Tools for Intel® Server Platform Services Firmware 3.0 E5

User Guide

February 2014

Revision: 1.01

Intel Confidential

I2C is a two-wire communications bus/protocol developed by Philips. SMBus is a subset of the I2C bus/protocol and was developed by Intel. Implementations of the I2C bus/protocol may require licenses from various entities, including Philips Electronics N.V. and North American Philips Corporation.

Intel® Active Management Technology requires the computer system to have an Intel® AMT-enabled chipset, network hardware and software, as well as connection with a power source and a corporate network connection. Setup requires configuration by the purchaser and may require scripting with the management console or further integration into existing security frameworks to enable certain functionality. It may also require modifications of implementation of new business processes. With regard to notebooks, Intel® AMT may not be available or certain capabilities may be limited over a host OS-based VPN or when connecting wirelessly, on battery power, sleeping, hibernating or powered off. For more information, see www.intel.com/technology/platform-technology/intel-amt/

Systems using Client Initiated Remote Access require wired LAN connectivity and may not be available in public hot spots or "click to accept" locations.

Code names featured are used internally within Intel to identify products that are in development and not yet publicly announced for release. Customers, licensees and other third parties are not authorized by Intel to use code names in advertising, promotion or marketing of any product or services and any such use of Intel's internal code names is at the sole risk of the user.

Contents

Figures

Tables

Revision History

Revision Number	Description	Revision Date
0.5	Initial draft	January 2013
0.9	Update MESDC, SpsINFO, spsFITC section	April 2013
0.91	Add EFI application notes	May 2013
0.92	Update screenshot with Alpha FW	June 2013
0.95	Update implementation for Beta FW (FITC, MESDC, spsMANUF)	September 2013
1.0	Update MESDC compliance tests section	February 2014
1.01	Dengate removal	February 2014

1 Introduction

The purpose of this document is to describe the tools that are used in the platform design, manufacturing, testing, and validation process.

1.1 Terminology

Acronym/Term	Definition
3PDS	3rd Party Data Storage
AC	Alternating Current
Agent	Software that runs on a client PC with OS running
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
BBBS	BIOS Boot Block Size
BIN	Binary file
BIOS	Basic Input Output System
BIOS-FW	Basic Input Output System Firmware
CLI	Command Line Interface
CPU	Central Processing Unit
CRB	Customer Reference Board
DHCP	Dynamic Host Configuration Protocol
DIMM	Dual In-line Memory Module
DLL	Dynamic Link Library
DNS	Domain Naming System
EC	Embedded Controller
EEPROM	Electrically Erasable Programmable Read Only Memory
EHCI	Enhanced Host Controller Interface
EID	Endpoint ID
End User	The person who uses the computer (either Desktop or Mobile). In corporate, the user usually does not have administrator privileges. The end user may not be aware to the fact that the platform is managed by Intel® AMT.
EOP	End Of Post
FCIM	Full Clock Integrated Mode
FCSS	Flex Clock Source Select
FDI	Flexible Display Interface

Acronym/Term	Definition
FITC	Flash Image Tool
FLOCKDN	Flash Configuration Lock-Down
FMBA	Flash Master Base Address
FPSBA	Flash PCH Strap Base Address
FPT	Flash Programming Tool
FPTW	Flash Programming Tool Window
FQDN	Fully Qualified Domain Name
FRBA	Flash Region Base Address
FW	Firmware
FWUpdate	Firmware Update
G3	A system state of Mechanical Off where all power is disconnected from the system. A G3 power state does not necessarily indicate that RTC power is removed.
GbE	Gigabit Ethernet
GMCH	Graphics and Memory Controller Hub
GPIO	General Purpose Input/Output
GUI	Graphical User Interface
GUID	Globally Unique Identifier
HECI (deprecated)	Host Embedded Controller Interface
Host or Host CPU	The processor running the operating system. This is different than the management processor running the Intel® ME FW.
Host Service/ Application	An application running on the host CPU
HostIF	Host Interface
HTTP	HyperText Transfer Protocol
HW	Hardware
IBEN	Input Buffer Enable
IBV	Independent BIOS Vendor
ICC	Integrated Clock Configuration
ID	Identification
IDER	Integrated Drive Electronics Redirection
INF	An information file (.inf) used by Microsoft operating systems that support the Plug & Play feature. When installing a driver, this file provides the OS with the necessary information about driver filenames, driver components, and supported hardware.
Intel® ME	Intel® Management Engine. The embedded processor residing in the chipset GMCH.
Intel® MEI	Intel® Management Engine Interface (renamed from HECI). The interface between the Intel® Management Engine and the Host system.

Acronym/Term	Definition
Intel® NM	Intel® Node Manager
spsINFO	Intel® ME information tool
spsInfoWin	Windows* version of MEINFO
spsManuf	spsManuf validates Intel® ME functionality on the manufacturing line
spsManufWin	Windows version of spsManuf
ISV	Independent Software Vendor
IT User	Information Technology User. Typically very technical and uses a management console to ensure multiple PCs on a network function.
JEDECID	Joint Electronic Device Engineering Councils ID. Standard Manufacturer's Identification Code that is assigned, maintained and updated by the JEDEC office
JTAG	Joint Test Action Group
KVM	Keyboard, Video, Mouse
LAN	Local Area Network
LED	Light Emitting Diode
LPC	Low Pin Count Bus
M0	Intel® ME power state where all HW power planes are activated. Host power state is S0.
M3	Intel® ME power state where all HW power planes are activated but the host power state is different than S0. (Some host power planes are not activated.) The Host PCIe* interface is unavailable to the host SW. The main memory is not available for Intel® ME use.
M-Off	No power is applied to the management processor subsystem. Intel® ME is shut down.
MAC address	Media Access Control address
NVM	Non-Volatile Memory
NVRAM	Non-Volatile Random Access Memory
OCKEN	Output Clock Enable
ODM	Original Device Manufacturer
OEM	Original Equipment Manufacturer
OEM ID	Original Equipment Manufacturer Identification
OOB	Out Of Band
OOB interface.	Out Of Band interface. An SOAP/XML interface over secure or non-secure TCP protocol.
OS	Operating System
OS Hibernate	OS state where the OS state is saved on the hard drive.
OS not Functional	The Host OS is considered non-functional in Sx power state in any one of the following cases when the system is in S0 power state: OS is hung After PCI reset OS watch dog expires OS is not present

Acronym/Term	**Definition**
OVR	Override
PC	Personal Computer
PCH	Platform Controller Hub
PCI	Peripheral Component Interconnect
PCIe*	Peripheral Component Interconnect Express
PDR	Platform Descriptor Region
PHY	Physical Layer
PID	Provisioning ID
PKI	Public Key Infrastructure
PM	Power Management
PRTC	Protected Real Time Clock
PSK	Pre-Shared Key
PSL	PCH Strap Length
RNG	Random Number Generator
ROM	Read Only Memory
RSA	A public key encryption method
RTC	Real Time Clock
S0	A system state where power is applied to all HW devices and the 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 (memory is in self refresh).
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 still connected.
SDK	Software Development Kit
SHA	Secure Hash Algorithm
SMBus	System Management Bus
SOL	Serial over LAN
SPI	Serial Peripheral Interface
SPI Flash	Serial Peripheral Interface Flash
Standby	OS state where the OS state is saved in memory and resumed from the memory when the mouse/keyboard is clicked.
Sx	All S states which are different than S0
SW	Software
System States	Operating System power states such as S0, S1, S2, S3, S4, and S5.
TCP/IP	Transmission Control Protocol/Internet Protocol
TLS	Transport Layer Security

Acronym/Term	Definition
UI	User Interface
UMA	Unified Memory Access
Un-configured state	The state of the Intel® ME FW when it leaves the OEM factory. At this stage the Intel® ME FW is not functional and must be configured.
UNS	User Notification Services
USB	Universal Serial Bus
USBr	Universal Serial Bus Redirection
VLAN	Virtual Local Area Network
VSCC	Vendor Specific Component Capabilities
Windows* PE	Windows* Preinstallation Environment
WIP	Work in Progress
XML	Extensible Markup Language. Intel® AMT's XML-based protocol has three parts: An envelope that defines a framework for describing what is in a message and how to process it A set of encoding rules for expressing instances of application-defined data types A convention for representing remote procedure calls and responses

1.2 Reference Documents

Document	Document No./Location
Intel® Server Platform Services Firmware Integration Guide	451994/CDI
Wellsburg Platform Controller Hub (PCH) SPI Programming Guide	516552/CDI
Wellsburg Platform Controller Hub (PCH) External Design Specification (EDS)	511555/CDI

2 Preface

2.1 Overview

This document covers the system tools used for creating, modifying, and writing binary image files, manufacturing testing, Intel® Management Engine (Intel$^{®}$ ME) setting information gathering, and debugging. The tools are located in **Kit directory\Tools**.

2.2 Operating System Support

Table 2-1. OS Support for Tools

Intel® ME and Manufacturing Tools	UEFI	FreeDOS	Windows XP	Windows 7 x86/x64	Windows 8 x64	Windows Server 2008 R2 SP1 x64	Windows Server 2012 x64	Windows PE x64 based on Windows 7	Linux RHEL 6.4 x86/x64	Linux SLES 11 SP3 x86/x64	Linux Fedora 18 x64
spsFITC			X	X	X	X					
MESDC			X	X	X	X	X				
spsMANUF	X	X				X	X	X	X	X	X
spsINFO	X	X			X	X	X	X	X	X	X
spsFPT	X	X				X	X	X			
SUT						X	X				

2.3 Error Return

Tools always return 0/1 for the error level (0 = success, 1= error). A detail error code is displayed on the screen and stored on an error.log file in the same directory as the tools. (See Appendix A for a list of these error codes.)

2.4 Running application under EFI

There is limitation on EFI application with assertion linked with empty current working directory. To avoid this kind of EFI assertion, user should either make sure already mounted device is used as current working directory (usually it's fs0, fs1,…), i.e.:

```
> fsx:
fsx:\> cd directory_with_sps_tools_package
…
```

or mount a block device first like below:

```
> mount blkx drive_name
> drive_name:
drive_name:\> cd dir_with_sps_tools_package
…
```

- also without naming the mounted file system:

```
> mount blkx
> blkx:
blkx:\> cd dir_with_sps_tools_package
...
```

where:

x – storage device id number as mapped by EFI

to see all mapped devices *map* command should be used.

2.5 Usage of the Double-Quote Character (")

The command shell used to invoke tools in both DOS and Windows* has a built-in CLI.

The command shell is intended to be used for invoking applications as well as running in batch mode and performing basic system and file operations. For this reason, the CLI has special characters that perform additional processing upon command.

The double-quote is the only character which needs special consideration as input. The various quoting mechanisms are the backslash escape character (/), single-quotes ('), and double-quotes ("). A common issue encountered with this is the need to have a double-quote as part of the input string rather than using a double-quote to define the beginning and end of a string with spaces.

For example, you may want these words – one two – to be entered as a single string for a vector instead of dividing it into two strings ("one", "two"). In that case, the entry – including the space between the words – must begin and end with double-quotes ("one two") in order to define this as a single string.

When double-quotes are used in this way in the CLI, they define the string to be passed to a vector, but are NOT included as part of the vector. The issue encountered with this is how to have the double-quote character included as part of the vector as well as bypassed during the initial processing of the string by the CLI. This can be resolved by preceding the double-quote character with a backslash (\").

For example, if you want these words to be input – input"string – the command line is: input\"string.

2.6 PMX Driver Limitation

Several tools (spsINFO, spsMANUF, and spsFPT) use the PMX library to get access to the PCI device. Only one tool can get access to the PMX library at a time because of library limitation. Therefore, running multiple tools to get access to PMX library will result in an error (failure to load driver).

The PMX driver is not designed to work with the latest Windows driver model (it does not conform to the new driver's API architecture).

In Windows* 7, the verifier sits in kernel mode, performing continual checks or making calls to selected driver APIs with simulations of well-known driver related issues.

Warning: Running the PMX driver with the Windows 7 driver verifier turned on causes the OS to crash. Do not include PMX as part of the verifier driver list if you are running Windows 7 with the driver verifier turned on.

3 Flash Image Tool

The Flash Image Tool (**spsFITC.exe**) creates and configures a complete SPI image file for platforms in the following way:

spsFITC creates and allows configuration of the Flash Descriptor Region, which contains configuration information for platform hardware and FW.

spsFITC assembles the following into a single SPI flash image:

Binary files of the following regions:

- BIOS
- Intel integrated LAN (GbE)
- Intel ME
- Platform Descriptor Region
- Device Extension Region
- The Flash Descriptor Region created by spsFITC

You can manipulate the completed SPI image via a GUI and change the various chipset parameters to match the target hardware. Various configurations can be saved to independent files, so you don't have to recreate a new image each time. Use of GUI is strongly advised while changing the settings as it uses internal logic to validate and cross-reference the new settings.

spsFITC supports a set of command line parameters that can be used to build an image from the CLI or from a batch file. When a previously stored configuration is used to define the image layout, you don't have to interact with the GUI.

Note: spsFITC only generates a complete SPI image file; it does not program the flash device. This complete SPI image must be programmed into the flash with spsFPT, any third-party flash burning tool, or some other flash programmer device.

3.1 System Requirements

spsFITC runs on Windows XP, Windows 7 x86/x64, Windows 8 x64, and Windows 2008 R2 SP1 x64. The tool does not have to run on an Intel ME-enabled system.

3.2 Flash Image Details

A flash image can be composed of six regions. The locations of these regions are referred to in terms of where they can be found within the total memory of the flash.

Figure 3-1. SPI Flash Image Regions

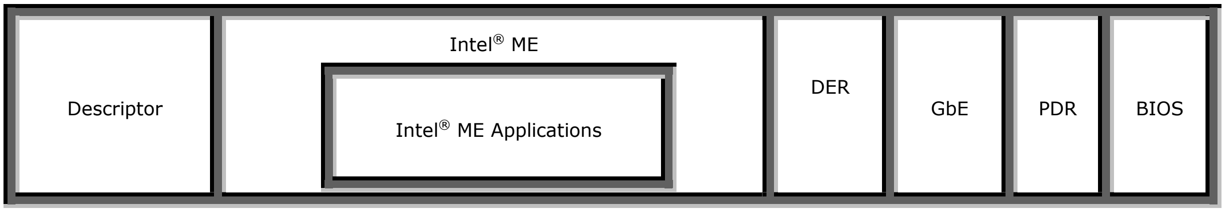


Table 3-1. Flash Image Regions – Description

Region	Description
Descriptor	This region contains information such as the space allocated for each region of the flash image, read-write permissions for each region, and a space which can be used for vendor-specific data. It takes up a fixed amount of space at the beginning of the flash memory. **Note:** This region MUST be locked before the serial flash device is shipped to end users. Please see section 3.3.4.3 for more information. Failure to lock the Descriptor Region leaves the Intel® ME device vulnerable to security attacks. This region is mandatory and enabled by default.
GbE	This region contains code and configuration data for an Intel Integrated LAN (Gigabit Ethernet). This region is not mandatory and disabled by default.
DER	Device Extension Region used by Intel Node Manager-PTU
PDR	This region lets system manufacturers describe custom features for the platform. This region is not mandatory and disabled by default.
Intel® ME	This region contains code and configuration data for Intel® ME applications. It takes up a variable amount of space up to the BIOS region. This region is mandatory and enabled by default.
BIOS	This region contains code and configuration data for the entire computer. This region is not mandatory and disabled by default.

3.2.1 Flash Space Allocation

Space allocation for each region is determined as follows:

1. Each region can be assigned a fixed amount of space. If a region is not assigned a fixed amount of space, it occupies only as much space as it requires.
2. If there is still space left in the flash after allocating space to all of the regions, the Intel ME region expands to fill the remaining space.

3.3 Required Files

The spsFITC main executable is **spsFITC.exe**. The following files must be in the same directory as **spsFITC.exe**:

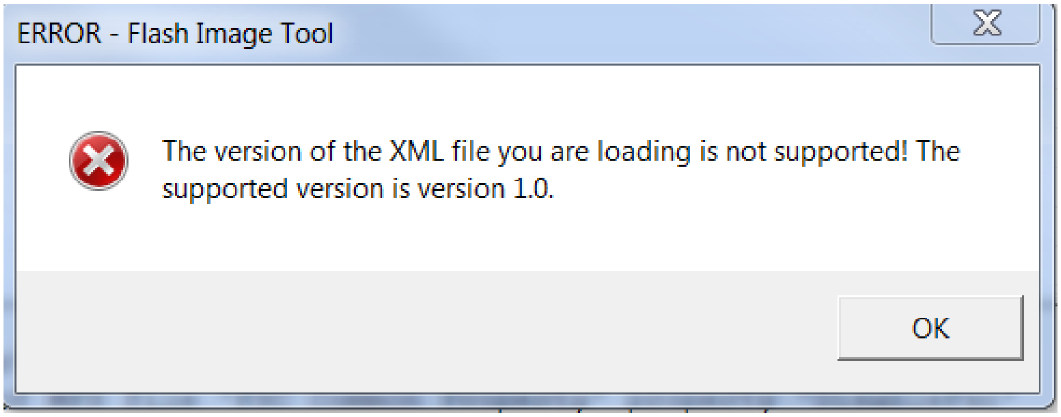
- spsfitc.ini - if this file is missing, spsFITC will create a new one filled with default values

Additional files required by spsFITC for work:

- configuration XML
- region specific binaries

spsFITC does not run correctly if any of the .xml and .bin files listed above are missing. spsFITC creates a blank **spsfitc.ini** file if there is no **spsfitc.ini** file in the folder.

Note: When load an XML file from previous kit releases spsFITC will display a message to the user that the file being used is older than the version spsFITC expecting (See the following example).



The version of the XML file you are loading is not supported! The supported version is version 1.0.

3.3.1 Configuration Files

The flash image can be configured in many different ways, depending on the target hardware and the required FW options. spsFITC lets you change this configuration in a graphical manner (via the GUI). Each configuration can be saved to an XML file. These XML files can be loaded at a later time and used to build subsequent flash images.

3.3.1.1 Creating a New Configuration

spsFITC provides default configuration files that you can use to build a new image. You should open one of the xml configurations files from the package to create a new configuration.

3.3.1.2 Opening an Existing Configuration

To open an existing configuration file:

1. Choose **File** > Open or click on Open icon on toolbar; the Open File dialog appears

3. Select the XML file you want to load
4. Click **Open**.

Note: You can also open a file by dragging and dropping a configuration file into the main window of the application.

3.3.1.3 Saving a Configuration

To save the current configuration in an XML file:

1. Choose **File > Save, click on Save icon on toolbar** or **File > Save As**; the **Save File** dialog appears if the configuration has not been given a name or if **File > Save As** was chosen.
2. Select the path and enter the file name for the configuration.
3. Click **Save**.

3.3.1.4 Merging XMLs

To merge old configuration with current XML:

1. Choose File > Merge XML
2. Select old XML configuration file in the active dialog
3. Select new (default) XML configuration provided with spsFITC in the next dialog

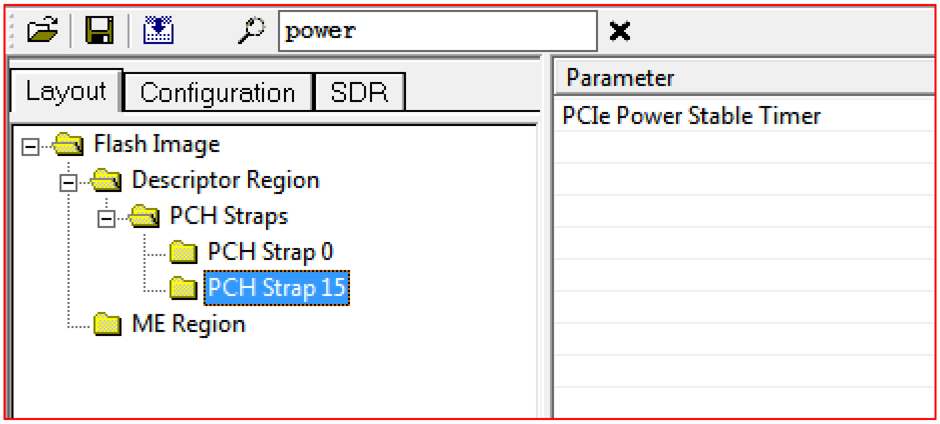
Values from old configuration that are different than XML defaults will be merged to new XML configuration if possible. Result of merging process will be saved in the current spsFITC working directory as MergeResult.xml. Afterwards merged configuration will be automatically opened by spsFTIC.

3.3.1.5 Using search box

To search for parameters containing specific phrase:

1. Type the phrase you're looking for in the search box. Searching begins as you type
2. Only nodes which contain found parameters will be visible on the left pane
3. Use the ☒ button to clear the search box and bring back visibility of all nodes

Note: The search phrase will be only matched with parameter names and not with nodes names.



Note: Example use of search box

3.3.2 Environment Variables

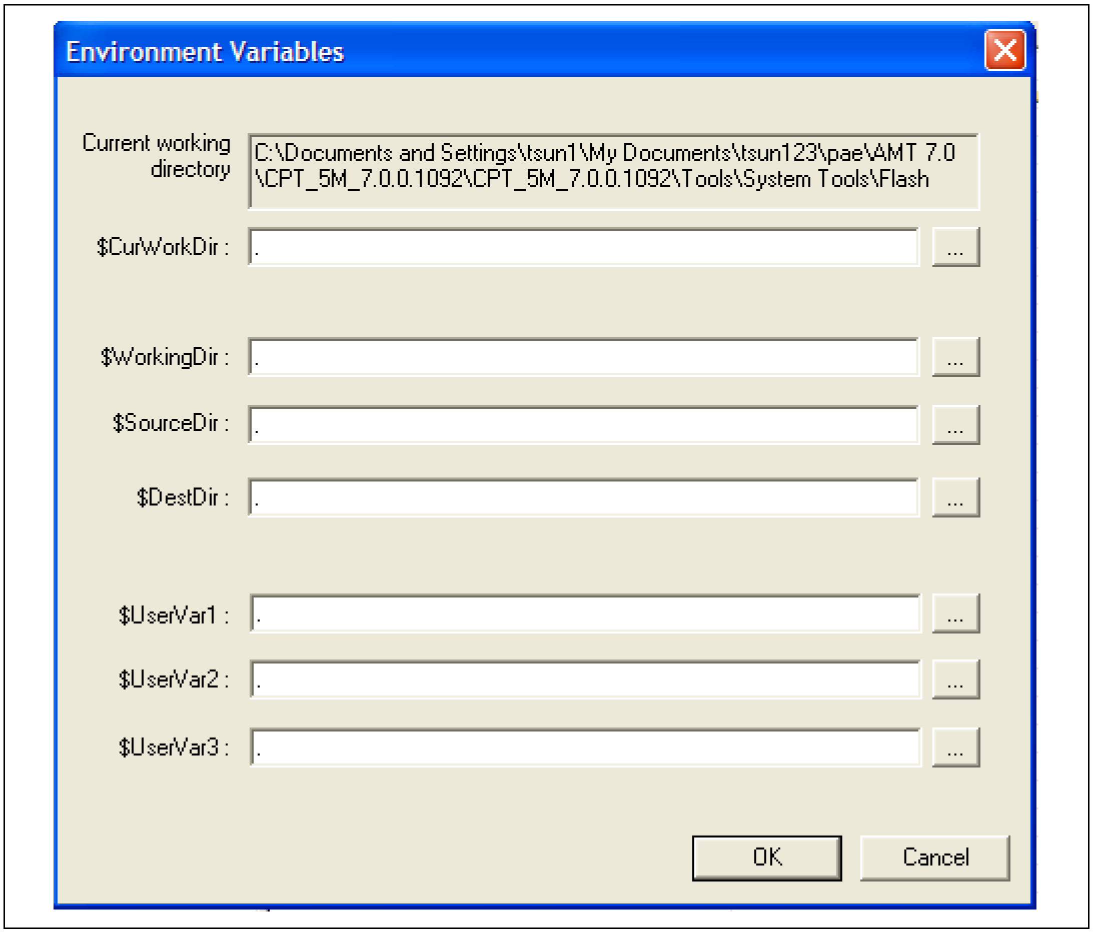
A set of environment variables is provided to make the image configuration files more portable. The configuration is not tied to a particular root directory structure because all of the paths in the configuration are relative to environment variables. You can set the environment variables appropriate for your computer, or override the variables with command line options.

