Policy Control is Globally Enabled (see C0h command).	

Net function = 2Eh-2Fh			
Code	**Command**	**Request, response data**	**Description**
		=0 – Otherwise. [5] – Policy Operational state 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 these events occurs: – 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 is administratively disabled. If request Byte 4 equals 01h or 02h state this bit is always set to 0 as it applies only for per policy statistics. [6] – Measurements state =1 – Measurements in progress (host CPU is in S0 state and there are no problems with the readings reported to the remote console). =0 – Measurements are suspended (host CPU in Sx state or in the S0 state problem with the readings reported to the remote console). [7] – Policy activation state 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. If request Byte 4 equals 01h or 02h state this bit is always set to 0 as it applies only for per policy statistics. For Host Response Time, Host Unhandled Requests, Host Communication Failure, CPU, and memory throttling statistics Byte 21 is always set to 0.	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. 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 doesn't affect execution of requests for other modes.
C9h	Get Intel® NM Capabilities	Request Byte 1:3 – Intel Manufacturer ID – 000157h, LS byte first. Byte 4 – Domain ID [0:3] – Domain ID (Identifies the domain that this Intel® NM policy applies to) =00h – Entire platform. =01h – CPU subsystem. =02h – Memory subsystem. =03h – Reserved. =04h – High Power I/O subsystem. Others – Reserved. [4:7] – Reserved. Write as 0000b. Byte 5 – Policy Type | Policy Trigger Type [0:3] – 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 Platform Reset Trigger in 1/10th of second. =4 – Boot time policy. Others – Reserved. [4:7] – Policy Type =1 – Power Control Policy. Others – Reserved.	Get Intel® NM capabilities. This command would be executed regardless of responder LUN value. Common setting is returned for all LUNs. Due to CPU limitation, Intel® ME FW is not able to read power range for memory domain on Denlow platforms. Note: On Denlow platforms, 0 watts is always reported for Min Power in CPU domain.

Net function = 2Eh-2Fh			
Code	**Command**	**Request, response data**	**Description**
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in section 2.11). =81h – Domain ID Invalid. =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 (No Policy Trigger) than 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 the field is equal to zero, it means that Intel® NM does not impose any limit. If Policy Trigger Type in the request equals 1 (Inlet Temperature Policy Trigger) than 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 Platform Reset) than 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) than 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 (No Policy Trigger) than 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 this field is equal to zero, it means that Intel® NM does not impose any limit. If Policy Trigger Type in the request equals 1 (Inlet Temperature Policy Trigger) than this field contains minimum 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 Platform Reset) than 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) than this field is not applicable and is set to 0. Byte 10:13 – Min Correction Time settable in milliseconds (ms) as unsigned integer value. Byte 14:17 – Max Correction Time settable in milliseconds	

Net function = 2Eh-2Fh			
Code	**Command**	**Request, response data**	**Description**
		(ms) 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® NM policy applies to) =00h – Entire platform. =01h – CPU subsystem. =02h – Memory subsystem. =03h – Reserved. =04h – High Power I/O subsystem. Others – Reserved. [4:6] – Reserved. Write as 000b. [7] – Limiting based on =0 – Wall input power – PSU input power. =1 – DC power – PSU output power or bladed system or direct DC reading from the CPU.	
CAh	Get Intel® NM Version	Request Byte 1:3 – Intel Manufacturer ID – 000157h, LS byte first.	Get Intel® NM 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 2.11). 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 – FFh – Reserved for future use. Byte 6 – IPMI interface version =01h – Intel® NM IPMI version 1.0. =02h – Intel® NM IPMI version 2.0 (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 response byte 4 – 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).	

Net function = 2Eh-2Fh			
Code	**Command**	**Request, response data**	**Description**
CBh	Set Intel® NM Power Draw Range	Request Byte 1:3 – Intel Manufacturer ID – 000157h, LS byte first. Byte 4 – Domain ID [0:3] – 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. [4:7] – 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. 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, minimum power draw will be obtained from CPU via PECI. This is an unsigned integer value.	Set the min/max power consumption ranges. This info is preserved in the persistent storage. After receiving the request, Intel® NM validates whether any policies with limit do not fit into the new power consumption range. If Intel® NM detects such policies, it sends Intel® NM Health Event with Policy Misconfiguration flag set. Additionally, Intel® NM disables all the policies with power limit below the Minimum Power Draw. The policy enters a nonoperational state in which power limit is kept not by Intel® NM policy but by Intel® NM Power Draw Range. Note that Get Intel® NM Policy function will return that the policy is “enabled” in a nonoperational state. Intel® NM does not enable them until power policy limit is below the Minimum Power Draw, Intel® NM Power Draw Range, can be used to override the existing policy power limit for a period of time. The same action is taken, when Intel® NM receives a new power draw range from CPU via PECI at POST.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in section 2.11). =81h – Invalid Domain ID. Byte 2:4 – Intel Manufacturer ID – 000157h, LS byte first.	

Net function = 2Eh-2Fh			
Code	**Command**	**Request, response data**	**Description**
			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.
CEh	Set Intel® NM Alert Destination	Request Byte 1:3 – Intel Manufacturer ID – 000157h, LS byte first. Byte 4 – Channel number [0:3] – 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] – Destination information operation =0 – Register alert receiver. =1 – Unregister alert receiver. Use this bit to invalidate the current destination configuration. Alerts will be blocked. Byte 5 – Destination Information **For channel medium – IPMB** [0] – reserved. Write as 0b. [1:7] – 7-bit I2C Slave Address. **For channel medium - 802.3 LAN** Destination Selector/ Operation [0:3] – Destination selector. Selects which alert destination should go to =0h – Use volatile destination info. =1h – Fh – Use nonvolatile destination info. Destination Selector definition is the same as in the "Set/Get LAN Configuration Parameters" command. [4:7] – Reserved. Write as 0000b. Byte 6 – Alert String Selector. Selects which Alert String, if any, to use with the alert. [0:6] – 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.	Provides alert destination info for Intel® NM to send direct alerts that bypass the BMC SEL. Destination Selector/Operation and Alert String Selector fields correspond to the associated LAN configuration parameters applicable to the BMC channel number over which to send the alert. Note that Intel® NM will determine the channel medium type in order to resolve the destination. If Intel® NM is implemented in an entity separate from the BMC, then this is done by querying the BMC using the Get Channel Info IPMI command. This command would be executed regardless of responder LUN value; however there is one global setting that would be updated. New value would modify settings for other responder LUNs.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in section 2.11). Byte 2:4 – Intel Manufacturer ID – 000157h, LS byte first.	