It is recommended that the environment variables be the first thing you set when working with a new configuration. This ensures that spsFITC can properly substitute environment variables into paths to keep them relative. Doing this also speeds up configuration because many of the **Open File** dialogs default to particular environment variable paths.

To modify the environment variables:

1. Choose **Build > Environment Variables**; a dialog appears displaying the current working directory on top, followed by the current values of all the environment variables:
    - $WorkingDir – the directory where the log file is kept and where the components of an image are stored when an image is decomposed.
    - $SourceDir – the directory that contains the base image binary files from which a complete flash image is prepared. Usually these base image binary files are obtained from Intel® VIP on the Web, a BIOS programming resource, or another source.
    - $DestDir – the directory in which the final combined image is saved, as well as all intermediate files generated during the build.
    - $UserVar1-3 – used when the above variables are not populated.

Figure 3-2. Environment Variables Dialog



Current working directory: C:\Documents and Settings\tsun1\My Documents\tsun123\pae\AMT 7.0\CPT_5M_7.0.0.1092\CPT_5M_7.0.0.1092\Tools\System Tools\Flash

$CurWorkDir : .

$WorkingDir : .

$SourceDir : .

$DestDir : .

$UserVar1 : .

$UserVar2 : .

$UserVar3 : .

2. Click the ... button next to an environment variable and select the directory where that variable's files will be stored; the name and relative path of that directory appears in the field next to the variable's name.
3. Repeat Step 2 until the directories of all relevant environment variables have been defined.
4. Click **OK**.

Note: The environment variables are saved in the application's INI file, not the XML configuration file. This allows the configuration files to be portable across different computers and directory structures.

3.3.3 Build Settings

spsFITC lets you set several options that control how the image is built. The options that can be modified are described in Table 3-2.

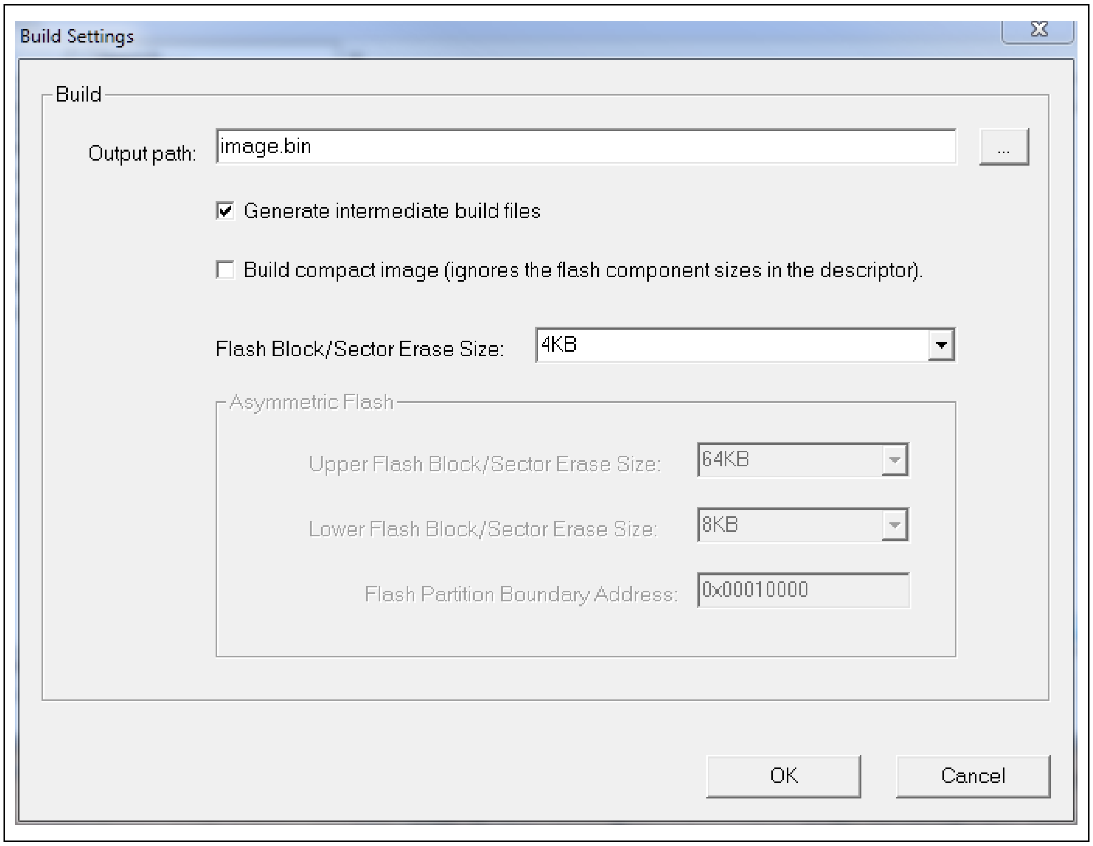
To modify the build setting:

1. Choose **Build** > **Build Settings**; a dialog appears showing the current build settings.
2. Modify the relevant settings in the **Build Settings** dialog.
3. Click **OK**; the modified build settings are saved in the XML configuration file.

Table 3-2. Build Settings Dialog Options

Option	Description
Output path	The path and filename where the final image should be saved after it is built. (**Note**: Using the $DestDir environment variable makes the configuration more portable.)
Generate intermediate build files	Causes the application to generate separate (intermediate) binary files for each region, in addition to the final image file (see Figure 3-3). These files are located in the specified output folder's INT subfolder. These image files can be programmed individually with the spsFPT.
Build Compact Image	Creates the smallest flash image possible. (By default, the application uses the flash component sizes in the Descriptor to determine the image length.)
Flash Block/Sector Erase Size	All regions in the flash conform to the 4 KB sector erase size. It is critical that this option is set correctly to ensure that the flash regions can be properly updated at runtime.
Assymetric Flash	Allows you to specify a different sector erase size for the upper and lower flash block. This option also lets you modify the flash partition boundary address. **Only 4 KB erase is supported for Intel® ME FW.** This option is disabled.

Figure 3-3. Build Settings Dialog



Build

Output path: image.bin

☑ Generate intermediate build files

☐ Build compact image (ignores the flash component sizes in the descriptor).

Flash Block/Sector Erase Size: 4KB

Asymmetric Flash

Upper Flash Block/Sector Erase Size: 64KB

Lower Flash Block/Sector Erase Size: 8KB

Flash Partition Boundary Address: 0x00010000

3.3.4 Modifying the Flash Descriptor Region

The FDR contains information about the flash image and the target hardware. This region contains the read/write values. It is important for this region to be configured correctly or the target platform may not function as expected. This region also needs to be configured correctly in order to ensure that the system is secure.

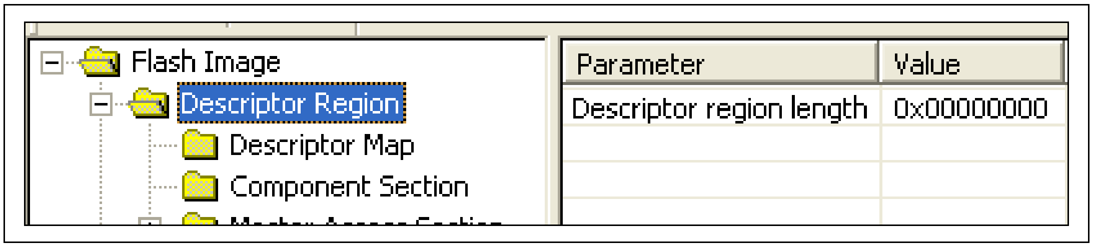
3.3.4.1 Descriptor Region Length

The Descriptor Region Length parameter sets the size of the Descriptor region.

To set the value of the Descriptor Region Length parameter:

1. Select **Descriptor Region** in the left pane; the **Descriptor Region Length** parameter appears in the right pane.
2. Double-click the **Descriptor Region Length** parameter; the **Descriptor Region Length** dialog appears.
3. Enter any non-zero value into the dialog to set the length of the region and click **OK**.

Figure 3-4. Descriptor Region Length Parameter



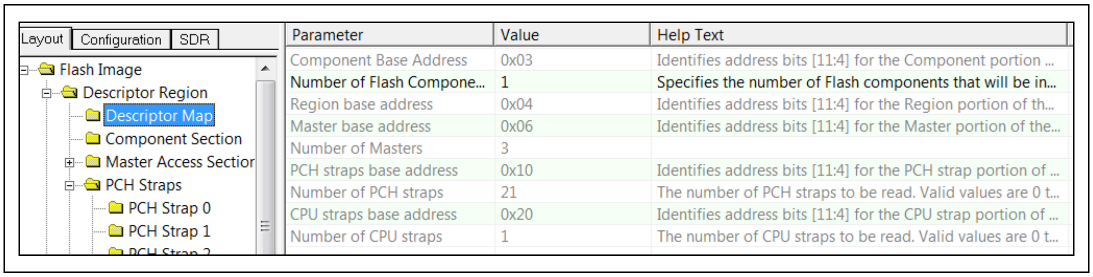
Parameter	Value
Descriptor region length	0x00000000

3.3.4.2 Setting the Number and Size of the Flash Components

To set the number of flash components:

1. Expand the **Descriptor Region** node of the tree in the left pane.
2. Select **Descriptor Map** (see Figure 3-5); all the parameters in the Descriptor Map section are listed in the right pane.

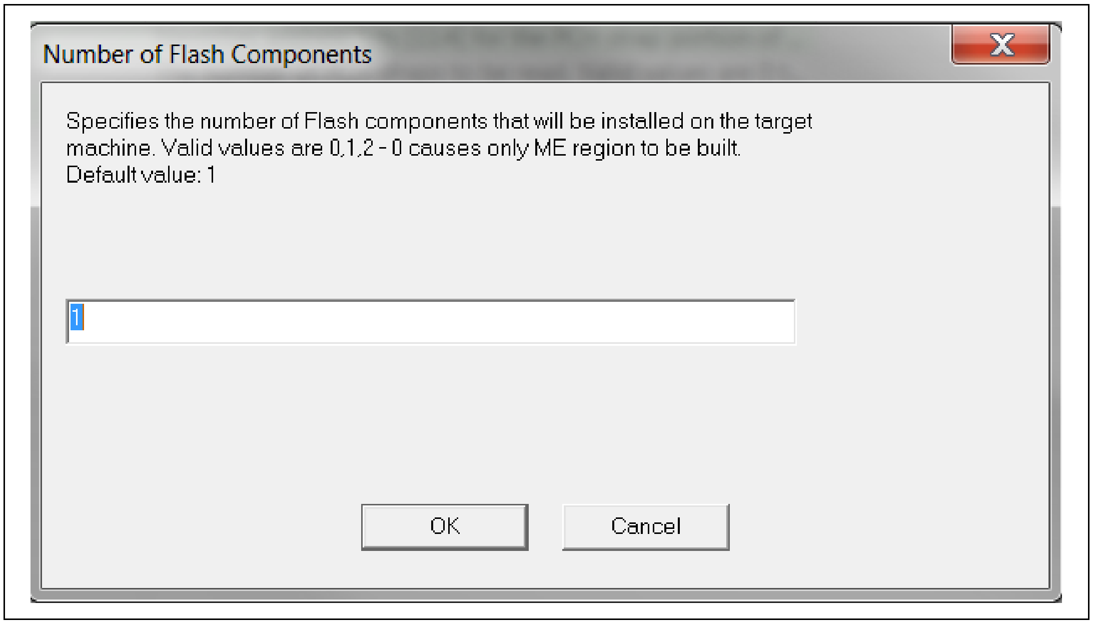
Figure 3-5. Descriptor Region > Descriptor Map Parameters



Parameter	Value	Help Text
Component Base Address	0x03	Identifies address bits [11:4] for the Component portion ...
Number of Flash Compone...	1	Specifies the number of Flash components that will be in...
Region base address	0x04	Identifies address bits [11:4] for the Region portion of th...
Master base address	0x06	Identifies address bits [11:4] for the Master portion of the...
Number of Masters	3	
PCH straps base address	0x10	Identifies address bits [11:4] for the PCH strap portion of ...
Number of PCH straps	21	The number of PCH straps to be read. Valid values are 0 t...
CPU straps base address	0x20	Identifies address bits [11:4] for the CPU strap portion of ...
Number of CPU straps	1	The number of CPU straps to be read. Valid values are 0 t...

3. Double-click **Number of Flash Components** in the right pane (see Figure 3-6); the Flash Components dialog appears.
4. Enter the number of flash components (valid values are 0, 1 or 2).
5. Click **OK**; the parameter is updated.

Figure 3-6. Flash Components Dialog



Specifies the number of Flash components that will be installed on the target machine. Valid values are 0,1,2 - 0 causes only ME region to be built.
Default value: 1

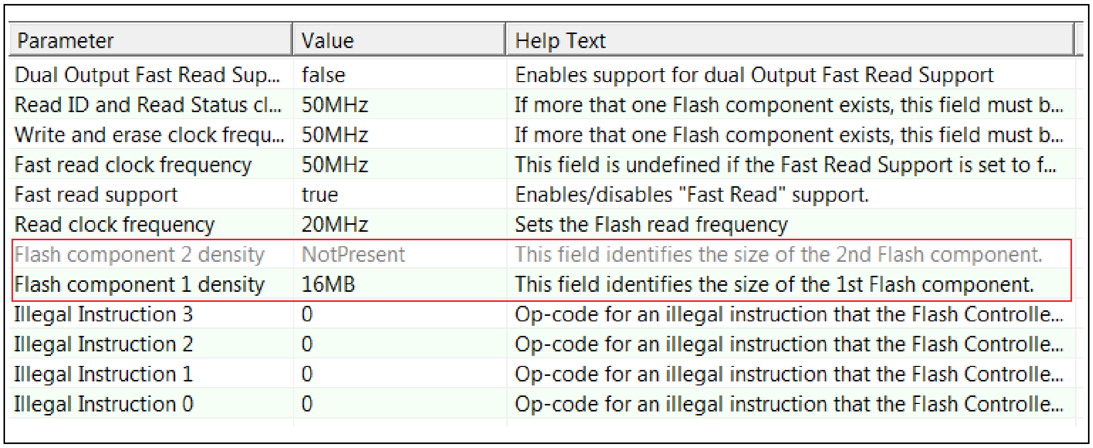
1

To set the size of each flash component:

1. Expand **Descriptor Region** node in the left pane and select **Component Section**; the Component Section parameters appear in the right pane. The **Flash component 1 density** and **Flash component 2 density** parameters specify the size of each flash component.
2. Double-click on one of these parameters; a dialog appears.
3. Select the correct component size from the dialog's drop-down list and click **OK**; that parameter is updated.
4. Repeat steps 2-3 for the other parameter.

Note: The size of the second flash component is only editable if the number of flash components is set to 2.

Figure 3-7. Descriptor Region > Component Section Parameters



Parameter	Value	Help Text
Dual Output Fast Read Sup...	false	Enables support for dual Output Fast Read Support
Read ID and Read Status cl...	50MHz	If more that one Flash component exists, this field must b...
Write and erase clock frequ...	50MHz	If more that one Flash component exists, this field must b...
Fast read clock frequency	50MHz	This field is undefined if the Fast Read Support is set to f...
Fast read support	true	Enables/disables "Fast Read" support.
Read clock frequency	20MHz	Sets the Flash read frequency
Flash component 2 density	NotPresent	This field identifies the size of the 2nd Flash component.
Flash component 1 density	16MB	This field identifies the size of the 1st Flash component.
Illegal Instruction 3	0	Op-code for an illegal instruction that the Flash Controlle...
Illegal Instruction 2	0	Op-code for an illegal instruction that the Flash Controlle...
Illegal Instruction 1	0	Op-code for an illegal instruction that the Flash Controlle...
Illegal Instruction 0	0	Op-code for an illegal instruction that the Flash Controlle...

3.3.4.3 Region Access Control

Regions of the flash can be protected from read or write access by setting a protection parameter in the Descriptor Region. The Descriptor Region must be locked before Intel ME devices are shipped. If the Descriptor Region is not locked, the Intel ME device is vulnerable to security attacks. The level of read/write access provided is at the discretion of the OEM/ODM. A cross-reference of access settings is shown below.

Table 3-3. Region Access Control Table

	Regions that can be accessed				
Region to Grant Access	**PDR**	**Intel® ME**	**GbE**	**BIOS**	**Descriptor**
Intel® ME	None/Read/Write	Intel® ME can always read from and write to Intel® ME Region	None/Read/Write	None/Read/Write	None/Read/Write
GbE	None/Read/Write	None/Read/Write	GbE can always read from and write to GbE Region	None/Read/Write	None/Read/Write
BIOS	None/Read/Write	None/Read/Write	None/Read/Write	Write only. BIOS can always read from and write to BIOS Region	None/Read/Write

There are three parameters in the Descriptor that specify access for each chipset. The bit structure of these parameters is shown below.

Key:

0 – denied access
1 – allowed access
NC – bit may be either 0 or 1 since it is unused.

Table 3-4. CPU/BIOS Access

	Read Access							
	Unused			PDR	GbE	Intel ME	BIOS	Desc
Bit Number	7	6	5	4	3	2	1	0
Bit Value	X	X	X	0/1	0/1	0/1	NC	0/1

	Write Access							
	Unused			PDR	GbE	Intel ME	BIOS	Desc
Bit Number	7	6	5	4	3	2	1	0
Bit Value	X	X	X	0/1	0/1	0/1	NC	0/1

Example:

If the CPU/BIOS needs read access to the GbE and Intel ME and write access to Intel ME, then the bits are set to:

Read Access – 0b 0000 1110 (0x 0E in hexadecimal)

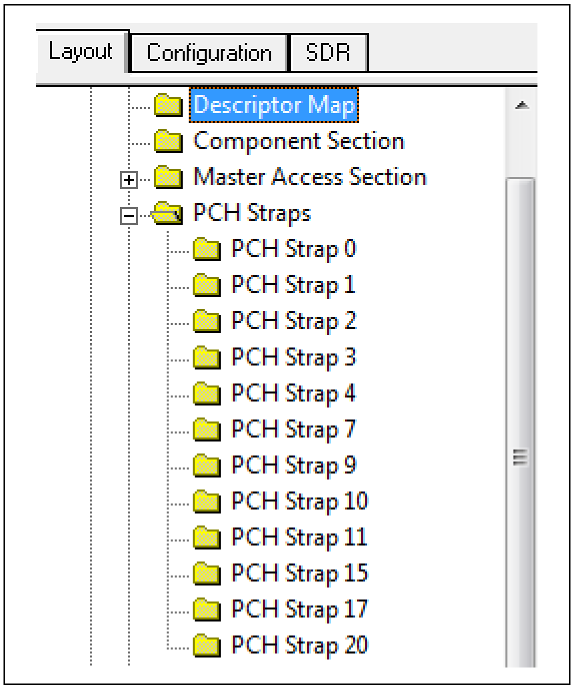
Write Access – 0b 0000 0110 (0x 06 in hexadecimal)

To set these access values in spsFITC:

1. Select **Descriptor Region > Master Access Section > CPU/BIOS** in the left pane; the access parameters are listed in the right pane.
2. Double-click on each parameter and set its access value in one of the following ways:
   - To generate an image for debug purposes or to leave the SPI region open: select 0xFF for both read and write access in all three sections.
   - To lock the SPI in the image creation phase: select the recommended setting for production (e.g., select 0x0D for Intel ME read access and 0x0C for Intel ME write access).

3.3.5 PCH Soft Straps

These sections contain configuration options for the PCH. The number of Soft Strap sections and their functionality differ based on the target PCH. Improper settings could lead to undesirable behavior from the target platform. (For more information on how to set them correctly, see the FW Bringup Guide or the PCH SPI programming guide, Appendix A.)



3.3.6 VSCC Table

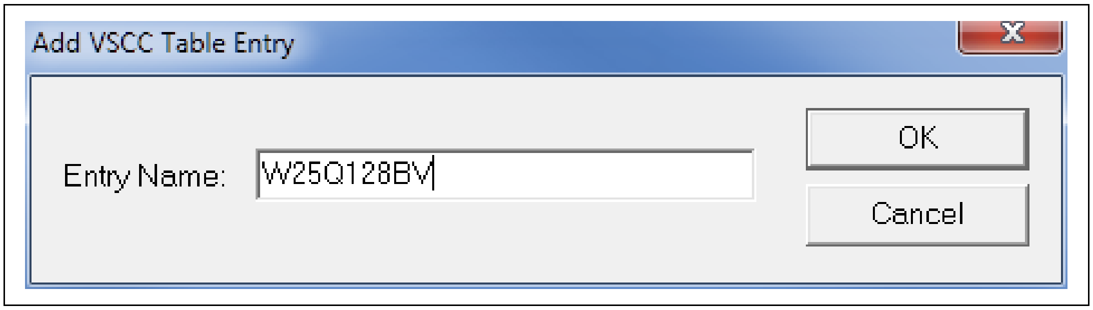
This section is used to store information to setup flash access for Intel ME. This does not have any effect on the usage of the spsFPT. **If the information in this section is incorrect, Intel ME FW may not communicate with the flash device.** The information provided is dependent on the flash device used on the system. (For more information, see the PCH SPI programming guide, Section 6.4.)

3.3.6.1 Adding a New Table Entry

To add a new table entry:

1. Right-click on Descriptor Region > VSCC table.
2. Choose **Add Table Entry** from the pop-up menu; the **Add Table Entry** dialog appears.

Figure 3-8. Add VSCC Table Entry Dialog

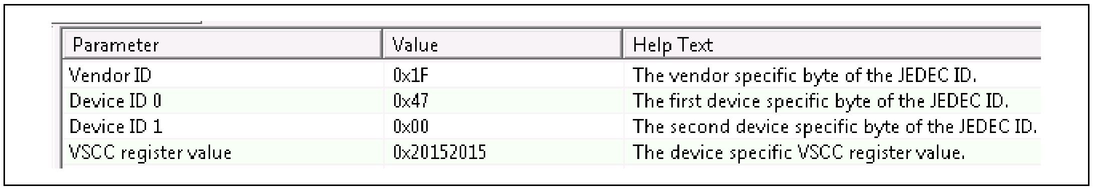


Entry Name: W25Q128BV

3. Enter a name into the **Entry Name** field. (**Note**: To avoid confusion it is recommended that each table entry name be unique. There is no checking mechanism in spsFITC to prevent table entries that have the same name and no error message is displayed in such cases.)
4. Click **OK**; the new table is listed in the left pane under **VSCC Table** and you can enter into it the values for the flash device. (Figure 3-9, which shows the parameters of a new VSCC table.)

Note: The values in the VSCC table can be found in the serial flash data sheet. You should use the CPT SPI Programming Guide to calculate the VSSC values.