Net function = 2Eh-2Fh			
Code	**Command**	**Request, response data**	**Description**
CFh	Get Intel® NM 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® NM. 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 2.11). Byte 2:4 – Intel Manufacturer ID – 000157h, LS byte first. Byte 5 – Channel number [0:3] – BMC channel number over which alert from BMC to management console will be sent. [4:6] – Reserved. Write as 000b. [7] – Destination information operation =0 – Configuration valid. Alert receiver registered. =1 – Configuration invalid. Alert receiver not registered. Alerts are blocked. Byte 6 – Destination Selector/ Operation [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 00000b. Byte 7 – Alert String Selector. Selects which Alert String, if any, to use with the alert. [0:6] – 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.	
F2h	**Get Limiting Policy ID**	Request Byte 1:3 = Intel manufacturers ID – 0x000157, LS byte first Byte 4 – Domain ID [0:3] = Domain ID(Identifies the domain for which the response is to be provided.) **00h** – Entire platform **01h** – CPU subsystem **02h** – Memory subsystem **Others** – Reserved [4:7] = Reserved. Write as 0000b.	This command fetches policy which is currently limiting. 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 [NM IPMI]) =A1h – No policy is currently limiting for the specified DomainID Byte 2:4 = Intel manufacturers ID – 0x000157, 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	

3.1.1 Get Intel® NM statistics current value field

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 in %. 100% means that the CPU/memory is throttled to maximum level that Intel® NM can enforce. 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 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® NM 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 in %. 100% means that the CPU/memory is throttled to maximum level that Intel® NM can enforce. 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 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® NM 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 in %. 100% means that the CPU/memory is throttled to maximum level that Intel® NM can enforce. 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 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 in %. 100% means that the CPU/memory is throttled to maximum level that Intel® NM can enforce. 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 allows you to modify some policy parameters while a policy is active. Modification of different parameters affects policy behavior in different ways. The following list shows the policy parameters from the *Set Intel® NM Policy* command and what the impact on policy behavior will be if the policy is 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, the number of policies of different types that may be created is limited.)
- **Policy Enable (Byte 4 [4])** – changing of this parameter is similar to execution of Enable/Disable Intel® NM 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, the number of policies of different types that may 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 stored in persistent memory.
- **Policy Exception Action (Byte 7 [1:0])** – If changed, failure action timeout is reset, and new failure 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, failure 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 would be checked. New threshold event would be generated only when threshold state would change (e.g. if before modification it was asserted and it's still asserted, no event would be generated). If new policy Alert Threshold would be configured, its current state would be treated as deasserted so if it would be exceeded, assertion event would be sent. In case of removing Alert Threshold, its current state would be lost; in case of recreation, it would be treated as deasserted.

If Set Intel® NM Policy Suspend Periods is executed on enabled policy, a new configuration would be applied in 1 second.

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: The following commands are not supported when Intel® NM Feature Enabled is set to 'false' using Flash Image Tool.

Net function = 2Eh-2Fh			
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 [0:3] – 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. [4:7] – Reserved. Write as 0000b. Byte 5:6 – Target power budget in [Watts] that should be maintained by the Power Budget Control Service.	Set total power budget for the CPUs. This command is optional and may be unavailable on certain implementations. This command controls the platform power limit using aggressive settings. **Note** – "Set Total Power Budget" function is only accessible if the Intel® NM policy control feature is disabled.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in section 2.11). =81h – Invalid Domain ID. =84h – Power Budget out of range. Byte 2:4 – Intel Manufacturer ID – 000157h, LS byte first. **Note** – When the Intel® NM policy control is enabled globally (see C0h command) the command should respond with Completion Code D5h – Cannot execute command – command not supported in present state.	
D2h	Set Max Allowed CPU P-state/T-state	Request Byte 1:3 – Intel Manufacturer ID – 000157h, LS byte first. Byte 4 – Domain ID [0:3] – 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. [4:5] – Control Knob =00b – Set max allowed CPU P-state/T-state. =01b – Set max allowed logical processors.	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

Net function = 2Eh-2Fh			
Code	**Command**	**Request, response data**	**Description**
		=10b – Reserved. =11b – Reserved. [6:7] – 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.	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.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in section 2.11). =81h – Invalid Domain ID. =8Ah – P-state or T-state 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. Byte 4 – Domain ID [0:3] – Domain ID (Identifies the domain that this setting applies to) 00h – Entire platform. 01h – CPU subsystem. 02h – Memory subsystem. Others – Reserved. [4:7] – Reserved. Write as 0000b.	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 2.11). =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.	

Net function = 2Eh-2Fh			
Code	**Command**	**Request, response data**	**Description**
D3h	Get Max Allowed CPU P-state/T-state	Request Byte 1:3 – Intel Manufacturer ID – 000157h, LS byte first. Byte 4 – Domain ID [0:3] – 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. [4:5] – Control Knob =00b – get max allowed CPU P-state/T-state.=01b – get max allowed logical processors. =10b – Reserved. =11b – Reserved. [6:7] – 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 2.11). =81h – Invalid Domain ID. 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 requested by Intel® ME and OSPM is not required to fulfill this request. **Note** – If any of the fields is set to FFh, it means that value is unavailable. **Note** – This command returns 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	Get Number Of P-states/T-states	Request Byte 1:3 – Intel Manufacturer ID – 000157h, LS byte first. Byte 4 – Domain ID [0:3] – 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. [4:5] – Control Knob =00b – Max allowed processor P-states/T-states. =01b – Max allowed logical processors. =10b – Reserved. =11b – Reserved. [6:7] – Reserved. Write as 00b.	This command is optional and may be unavailable on certain implementations.

Net function = 2Eh-2Fh			
Code	**Command**	**Request, response data**	**Description**
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in section 2.11). =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 will be always be 1 or more, even if BIOS will pass information that 0 P-states are supported. Byte 6 – Current Number of T-states available on the platform. This number will be always 1 or more, even if BIOS will pass information that T-states are supported. 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 – 0x000157, LS byte first. Byte 4 – Domain ID [0:3] – Domain ID (Identifies Domain which uses the defined PSU set). Currently, FW supports only one domain – Domain 0). [4:7] – Reserved. Write as 0000b. Byte 5 – PMBUS PSU address 1. Intel® NM 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. [1:7] – 7-bit PSU SMBUS address. Set to 00h if address is not used. Byte 6:12 – PMBUS PSU address 2 to PMBUS PSU address 8 encoded as in the Byte 5.	This command may override the supported set of PSUs by defining a set of all supported PSUs. Only the PSU SMBUS addresses are stored in the persistent storage. This command should be send to Intel® NM by BMC if the lack of reading from the defined PSU should be reported to the Management Console using Intel® NM Health Event. Otherwise, Intel® NM will send a notification only if all PSUs will disappear and there will be no power readings available.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in section 2.11). =81h – Invalid Domain ID. Byte 2:4 – Intel manufacturer ID – 0x000157, LS byte first.	
D8h	**Get PSU Configuration**	**Request** Byte 2:4 – Intel manufacturer ID – 0x000157, LS byte first. Byte 4 – Domain ID [0:3] – Domain ID (Identifies Domain which uses the defined PSU set). Currently, FW supports only one domain – Domain 0). [4:7] – Reserved. Write as 0000b.	
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in section 2.11). =81h – Invalid Domain ID.	