Figure 3-9. Sample VSCC Table Entry



Parameter	Value	Help Text
Vendor ID	0x1F	The vendor specific byte of the JEDEC ID.
Device ID 0	0x47	The first device specific byte of the JEDEC ID.
Device ID 1	0x00	The second device specific byte of the JEDEC ID.
VSCC register value	0x20152015	The device specific VSCC register value.

3.3.6.2 Removing an Existing VSCC Table

To remove an existing table:

1. Right-click on the name of the table in the left pane that you want to remove.
2. Choose Remove Table Entry; the table and all the information in it is removed.

3.3.7 Modifying the Intel® Management Engine (Intel® ME) Region

The Intel ME Region contains all of the FW data for the Intel ME.

Setting the Intel ME Region Binary File

To select the Intel ME region binary file:

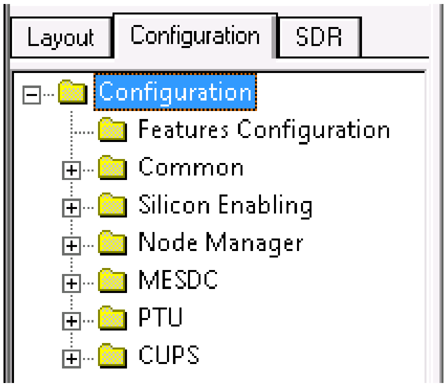
1. Select the Intel ME Region tree node.
2. Right-click on the **Intel ME Region,** Click Edit Region; a dialog appears that lets you select the Intel ME binary file to be used.

3. Click **OK** to update the parameter; when the flash image is built, the contents of this file is copied into the Intel ME Region.

3.3.8 Intel ME FW Configurations

This section contains Intel ME related configurations.

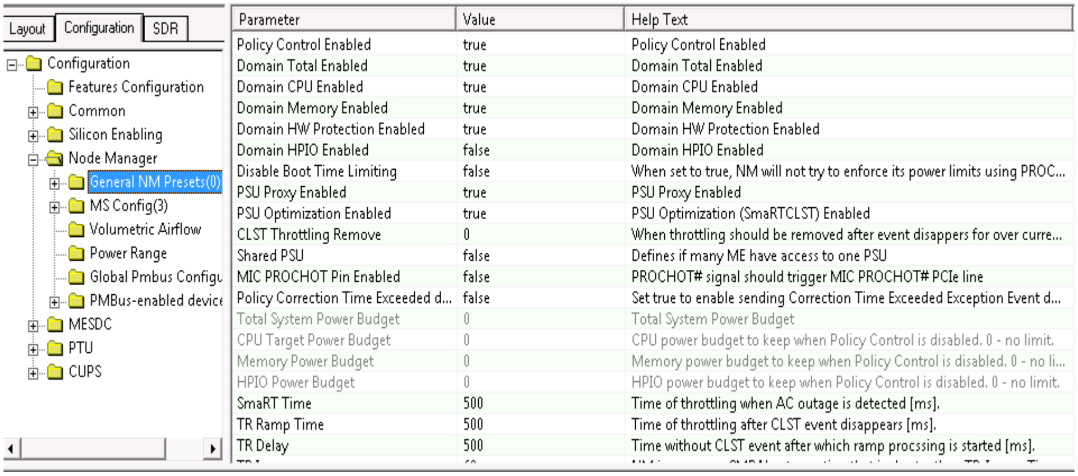
Figure 3-10. Configuration Tag—Intel ME Configuration



This section contains the setting for all Intel ME related feature configurations that apply to all FW configurations including SiEn, Intel® Node Manager (Intel® NM).

3.3.8.1 General Intel® Node Manager Presents

Figure 3-11. Configuration ->Node Manager ->General NM Presets



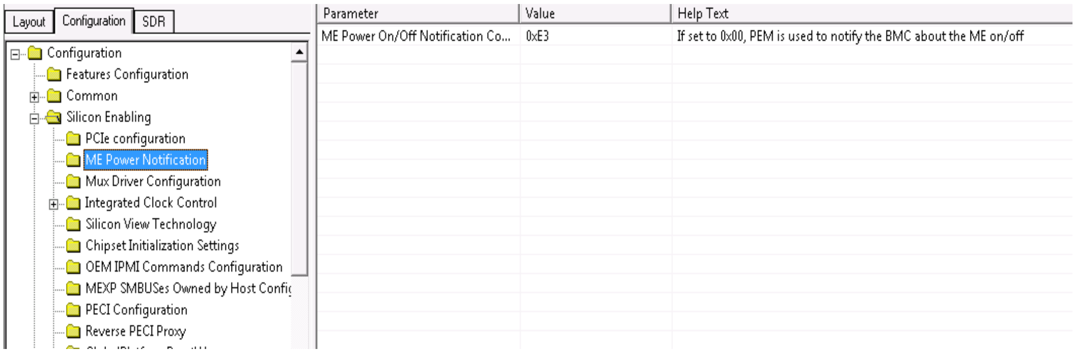
Parameter	Value	Help Text
Policy Control Enabled	true	Policy Control Enabled
Domain Total Enabled	true	Domain Total Enabled
Domain CPU Enabled	true	Domain CPU Enabled
Domain Memory Enabled	true	Domain Memory Enabled
Domain HW Protection Enabled	true	Domain HW Protection Enabled
Domain HPIO Enabled	false	Domain HPIO Enabled
Disable Boot Time Limiting	false	When set to true, NM will not try to enforce its power limits using PROC...
PSU Proxy Enabled	true	PSU Proxy Enabled
PSU Optimization Enabled	true	PSU Optimization (SmaRTCLST) Enabled
CLST Throttling Remove	0	When throttling should be removed after event disappers for over curre...
Shared PSU	false	Defines if many ME have access to one PSU
MIC PROCHOT Pin Enabled	false	PROCHOT# signal should trigger MIC PROCHOT# PCIe line
Policy Correction Time Exceeded d...	false	Set true to enable sending Correction Time Exceeded Exception Event d...
Total System Power Budget	0	Total System Power Budget
CPU Target Power Budget	0	CPU power budget to keep when Policy Control is disabled. 0 - no limit.
Memory Power Budget	0	Memory power budget to keep when Policy Control is disabled. 0 - no li...
HPIO Power Budget	0	HPIO power budget to keep when Policy Control is disabled. 0 - no limit.
SmaRT Time	500	Time of throttling when AC outage is detected [ms].
TR Ramp Time	500	Time of throttling after CLST event disappears [ms].
TR Delay	500	Time without CLST event after which ramp procssing is started [ms].

This section contains the general Intel Node Manager configuration including policy control, domains state, pre-configured policies and SmaRT&CLST configurations. This only applies to Intel Node Manager SKU.

Note: Any settings which are different than default value are highlighted.

3.3.8.2 Intel ME Power Notification

Figure 3-12. Configuration -> Silicon Enabling -> ME Power Notification

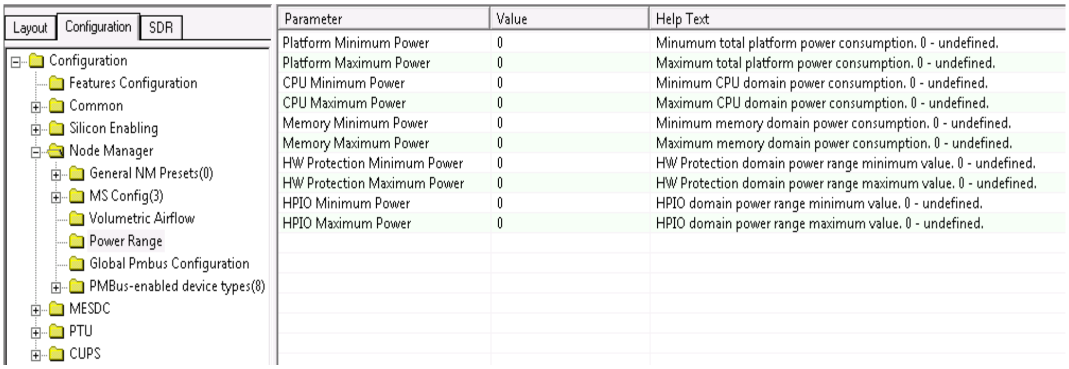


Parameter	Value	Help Text
ME Power On/Off Notification Co...	0xE3	If set to 0x00, PEM is used to notify the BMC about the ME on/off

This section shows the configuration needed for a command telling BMC which address to use for the Intel ME power on/off notification. This is only relevant to the Intel Node Manager configuration.

3.3.8.3 Power Range

Figure 3-13. Configurations-> Node Manager -> Power Range

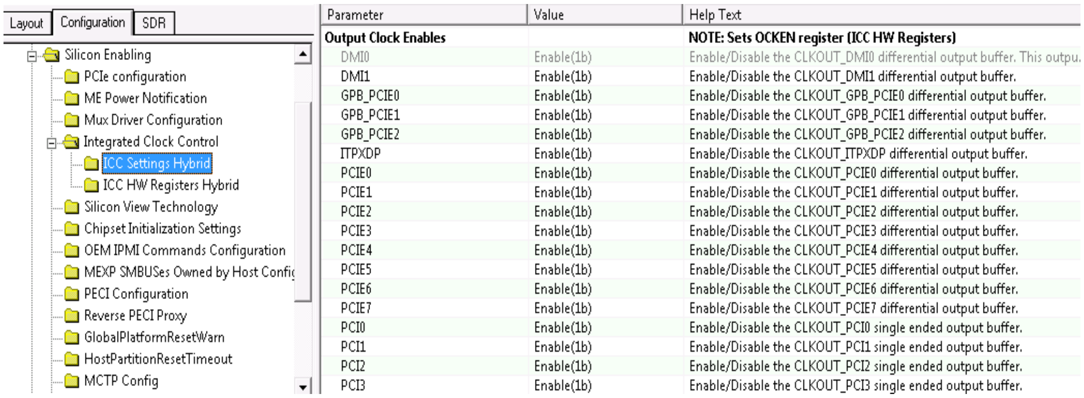


Parameter	Value	Help Text
Platform Minimum Power	0	Minumum total platform power consumption. 0 - undefined.
Platform Maximum Power	0	Maximum total platform power consumption. 0 - undefined.
CPU Minimum Power	0	Minimum CPU domain power consumption. 0 - undefined.
CPU Maximum Power	0	Maximum CPU domain power consumption. 0 - undefined.
Memory Minimum Power	0	Minimum memory domain power consumption. 0 - undefined.
Memory Maximum Power	0	Maximum memory domain power consumption. 0 - undefined.
HW Protection Minimum Power	0	HW Protection domain power range minimum value. 0 - undefined.
HW Protection Maximum Power	0	HW Protection domain power range maximum value. 0 - undefined.
HPIO Minimum Power	0	HPIO domain power range minimum value. 0 - undefined.
HPIO Maximum Power	0	HPIO domain power range maximum value. 0 - undefined.

This section can set pre-configured power budget applied when policy control is disabled. This is only relevant to Intel Node Manager configuration.

3.3.8.4 Integrated Clock Control Registers

Figure 3-14. Configuration-> Silicon Enabling -> Integrated Clock Control

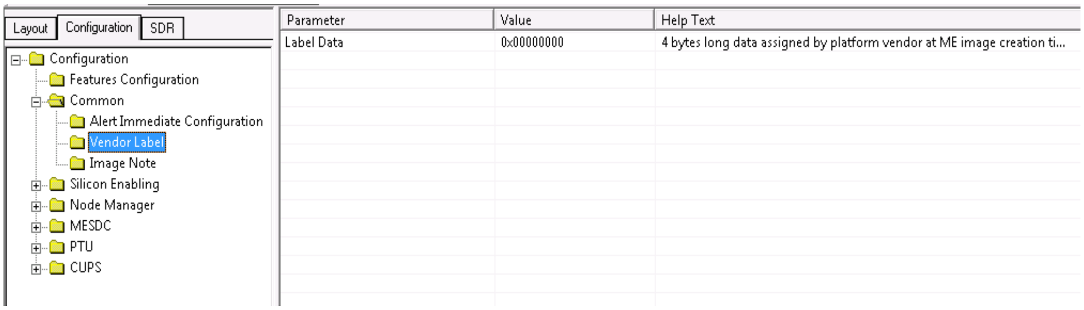


Parameter	Value	Help Text
Output Clock Enables		**NOTE: Sets OCKEN register (ICC HW Registers)**
DMI0	Enable(1b)	Enable/Disable the CLKOUT_DMI0 differential output buffer. This outpu.
DMI1	Enable(1b)	Enable/Disable the CLKOUT_DMI1 differential output buffer.
GPB_PCIE0	Enable(1b)	Enable/Disable the CLKOUT_GPB_PCIE0 differential output buffer.
GPB_PCIE1	Enable(1b)	Enable/Disable the CLKOUT_GPB_PCIE1 differential output buffer.
GPB_PCIE2	Enable(1b)	Enable/Disable the CLKOUT_GPB_PCIE2 differential output buffer.
ITPXDP	Enable(1b)	Enable/Disable the CLKOUT_ITPXDP differential output buffer.
PCIE0	Enable(1b)	Enable/Disable the CLKOUT_PCIE0 differential output buffer.
PCIE1	Enable(1b)	Enable/Disable the CLKOUT_PCIE1 differential output buffer.
PCIE2	Enable(1b)	Enable/Disable the CLKOUT_PCIE2 differential output buffer.
PCIE3	Enable(1b)	Enable/Disable the CLKOUT_PCIE3 differential output buffer.
PCIE4	Enable(1b)	Enable/Disable the CLKOUT_PCIE4 differential output buffer.
PCIE5	Enable(1b)	Enable/Disable the CLKOUT_PCIE5 differential output buffer.
PCIE6	Enable(1b)	Enable/Disable the CLKOUT_PCIE6 differential output buffer.
PCIE7	Enable(1b)	Enable/Disable the CLKOUT_PCIE7 differential output buffer.
PCI0	Enable(1b)	Enable/Disable the CLKOUT_PCI0 single ended output buffer.
PCI1	Enable(1b)	Enable/Disable the CLKOUT_PCI1 single ended output buffer.
PCI2	Enable(1b)	Enable/Disable the CLKOUT_PCI2 single ended output buffer.
PCI3	Enable(1b)	Enable/Disable the CLKOUT_PCI3 single ended output buffer.

This section allows user to config registers for clock control profiles' setting. Only relevant to ICC enabled platform (Grantley platform).

3.3.8.5 Vendor Label

Figure 3-15. Configuration -> Common -> Vendor Label

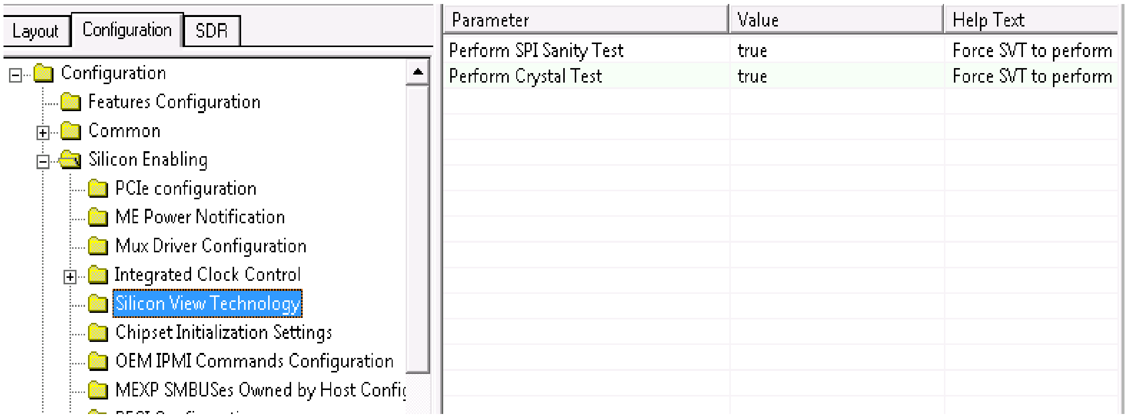


Parameter	Value	Help Text
Label Data	0x00000000	4 bytes long data assigned by platform vendor at ME image creation ti...

This section shows the users how to configure a 4-byte label that can be used by platform owner.

3.3.8.6 Silicon View Technology

Figure 3-16. Configuration-> Silicon Enabling -> Silicon View Technology

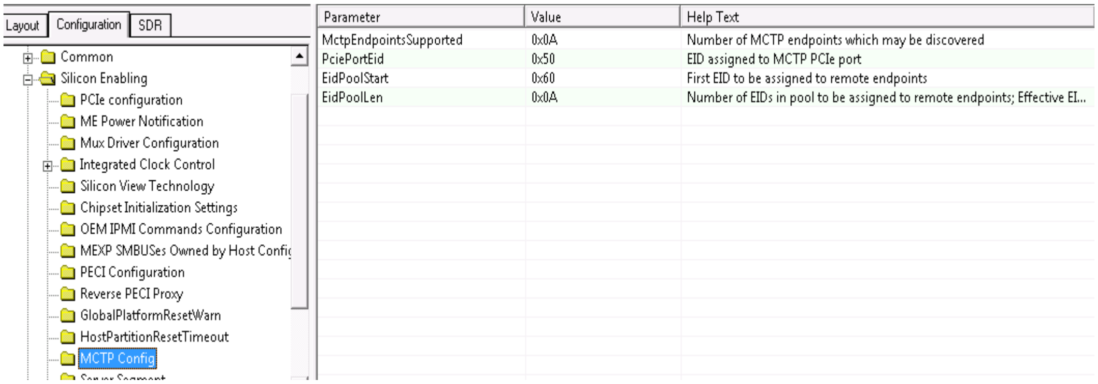


Parameter	Value	Help Text
Perform SPI Sanity Test	true	Force SVT to perform
Perform Crystal Test	true	Force SVT to perform

This section shows the user how to configure a Silicon View Technology boot configuration (perform SPI Sanity test, perform crystal frequency test).

3.3.8.7 MCTP

Figure 3-17. Configuration-> Silicon Enabling -> MCTP Config

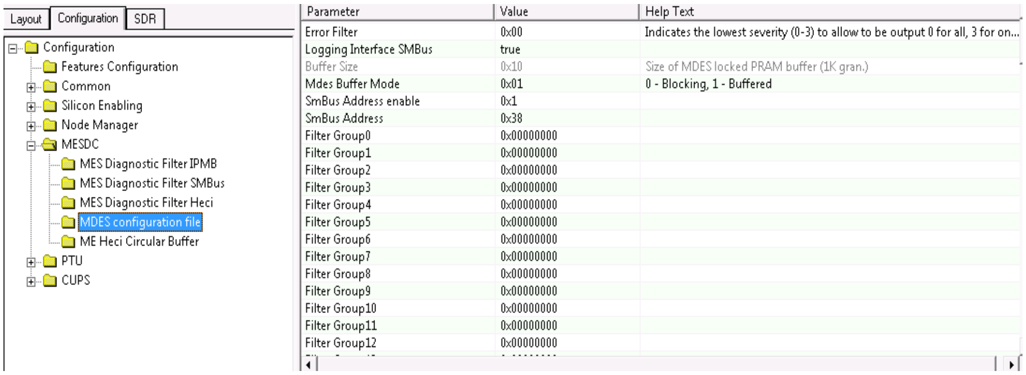


Parameter	Value	Help Text
MctpEndpointsSupported	0x0A	Number of MCTP endpoints which may be discovered
PciePortEid	0x50	EID assigned to MCTP PCIe port
EidPoolStart	0x60	First EID to be assigned to remote endpoints
EidPoolLen	0x0A	Number of EIDs in pool to be assigned to remote endpoints; Effective EI...

This section allows customer to set MCTP related settings.

3.3.8.8 MDES configuration file

Figure 3-18. Configuration-> MESDC -> MESDC Configuration File

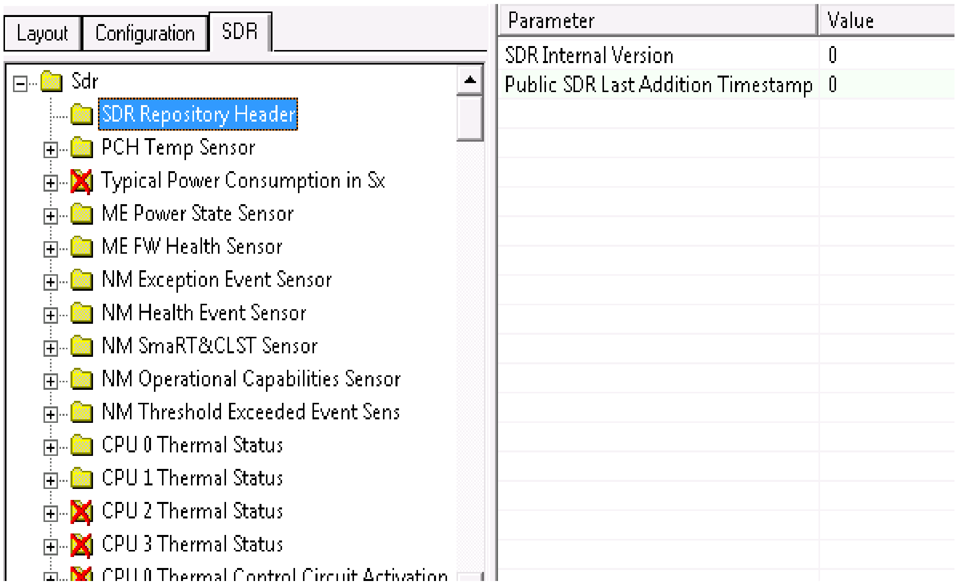


Parameter	Value	Help Text
Error Filter	0x00	Indicates the lowest severity (0-3) to allow to be output 0 for all, 3 for on...
Logging Interface SMBus	true	
Buffer Size	0x10	Size of MDES locked PRAM buffer (1K gran.)
Mdes Buffer Mode	0x01	0 - Blocking, 1 - Buffered
SmBus Address enable	0x1	
SmBus Address	0x38	
Filter Group0	0x00000000	
Filter Group1	0x00000000	
Filter Group2	0x00000000	
Filter Group3	0x00000000	
Filter Group4	0x00000000	
Filter Group5	0x00000000	
Filter Group6	0x00000000	
Filter Group7	0x00000000	
Filter Group8	0x00000000	
Filter Group9	0x00000000	
Filter Group10	0x00000000	
Filter Group11	0x00000000	
Filter Group12	0x00000000	

This section allows customer to set MESDC related settings.

3.3.9 SDR Configuration

Figure 3-19. SDR Configuration



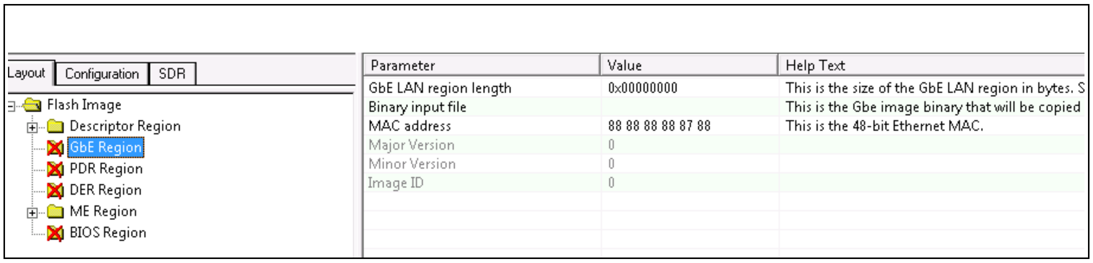
Parameter	Value
SDR Internal Version	0
Public SDR Last Addition Timestamp	0

This section shows the configuration for sensors data, HW descriptions (PIA). These settings are relevant to Intel Node Manager configurations.