Net function = 2Eh-2Fh			
Code	**Command**	**Request, response data**	**Description**
		Byte 2:4 – Intel manufacturer ID – 0x000157, LS byte first. Byte 5 – Domain ID [0:3] – Domain ID (Identifies Domain which uses the defined PSU set). Currently, FW supports only one domain – Domain 0). [4:7] – Reserved. Return as 0000b. Byte 6 – PMBUS PSU address 1. Intel® NM will monitor the presence of the defined PSU [7:1] – 7-bit PSU SMBUS address. Set to 00h if address is not used. [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. 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 [7] = 1b – Enable PEC. [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. [5:4] – Reserved. Write as 00b. [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. [0] – Reserved. Write as 0b. Byte 5 – Target PSU Address [7:1] – 7-bit PSU SMBUS address. [0] – Reserved. Write as 0b. Byte 6 – Write Length. Byte 7 – Read Length. This filed is used to validate if the slave returns proper number of bytes. Read Length does not include PEC byte. Byte 8:M – PMBUS command.	This command sends one PMBUS command to the specified address. Address is validated against factory presets.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in section 2.11). =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.	

Net function = 2Eh-2Fh			
Code	**Command**	**Request, response data**	**Description**
		=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 7 of request. =A5h – Unsupported Write Length. =A6h – Unsupported Read Length. =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, A2h, A3h.	
EAh	**Get Host CPU data**	Request Byte 1:3 – Intel Manufacturer ID – 000157h, LS byte first. Byte 4 – Domain ID [0:3] – Domain ID (Identifies the set of processors supported by the domain. Currently FW supports only one domain – Domain 0). [4:7] – 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 2.11). =81h – Invalid Domain ID. Byte 2:4 – Intel Manufacturer ID – 000157h, LS byte first. Byte 5 – Host CPU data [7] – Set to 1 if End of POST notification was received. [6:5] – Reserved. Write as 00b. [4] – Set to 1 if Host CPU discovery data provided with that command is valid. [3] – Set to 1 if Intel® NM already activated regular power limiting policies after Host startup. [2:0] – Reserved. Write as 000b. **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	

Net function = 2Eh-2Fh			
Code	**Command**	**Request, response data**	**Description**
		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 9:16 – Processor Discovery Data for the lowest number processor in LSByte-first order. Turbo power current Limit MSR 1Ach for the lowest number processor passed by BIOS. Bytes 17:24 – Processor Discovery Data 2 for the lowest number processor in LSByte-first order. Platform Info MSR 0Ceh for the lowest number processor passed by BIOS. Byte 25 – Reserved. Write as 00000000b.	

3.3 External DCMI power management commands

DCMI provides an API for setting maximum power limit. It is assumed that this limit applies only to Platform Power Domain. For this purpose, an additional power policy is created when requested through DCMI Set Power Limit command. DCMI policy does not use suspend periods, thresholds, or failure actions. In order to distinguish it from other policies, a special status bit DCMI Specific is set in policy configuration.

This policy is visible on the IPMI interface, but it cannot be modified or changed using IPMI. Policy modification, activation, and deactivation are possible using DCMI API only. DCMI statistics are compatible with IPMI statistics. Apart from different IPMI interface, this policy does not differ from other policies and is equally treated when it comes to limiting.

The standard DCMI Power Management commands defined in [DCMI] offer functionality that is a subset of Intel® Intelligent Power Technology (see section 3.1). In DM and DNM firmware and on Romley, Brickland, and Denlow platforms also in Intel® NM firmware, the following commands are supported:

Net function = DCGRP (0x2C)			
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

Net function = DCGRP (0x2C)			
Code	**Command**	**Request, response data**	**Description**
		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 = 5. 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 [7:6] – Time duration units =00b – Seconds. =01b – Minutes. =10b – Hours. =11b – Days. [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.	Manager capabilities.
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 – Reserved. Write 00h. 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.	

Net function = DCGRP (0x2C)			
Code	**Command**	**Request, response data**	**Description**
		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 [0:5] – Reserved. [6] =1b – Power Measurement active. =0b – No Power Measurement is available. [7] – Reserved. **Note** – Sampling duration depends on Mode selection. For Mode 1 global system statistics collected over system up time are returned.	
03h	Get Power Limit	Request Byte 1 – Group Extension Identification = DCh. Byte 2:3 – Reserved for future use. Write 0000h.	
		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 and log event to SEL. =11h – Log event to SEL. 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 and log event to SEL. =11h – Log event to SEL. 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	The following defaults are used by Intel® NM during policy creation (see command "Set Intel® NM Policy"): Domain=0 Is Dcmi=TRUE Policy Trigger Type=0 Aggressive CPU Power Correction=0 Trigger Limit=0.

Net function = DCGRP (0x2C)			
Code	**Command**	**Request, response data**	**Description**
		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.	
		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 IPMI sensors implemented by Intel® NM firmware

Table 3-1 lists events and sensors implemented by Intel® NM firmware.

Table 3-1 IPMI sensors implemented by Intel® NM FW

Event	Sensor type	Event dir	Event type	Alert immediate (see section 4.5)
Intel® NM Exception Event	DCh – OEM	0 – assertion	72h – OEM	No
Intel® NM Health Event	DCh – OEM	0 – assertion 1 – deassertion	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 Platform power consumption in Sx state	DCh – OEM	N/A	N/A	N/A
PSU Status	08h – PSU	0 – assertion 1 – deassertion	6Fh – Sensor Spec	No
DCMI Power Threshold Event	09h – Power Unit	0 – assertion 1 – deassertion	05h – Generic – limit status	No
DCMI Power Off Event	12h – System Event	0 – assertion	0Ah – Generic - system state	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® NM 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 Intel® ME startup after G3 condition on Global Platform Reset (see definition in PCH EDS).