3.3.10 Modifying the GbE (LAN) Region

The GbE Region contains various configuration parameters (e.g., the MAC address) for the embedded Ethernet controller.

Figure 3-20. GbE Region Options



Parameter	Value	Help Text
GbE LAN region length	0x00000000	This is the size of the GbE LAN region in bytes. S
Binary input file		This is the Gbe image binary that will be copied
MAC address	88 88 88 88 87 88	This is the 48-bit Ethernet MAC.
Major Version	0	
Minor Version	0	
Image ID	0	

Note: This setting can be also overwritten from the CLI invocation (it applies to GUI settings also) by calling /GbE flag with the valid path to the GbE file with the filename. For example, “spsFITC.exe /GbE gbe.bin mycfg.xml” will show the GUI interface with the GBE region enabled and binary input file set to gbe.bin. More examples included below.
Setting the GbE Region Length Option

The GbE Region length option should not be altered. A value of 0x00000000 indicates that the GbE Region will be auto-sized as described in Section 3.2.1.

3.3.10.1 Setting the GbE Region Binary File

To select the GbE Region binary file:

1. Select **GbE Region** in the left pane; the GbE Region parameters are listed in the right pane.
2. Double-click on the **Binary input file** parameter; a dialog appears that lets you select the GbE file to use.
3. Select a file.
4. Click **OK** to update the parameter; when the flash image is built, the contents of this file is copied into the GbE Region.

3.3.10.2 Enabling/Disabling the GbE Region

The GbE Region can be excluded from the flash image by disabling it in the spsFITC.

To disable the GbE Region:

1. Right-click on **GbE Region** in the left pane.
2. Choose **Disable Region** from the pop-up menu; when the flash image is built it will not contain a GbE Region

To enable the GbE Region:

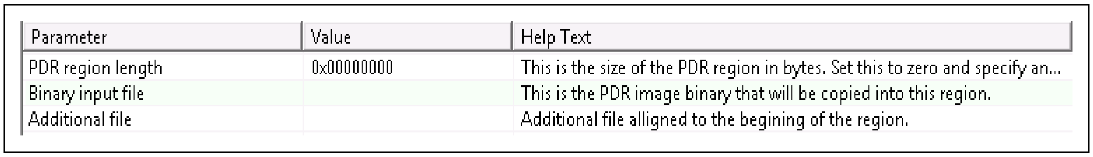
1. Right-click on **GbE Region** in the left pane.

2. Choose **Enable Region** from the pop-up menu.

3.3.11 Modifying the PDR Region

The PDR Region contains various configuration parameters that let you customize the computer's behavior.

Figure 3-21. PDR Region Options



Parameter	Value	Help Text
PDR region length	0x00000000	This is the size of the PDR region in bytes. Set this to zero and specify an...
Binary input file		This is the PDR image binary that will be copied into this region.
Additional file		Additional file alligned to the begining of the region.

Note: This setting can be also overwritten from the CLI invocation (it applies to GUI settings also) by calling /PDR flag with the valid path to the bios file with the filename. For example: "spsFITC.exe /pdr pdr.bin mycfg.xml" will show the GUI interface with the PDR region **enabled** and binary input file set to bios.bin. More examples included.

3.3.11.1 Setting the PDR Region Length Option

The PDR Region length option should not be altered. A value of 0x00000000 indicates that the PDR Region will be auto-sized as described in Section 3.2.1.

3.3.11.2 Setting the PDR Region Binary File

To select the PDR region binary file:

1. Select PDR Region in the left pane; the PDR Region parameters are listed in the right pane.
2. Double-click the **Binary input file** parameter; a dialog appears that lets you specify which PDR file to use.
3. Click **OK** to update the parameter; when the flash image is built, the contents of this file is copied into the PDR region.

3.3.11.3 Enabling/Disabling the PDR Region

The PDR Region can be excluded from the flash image by disabling it in spsFITC.

To disable the PDR Region:

1. Right-click on **PDR Region** in the left pane.
2. Choose **Disable Region** from the pop-up menu; when the flash image is built, there is no PDR Region in it.

Note: This region is disabled by default.

To enable the PDR Region:

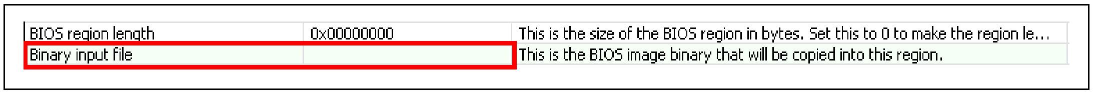
1. Right-click on **PDR Region** in the left pane.

2. Choose **Enable Region** from the pop-up menu.

3.3.12 Modifying the BIOS Region

The BIOS Region contains the BIOS code run by the host processor. spsFITC always aligns this region with the end of the flash image. This is done so that if the flash descriptor becomes corrupt for any reason, the PCH defaults to legacy mode and looks for the reset at the end of the flash memory. By placing the BIOS Region at the end there is a chance the system will still boot. It is also important to note that the BIOS binary file is aligned with the end of the BIOS Region so that the reset vector is in the correct place. This means that if the binary file is smaller than the BIOS Region, the region is padded at the beginning instead of at the end.

Figure 3-22. BIOS Region Parameters



BIOS region length	0x00000000	This is the size of the BIOS region in bytes. Set this to 0 to make the region le...
Binary input file		This is the BIOS image binary that will be copied into this region.

Note: This setting can be also overwritten from the CLI invocation (it applies to GUI settings also) by calling /bios flag with the valid path to the bios file with the filename. For example: "spsFITC.exe /bios bios.bin mycfg.xml" shows the GUI interface with the bios region enabled and binary input file set to BIOS.bin. More examples included.

3.3.13 Setting the BIOS Region Length Parameter

The value of the BIOS Region length parameter should not be altered. A value of 0x00000000 indicates that the BIOS Region will be auto-sized to fit the least possible space aligned to 4 kb of the binary image file set by the user.

3.3.13.1 Setting the BIOS Region Binary File

To select the BIOS region binary file:

1. Select **BIOS Region** in the left pane; the BIOS Region parameters are listed in the right pane.
2. Double-click the **Binary input file** parameter; a dialog appears that lets you specify which BIOS file to use.
3. Click **OK** to update the parameter; when the flash image is built, the contents of this file are copied into the BIOS region.

3.3.13.2 Enabling/Disabling the BIOS Region

The BIOS Region can be excluded from the flash image by disabling it in spsFITC.

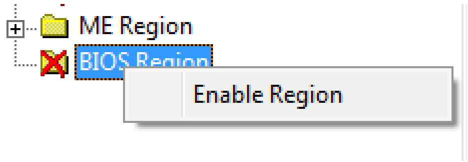
To disable the BIOS Region:

1. Right-click on **BIOS Region** in the left pane.

2. Choose **Disable Region** from the pop-up menu; when the flash image is built, there is no BIOS Region in it.

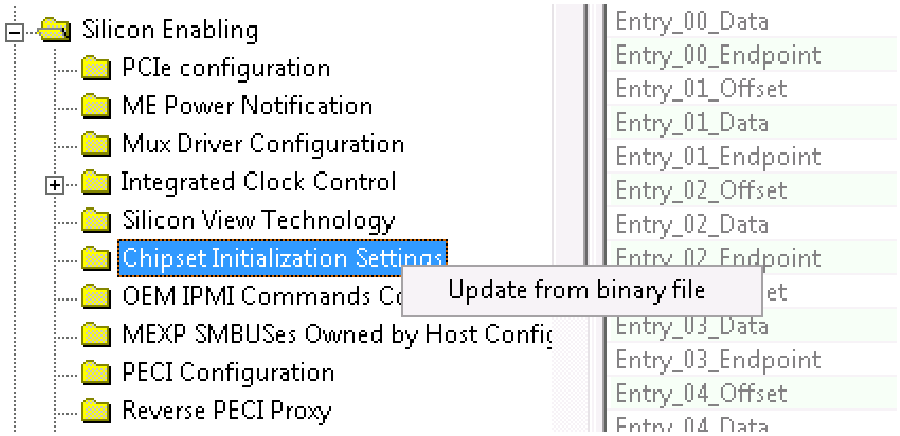
To enable the PDR Region:

1. Right-click on **BIOS Region** in the left pane.
2. Select **Enable Region** from the pop-up menu.



3.3.14 Loading settings for MFS file from a binary file

Some settings can be loaded from a binary file. This capability is assigned to a node in the left pane (not to a single parameter in the right pane). To load settings right-click on a node and choose "Update from binary file" option from a context menu. Node that the node which has this capability is: Configuration\Silicon Enabling\Chipset Initialization Settings.



Binary file must be the appropriate size (number of bytes in the file must be equal or less than number of bytes in settings).

3.3.15 Building a Flash Image

The flash image can be built with the spsFITC GUI interface.

To build a flash image with the currently loaded configuration:

- Choose Build > Build Image.

– OR –

- Specify an XML file with the `/b` option in the command line.

spsFITC uses an XML configuration file and the corresponding binary files to build the SPI flash image. The following is produced when an image is built:

- Binary file representing the image
- Text file detailing the various regions in the image
- Optional set of intermediate files.
- Multiple binary files containing the image broken up according to the flash component sizes (**Note**: These files are only created if two flash components are specified.)

The individual binary files can be used to manually program independent flash devices using a flash programmer. However, you should select the single larger binary file when using spsFPT.

3.3.16 Decomposing an Existing Flash Image

spsFITC is capable of taking an existing flash image and decomposing it in order to create the corresponding configuration. This configuration can be edited in the GUI like any other configuration (see below). A new image can be built from this configuration that is almost identical to the original, except for the changes you made to it.

To decompose an image:

1. Chose **File** > **Open.**
2. Change the file type filter to the appropriate file type.
3. Select the required file and click **Open**; the image is automatically decomposed, the GUI is updated to reflect the new configuration, and a folder is created with each of the regions in a separate binary file.

Note: It is also possible to decompose an image by simply dragging and dropping the file into the main window. When decomposing an image, there are some settings will not be able to be decomposed by spsFITC. spsFITC will use Intel default value instead. User might want to check the log file to find out which settings were not parsed.

3.4 Command Line Interface

spsFITC supports command line options.

To view all of the supported options: Run the application with the `-?` option.

The command line syntax for spsFITC is:

```
spsFITC.exe [/?] [/v] [/b] [/o <file>] [/idlm <file> [/me <file>][/me_offset <num>]
[/gbe <file>] [/bios <file>] [/sec_bios <file>][/pdr <file>] [/der <file>]
[/w <path>] [/s <path>] [/d <path>][/u1 <value>] [/u2 <value>] [/u3 <value>]
```

```
[/i <enable|disable>][/sdr <path>] [/sdrentinst 00] [/sdrch 00] [/DualImage]
[/flashcount <1|2>] [/flashsize1 <size>] [/flashsize2 <size>] [/fsc] [/save]
[XML or BIN file]
```

Table 3-5. spsFITC Command Line Options

Option	Description
/?	Display command line options.
/v	Display Flash Image Tool for Server Platform Services version.
/b	Build the flash image. Does NOT display the GUI.
/o <file>	Overrides the output file path.
/idlm <file>	Include IDLM file in appropriate place.
/me <file>	Overrides the binary source file for the ME region with the specified binary file.
/me_offset	Overrides the offset for the ME region.
/gbe <file>	Overrides the binary source file for the GbE region with the specified binary file.
/bios <file>	Overrides the binary source file for the BIOS Region with the specified binary file.
/sec_bios<file>	Overrides the binary source file for the Secondary BIOS region with the specified binary file.
/pdr <file>	Overrides the binary source file for the PDR region with the specified binary file.
/der <file>	Overrides the binary source file for the DER region with the specified binary file.
/w <path>	Overrides the working directory environment variable $WorkingDir. It is recommended that you set these environmental variables first. (Suggested values can be found in the OEM Bringup Guide.)
/s <path>	Overrides the source file directory environment variable $SourceDir. It is recommended that you set these environmental variables before starting a project.
/d <path>	Overrides the destination directory environment variable $DestDir. It is recommended that you set these environmental variables before starting a project.
/u1 <value>	Overrides the $UserVar1 environment variable with the value specified. Can be any value required.
/u2 <value>	Overrides the $UserVar2 environment variable with the value specified. Can be any value required.
/u3 <value>	Overrides the $UserVar3 environment variable with the value specified. Can be any value required.
/i <enable|disable>	Overrides intermediate file generation.
/sdr <path>	Create SDR file. Does NOT display the GUI.
/sdrentinst	While creating SDR file overrides 'Entity instance' values.
/sdrch	While creating SDR file overrides 'Channel' values.
/DualImage	Overrides the single image configuration

Option	Description
/flashsize1 <0, 1, 2, 3, 4 or 5>	Overrides the size of the first flash component with the size of the option selected as follows: 0 = 512 KB 1 = 1 MB 2 = 2 MB 3 = 4 MB 4 = 8 MB 5 = 16 MB.
/flashsize2 <0, 1, 2, 3, 4 or 5>	Overrides the size of the second flash component with the size of the option selected as follows: 0 = 512 KB 1 = 1 MB 2 = 2 MB 3 = 4 MB 4 = 8 MB 5 = 16 MB.
/fsc	Build Fan Speed Control binary file. Does NOT display the GUI.
/save	Open and save xml file to prepare appropriate format. Does NOT display the GUI.
[XML or BIN file]	The XML configuration (create) or BIN (decompose) file.

3.5 Example – Decomposing an Image and Extracting Parameters

The current value parameters of an image can be viewed by dragging and dropping the image into the main window, which then displays the current values of the image's parameters.

An image's parameters can also be extracted by entering the following commands into the command line:

```
spsFITC.exe output.bin /b
```

This command creates a folder named "output". The folder contains the individual regions (Descriptor, GBE, Intel ME, BIOS) and the Map file (**<FILENAME>. MAP**).

The xml file contains the current Intel ME parameters.

The Map file contains the start, end, and length of each region.

3.6 More Examples of spsFITC CLI

Note: If using paths defined in the KIT, be sure to put "" around the path as the spaces cause issues.

Take an existing (dt_ori.bin) image and put in a new BIOS binary:

```
spsFITC.exe /b /bios "..\..\..\Image Components\BIOS\BIOS.ROM" <file.bin
or file.xml>
```

Take an existing image and put in a different Intel® ME region:

```
spsFITC.exe /b /me ".\..\..\Image
Components\Firmware\PCH_REL_BYP_ME_UPD_PreProduction_0xB0.BIN" <file.bin
or file.xml>
```

Take an existing image and put in a different Intel® ME region:

```
spsFITC.exe /b /gbe ".\..\..\Image
Components\GbE\82577_A2_CPT_A1_VER0PT21_MOBILE.bin" <file.bin or file.xml>
```

4 MESDC Tool

4.1 MESDC Tool Overview

Intel ME Debug and Compliance Console (MESDC) is an application to diagnose Intel ME Firmware. This tool performs run-time tests, receives Intel ME Firmware Status (defined in [ME_BIOS_Interface]) and Trace Logs, processes and presents received data.

MESDC consists of the following elements:

SOFTWARE

- MESDC.exe – Windows (Net Framework 4.0) based application;
- Aadrvark.dll – dynamic library to communicate with Aardvark adapter;
- Common.dll – dynamic library with MESDC and MESDC Agent common functions
- Heci.dll – dynamic library to communicate with ME FW via HECI
- TransportRmcpp.dll – dynamic library to communicate with BMC
- Total Phase USB driver for Windows;

HARDWARE

- USB A(M)-B(M) cable;
- Aardvark* I^{2}C* host adapter;
- 3-wire SMBus cable;
- MDDD Mobile DIMM Adapter / MDDD Desktop DIMM Adapter;

4.2 Installation and Initialization

This section covers installing software and hardware components described in the following sections.

4.2.1 Software installation – USB Driver

To install the appropriate USB communication driver under Windows*, use the Total Phase USB Driver Installer before plugging in any device. The driver installer can be found either on the CD-ROM (use the HTML based guide that is opened when the CD is first loaded to locate the Windows installer), or in the Downloads section of the Aardvark adapter product page on the Total Phase website.

After the driver has been installed, plugging in an Aardvark adapter for the first time will cause the adapter to be installed and associated with the correct driver.

4.2.2 Software installation – MESDC Application

MESDC application is compatible with Windows XP, Windows 7 x86/x64, Windows 8 x64, Windows Server 2008 R2 SP1 x64, Windows Server 2012 x64 with installed .NET Framework 4.0 or higher.

To communicate with Intel ME FW MESDC uses Aardvark.dll, Heci.dll or TransportRmcpp.dll library which should be located in one of the following places:

- The directory from which the application binary was loaded;
- The application’s directory;
- System directory, for example, C:\Windows\System32;
- The windows directory, for example, C:\Windows
- The directory listed in the PATH environmental variable.

4.2.3 Intel ME Image Preparation

When building the image with spsFITC check that the below soft straps are set correctly:

- Intel ME SMBus (aka SMT1 or Host SMBus) I2C* Address Enable is TRUE
- Intel ME SMBus (aka SMT1 or HostSMBus) I2C Address is 0x48. If 0x48 conflicts with another device on SMBus, this address could be set to a different value. If set to value other than 0x48, specify it in the “ME I2C Address [hex]” field in MESDC console under the SMBus Settings in the Configuration menu.
- ME Debug SMBus Emergency Mode Enable = false (default). This enables diagnostics service in normal mode. Set it to True to enable diagnostic service emergency mode that allows monitoring the very initial steps of Intel ME firmware boot.
  Note that Emergency Mode Enabled enforces non-buffered message transmission and fixed set of event types is traced.
- ME Debug SMBus Emergency Mode Address = 0x38. If 0x38 conflicts with another device on SMBus, this address could be set to a different value. If set to value other than 0x38, specify it in the “MESDC I2C Address [hex]” field in MESDC console under the SMBus Settings in the Configuration menu.

“ME Debug SMBus Emergency Mode Enable” determines the mode of the diagnostics service operation for both the recovery boot loader and operational code. “ME Debug SMBus Emergency Mode Address” applies to the recovery boot loader and operational code in emergency mode. For operational code in normal mode, a non-zero address has still to be set in straps to enable the diagnostics service, but the “ME Debug SMBus Emergency Mode Address” and other diagnostics settings are to be configured from spsFITC (the Configuration-> MESDC-> MDES configuration file menu).The table below summarizes the modes of the diagnostics service operation, depending on the “ME Debug SMBus Emergency Mode Address” and “ME Debug SMBus Emergency Mode Enable” strap settings:

Intel ME Debug SMBus Emergency Mode Enable	Intel ME Debug SMBus Address	Operation Mode
false (zero)	Zero	Diagnostic service is disabled – this is the default setting that shall be used on end-user platforms and this is the default in the release package provided by Intel. With this setting, the diagnostic service can only be enabled by an update of this strap (that is, setting the address to a non-zero value). Such an update of a strap is possible only after overriding the security limitations defined in SPI flash descriptor with use of the Security Override Jumper described in section 3.3.
false (zero)	Non-zero (default is 0x00)	Diagnostics service is enabled in normal mode – the default set of diagnostic event messages is reported by Intel ME FW in buffered mode, with the exception of the initial MDDD messages sent during FW boot by ROM code. This mode shall be normally enabled during the integration phase by the OEM. This mode makes it possible to perform all the Compliance tests. The diagnostic service functions outside of the recovery boot loader can still be reconfigured via the Diagnostic Console in run-time (or via spsFITC). The configuration used by the recovery boot loader is fixed and cannot be reconfigured.
true (one)	Non-zero (default is 0x00)	Diagnostics service is enabled in emergency mode – a fixed set of diagnostic event message types is reported by Intel ME FW in blocking mode (non-buffered). This mode is mostly useful only when debugging very low-level problems related to Intel ME FW boot and it shall not be used when Intel ME FW is able to normally boot and start during the integration phase. The diagnostic service functions in the emergency mode are hardcoded and cannot be reconfigured.

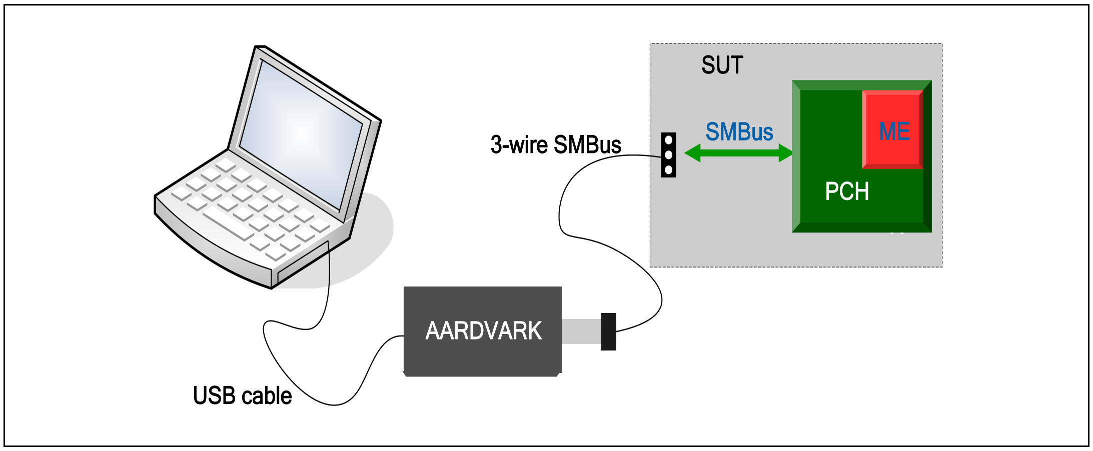
4.2.4 Hardware Configuration

4.2.4.1 3-wire SMBus Configuration

The following steps must be taken to set proper SMBus configuration:

1. Disconnect AC power,
2. Connect Aardvark adapter to 3 wire SMBus,
3. Connect Aardvark adapter to host computer,
4. Connect AC power.

Figure 4-1. 3-Wire SMBus Configuration



4.2.4.2 IPMB Configuration

The following steps must be taken to set proper IPMB configuration:

1. Disconnect AC power,
2. Connect Aardvark adapter to 3 wire SMLink0,
3. Connect Aardvark adapter to host computer,
4. Connect AC power.

4.2.4.3 RMCP+ Configuration

The following steps must be taken to set proper RMCP+ configuration:

1. Configure BMC IP address,
2. Add new BMC user: username and password must be non-empty,
3. Connect platform to LAN network.

4.2.4.4 Remote Agent Configuration

The following steps must be taken to set proper Remote Agent configuration:

1. Install Remote Agent on tested platform and copy required SSL keys. Remote Agent application is compatible with Windows Server 2008 R2 SP1 x64, Windows Server 2012 x64, Red Hat 6.4 x86/x64
2. Connect platform to LAN network

4.2.4.5 HECI Configuration

The following steps must be taken to set proper HECI configuration:

1. MESDC Application should be run on tested platform.

4.3 MESDC Application

4.3.1 Introduction

Intel ME Debug and Compliance Console (MESDC) is an application to diagnose Intel ME Firmware.

These are the basic steps for start working with MESDC application. For more detailed information, please refer to the specific sections in this manual.