Table 3-2 Sensors Exposed by Intel® NM Firmware

Sensor #	Description	Firmware SKU availability	Reading availability	Defaults configurable in flash image tool	Notes
21	Typical Platform power consumption in Sx state	Intel® NM/ DNM/DM	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® NM uses this data to estimate the power consumed in Sx state.
24	Dynamic Power Intel® NM Exception event Sensor	Intel® NM/DNM	E	No	OEM Event only sensor. Event will be sent when maintained policy power limit exceeds the Correction Time Limit. If the policy maintained power limit exceeds multiple Correction Time Limits, then the event will be sent only once after the first Correction Time, unless "Repeat Alert Exception Action" in FITc is set to repeat the event every Correction Time until the power limit is exceeded. Events are sent no faster than every 300 ms. First occurrence of unacknowledged event will be retransmitted according to IPMB specification after ~230 ms. "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.

Sensor #	Description	Firmware SKU availability	Reading availability	Defaults configurable in flash image tool	Notes
25	Dynamic Power Intel® NM Health event Sensor	Intel® NM/DNM	E	No	OEM Event only sensor used to send events about integrity of Intel® NM policy or necessary sensor readings. Events are sent no faster than every 300 ms. First occurrence of unacknowledged event will be retransmitted according to IPMB specification after ~230 ms. "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.
26	Dynamic Power Intel® NM Operational Capabilities sensor	Intel® NM/DNM	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. Please 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. Events are sent no faster than every 300 ms. Current value of not acknowledged capabilities sensor will be retransmitted according to IPMB specification after ~230 ms.
27	Intel® NM Alert Threshold Exceeded sensor	Intel® NM/DNM	E	No	OEM Event only sensor used to send events when Intel® NM detects that a specified alert threshold for one of the policies is exceeded. Events are sent no faster than every 300 ms. First occurrence of Threshold exceeded event assertion/deassertion will be retransmitted according to IPMB specification after ~230 ms. "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.
58	DCMI Power Off Sensor	Intel® NM/ DNM/DM	A	Yes	Sensor is defined in [DCMI].
59	DCMI Power Threshold Sensor	Intel® NM/ DNM/DM	A	Yes	Sensor is defined in [DCMI].

Sensor #	Description	Firmware SKU availability	Reading availability	Defaults configurable in flash image tool	Notes
102-109	PSU#0 to PSU#7 Status	Intel® NM/ DNM/DM	A	Yes	Discrete sensor. This sensor is used for getting the Power Supply status.

3.4.1 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.

3.4.2 Intel® NM exception 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 Platform Event messages to BMC when policy limit is exceeded.

3.4.3 Intel® NM alert threshold exceeded 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.

3.4.4 Intel® NM health sensor

This OEM Event only sensor is used to send events about integrity of Intel® NM Policy or necessary sensor readings.

3.4.5 Thermal sensor on DIMM (TSOD)

This sensor is an event sensor used to send assertion and deassertion events whenever the Host SMBus attached to the Memory Controller within a CPU hangs.

Embedded HW logic periodically checks the status of this Host SMBus. If any communication issue occurs, it automatically sets a Host SMBus error status, which is constantly monitored by Intel® ME FW. If Host SMBus error status is set, Intel® ME FW will initiate Host SMBus recovery procedure (Host SMBus soft reset). Successful reset will return Host SMBus to the proper, responding state. If not, the procedure will initiate again. During the Host SMBus recovery process, the TSOD sensor event described in section 3.4.8 will be sent.

Note: For more about the recovery procedure, see the Intel® SPS Integration Guide (#451994).

3.4.6 Intel® NM operational capabilities change sensor

This OEM sensor indicates the operational capabilities of the Intel® NM service. Whenever the sensor value changes, an immediate alert is also sent. See the event description for the description of the values of the sensor.

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 wearout protection mechanism did not lock access to SPI flash.
Monitoring capability	Intel® NM is monitoring total system power consumption and CPU power and Memory power.	1. Platform is in S0/S1 state. 2. Intel® NM did not detect power reading failure conditions.
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® NM enforces nonzero CPU or memory throttling level. 3. OSPM driver is responding to requests from Intel® NM. 4. PECI interface is working properly. **Note** – When Intel® NM does actively limits 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.

3.4.7 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.

PSU Status Sensors are created only for PSUs that have responded to Intel® ME Discovery commands. This means that no events will be sent to BMC for PSUs that were never populated, even if such PSUs were configured in the xml file. However, such events will be sent for PSUs that had been removed during platform operation as once created PSU Status Sensors will remain in the SDR.

Table 3-3 shows the bits each sensor supports.

Table 3-3 PSU Status Sensors

PSU status assertion bit	Description
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).
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 Fault (STATUS_FANS_1_2).
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 status_xxx bits such as Iout OC warning, Pout OP warning (STATUS_IOUT), Vin UV warning, Iin over current warning, Pin over power warning (STATUS_INPUT), OT warning (STATUS_TEMPERATURE), Fan 1 warning, Fan 2 warning (STATUS_FANS_1_2).
AC lost	Asserted if there is no AC power input to a power supply module.
Configuration Error	Asserted for not supported PSU, when other information cannot be read.

PSU status assertion bit	Description
PSU optimization triggered	Asserted if PSU optimization feature is throttling the total system power to prevent the system failure. PSU optimization could be activated by one of the following warnings Iout OC warning (STATUS_IOUT), Vin UV warning (STATUS_INPUT) or OT warning (STATUS_TEMPERATURE). "Rearm Sensor Events" IPMI command resets CLST events (overtemperature and overcurrent) 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.

3.4.8 Event messages definition

Net function = S/E (04h)			
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® NM 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. [3] – Intel® NM 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® NM policy applies to) =00h – Entire platform. =01h – CPU subsystem. =02h – Memory subsystem. =03h – Reserved. =04h – High Power I/O subsystem. Others – Reserved. [4:7] – Reserved. Byte 7 – Event Data 3 = <Policy ID>	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. **Note** – In DNM SKU this event is logged to SEL.