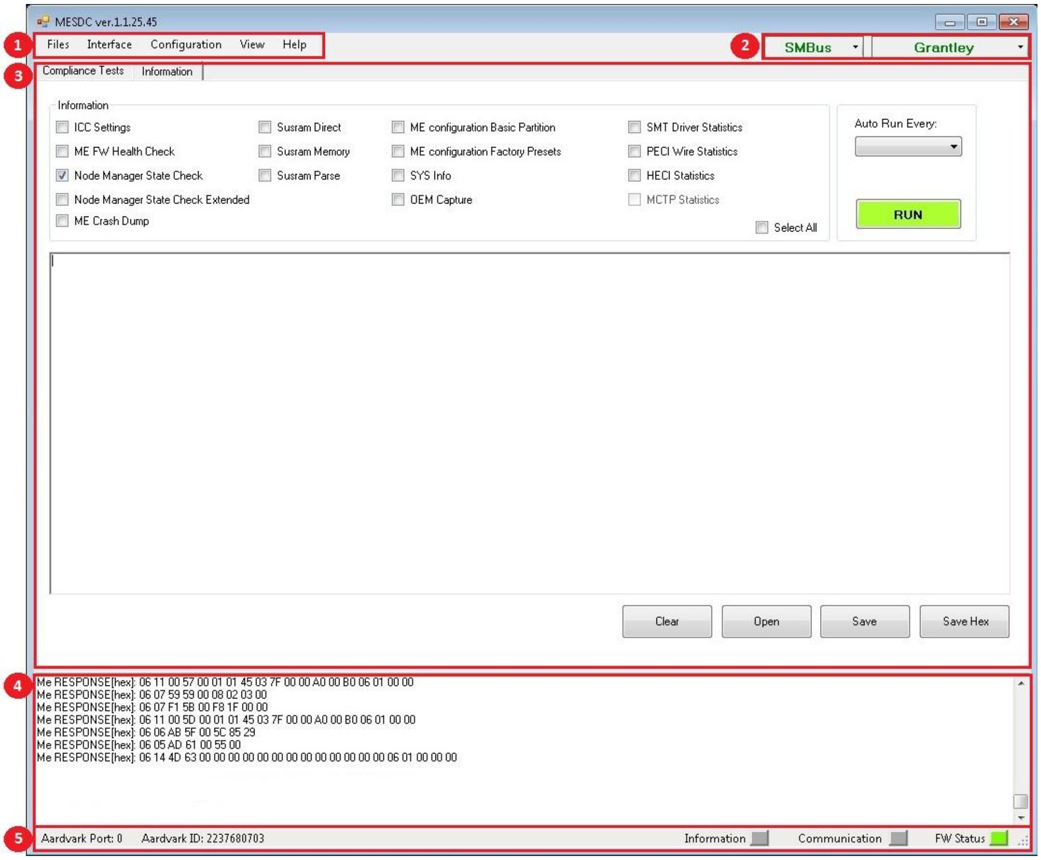
Select Interface (4.4 Initialization of MESDC Application)

For proper MESDC initialization current Intel ME FW Configuration must be read. (more details in section 4.3.3.)

The main application form is divided into five sections:

1. Main menu
2. Toolbar
3. Diagnostic modules in tabbed pages
4. Operation Log, which keeps track of all operations.
5. Statusbar

4.



```
Me RESPONSE[hex]: 06 11 00 57 00 01 01 45 03 7F 00 00 A0 00 B0 06 01 00 00
Me RESPONSE[hex]: 06 07 59 59 00 08 02 03 00
Me RESPONSE[hex]: 06 07 F1 5B 00 F8 1F 00 00
Me RESPONSE[hex]: 06 11 00 5D 00 01 01 45 03 7F 00 00 A0 00 B0 06 01 00 00
Me RESPONSE[hex]: 06 06 AB 5F 00 5C 85 29
Me RESPONSE[hex]: 06 05 AD 61 00 55 00
Me RESPONSE[hex]: 06 14 4D 63 00 00 00 00 00 00 00 00 00 00 00 00 00 06 01 00 00 00
```

Aardvark Port: 0   Aardvark ID: 2237680703

4.3.1.1 Main menu

MESDC main menu contain 5 items

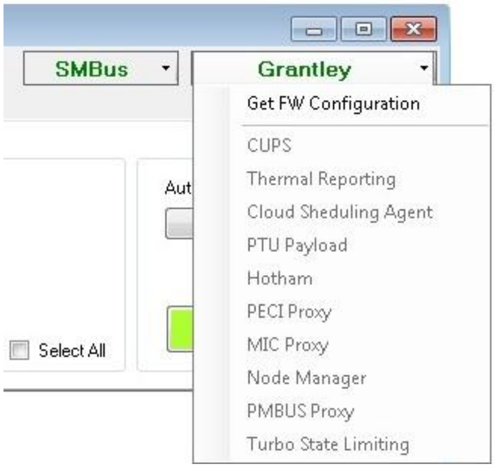
1. Files
    - 1.1. Save
        - 1.1.1. Gather Debug Information: Run all available reports and save their results to zip file, additionally saves operational log.
        - 1.1.2. Save Current Information: Gets logs from Reports, Trace Console and Operational log and save them to zip file.
        - 1.1.3. Save trace logs: save trace logs into a file for future analysis
        - 1.1.4. Save information logs: save information logs into a file for future analysis
    - 1.2. FW Configuration:
        - 1.2.1. Get FW Configuration: read FW configuration from connected platform
        - 1.2.2. Override All Features to ON: enables all features in MESDC for specific platform type, available only for advanced view

    1.3. Exit: exit the MESDC application

2. Interface: MESDC can communicate with Intel ME through one of the flowing interfaces:
    2.1. SMBus
    2.2. MEI (HECI)
    2.3. IPMI (RMCP+)
    2.4. IPMB (Aardvark)
    2.5. Agent (TCP)
3. Configuration: details configuration settings (see section 4.4 Initialization of MESDC Application):
    3.1. SMBus Aardvark
    3.2. MEI (HECI)
    3.3. IPMI (RMCP+)
    3.4. IPMB (Aardvark)
    3.5. Agent (TCP)
    3.6. AutoConnect: Available only for advanced view
    3.7. Auto Get FW Configuration: Available only for advanced view
    3.8. Auto Fetch Report: Available only for advanced view
4. View
    4.1. Basic: Basic view will only provide 2 modules (Compliance Test and Information) for simple usage of the MESDC
    4.2. Advanced: Advanced view will provide all the stages for advanced user
5. Help
    5.1. About
    5.2. Platform Type Information

4.3.1.2 MESDC Toolbar

MESDC shows state of selected interface and identified platform type with current ME features.



4.3.1.3 Diagnostic modules

In Basic View MESDC application consists of two stages of operation to diagnose Intel ME FW at run-time. All modules are described in section 4.5 MESDC Application Modules.

MESDC modules in Basic View:

1. Compliance Tests
2. Information

In Advanced View MESDC application consists of six stages of operation to diagnose Intel ME FW at run-time. All modules are described in section 4.5 MESDC Application Modules.

MESDC modules in Advanced View:

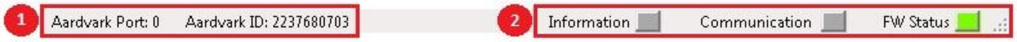
1. Trace Console
2. Communication
3. Compliance Tests
4. IDLM
5. Information
6. Volumetric
7. MIC Monitoring

4.3.1.4 Operation Log

Operation Log keeps track of all received response frames, communication errors and Intel ME Firmware state.

4.3.1.5 Statusbar

StatusBar contains basic information about chosen interface (1), read FW Status and selected Auto Run (2).



Aardvark Port: 0 Aardvark ID: 2237680703

FW Status indicator colors:

- green – FW active
- yellow – recovery, test or init state
- red- disabled, transition, wait or reset state
- grey – unrecognized state

Information and Communication indicators blink with every iteration of Auto Run feature.

4.3.2 Autorun features

Autorun features lunch when MESDC starts and provide basic information about current platform. They can be disabled only in advanced view. Received ME features state and platform type allows to enable specified reports, diagnostic commands and compliance tests. When ME configuration status is unavailable there is possibility to override all features for specified platform.

4.3.2.1 AutoConnect

Auto Connect: if this is selected, at the time launch of MESDC, MESDC will try to connect ME through previous setting of interface automatically.

In basic view there is no possibility to disable this feature.

4.3.2.2 AutoGet

Auto Get FW configuration: If this is selected, at the time launch of MESDC, MESDC will get the FW configuration automatically and apply that to the compliance. If user wants to override this configuration, user can use following option in the menu to override the configuration.

In basic view there is no possibility to disable this feature.

4.3.2.3 Auto Fetch Report

Fetch Report: If this is selected, at the time launch of MESDC, MESDC will run the report in information tag automatically.

In basic view there is no possibility to disable this feature.

4.3.2.4 Override Features Functionality

This functionality allows to override all features to ON for specified platform type. It can be used when there is no possibility to read ME FW configuration and platform depends diagnostic commands, compliance tests or information reports are unavailable.

4.3.2.5 Setting MDES logging interface

Intel ME Firmware is able to send MESDC Traces via Host SMBus or write them to SPI Flash. MDES Logging Interface option can be enabled/disabled by changing proper MFS MDES fields in the XML configuration file or via diagnostic command. For RMCP+ or HECI interfaces Flash Logging enables when ‘Write MESDC Traces’ checkbox is selected. When SMBus Interface is chosen, sending MESDC Traces via Host SMBus are automatically enabled. To apply new MDES Logging Interface settings ME Reset is required.

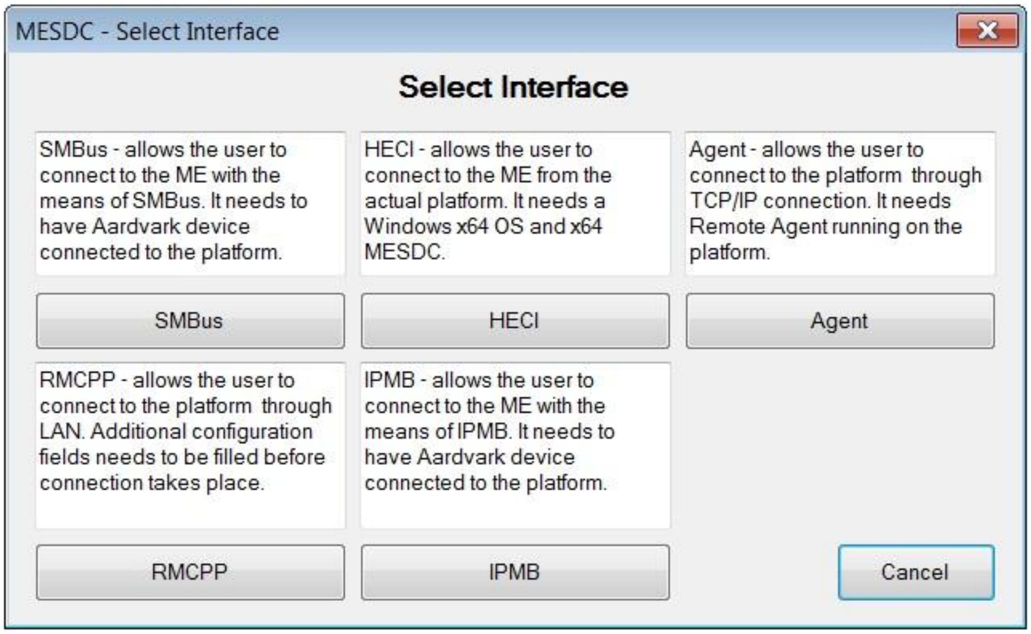
In basic view there is no possibility to disable this feature.

4.4 Initialization of MESDC Application

There are five communication options between MESDC application and target testing system:

1. SMBUS with Aardvark
2. MEI (HECI) on local platform
3. IPMB with Aardvark
4. RMCP+
5. Remote Agent

The first time launch the MESDC tool, following dialog will pop-up for user to choose which interface user wants to connect with Intel ME



SMBus - allows the user to connect to the ME with the means of SMBus. It needs to have Aardvark device connected to the platform.

HECI - allows the user to connect to the ME from the actual platform. It needs a Windows x64 OS and x64 MESDC.

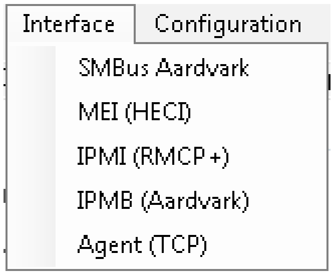
Agent - allows the user to connect to the platform through TCP/IP connection. It needs Remote Agent running on the platform.

RMCPP - allows the user to connect to the platform through LAN. Additional configuration fields needs to be filled before connection takes place.

IPMB - allows the user to connect to the ME with the means of IPMB. It needs to have Aardvark device connected to the platform.

When user selects one of the interfaces, some detail level of the setting may be needed for MESDC to work with that interface. The detail configuration setting is available in the configuration menu

Figure 4-2. Communication Configuration for MESDC



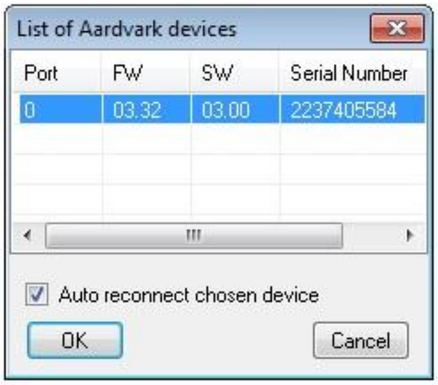
4.4.1 SMBus

To communicate with Intel ME FW via SMBUS, a proper Aardvark adapter has to be chosen. After launching the MESDC application ConfigAA form must be opened from menu Interface->SMBus Aardvark. There is a list of all available Aardvark adapters connected to the computer. If there are no available units, then application displays warning message: “Aardvark ERROR: NOT_CONNECTED Aardvark adapter”. The list provides the following information:

- Port – the port that Aardvark adapter occupies, zero-based number;
- FW – Firmware version of Aardvark adapter;
- SW – Software version of Aardvark adapter;

- Serial Number of Aardvark adapter.

Figure 4-3. Aardvark Adapter Choosing form of MESDC


List of Aardvark devices

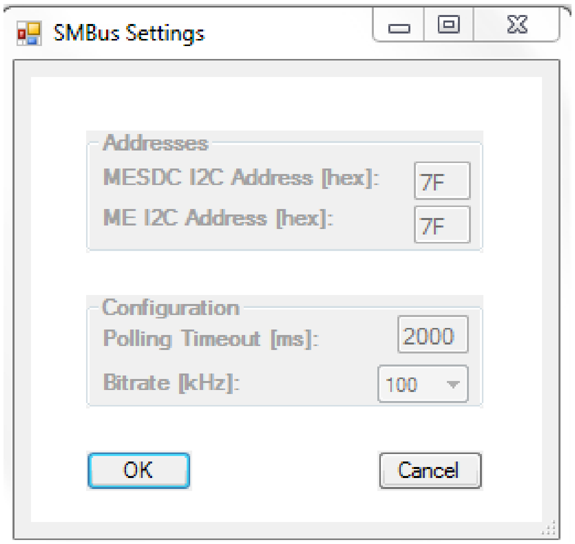
Port	FW	SW	Serial Number
0	03.32	03.00	2237405584

☑ Auto reconnect chosen device

It is possible to reinitialize the Aardvark adapter.

After successful connection information about connection parameters are displayed on StatusBar: Aardvark Port, Aardvark ID

User can change some of the SMBus configuration at SMBus Setting dialog. The default setting would work for most of the cases.


SMBus Settings

Addresses

MESDC I2C Address [hex]: 7F

ME I2C Address [hex]: 7F

Configuration

Polling Timeout [ms]: 2000

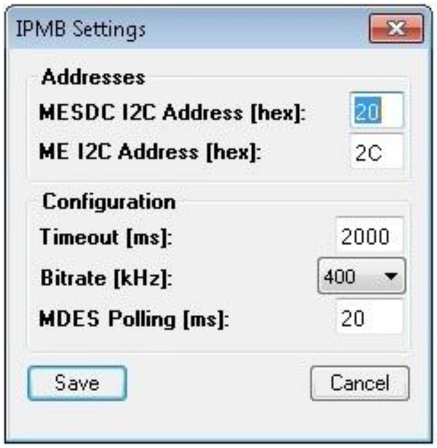
Bitrate [kHz]: 100

4.4.2 IPMB

To communicate with Intel ME FW via IPMB, a proper Aardvark adapter has to be chosen. After launching the MESDC application ConfigAA form must be opened from menu Interface->IPMB Aardvark. There is a list of all available Aardvark adapters connected to the computer. If there are no available units, then application displays warning message: “Aardvark ERROR: NOT_CONNECTED Aardvark adapter”. The list provides the following information:

- Port – the port that Aardvark adapter occupies, zero-based number;
- FW – Firmware version of Aardvark adapter;
- SW – Software version of Aardvark adapter;
- Serial Number of Aardvark adapter.

User can change some of the IPMB configuration at IPMB setting dialog. The default setting would work for most of the cases.



Addresses

MESDC I2C Address [hex]: 20

ME I2C Address [hex]: 2C

Configuration

Timeout [ms]: 2000

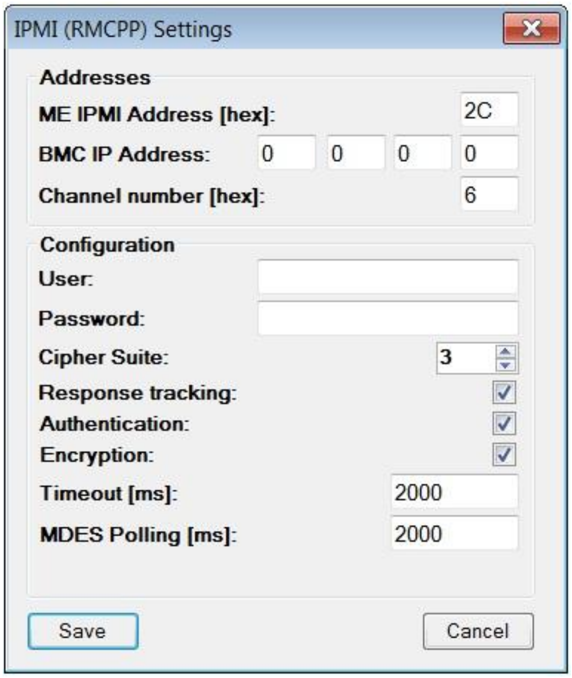
Bitrate [kHz]: 400

MDES Polling [ms]: 20

4.4.3 RMCP+

To communicate with Intel ME FW via IPMI proper BMC configuration is needed. User should provide ME IPMI and BMC addresses, account data and encryption settings in the IPMI configuration section to make MESDC work through RMCP+ interface.

Figure 4-4. IPMI Setting for MESDC



IPMI (RMCPP) Settings

Addresses

ME IPMI Address [hex]: 2C

BMC IP Address: 0 0 0 0

Channel number [hex]: 6

Configuration

User:

Password:

Cipher Suite: 3

Response tracking: ☑

Authentication: ☑

Encryption: ☑

Timeout [ms]: 2000

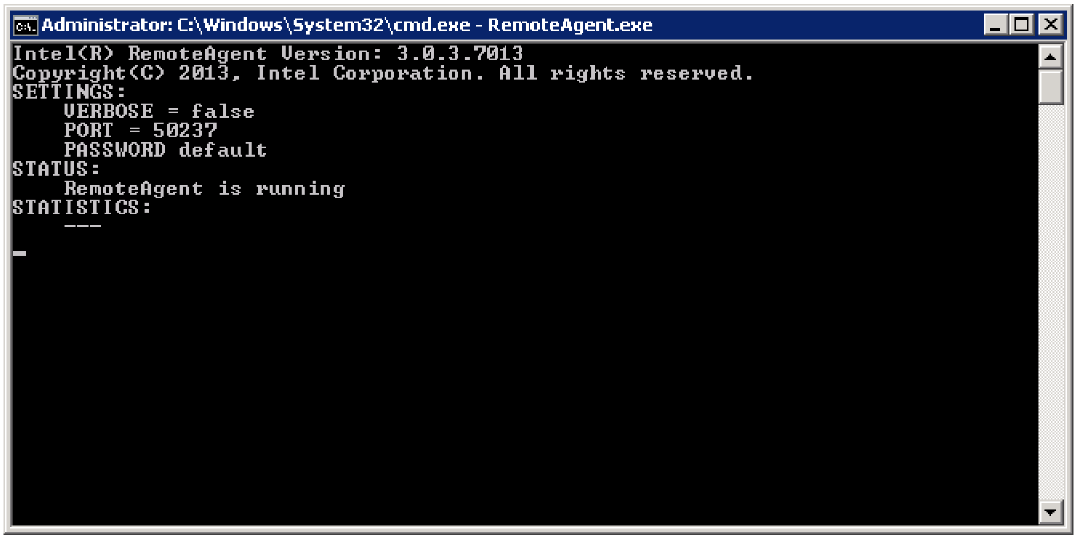
MDES Polling [ms]: 2000

After successful connection information about connection parameters are displayed on StatusBar: IP and user

4.4.4 Remote Agent

To communicate with Intel ME Firmware via HECI interface using Remote Agent, RemoteAgent.exe application must be running on the platform.

Although Remote Agent uses HECI interface only diagnostic commands can be sent. HECI commands are not available.

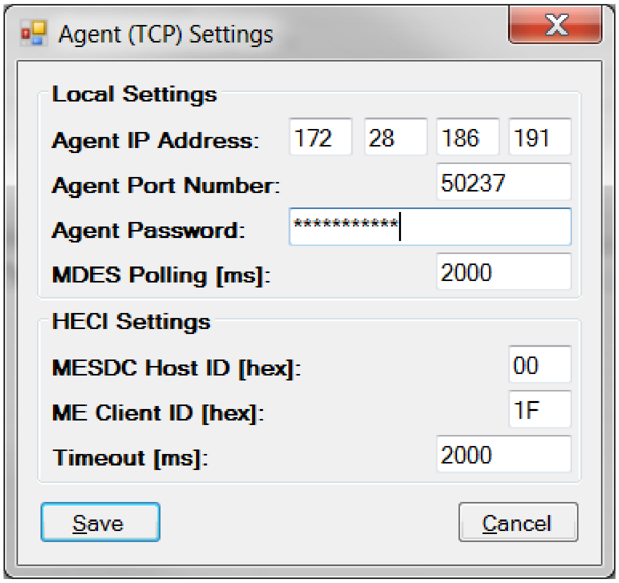

Administrator: C:\Windows\System32\cmd.exe - RemoteAgent.exe

```
Intel(R) RemoteAgent Version: 3.0.3.7013
Copyright(C) 2013, Intel Corporation. All rights reserved.
SETTINGS:
    VERBOSE = false
    PORT  = 50237
    PASSWORD default
STATUS:
    RemoteAgent is running
STATISTICS:
    ---
_
```

Remote Agent has the following optional startup parameters:

- -?|-H|-HELP
  Displays help screen.
- -VER|-VERSION
  Displays version information.
- -V|-VERB|-VERBOSE [filename]
  Displays the debug information of the tool.
  Debug information is shown in the form of ten raw frames lately exchanged with client. Additionally if filename is specified a full log file from Remote Agent work is created.
- -P|-PORT [number]
  Sets the port on which Server will listen for incoming requests.
  If no port number is specified the default value will be used.
- -PASS|-PASSWORD [password]
  If password is specified overwrites the default Agent password,
  else a prompt for password is shown first.

To connect with Remote Agent following TCP configuration has to be done.


Agent (TCP) Settings

Local Settings

Agent IP Address: 172 28 186 191

Agent Port Number: 50237

Agent Password: ***********

MDES Polling [ms]: 2000

HECI Settings

MESDC Host ID [hex]: 00

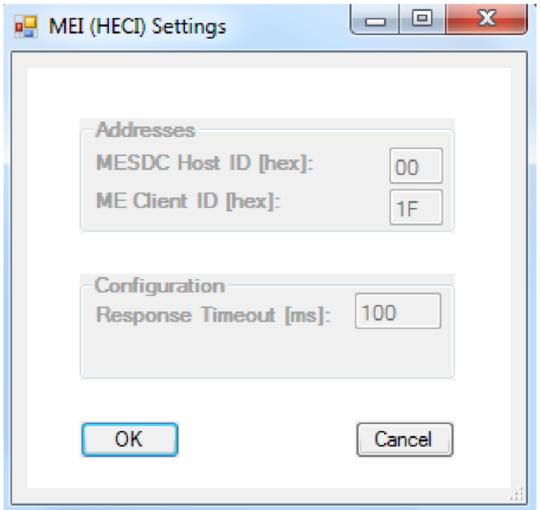
ME Client ID [hex]: 1F

Timeout [ms]: 2000

After successful connection IP Address and Port Number are displayed on the Status Bar.

4.4.5 HECI

User can change some of the HECI configuration at HECI setting dialog. The default setting would work for most of the cases.


MEI (HECI) Settings

Addresses

MESDC Host ID [hex]: 00

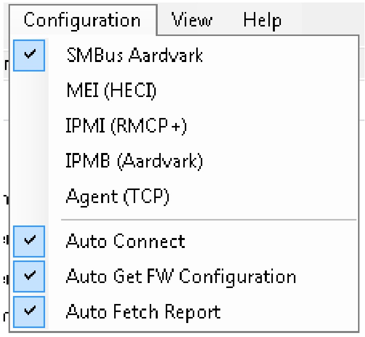
ME Client ID [hex]: 1F

Configuration

Response Timeout [ms]: 100

4.5 MESDC Application Modules

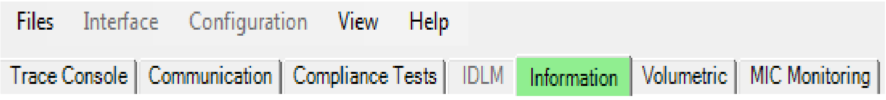
Configuration



MESDC application consists of stages of operation to diagnose Intel ME FW at run-time.

If any of modules below is working, tab page header of this module is highlighted.

Example for "Information" tab while reports are gathering:

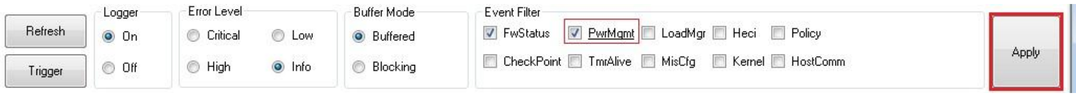


4.5.1 Trace Console

In this mode of operation, MESDC receives and parses run-time SMBus messages. Trace Console allows analyzing a firmware flow, in particular the initialization sequence. There are several basic messages types: Firmware Status, Checkpoints, Power Management, Timer Alive, Load Manager, Misconfiguration, HECI, Kernel, Policy and Hostcomm (Appendix - MDES messages).

MESDC is configured as an SMBus Slave. User can use the Trace Console tab to configure the ME Debug Event Service (MDES) in Intel ME FW so as to report appropriate types of messages for an analysis. User can set logger on/off, set error level to log Critical, High, Low Errors, , or Information, set event filter to log selected events. User can set buffer mode to determine the way how MDES reports the messages. By default the messages are sent in buffer mode in order not to introduce much load to the Intel ME FW. The blocking mode is designed to help investigate problems during boot time. To apply changes in MDES ME FW need to be restarted. Settings which are not applied are underlined. User need to click Apply button to set new logger configuration.

User can press the 'Refresh' button to receive the logger configuration from Intel ME FW.



Logger: ● On ○ Off

Error Level: ○ Critical ○ Low ○ High ● Info

Buffer Mode: ● Buffered ○ Blocking

Event Filter: ☑ FwStatus ☑ PwrMgmt ☐ LoadMgr ☐ Heci ☐ Policy ☐ CheckPoint ☐ TmrAlive ☐ MisCfg ☐ Kernel ☐ HostComm

User can start/stop/clear displaying messages by pressing the 'Capture', 'Stop', 'Clear' buttons. User can switch between tabs and perform other operations while a trace is being captured.

1. User can save parsed messages in a file by selecting the 'Files->Save Logs' menu. MESDC will create a file called *log_YYYY_MM_DD_HH_MM_SS.rtf* in the same folder where MESDC tool is located.

User can set SMBus settings such as MESDC and Intel ME I2C address by selecting the 'Configuration->SMBus Settings' menu. There are following default values of I2C addresses (7-bit format):

- MESDC trace address = 0x38;
- Intel ME address = 0x48;

Note: The filter settings in the MDES emergency mode are fixed and can't be changed with use of MESDC.

Note: CheckBox TmrAlive is available only for SMBus.

4.5.1.1 Trace Logger Mode of Data Reception

4.5.1.2 Majority of the logs will be output during boot time. It is allowed to catch trace logs via Ipmb, Rmcp+, Heci interface. In this case special settings in MDES are needed (see section 4.3.2.5 Setting MDES logging interface

).).

In order to receive the log at boot time via SMBus:

1. Once all the hardware is set up and Intel ME image is prepared to enable diagnostic service, start the MESDC on a Host Platform while DUT is off (G3 or S5).
1. Press the 'Capture' button to enable displaying logs.
2. Power on the DUT and you will see the boot time logs displayed on MESDC.

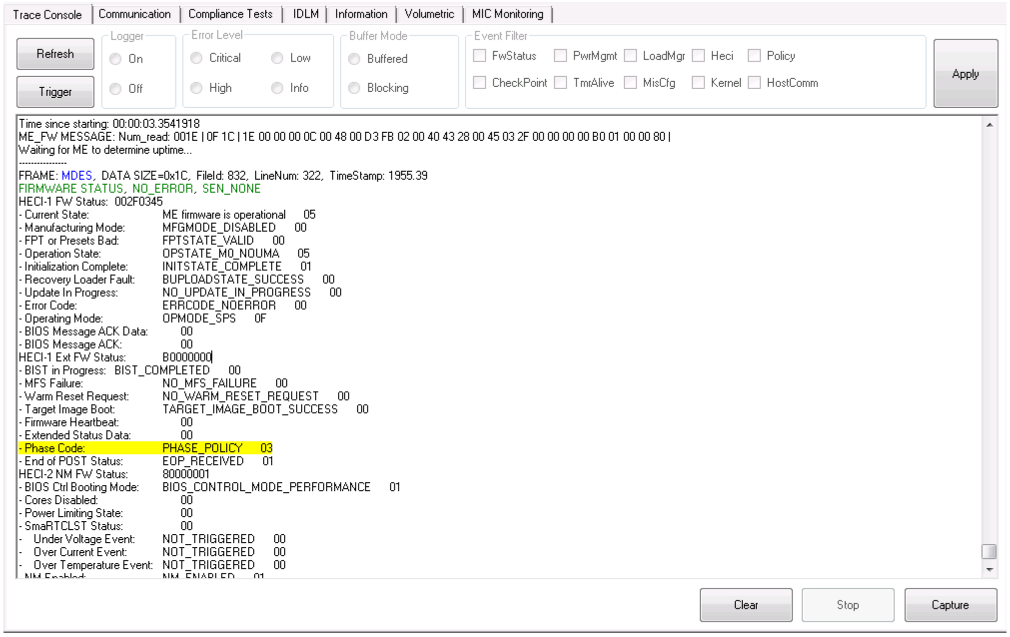
In order to receive the log at boot time via RMCP+/HECI/Agent:

1. Press the 'Capture' button to enable displaying logs.
2. If all logs are read MESDC automatically STOP.

Note: Changes in FwStatus Events (Firmware Status frame)are highlighted.

3.

Figure 4-5.MESDC GUI – Trace Console Tabbed Page

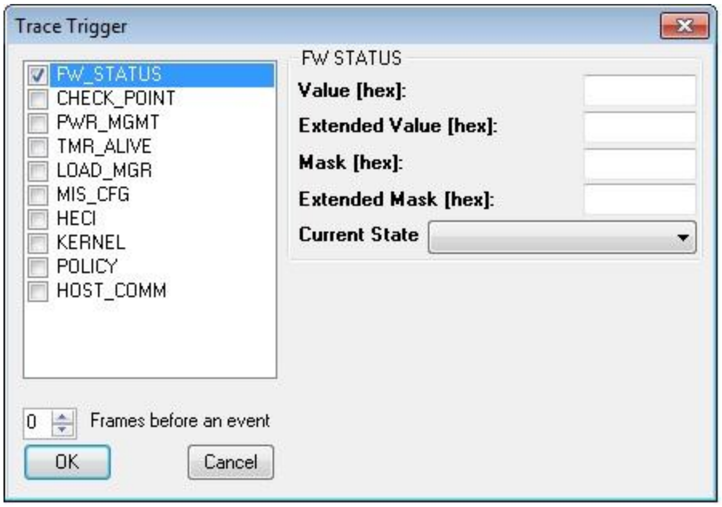


```
Time since starting: 00:00:03.3541918
ME_FW MESSAGE: Num_read: 001E | 0F 1C | 1E 00 00 00 0C 00 48 00 D3 FB 02 00 40 43 28 00 45 03 2F 00 00 00 00 B0 01 00 00 80 |
Waiting for ME to determine uptime...
---------------
FRAME: MDES,  DATA SIZE=0x1C,  FileId: 832,  LineNum: 322,  TimeStamp: 1955.39
FIRMWARE STATUS,  NO_ERROR,  SEN_NONE
HECI-1 FW Status:  002F0345
- Current State:              ME firmware is operational     05
- Manufacturing Mode:         MFGMODE_DISABLED     00
- FPT or Presets Bad:         FPTSTATE_VALID     00
- Operation State:            OPSTATE_M0_NOUMA     05
- Initialization Complete:    INITSTATE_COMPLETE     01
- Recovery Loader Fault:      BUPLOADSTATE_SUCCESS     00
- Update In Progress:         NO_UPDATE_IN_PROGRESS     00
- Error Code:                 ERRCODE_NOERROR     00
- Operating Mode:             OPMODE_SPS     0F
- BIOS Message ACK Data:         00
- BIOS Message ACK:              00
HECI-1 Ext FW Status:         B0000000
- BIST in Progress:   BIST_COMPLETED     00
- MFS Failure:                NO_MFS_FAILURE     00
- Warm Reset Request:         NO_WARM_RESET_REQUEST     00
- Target Image Boot:          TARGET_IMAGE_BOOT_SUCCESS     00
- Firmware Heartbeat:            00
- Extended Status Data:          00
- Phase Code:                 PHASE_POLICY     03
- End of POST Status:         EOP_RECEIVED     01
HECI-2 NM FW Status:          80000001
- BIOS Ctrl Booting Mode:     BIOS_CONTROL_MODE_PERFORMANCE     01
- Cores Disabled:                00
- Power Limiting State:          00
- SmaRTCLST Status:              00
-   Under Voltage Event:      NOT_TRIGGERED     00
-   Over Current Event:       NOT_TRIGGERED     00
-   Over Temperature Event:   NOT_TRIGGERED     00
- NM Enabled:                 NM_ENABLED     01
```

The log can be triggered if user enable trigger from the interface Trigger

Trigger can be enabled when one of the trigger condition is satisfied. FW_Ststus trigger can be configured in three options

1. Exactly Value customer want to trigger. Log will start when FW_STS is the same as customer configured



☑ FW_STATUS
☐ CHECK_POINT
☐ PWR_MGMT
☐ TMR_ALIVE
☐ LOAD_MGR
☐ MIS_CFG
☐ HECI
☐ KERNEL
☐ POLICY
☐ HOST_COMM

0 Frames before an event

2. Mask. Log will be triggered when the bits user configured match the mask
3. FW status State, Log will be triggered when FW is in the state user is configured

The log will be triggered while one of the conditions gets satisfied.

4.5.2 Communication

4.5.2.1 Diagnostics tab

In this mode MESDC start a conversation with Intel ME FW to execute diagnostics by Intel ME FW and receives a response with results.

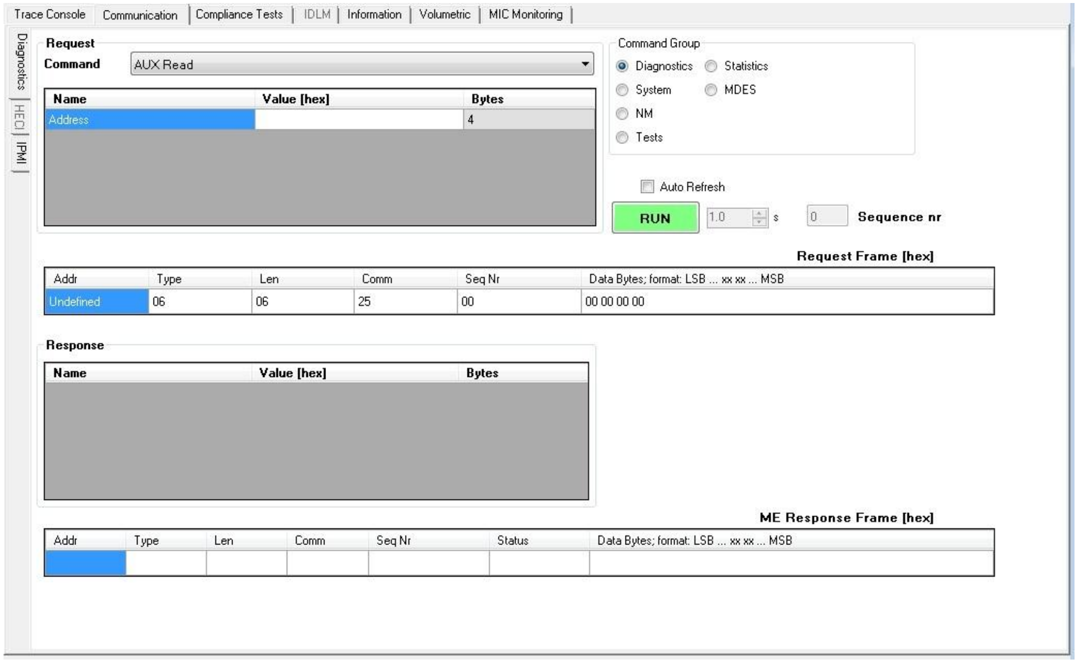
Diagnostic tab page consists of the following elements:

Element	Description
Command Group	Command Group is used to select a group with the command to be executed. User can select one of the following groups: - Diagnostics - System - Intel Node Manager - Tests - Statistics - MDES
Request	Request group is used to configure SMBus request. User can choose a proper request (like AUX or Memory read) from Command drop down menu and specify the value in hex. Then a request is sent to Intel ME by clicking RUN button
Request Frame	Displays the entire Request: Addr - SMBus slave address of Intel ME Type - type of SMBus message Len - number of data bytes in SMBus frame Comm - command ID Seq.Nr - sequence number of request Data Bytes - data in SMBus frame
Response	Displays the result of executed diagnostics in a decoded format
Intel ME Response Frame	Displays the entire Intel ME Response: Addr - SMBus slave address of receiver Type - type of SMBus message Len - number of data bytes in SMBus frame Comm - command ID Seq.Nr - sequence number of response Status - generic status code Data Bytes - data in SMBus frame
Auto Refresh	User can keep sending MESDC command by checking Auto Refresh. If Auto Refresh is enabled Communication indicator on statusbar blinks green every iteration.
Operation Log	Displays communication errors

More detail MESDC supported commands are available in Appendix B.

Note: 0x3CC3A55A is the magic number sending to Intel ME for Intel ME reset command.

Figure 4-6. MESDC GUI–Communication Page



Request

Command AUX Read

Name	**Value [hex]**	**Bytes**
Address		4

Command Group

● Diagnostics ○ Statistics
○ System ○ MDES
○ NM
○ Tests

☐ Auto Refresh

RUN 1.0 s 0 **Sequence nr**

Request Frame [hex]

Addr	Type	Len	Comm	Seq Nr	Data Bytes; format: LSB ... xx xx ... MSB
Undefined	06	06	25	00	00 00 00 00

Response

Name	**Value [hex]**	**Bytes**

ME Response Frame [hex]

Addr	Type	Len	Comm	Seq Nr	Status	Data Bytes; format: LSB ... xx xx ... MSB

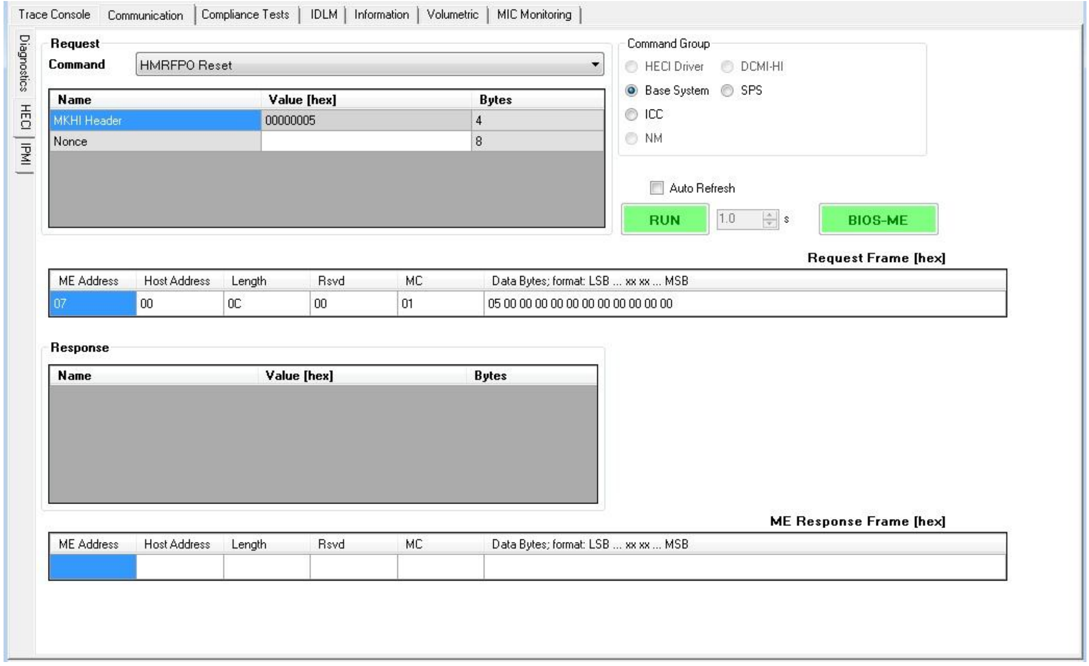
4.5.2.2 HECI tab

HECI tab allows user to send HECI command to Intel ME FW.

HECI tab page consists of the following elements:

Element	**Description**
Command Group	Command Group is used to select a group with the command to be executed. User can select one of the following groups: - Base System - ICC - Intel Server Platform Services
Request	Request group is used to configure HECI request. User can choose a proper request (like ICC Set Clock Enables) from Command drop down menu and specify the value in hex. Then a request is sent to Intel ME by clicking RUN button
Request Frame	Displays the entire Request: Me Address – Intel ME client ID, part of HECI Header Host Address – Host Client ID, part of HECI Header Length - number of data bytes in HECI frame, part of HECI Header Rsvd – reserved bits MC – Message Complete, part of HECI Header Data Bytes - data in HECI frame

Element	Description
Command Group	Command Group is used to select a group with the command to be executed. User can select one of the following groups: - Base System - ICC - Intel Server Platform Services
Response	Displays the result of executed command in a decoded format
Intel ME Response Frame	Displays the entire Intel ME Response: Me Address – ME client ID, part of HECI Header Host Address – Host Client ID, part of HECI Header Length - number of data bytes in HECI frame, part of HECI Header Rsvd – reserved bits MC – Message Complete, part of HECI Header Data Bytes - data in HECI frame
ME-BIOS	User can simulate Intel ME-BIOS communication by invoking 'ME-BIOS' button. MESDC application sends commands: - MKHI Get FW Version - Get Intel ME-BIOS Interface - HMRFPO Lock - End Of Post
Auto Refresh	Checking "Auto Refresh" checkbox will make the MESDC automatically send the command every n seconds, where n is a specified time interval left to the Run button. If Auto Refresh is enabled Communication indicator on statusbar blinks green every iteration.
Operation Log	Displays communication errors



Request

Command HMRFPO Reset

Name	Value [hex]	Bytes
MKHI Header	00000005	4
Nonce		8

Request Frame [hex]

ME Address	Host Address	Length	Rsvd	MC	Data Bytes; format: LSB ... xx xx ... MSB
07	00	0C	00	01	05 00 00 00 00 00 00 00 00 00 00 00

Response

Name	Value [hex]	Bytes

ME Response Frame [hex]

ME Address	Host Address	Length	Rsvd	MC	Data Bytes; format: LSB ... xx xx ... MSB

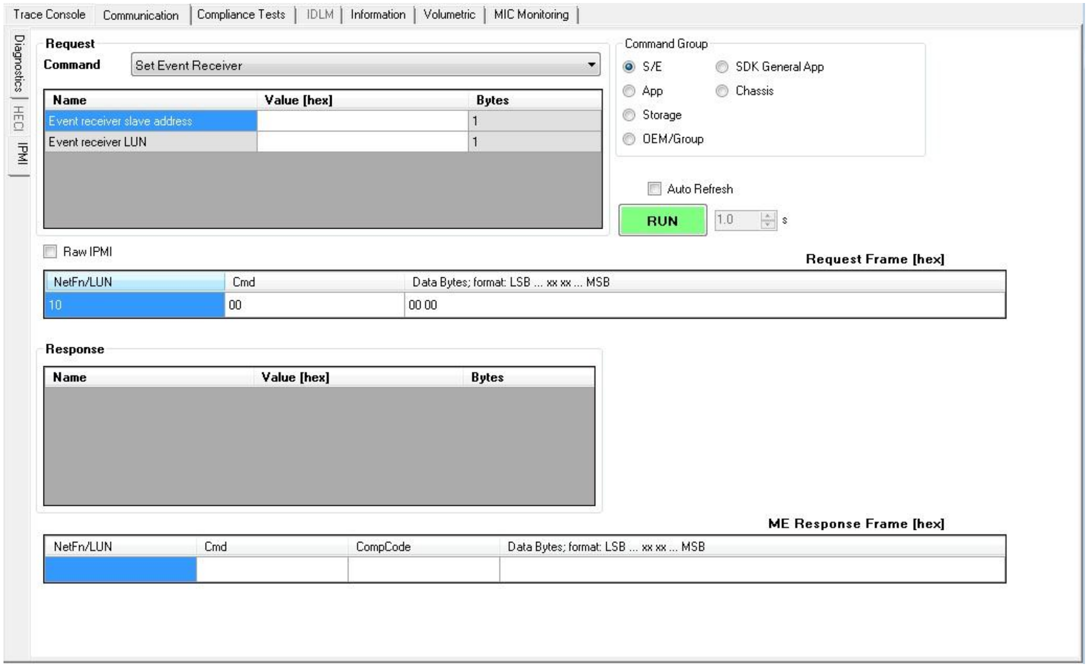
4.5.2.3 IPMI tab

IPMI tab allows user to send IPMI command to Intel ME FW.