Net function = S/E (04h)			
Code	**Command**	**Request, response data**	**Description**
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in section 2.11). Others – Error.	
02h	Thermal sensor on DIMM (TSOD)	**Request** Byte 1 – EvMRev =04h – IPMI2.0 format. Byte 2 – Sensor Type =28h – OEM. Byte 3 – Sensor Number =6Eh –TSOD SMBus Status. Byte 4 – Event Dir | Event Type [0:6] – Event Type =76h –OEM. [7] – Event Dir =0 – Assertion Event. =1 – Deassertion Event Byte 5 – Event Data 1 [0:3] – Status of Memory Controllers and Host SMBus for CPU0 and CPU1. [0] – CPU0 / Memory Controller 0 [1] – CPU0 / Memory Controller 1 [2] – CPU1 / Memory Controller 0 [3] – CPU1 / Memory Controller 1 =0 – Available. Host SMBus is in operation and responding. =1 – Not Available, Host SMBus hung and does not respond. [4:7] – Status of Memory Controllers and Host SMBus for CPU2 and CPU3 if are present. [4] – CPU2 / Memory Controller 0 [5] – CPU2 / Memory Controller 1 [6] – CPU3 / Memory Controller 0 [7] – CPU3 / Memory Controller 1 =0 – Available. Host SMBus is in operation and responding. =1 – Not Available, Host SMBus hung and does not respond. Byte 6 – Event Data 2 =FFh – or not present if unspecified. Byte 7 – Event Data 3 =FFh – or not present if unspecified.	NOTE: Each CPU has two separate memory controllers. DIMMs are connected to the CPU by Host SMBus. In some cases, Host SMBus can hang. When it does, recovery process of Host SMBus is initiated and followed by this event. Intel® ME FW will send the event with assertion mask plus specific set of Byte 5 when Host SMBus within CPU hangs and Intel® ME FW will initiate recovery process of this bus. It depends on which MC hung. Intel® ME FW will send the event with deassertion mask plus specific set of Byte 5 when Host SMBus within the CPU is restored and responding. It depends on which MC hung.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in section 2.11). Others – Error.	

Net function = S/E (04h)			
Code	**Command**	**Request, response data**	**Description**
16h	Alert Immediate Message Intel® NM Health Event	**Request** Bytes 1:3 – As set in the "Set Intel® NM 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® NM Health sensor. Byte 8 – Event Dir | Event Type [0:6] – Event Type =73h – OEM. [7] – Event Dir =0 – Assertion Event. =1 – Deassertion Event Byte 9 – Event Data 1 [0:3] – Health Event Type =02h – Sensor Intel® NM. [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. [4:7] – Error Type 0 – 9 – Reserved. =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 Intel® NM Policy" command Byte 7 bit [1]. 16 – Reserved. Byte 11 – Event Data 3 For Error Type = 10 or 15 – <Policy ID>. For Error Type = 11 <PowerSensorAddress>. For Error Type = 12 <Inlet Sensor Address>. **Response** Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in section 2.11). Others – Error.	This message provides a run-time error indication about Intel® NM'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. 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. The assertion of 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 a domain. In this case Event Data 3 byte is set to zero. In case of lack of readings from all sensors event with Date Event 3 byte equal to zero is asserted and deasserted only if the platform is in S0 power state. First occurrence of not acknowledged event will be retransmitted no faster than every 300 milliseconds. **Note** – In DNM SKU this message is logged to SEL.

Net function = S/E (04h)			
Code	**Command**	**Request, response data**	**Description**
16h	Alert Immediate Message Intel® NM Operational Capabilities Change	**Request** Bytes 1:3 – As set in the "Set Intel® NM 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 =26 – Intel® NM Operational Capabilities sensor. Byte 8 – Event Dir | Event Type [0:6] – Event Type =74h – OEM. [7] – Event Dir =0 – Assertion Event. =1 – Deassertion Event. Byte 9 – Event Data 1 [0:2] – Current state of Operational Capabilities. The same value is also returned by the Get Sensor Reading command invoked for 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. [3:7] – Reserved. **Response** Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in section 2.11). Others – Error.	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 3.4.6 for detailed description of the conditions triggering changes of the capabilities bitmask. Current value of not acknowledged capabilities sensor will be retransmitted no faster than every 300 milliseconds. **Note** – In DNM SKU this message is logged to SEL.
16h	Alert Immediate Message Intel® NM Alert Threshold Exceeded	**Request** Bytes 1:3 – As set in the "Set Intel® NM 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 =27 – Intel® NM Event Sensor. Byte 8 – Event Dir | Event Type [0:6] – Event Type =72h – OEM. [7] – Event Dir =0 – Assertion Event.	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 ms. **Note** – In DNM SKU this

Net function = S/E (04h)			
Code	**Command**	**Request, response data**	**Description**
		=1 – Deassertion Event. Byte 9 – Event Data 1 [0:1] – Threshold Number =0 – 2 – Threshold index. [2] – Reserved. [3] – Intel® NM 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. [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 (Identifies the domain that this Intel® NM policy applies to) =00h – Entire platform. =01h – CPU subsystem. =02h – Memory subsystem. =03h – Reserved. =04h – High Power I/O subsystem. Others – Reserved [4:7] – Reserved. Byte 11 – Event Data 3 – <Policy ID>.	message is logged to SEL.
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in section 2.11). 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 [7:6] =00b – Unspecified byte 2. [5:4] =00b – Unspecified byte 3. [3:0] – Offset from Event/Reading Code for discrete event state. (Assertion and Deassertion refer to Byte 4, bit [7] - Event Dir.)=0 – Presence detected Assertion:Power supply module is present. Deassertion: Power supply module is not present. =1 – Power supply failure Assertion: No Power Supply failure detected. Deassertion: One of faults detected: IOut OC Fault, Pout OP fault (STATUS_IOUT), OT fault	Event will be sent each time when monitored PSU Status changes. First occurrence of not acknowledged event will be retransmitted no faster than every 300 milliseconds. "Rearm 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 DNM SKU, this event is logged to SEL.

Net function = S/E (04h)			
Code	**Command**	**Request, response data**	**Description**
		(STATUS_TEMPERATURE), Fan Fault (STATUS_FANS_1_2). =2 – Predictive failure Assertion: No Power Supply warning detected. Deassertion: One of warnings detected: Iout OC warning, Pout OP warning (STATUS_IOUT), Vin UV warning, Iin over current warning, Pin over power warning (STATUS_INPUT), OT warning (STATUS_TEMPERATURE), Fan 1 warning, Fan 2 warning (STATUS_FANS_1_2). =3 – AC Lost Assertion: AC Power OK. Deassertion: No AC power input to a power supply module (Unit Off For Low Input Voltage bit in STATUS_INPUT is set). =4 – Configuration Error Assertion: PSU configuration OK. Deassertion: – Not supported PSU, when other information cannot be read (PSU not responded for PMBus requests). =5 – PSU optimization feature triggered. Throttling to the system power applied to prevent the system failure. PSU optimization could be activated by one of the following warnings: Iout OC warning (STATUS_IOUT), Vin UV warning (STATUS_INPUT) or OT warning (STATUS_TEMPERATURE) Assertion: PSU optimization feature activated to prevent the system failure. Deassertion: No Power Supply warning detected. =6--15 – Reserved. Byte 6 – Event Data 2 [7:4] - Optional offset from 'Severity' Event/Reading Code. (0Fh if unspecified). [3:0] - Optional offset from Event/Reading Type Code for previous discrete event state. (0Fh if unspecified.) Byte 7 – Event Data 3 Optional OEM code. FFh or not present if unspecified.	
		Response Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in section 2.11). Others – Error.	

3.5 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, Intel® ME expects BMC at address 10h in 7-bit format (20h in 8-bit). This address is fixed.

To support initialization of Intel® Management Engine owned sensors based on associated SDRs, the 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 the 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 platform reset or startup, Intel® NM will activate "Time After Platform Reset" policy.

4.1 IPMI device "global" commands

Net function = Chassis (00h)			
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 a 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 2.11).	

4.2 Sensor device commands

Net function = S/E (04h)			
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 2.6.4 and 3.4.8

4.3 Alert immediate command

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

Net function = S/E (04h)			
Code	**Command**	**Request, response data**	**Description**
16h	Alert Immediate	For command description see [IPMI]	This is a 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.

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 Platform Power consumption. Single power reading, subtotal of per-rail readings from one PSU as well as total of single run of readings across all attached PSUs 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, the 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 Intel® ME 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 Intel® ME firmware. BMC should average the power over 1 sec to allow Intel® ME firmware to read the power two to ten times per second[3]. 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:

OEM command	Description	Encoding
XXh (default E2h[4])	'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.

3 *The OEM should define the frequency of power readings (10, 5, or 2 per second) 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.*

4 *OEM command code value may be different for Power Consumption reading.*

4.4.2 Inlet air temperature readings

Depending on the configuration Intel® ME firmware may use one of the following sources for inlet temperature readings:

1. On Romley and Bromolow: SST sensors attached directly to PCH. In that case, there is no need to implement any support in the BMC.
2. On all platforms: BMC sensor read by Intel® ME 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 Intel® ME firmware. Additionally, the OEM command code may be configured using factory presets:

OEM command	Description	Encoding
XXh (default E2h[5])	'OEM Get Reading' with type "Inlet Air Temperature"	The value of the sensor returned in "Get Inlet Air Temperature" is encoded on 16-bit encoding 2s-complement signed integer. Values below -128 degrees centigrade and above +127 degrees centigrade are ignored and treated as a temperature reading failure.

Note: OEM command code value may be different for Inlet Air Temperature readings and Power Consumption reading)

1. With "Get Sensor Reading" IPMI command Intel® ME firmware expects the sensor reading to be temperature in range of 0 to 100 centigrade. Sensor readings outside of the specified range are ignored and treated as a temperature reading failure.
2. Inlet Air Temperature reading can be disabled. In that case, the inlet temperature statistics and Intel® NM Policies using temperature trigger will be not available.

4.4.3 OEM management engine power state change

Optionally, instead of the event, Intel® SPS firmware may send an IPMI command with information as defined above. Using factory presets, the OEM may use an event and/or OEM command.

OEM command	Description	Encoding
XXh (default 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.

5 *OEM command code value may be different for Inlet Air Temperature readings.*

4.4.4 OEM command definition

Net function = SDK general application (0x30)			
Code	**Command**	**Request, response data**	**Description**
E2h	OEM Get Reading	Request Byte 1 – Domain ID/Reading Type [0:3] – Domain ID. Currently only domain 0 will be queried. Other domains are queried internally by Intel® ME using. [4:7] – Reading Type =00h – Platform Power Consumption. For platform power consumption the Domain ID will be set to 0. Values from 1 to 15 could be used to address power rails. Per-rail readings are optional and firmware needs to be preconfigured in the factory settings. =01h – Inlet Air Temperature. For Inlet Air Temperature the Domain ID will be set to 0. =03h – 0Fh – Reserved. Byte 2:3 – Reserved. Write as 0000h.	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 2.11). Byte 2 – Reading Type [0:3] – Domain ID copied from request (depending on the Byte 2 identifies the processor which should be queried for I_{CC_TDC} or 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 – I_{CC_TDC} reading from PECI. Values below 0 and above 255 will be ignored and treated as a PECI reading failure. =03h – FFh – Reserved. Byte 3:4 – Reading value 16-bit encoding 2s-complement signed integer.	
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 2.11).	

4.4.5 Summary of options

Summary of implementation options		
Functionality	**Type**	**Description**
See section 4.4.1.	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.
See section 4.4.2.	Factory presets	Required if inlet temperature statistics and Intel® NM Policies using temperature trigger will be exposed by the platform.
See section 4.4.3.	Factory presets	Intel® ME firmware is able to send notification about the power state change using a 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[6].

The first step in the Intel® NM 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[7]. 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 subtype field in them as the first byte of the OEM data indicates the type of record following.

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

7 *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
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
	10	Intel® NM Alert Threshold Exceeded sensor	=27

4.6 Alerts

Events mentioned in section 3.4 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.

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 Node Manager Alert Destination" OEM command. Section 3.1 contains the description of this OEM command.

Setting alert destination using "Set 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 later.

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

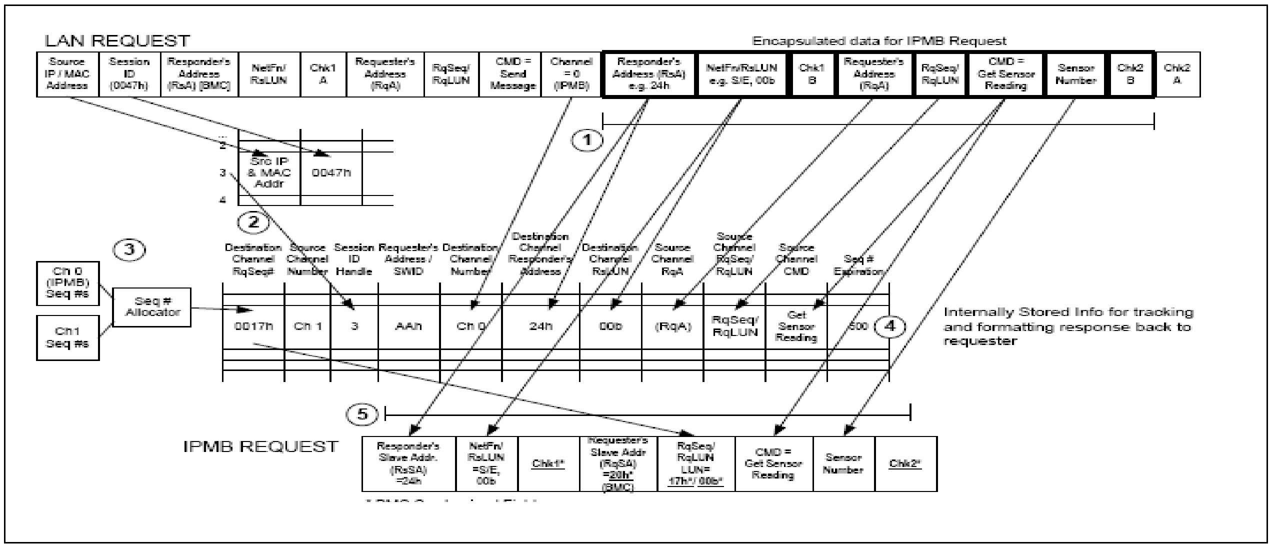
If Intel® NM is implemented on BMC, then external SW will send the commands directly addressed to BMC (no bridging).