IPMI tab page consists of the following elements:

Element	Description
Command Group	Command Group is used to select a group with the command to be executed. User can select one of the following groups: - S/E - App - Storage - OEM/Group - SDK General App - Chassis
Request	Request group is used to configure IPMI request. User can choose a proper request (like Get Device ID) from Command drop down menu and specify the value in hex. Then a request is sent to Intel ME by clicking RUN button
Request Frame	Displays the entire Request: NetFn/LUN Cmd - command ID Data Bytes - data in IPMI frame
Response	Displays the result of executed command in a decoded format
Intel ME Response Frame	Displays the entire Intel ME Response: NetFn/LUN

Element	Description
Command Group	Command Group is used to select a group with the command to be executed. User can select one of the following groups: - S/E - App - Storage - OEM/Group - SDK General App - Chassis
	Cmd - command ID Data Bytes - data in IPMI frame
Raw IPMI	User can send raw IPMI frame by checking "Raw IPMI" checkbox. Request Frame fields will be editable and can be overwritten by user.
Auto Refresh	Checking "Auto Refresh" checkbox will make the MESDC automatically send the command every n seconds, where n is a specified time interval left to the Run button. If Auto Refresh is enabled Communication indicator on statusbar blinks green every iteration.
Operation Log	Displays communication errors



Command Set Event Receiver

Name	Value [hex]	Bytes
Event receiver slave address		1
Event receiver LUN		1

Command Group

● S/E
○ App
○ Storage
○ OEM/Group
○ SDK General App
○ Chassis

☐ Auto Refresh

1.0 s

☐ Raw IPMI

Request Frame [hex]

NetFn/LUN	Cmd	Data Bytes; format: LSB ... xx xx ... MSB
10	00	00 00

Response

Name	Value [hex]	Bytes

ME Response Frame [hex]

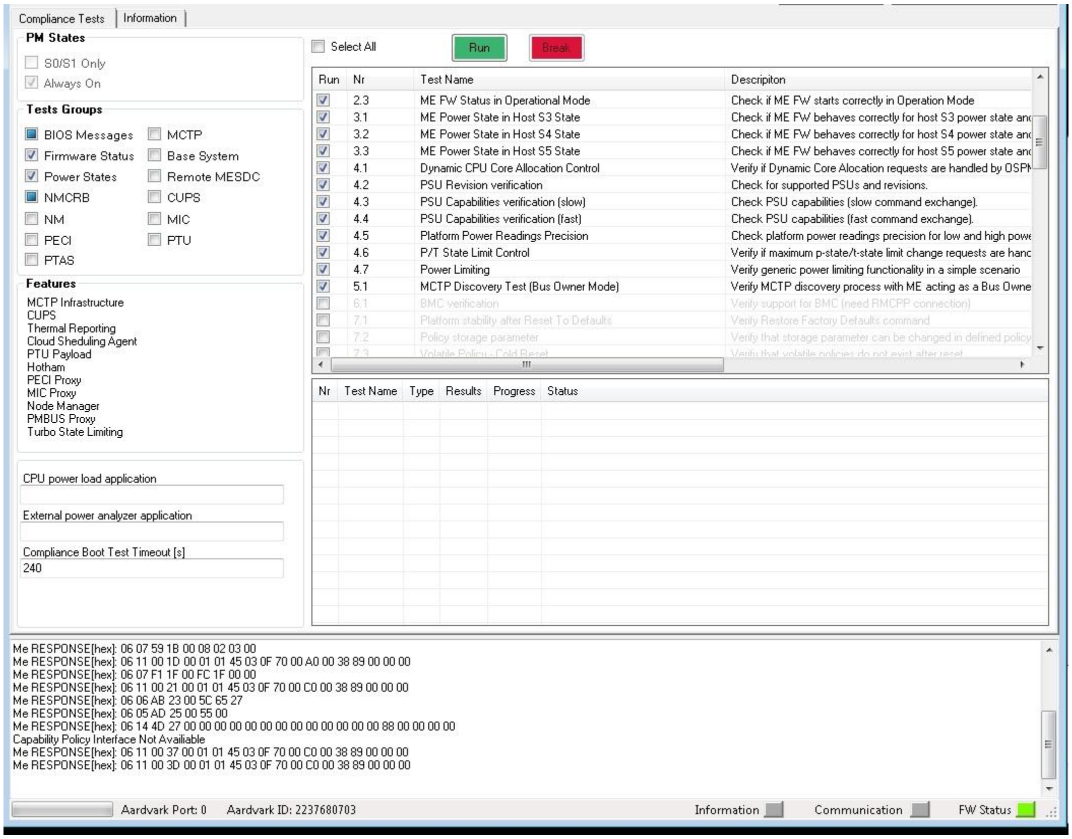
NetFn/LUN	Cmd	CompCode	Data Bytes; format: LSB ... xx xx ... MSB

4.5.3 Compliance Tests

This tab is design for customer to perform compliance tests. List of available compliance tests depends on Platform Type and enabled FW features. It is for user convenience to make some of the tests into tests Groups as BIOS, FW status and power States.

The test result and detail log is available in the same directory as MESDC named as report-XXXX and log-XXXX.

For more detailed information about every compliance test, please refer to the specific sections in this manual (4.6 Intel® Management Engine (Intel® ME) FW Compliance Tests).



PM States

☐ S0/S1 Only
☑ Always On

Tests Groups

■ BIOS Messages
☑ Firmware Status
☑ Power States
■ NMCRB
☐ NM
☐ PECI
☐ PTAS
☐ MCTP
☐ Base System
☐ Remote MESDC
☐ CUPS
☐ MIC
☐ PTU

Features

MCTP Infrastructure
CUPS
Thermal Reporting
Cloud Sheduling Agent
PTU Payload
Hotham
PECI Proxy
MIC Proxy
Node Manager
PMBUS Proxy
Turbo State Limiting

CPU power load application:

External power analyzer application:

Compliance Boot Test Timeout [s]: 240

☐ Select All

Run	Nr	Test Name	Descripiton
☑	2.3	ME FW Status in Operational Mode	Check if ME FW starts correctly in Operation Mode
☑	3.1	ME Power State in Host S3 State	Check if ME FW behaves correctly for host S3 power state anc
☑	3.2	ME Power State in Host S4 State	Check if ME FW behaves correctly for host S4 power state anc
☑	3.3	ME Power State in Host S5 State	Check if ME FW behaves correctly for host S5 power state anc
☑	4.1	Dynamic CPU Core Allocation Control	Verify if Dynamic Core Alocation requests are handled by OSPN
☑	4.2	PSU Revision verification	Check for supported PSUs and revisions.
☑	4.3	PSU Capabilities verification (slow)	Check PSU capabilities (slow command exchange).
☑	4.4	PSU Capabilities verification (fast)	Check PSU capabilities (fast command exchange).
☑	4.5	Platform Power Readings Precision	Check platform power readings precision for low and high powe
☑	4.6	P/T State Limit Control	Verify if maximum p-state/t-state limit change requests are hanc
☑	4.7	Power Limiting	Verify generic power limiting functionality in a simple scenario
☑	5.1	MCTP Discovery Test (Bus Owner Mode)	Verify MCTP discovery process with ME acting as a Bus Owne
☐	6.1	BMC verification	Verify support for BMC (need RMCPP connection)
☐	7.1	Platform stability after Reset To Defaults	Verify Restore Factory Defaults command
☐	7.2	Policy storage parameter	Verify that storage parameter can be changed in defined policy
☐	7.3	Volatile Policy - Cold Reset	Verify that volatile policies do not exist after reset

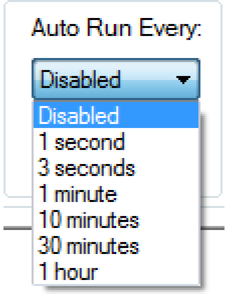
Nr	Test Name	Type	Results	Progress	Status

```
Me RESPONSE[hex]: 06 07 59 1B 00 08 02 03 00
Me RESPONSE[hex]: 06 11 00 1D 00 01 01 45 03 0F 70 00 A0 00 38 89 00 00 00
Me RESPONSE[hex]: 06 07 F1 1F 00 FC 1F 00 00
Me RESPONSE[hex]: 06 11 00 21 00 01 01 45 03 0F 70 00 C0 00 38 89 00 00 00
Me RESPONSE[hex]: 06 06 AB 23 00 5C 65 27
Me RESPONSE[hex]: 06 05 AD 25 00 55 00
Me RESPONSE[hex]: 06 14 4D 27 00 00 00 00 00 00 00 00 00 00 00 00 00 88 00 00 00 00
Capability Policy Interface Not Availiable
Me RESPONSE[hex]: 06 11 00 37 00 01 01 45 03 0F 70 00 C0 00 38 89 00 00 00
Me RESPONSE[hex]: 06 11 00 3D 00 01 01 45 03 0F 70 00 C0 00 38 89 00 00 00
```

Aardvark Port: 0   Aardvark ID: 2237680703

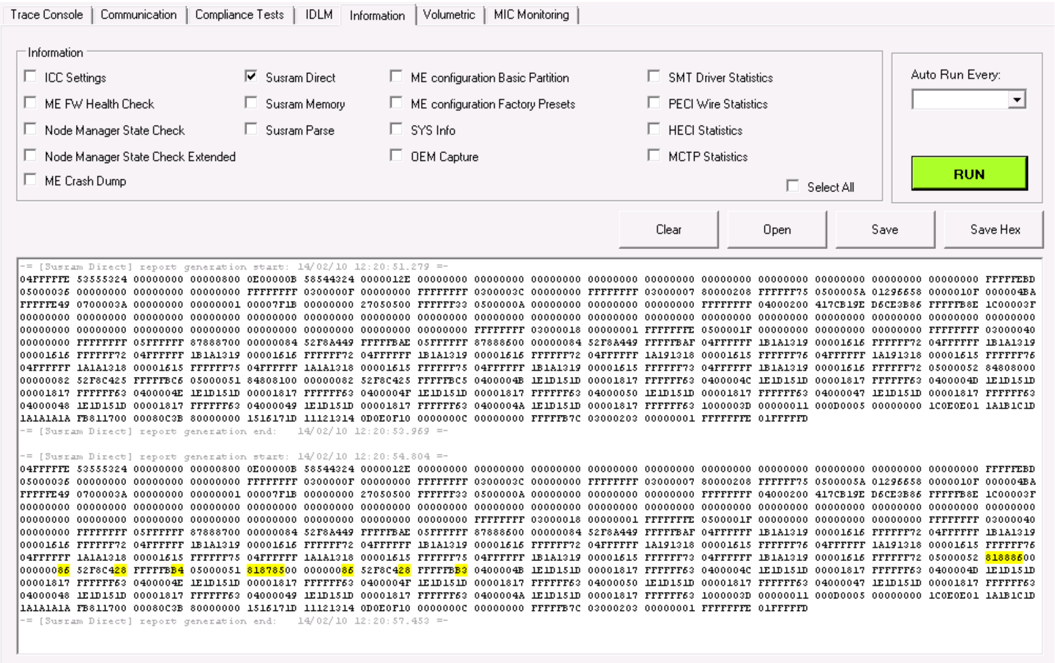
4.5.4 Information

This tab is design for customer to capture useful information from the platform. All/any of the option can be run once by clicking RUN button or run multiple times by change the Auto Run configuration. If Auto Run is enabled Information indicator on statusbar blinks green every iteration.



The result will be show in log frame with highlighted changes and can be saved by click Save button. Each section of the information will have a time stamp attached in the log.

The different bytes will be highlighted for Susram Direct, Susram Memory, OEM Capture and HECI Statistics reports.



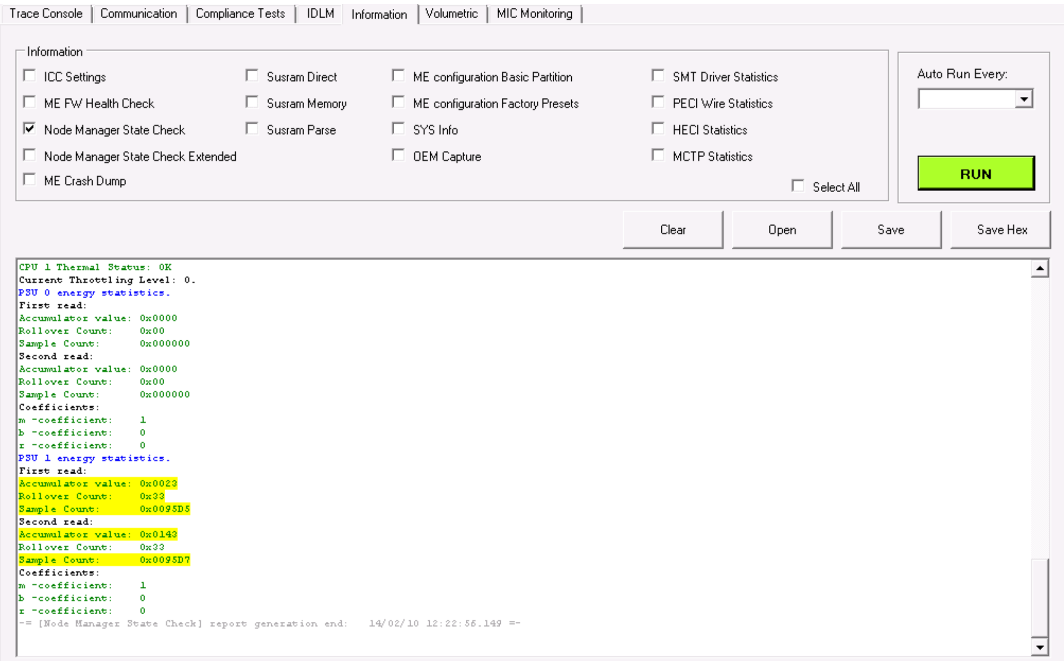
Information

☐ ICC Settings
☐ ME FW Health Check
☐ Node Manager State Check
☐ Node Manager State Check Extended
☐ ME Crash Dump
☑ Susram Direct
☐ Susram Memory
☐ Susram Parse
☐ ME configuration Basic Partition
☐ ME configuration Factory Presets
☐ SYS Info
☐ OEM Capture
☐ SMT Driver Statistics
☐ PECI Wire Statistics
☐ HECI Statistics
☐ MCTP Statistics
☐ Select All

```
-= [Susram Direct] report generation start: 14/02/10 12:20:51.279 =-
04FFFFFE 53555324 00000000 00000800 0E00000B 58544324 0000012E 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 FFFFFEBD
05000036 00000000 00000000 00000000 FFFFFFFF 0300000F 00000000 FFFFFFFF 0300003C 00000000 FFFFFFFF 03000007 80000208 FFFFFF75 0500005A 01296658 0000010F 000004BA
FFFFFE49 0700003A 00000000 00000001 00007F1B 00000000 27050500 FFFFFF33 0500000A 00000000 00000000 00000000 FFFFFFFF 04000200 417CB19E D6CE3B86 FFFFFB8E 1C00003F
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 FFFFFFFF 03000018 00000001 FFFFFFFE 0500001F 00000000 00000000 00000000 FFFFFFFF 03000040
00000000 FFFFFFFF 05FFFFFF 87888700 00000084 52F8A449 FFFFFBAE 05FFFFFF 87888600 00000084 52F8A449 FFFFFBAF 04FFFFFF 1B1A1319 00001616 FFFFFF72 04FFFFFF 1B1A1319
00001616 FFFFFF72 04FFFFFF 1B1A1319 00001616 FFFFFF72 04FFFFFF 1B1A1319 00001616 FFFFFF72 04FFFFFF 1A191318 00001615 FFFFFF76 04FFFFFF 1A191318 00001615 FFFFFF76
04FFFFFF 1A1A1318 00001615 FFFFFF75 04FFFFFF 1A1A1318 00001615 FFFFFF75 04FFFFFF 1B1A1319 00001615 FFFFFF73 04FFFFFF 1B1A1319 00001616 FFFFFF72 05000052 84808000
00000082 52F8C425 FFFFFBC6 05000051 84808100 00000082 52F8C425 FFFFFBC5 0400004B 1E1D151D 00001817 FFFFFF63 0400004C 1E1D151D 00001817 FFFFFF63 0400004D 1E1D151D
00001817 FFFFFF63 0400004E 1E1D151D 00001817 FFFFFF63 0400004F 1E1D151D 00001817 FFFFFF63 04000050 1E1D151D 00001817 FFFFFF63 04000047 1E1D151D 00001817 FFFFFF63
04000048 1E1D151D 00001817 FFFFFF63 04000049 1E1D151D 00001817 FFFFFF63 0400004A 1E1D151D 00001817 FFFFFF63 1000003D 00000011 000D0005 00000000 1C0E0E01 1A1B1C1D
1A1A1A1A FB811700 00080C3B 80000000 1516171D 11121314 0D0E0F10 0000000C 00000000 FFFFFB7C 03000203 00000001 FFFFFFFE 01FFFFFD
-= [Susram Direct] report generation end:   14/02/10 12:20:53.969 =-

-= [Susram Direct] report generation start: 14/02/10 12:20:54.804 =-
04FFFFFE 53555324 00000000 00000800 0E00000B 58544324 0000012E 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 FFFFFEBD
05000036 00000000 00000000 00000000 FFFFFFFF 0300000F 00000000 FFFFFFFF 0300003C 00000000 FFFFFFFF 03000007 80000208 FFFFFF75 0500005A 01296658 0000010F 000004BA
FFFFFE49 0700003A 00000000 00000001 00007F1B 00000000 27050500 FFFFFF33 0500000A 00000000 00000000 00000000 FFFFFFFF 04000200 417CB19E D6CE3B86 FFFFFB8E 1C00003F
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 FFFFFFFF 03000018 00000001 FFFFFFFE 0500001F 00000000 00000000 00000000 FFFFFFFF 03000040
00000000 FFFFFFFF 05FFFFFF 87888700 00000084 52F8A449 FFFFFBAE 05FFFFFF 87888600 00000084 52F8A449 FFFFFBAF 04FFFFFF 1B1A1319 00001616 FFFFFF72 04FFFFFF 1B1A1319
00001616 FFFFFF72 04FFFFFF 1B1A1319 00001616 FFFFFF72 04FFFFFF 1B1A1319 00001616 FFFFFF72 04FFFFFF 1A191318 00001615 FFFFFF76 04FFFFFF 1A191318 00001615 FFFFFF76
04FFFFFF 1A1A1318 00001615 FFFFFF75 04FFFFFF 1A1A1318 00001615 FFFFFF75 04FFFFFF 1B1A1319 00001615 FFFFFF73 04FFFFFF 1B1A1319 00001616 FFFFFF72 05000052 81888600
00000086 52F8C428 FFFFFBB4 05000051 81878500 00000086 52F8C428 FFFFFBB3 0400004B 1E1D151D 00001817 FFFFFF63 0400004C 1E1D151D 00001817 FFFFFF63 0400004D 1E1D151D
00001817 FFFFFF63 0400004E 1E1D151D 00001817 FFFFFF63 0400004F 1E1D151D 00001817 FFFFFF63 04000050 1E1D151D 00001817 FFFFFF63 04000047 1E1D151D 00001817 FFFFFF63
04000048 1E1D151D 00001817 FFFFFF63 04000049 1E1D151D 00001817 FFFFFF63 0400004A 1E1D151D 00001817 FFFFFF63 1000003D 00000011 000D0005 00000000 1C0E0E01 1A1B1C1D
1A1A1A1A FB811700 00080C3B 80000000 1516171D 11121314 0D0E0F10 0000000C 00000000 FFFFFB7C 03000203 00000001 FFFFFFFE 01FFFFFD
-= [Susram Direct] report generation end:   14/02/10 12:20:57.453 =-
```

The different lines will be highlighted for ICC settings, Intel ME FW Health Check, Intel Node Manager State Check, Intel Node Manager State Check Extended, Intel ME Crush Dump, Susram Parse, SMT Driver Statistics, PECI Wire Statistics, MCTP Statistics reports.



☐ ICC Settings
☐ ME FW Health Check
☑ Node Manager State Check
☐ Node Manager State Check Extended
☐ ME Crash Dump
☐ Susram Direct
☐ Susram Memory
☐ Susram Parse
☐ ME configuration Basic Partition
☐ ME configuration Factory Presets
☐ SYS Info
☐ OEM Capture
☐ SMT Driver Statistics
☐ PECI Wire Statistics
☐ HECI Statistics
☐ MCTP Statistics
☐ Select All

```
CPU 1 Thermal Status: OK
Current Throttling Level: 0.
PSU 0 energy statistics.
First read:
Accumulator value: 0x0000
Rollover Count:    0x00
Sample Count:      0x000000
Second read:
Accumulator value: 0x0000
Rollover Count:    0x00
Sample Count:      0x000000
Coefficients:
m -coefficient:    1
b -coefficient:    0
r -coefficient:    0
PSU 1 energy statistics.
First read:
Accumulator value: 0x0023
Rollover Count:    0x33
Sample Count:      0x0095D5
Second read:
Accumulator value: 0x0143
Rollover Count:    0x33
Sample Count:      0x0095D7
Coefficients:
m -coefficient:    1
b -coefficient:    0
r -coefficient:    0
-= [Node Manager State Check] report generation end:   14/02/10 12:22:56.149 =-
```

4.5.4.1 ICC setting

MESDC will retrieve ICC setting exposed by Intel ME FW.

Detail reference for ICC register, please refer to Wellsburg EDS. The ICC information retrieve from Pre-production silicon might be different than the ones from post production silicon.

4.5.4.2 Example of ICC report is available in Appendix C. Intel ME FW Health Check

This will retrieve Intel ME FW version, operational mode, FW exception number, FW self-test result, last Global Reset Cause and Intel ME FW reset counter.

Example of ME FW Health Check report is available in Appendix C.

4.5.4.3 Intel Node Manager State Check

This will retrieve Intel Node Manager Features information: e.g. Ptam State, Total Power Budget, Intel NM Statistics. The output of report is displayed in human-readable form.

Example of Intel Node Manager State Check report is available in Appendix C.

4.5.4.4 Intel Node Manager State Check Extended

This is the reserved data for Intel to analysis. If needed, user should capture this and send Intel for next level of analysis based on Intel guidance.

4.5.4.5 Intel ME Crash Dump

This is the reserved data for Intel to analysis. If needed, user should capture this and send Intel for next level of analysis based on Intel guidance.

4.5.4.6 Susram direct/Memory/Parse

MESDC will retrieve Intel ME related information which is stored in SUSRAM and based on the option to show it in HEX or parsed.

Intel ME recovery reason, Intel ME exceptions, reset reason, power management event etc. are stored at SUSRAM and will not be changed during boot cycle.

An example of SUSRAM dump in human readable format (Parse) is available in Appendix C.

4.5.4.7 Intel ME Configuration Basic Partition/ Factory Presets

MESDC allows user to read the file system from Intel ME FW and display the contents of physical partitions:

- Basic Partition
- Factory Presets

The user can fetch the list of files by selecting one of the report. The contents of each file can be viewed in hex format by clicking on a file from the list. Basic partition contains runtime Intel ME FW data, and factory presets contain the factory default setting for Intel ME FW.

Example of ME Configuration Basic Partition report is available in Appendix C.

4.5.4.8 SYS Info

Sys Info report contains most of the information that is available in Information tab and is intent to get all the data for general debug purpose.

Sys Info report contains SUSRAM, Me Configuration Basic Partition Report and info about HW Data Registers

4.5.4.9 OEM Capture

OEM capture will make MESDC run several MESDC commands based on OEM capture .xml file. MESDC will run the MESDC command one by one and capture the result in log. User can also run the OEM capture file multiple times with selection autorun feature. An example of xml file is as following

```
<?xml version="1.0" encoding="utf-8"?>

<OEMCapture>
```

<Command Group="System" Name="Get Version" Arguments="0x0000" Info="Get Version Command in System group." />

</OEMCapture>

The Command Group, name should align with MESDC command in GUI interface.