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 that is being bridged to.

This will be in the form an 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. Please 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
Ch 0 (IPMB) Seq #s
Ch1 Seq #s
Seq # Allocator
3
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 below 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).

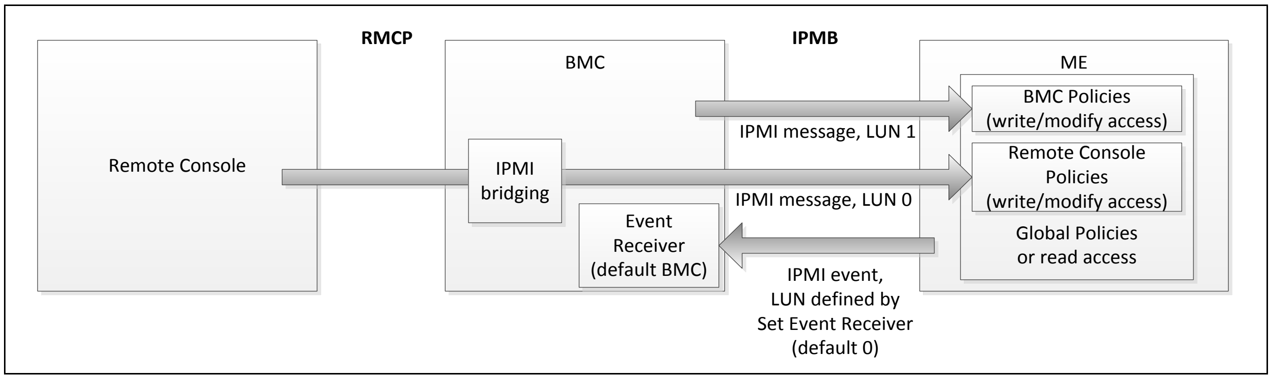
In addition, the actual medium to communicate the bridged packet on IPMB is also needed. This is done as part of its initial discovery by query of the BMC for channel protocol using the IPMI command "Get Channel Info". For example, if Intel® NM is implemented in the Intel® ME (satellite controller), 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.

4.8 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 Intel® NM Policy Suspend Periods
- Reset 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® NM 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 LUN. If they match, the command is processed normally. If they do not match, the command will not be executed and error code D4h – Insufficient Privilege Level – is returned.
- When a request doesn't modify the policy, it is executed as if the LUN from the request matches the LUN stored with the policy. The 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 match the one stored with the policy. This bit is cleared when the LUNs match.

The LUN numbers are used only for verification of the policy owner. These LUN numbers do not influence the number of available policies or triggers, nor do the LUN numbers influence the way policies or triggers 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 SSIF, IPMB, and IPMI over RMCP+.

4.9 IPMB reset scenarios

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

4.10 MIC connectivity

4.10.1 MIC reverse proxy

In order for the Intel® NM to communicate with MICs it is required that the BMC implements the reverse PCIe*-SMB proxy.

Normally the PCIe* SMBus signals from each slot on a baseboard form a single bus. This prevents multiple PCIe* devices from using the same address. Brickland BMC platforms should implement a SMBus mux between the BMC and several of the slots allowing communication with each device independent of the cards in other slots. The BMC must be aware of this mux and prevent collisions in usage.

The following two commands support IPMB and generic I2C transactions.

Net function = SDK general application (0x3e)			
Code	**Command**	**Request, response data**	**Description**
51h	Slot IPMB	**Request** Byte 1 [7:6] – Address Type =00b – Bus/Slot/Address =01b – Reserved for Unique identifier [5:4] Reserve [3:0] Bus# Set to 0 for "Address Type" is different from "Bus/Slot/Address" Byte 2 – Identifier/Slave-address. This byte holds either the unique ID or the slave address (8 bit "write" address), dependent on the "Address Type" field. Byte 3 – Net Function Byte 4 – Command Byte 5:n – Command Data (optional)	This command may be sent at any time. It will block until a response is received or an IPMB timeout occurs. If a SlotIPMB command is already in progress, a subsequent command will block until the first has released control of the mux .
		Response Byte 1 – Completion Code =0x00 – Normal =0Xc1 – Command not supported on this platform. =0xc7 – Command data invalid length. =0xc9 – Parameter not implemented or supported. =0x82 – Bus error. =0x85 – Invalid PCIe* slot number. Byte 2:n – Response Data	

Net function = SDK general application (0x3e)			
Code	**Command**	**Request, response data**	**Description**
52h	Slot I2C Master Write Read	Request Byte 1 [7:6] – Address Type =00b – Bus/Slot/Address =01b – Reserved for Unique identifier [5:4] Reserve [3:0] Bus# Set to 0 for "Address Type" is different from "Bus/Slot/Address" Byte 2 – Slot # =0x00 – if "Address Type" is different from "Bus/Slot/Address" =0xfe – Slot not specified Byte 3 – Identifier/Slave-address. This byte holds either the unique ID or the slave address (8 bit "write" address), Byte 4 – Number of bytes to read Byte 5:n – Data to write (optional)	This command may be sent at any time. It will block until all data has been written and read, or a bus error or timeout has occurred. If a SlotI2CMasterWriteRead command is already in progress, a subsequent command will block until the first has released control of the mux .
		Response Byte 1 – Completion Code =0x00 – Normal =0Xc1 – Command not supported on this platform. =0xc7 – Command data invalid length. =0xc9 – Parameter not implemented or supported. =0x82 – Bus error. =0x83 – NAK on write. =0x84 – Truncated read. =0x85 – Invalid PCIe* slot number. Byte 2:n – Data Read	

4.10.2 MIC discovery

This section describes IPMI commands that Intel® ME FW sends to the BMC during MIC discovery process. The BMC is required to support these commands and, upon request from Intel® ME FW, to provide an appropriate response as defined in below.

Net function = SDK general application (0x30)			
Code	**Command**	**Request, response data**	**Description**
E8h	Get MIC Card Info	Byte 1 - Card instance (1-based) for which information is requested. If this byte is zero only the total number of cards detected will be returned.	This command returns information about management-capable PCIe* cards that are discovered by the BMC, including protocol support and addressing information that can be used in MIC IPMB Request command. Note: E8h is the default value; it may be configured in FITC.
		Byte 1 – Completion Code =00h – Success (Remaining standard Completion Codes are shown in section 2.11.) =CBh "Requested sensor, data, or record not present" the requested card instance is greater than the number of cards detected. The following bytes are only returned if there are any management-capable cards detected by the Intel® ME. Byte 2 – Total number of MIC devices detected. The following bytes are only returned if the specified management-capable card is detected by the BMC. Byte 3 – Command Protocol Detection Support	

Net function = SDK general application (0x30)			
Code	Command	Request, response data	Description
		[7:4] – Reserved [3] - MCTP over SMBus [2] – IPMI on PCIe* SMBus (refer to IPMI 2.0 spec) [1] - IPMB [0] – Unknown A value of 1b indicates detection of a protocol is supported. Support for detection of specific protocols is OEM specific. NOTE: Intel® ME firmware for Brickland only support detection of IPMB Byte 4 – Command Protocols Supported by Card [7:4] – Reserved [3] – MCTP over SMBus [2] – IPMI on PCIe SMBus [1] - IPMB [0] – Unknown Byte 5 – Address/Protocol/Bus# [7:6] Address Type 00b – Bus/Slot/Address Other values reserved [3:0] Bus Number – Identifies SMBus interface on which the MIC device was detected. Byte 6 - Slot Number – Identifies PCIe* slot in which the MIC device is inserted. Byte 7 - Slave Address - The I2C slave address (8-bit “write” address) of the MIC device.	

§

5. IPMI OEM Command Summary

Table 5-1 summarizes the OEM commands used by Intel® NM firmware for Intel® ME-BMC communication. All the commands use Network Function 2Eh/2Fh and LUN 00b with Intel IANA Number 000157h.

Table 5-1 IPMI OEM commands

NET function	Code	Command name	Implemented by	SKU
2Eh + 000157h	A0-A8h	Online Firmware Update Commands	Intel® ME	SiEn/Intel® NM
2Eh + 000157h	C0-CDh	Intel® NM Commands	Intel® ME	Intel® NM/DNM
2Eh + 000157h	CEh	Set Node Manager Alert Destination	Intel® ME	Intel® NM
2Eh + 000157h	CFh	Get Node Manager Alert Destination	Intel® ME	Intel® NM
2Eh + 000157h	D0h	Set Total Power Budget	Intel® ME	Intel® NM
2Eh + 000157h	D1h	Get Total Power Budget	Intel® ME	Intel® NM
2Eh + 000157h	D2h	Set Max Allowed CPU P-state/T-state	Intel® ME	Intel® NM
2Eh + 000157h	D3h	Get Max Allowed CPU P-state/T-state	Intel® ME	Intel® NM
2Eh + 000157h	D4h	Get Number Of P-states/T-states	Intel® ME	Intel® NM
2Eh + 000157h	D7h	Set PSU Configuration	Intel® ME	Intel® NM/DNM
2Eh + 000157h	D8h	Get PSU Configuration	Intel® ME	Intel® NM/DNM
2Eh + 000157h	D9h	Send Raw PMBus	Intel® ME	Intel® NM
2Eh + 000157h	DCh	Set ME Power State	Intel® ME	SiEn/Intel® NM
2Eh + 000157h	40h-5Fh	PECI Proxy and PECI related commands in Intel® NM 2.0	Intel® ME/BMC	SiEn/Intel® NM
2Eh + 000157h	DFh	Force ME Recovery	Intel® ME	SiEn/Intel® NM/DM/DNM
2Eh + 000157h	E0h	Get ME Factory Presets Signature	Intel® ME	SiEn/Intel® NM/DM/DNM
2Eh + 000157h	E2-E3h	MIC Proxy commands	Intel® ME/BMC	SiEn/Intel® NM
2Eh + 000157h	EAh	Get Host CPU Data	Intel® ME	Intel® NM/DNM
2Ch (DCGRP) + DCh	01-05h	DCMI Power management Commands	Intel® ME	DNM/DM/ on Romley/Denlow/ Brickland Intel® NM
30h (SDK General Application)	A0-A7h	Online Firmware Update Commands (backward compatibility with Intel® NM 1.5)	Intel® ME	SiEn/Intel® NM
30h (SDK General Application)	XXh	OEM Management Engine Power State Change	Intel® ME	SiEn/Intel® NM

For a list of commands supported by recovery boot loader, see section 2.8.

§

A. Intel® NM 2.0 Platform Changes

This appendix describes changes between Intel® Intelligent Power Node Manager External Interface Specification using IPMI for Intel® NM 1.5 and Intel® NM 2.0.

A.1 IPMI commands not supported on Intel® NM 2.0 platform

Table A-2 presents IPMI commands not supported by Intel® NM 2.0 firmware in comparison with Intel® NM 1.5 implementations on Tylersburg-EP and Boxboro-EX. Intel® NM responds to the commands with C1h – Invalid Command Completion Code.

Table A-2 Supported IPMI Commands

NET function	Command code	Command name	Implemented by Intel® ME or BMC
30h SDK General App	DAh	Set Cooling Coefficient	Intel® ME
2Eh + 000157h (Intel IANA)	D6h	Set Host CPU data	Intel® ME

A.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.

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 Alert Threshold Exceeded sensor	DCh – OEM	0 – Assertion 1 – Deassertion	72h – OEM	Alert Immediate
Intel® NM Health Event	DCh – OEM	0 – Assertion 1 – Deassertion	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

A.3 IPMI sensors implemented by Intel® NM FW

List of sensors implemented by the Intel® ME firmware has changed see section 2.10 IPMI OEM PECI Proxy Sensors for details.

Average CPU Temperature sensors are not implemented.

A.4 Event generation control

Table A-3 summarizes how event generation can be enabled/disabled for each sensor implemented by Intel® NM firmware.

Table A-3 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 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 Node Manager Alert Destination – clearing alert destination parameters 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 Sensor Intel® NM Alert Threshold Exceeded Sensor	Alert Immediate	Event generation can be enabled/disabled using the IPMI command: Set Node Manager Alert Destination – clearing alert destination parameters 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: • 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.

§