4.5.4.10 Interface Statistics

4.5.4.10.1 SMT Driver Statistics

SMT driver is the driver for additional SMBUS available in Wellsburg PCH. This function will give you the statistics for SMT driver

4.5.4.10.2 PECI Wire Statistics

MESDC supports collecting basic and extended communication statistics for PECI interface.

4.5.4.10.3 HECI Statistics

MESDC supports collecting communication statistics for HECI interfaces (HECI-1 and HECI-2).

4.5.4.10.4 MCTP Statistics

This function gives the MCTP statistics in user friendly form.

4.5.5 Volumetric

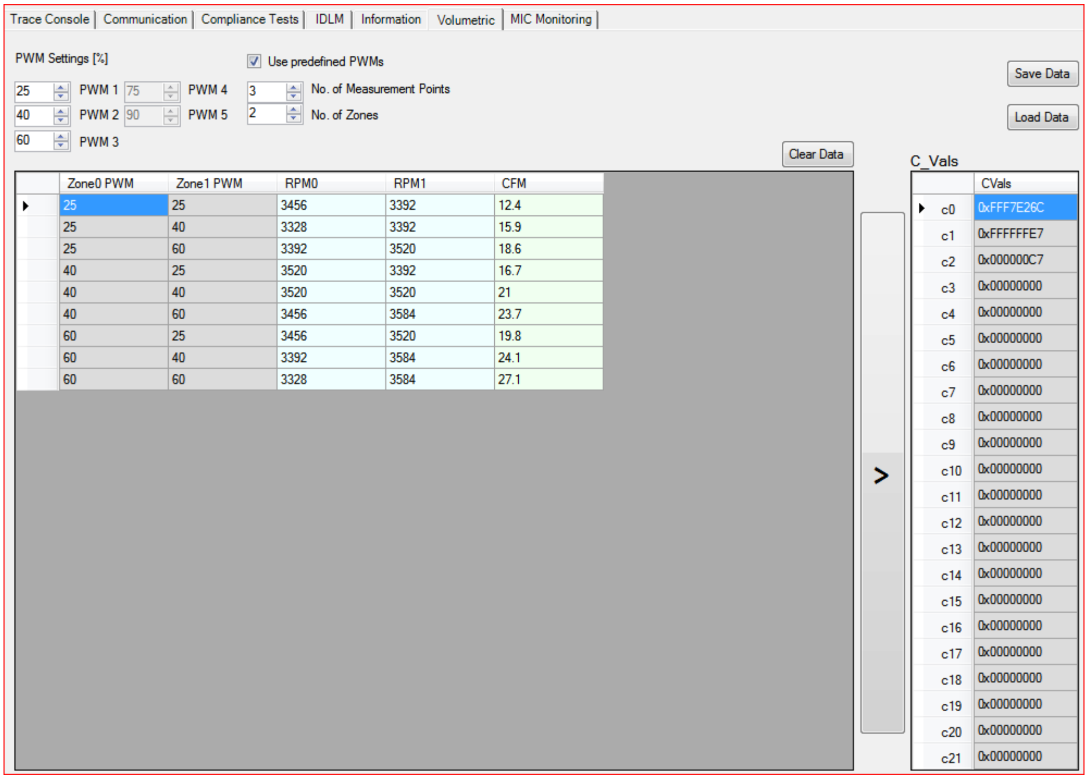
Volumetric airflow calculator tab is designed to help obtain appropriate coefficients which need to be set in spsFITC.

Interface of this tab consist of:

- Main input table which allows user to enter airflow measured in ft^3/min at various fan speeds (average in RPM) in different zones
- Calculate button - right to the input table
- Output coefficients table (on the right) – data is being calculated in format directly accepted by spsFITC
- “Number of measurement points” and PWM settings entries – allows to declare up to five predefined PWM values for fans at which airflow will be measured (if “Use predefined PWMs” checkbox is unchecked then user can enter any values directly in the main table)
- Number of zones – to declare number of fan zones (up to six)

- Load/Save data buttons – allows to save data from input table in xml file and load those files later

Input table can have 15625 rows when there are 5 measurement points and 6 zones set.



PWM Settings [%]

☑ Use predefined PWMs

25 PWM 1
40 PWM 2
60 PWM 3
75 PWM 4
90 PWM 5

3 No. of Measurement Points
2 No. of Zones

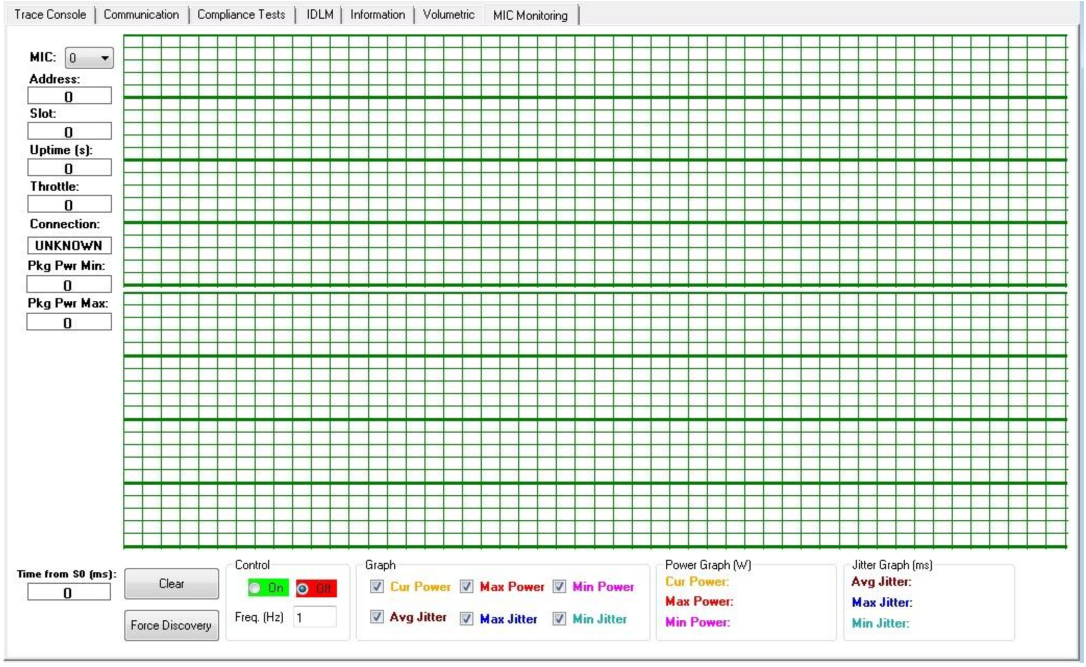
Zone0 PWM	Zone1 PWM	RPM0	RPM1	CFM
25	25	3456	3392	12.4
25	40	3328	3392	15.9
25	60	3392	3520	18.6
40	25	3520	3392	16.7
40	40	3520	3520	21
40	60	3456	3584	23.7
60	25	3456	3520	19.8
60	40	3392	3584	24.1
60	60	3328	3584	27.1

C_Vals

	CVals
c0	0xFFF7E26C
c1	0xFFFFFFE7
c2	0x000000C7
c3	0x00000000
c4	0x00000000
c5	0x00000000
c6	0x00000000
c7	0x00000000
c8	0x00000000
c9	0x00000000
c10	0x00000000
c11	0x00000000
c12	0x00000000
c13	0x00000000
c14	0x00000000
c15	0x00000000
c16	0x00000000
c17	0x00000000
c18	0x00000000
c19	0x00000000
c20	0x00000000
c21	0x00000000

Details please refer to Intel Server Platform Services FW integration guide.

4.5.6 Mic Monitoring



MIC: 0
Address: 0
Slot: 0
Uptime (s): 0
Throttle: 0
Connection: UNKNOWN
Pkg Pwr Min: 0
Pkg Pwr Max: 0
Time from S0 (ms): 0
Control: ○ On ● Off
Freq. (Hz) 1
Graph: ☑ Cur Power ☑ Max Power ☑ Min Power ☑ Avg Jitter ☑ Max Jitter ☑ Min Jitter

MIC Monitoring tab allows user to monitor power usage of MIC cards. To start monitoring select appropriate MIC (left top corner) and switch on monitoring in ‘Control’ section.

Two graphs are presented to the user:

- power usage (upper chart)
- jitter (bottom chart).

For measurements of power usage three values are presented:

- Current power usage (yellow)
- Maximum power usage (red)
- Minimum power usage (violet)

For measurements of jitter three values are presented:

- Average jitter (brown)
- Maximum jitter (blue)
- Minimum jitter (celadon)

Each value can be hidden by deselecting it in ‘Graph’ section.

Readings are also shown in two sections: Power Graph and Jitter Graph in right bottom corner. These sections show most recent readings unless the mouse cursor is pointing at any of the graph. Pointing at any point of the graph with mouse cursor will show value of measurements at this point in Power Graph or Jitter Graph section (depending on which graph the user is pointing at).

4.6 Intel® Management Engine (Intel® ME) FW Compliance Tests

4.6.1 Compliance Tests

In this mode of operation, MESDC executes a configurable set of tests of Intel ME integration on tested platform. The tests verify:

- Basic Intel ME firmware health
- Basic Intel ME system functionality
- Basic Intel ME features functionality

Detailed description of all compliance tests is provided in later sections of this chapter.

MESDC runs two types of tests: Boot and Interactive. For Boot type tests, MESCD collects and analyzes the data that checks correct BIOS interaction with SPS Firmware before End Of POST (EOP). For Interactive tests MESDC may prompt user input.

After the tests have been finished, MESDC saves results in a detailed report.

There are 3 sections to allow the user to configure the list of tests and present the results:

- Configuration
- Compliance tests
- Tests results

Configuration section contains a list of options, supported in platform design and may affect a set of compliance tests. Configuration options such as Platform type, FW SKU and PM States as well as default Tests Groups are to be provided by the platform when FW Configuration is retrieved. Configuration options such as Tests Groups allow user to enable or disable a specific group of tests to be run on SUT (system under test). Selecting/deselecting one tests group may result in a cluster of connected compliancy tests selected/deselected.

Compliance tests section contains a table of all tests and an execution part. A table consists of the following columns:

- *Run* – checkbox to enable/disable a test
- Nr – test number
- Test Name
- Description – description of test
- Type – type of test (Boot or Interactive)

- Progress – progress of test (NotRunning / Running / Done)
- Status – status of test (N/A- not available / Pass / Failed)

Aside from selecting/deselecting test sequences in the 'Configuration' section, users also have an option of enabling/disabling individual tests by clicking individual check boxes. Selecting/deselecting one test in a list of tests may result in a change of Tests Groups check boxes. There is also a separate checkbox to enable/disable all tests together.

Tests Results section shows the results of each executed test. The tests that have passed are presented in green, whilst the tests that have failed are marked in red. The tests that couldn't be executed and therefore the result is unknown are shown in yellow.

The generic execution of tests sequence (boot and interactive tests included) is as follows:

1. After first clicking the 'Run' button user may be informed that the logger settings must be changed and the Intel ME FW needs resetting for new settings to get applied.
2. After clicking the 'Run' button MESDC sends a set of commands to check logger settings in Intel ME and starts tests execution if they are correct.
3. For boot tests execution user is asked to restart platform.
4. Application is waiting for 60 seconds and, after receiving first boot log, a "Collecting boot tests data from FW. Please wait…" message is displayed.
5. After receiving last boot log or when 240 seconds have passed (configurable – Compliance Boot Test Timeout) MESDC executes boot tests and starts interactive tests.
6. During interactive tests user is asked to perform different activities depending on particular test (for example, platform reboot, OS hibernation, recovery jumper activation, and so forth).
7. At the end of tests sequence MESDC sends a set of commands to restore logger settings in Intel ME, displays results of tests, saves results in files report-yyyy-mm-dd-hh-mm.txt and diagnostic logs in files log-yyyy-mm-dd-hh-mm.txt.
8. In case of unhandled application exception an AppException.log file is created.

Additionally, for boot compliance tests (Group1-3) there is a possibility to define OS boot timeout. Default value is 240 s and when it expires test fails.

Note: The Compliance Tests cannot be run when diagnostic service is enabled in emergency mode, i.e. when Intel ME Debug SMBus Emergency Mode Enable strap setting is set to true.

4.6.2 Intel ME FW Compliance Tests

4.6.2.1 Intel ME-BIOS Integration Tests

Generally this set of tests is applicable to all the Intel ME firmware variants. However, several detailed tests apply to specific variants only. This is a test suite that verifies

Intel ME-BIOS communication flow by testing HECI messages exchanged between Intel ME and BIOS over HECI interface during system boot.

Note that all of the Intel ME-BIOS integration tests may be executed during one system boot only.

4.6.2.1.1 Test 1.1: BIOS HECI Interfaces initialization

Note: This test is applicable to the following Intel ME firmware variants: SiEn, Intel Node Manager, This test checks if BIOS initializes HECI interfaces. It is not started automatically by MESDC tool unless Tests Group of BIOS messages is selected. Just follow the instructions provided by MESDC.

Procedure:

- After starting the test perform full AC cycle.

Success Criteria:

- BIOS initializes HECI interfaces.

4.6.2.1.2 Test 1.2: BIOS Get Interface Version message

Note: This test is applicable to the following Intel ME firmware variants: SiEn, Intel Node Manager, This is a detailed test for Intel ME-BIOS communication flow that checks if BIOS sends Get Interface Version message to Intel ME. It is not started automatically by MESDC tool unless Tests Group of BIOS messages is selected. Just follow the instructions provided by MESDC.

Procedure:

- Same as above

Success Criteria:

- Exactly one Get Interface Version message has been sent by BIOS to Intel ME.

4.6.2.1.3 Test 1.3: BIOS ICC Set Clock Enable Message

Note: This test is applicable to Denlow platform only, to the following Intel ME firmware variants: SiEn, Intel Node Manager

This is a detailed test for Intel ME-BIOS communication flow that checks if BIOS sends ICC Set Clock Enable message to Intel ME. It is not started automatically by MESDC tool unless Tests Group of BIOS messages is selected. Just follow the instructions provided by MESDC.

Procedure:

Same as above

Success Criteria:

Exactly one ICC Set Clock Enable message has been sent by BIOS to Intel ME.

4.6.2.1.4 Test 1.4: BIOS NM CPU Discovery message

Note: This test is applicable to the following Intel ME firmware variants: NM.

This is a detailed test for Intel ME-BIOS communication flow that checks if BIOS sends Intel Node Manager CPU Discovery message to Intel ME. It is not started automatically by MESDC tool unless Tests Group of BIOS messages is selected. Just follow the instructions provided by MESDC.

Procedure:

Same as above

Success Criteria:

4. Exactly one Intel Node Manager CPU Discovery message has been sent by BIOS to Intel ME.

4.6.2.1.5 Test 1.5: BIOS HMRFPO Lock message

Note: This test is applicable to the following Intel ME firmware variants: SiEn, Intel Node Manager.

This is a detailed test for Intel ME-BIOS communication flow that checks if BIOS sends HMRFPO Lock message to Intel ME. It is not started automatically by MESDC tool unless Tests Group of BIOS messages is selected. Just follow the instructions provided by MESDC.

Procedure:

- Same as above

Success Criteria:

- Exactly one HMRFPO Lock message has been sent by BIOS to Intel ME.

4.6.2.1.6 Test 1.6: BIOS End Of Post message

Note: This test is applicable to the following Intel ME firmware variants: SiEn, Intel NM.

This is a detailed test for Intel ME-BIOS communication flow that checks if BIOS sends End Of POST message to Intel ME. It is not started automatically by MESDC tool unless Tests Group of BIOS messages is selected. Just follow the instructions provided by MESDC.

Procedure:

- Same as above

Success Criteria:

- Exactly one End Of POST message has been sent by BIOS to Intel ME.

4.6.2.2 Intel ME FW Status Tests

Note: This set of tests is applicable to the following Intel ME FW configurations: SiEn, Intel Node Manager.

This is a test suit that verifies basic Intel ME health in manufacturing, recovery and operational modes.

4.6.2.2.1 Test 2.1: Intel ME FW Status in Manufacturing mode

This test checks whether Intel ME FW correctly detects Security Strap Override condition activated on the tested platform. It is started automatically by MESDC tool whenever Tests Group of Firmware Status is selected. Just follow the instructions provided by MESDC.

Procedure:

1. Turn off SUT AC power.
2. Set the Security Strap Override jumper on.
3. If recovery jumper is implemented set it off.
4. Turn on SUT AC power; put it into S0/M0.

Success Criteria:

1. MESDC console can connect to Intel ME and reports the status is correct for manufacturing mode.

Intel ME Status bits	Expected Result	Actual Result
3:0	0010	
4	1	
5	0	
8:6	110	
9	1	

Intel ME Status bits	Expected Result	Actual Result
10	0	
11	0	
15:12	0000	
31:16	n/a	

4.6.2.2.2 Test 2.2: Intel ME FW Status in Recovery Mode

This test checks whether Intel ME FW correctly detects recovery jumper activation on the tested platform. It is started automatically by MESDC tool whenever Tests Group of Firmware Status is selected, but can be skipped if recovery jumper is not configured in Intel ME FW factory presets. Just follow the instructions provided by MESDC.

Procedure:

1. Turn off SUT AC power.
2. Set the Security Strap Override jumper off.
3. Set the recovery jumper on.
4. Turn on SUT AC power; put it into S0/M0.

Success Criteria:

- MESDC console can connect to Intel ME and reports the status is correct for recovery boot loader.

Intel ME Status bits	Expected Result	Actual Result
3:0	0010	
4	0	
5	0	
8:6	110	
9	1	
10	0	
11	0	
15:12	0000	
31:16	n/a	

4.6.2.2.3 Test 2.3: Intel ME FW Status in Operational mode

This test checks whether Intel ME firmware has started on Intel ME in operational mode. It is started automatically by MESDC tool whenever Tests Group of Firmware Status is selected. Just follow the instructions provided by MESDC.

Procedure:

1. Make sure Security Strap Override jumper and recovery jumper, if implemented, are off.
2. Turn on AC power and put SUT into S0/M0.
3. Using mm command in EFI Shell, or Intel ME Info in OS, read Intel ME FW Status.
4. Using MESDC.exe console start Intel ME FW Status Test.

Success Criteria:

1. Intel ME Firmware Status read with the EFI or OS tool reports that Intel ME is up in operational mode.
2. MESDC console can connect to Intel ME and reports the same Intel ME FW status value for operational mode.
3.

Intel ME Status bits	Expected Result	Actual Result
3:0	0101	
4	0	
5	0	
8:6	101	
9	1	
10	0	
11	0	
15:12	0000	
31:16	n/a	

4.6.2.3 Power States Tests

4.6.2.3.1 Test 3.1: Intel ME Power State in Host S3 State

This test checks whether Intel ME FW behaves correctly for host S3 power state and after host exits S3. It is started automatically by MESDC tool whenever Tests Group of Power States is selected, but can be skipped if S3 is not supported by the tested system. Just follow the instructions provided by MESDC. Depending on Intel ME power mode configuration in Intel ME FW factory presets Intel ME may enter MOff or M3 when host enters S3.

Procedure:

1. Make sure the Security Strap Override and the Recovery jumpers are off. If Intel ME has entered M0 state.
2. Put SUT into S3, check if Intel ME has entered MOff/M3 state.
3. Wakeup SUT to S0, check if Intel ME has entered M0 state.

Success Criteria:

1. For initial S0 state Intel ME is in M0 state.
2. For S3 state

If Intel ME is configured to work in S0/S1 only state it has entered MOff state and doesn't report heartbeat.

If Intel ME is configured to work in all Sx states, it has entered M3 state and reports heartbeat.

3. Intel ME has entered M0 state.

4.6.2.3.2 Test 3.2: Intel ME Power State in Host S4 State

This test checks whether Intel ME FW behaves correctly for host S4 power state and after host exits S4. It is started automatically by MESDC tool whenever Tests Group of Power States is selected, but can be skipped if S4 is not supported by the tested system. Just follow the instructions provided by MESDC. Depending on Intel ME power mode configuration in Intel ME FW factory presets Intel ME may enter MOff or M3 when host enters S4.

Procedure:

1. Make sure Security Strap Override and recovery jumpers are off.
2. Put SUT into S0/M0, check if Intel ME has entered M0 state.
3. Put SUT into S4, check if Intel ME has entered MOff/M3 state.
4. Wakeup SUT to S0, check if Intel ME has entered M0 state.

Success Criteria:

1. For initial S0 state Intel ME is in M0 state.
2. For S4 state

If Intel ME is configured to work in S0/S1 only state, it has entered MOff state and doesn't report heartbeat.

If Intel ME is configured to work in all Sx states, it has entered M3 state and reports heartbeat.

3. Intel ME has entered M0 state.

4.6.2.3.3 Test 3.3: Intel ME Power State in Host S5 State

This test checks whether Intel ME FW behaves correctly for host S5 power state and after host exits S5. It is started automatically by MESDC tool whenever Tests Group of Power States is selected. Just follow the instructions provided by MESDC. Depending on Intel ME power mode configuration in Intel ME FW factory presets Intel ME may enter MOff or M3 when host enters S5.

Procedure:

1. Make sure Security Strap Override and recovery jumpers are off.
2. Put SUT into S0, check if Intel ME has entered M0 state.
3. Put SUT into S5, check if Intel ME has entered MOff/M3 state.
4. Wakeup SUT to S0 check if Intel ME has entered M0 state.

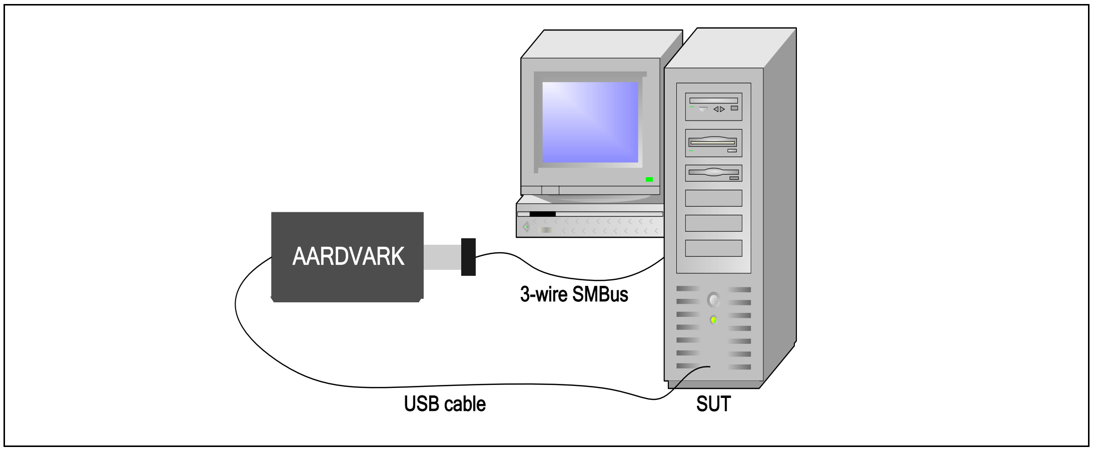
Success Criteria:

1. For initial S0 state Intel ME is in M0 state.
2. For S5 state
3. If Intel ME is configured to work in S0/S1 only state, it has entered MOff state and doesn't report heartbeats.
4. If Intel ME is configured to work in all Sx states, it has entered M3 state and reports heartbeats.
5. For the final S0 state Intel ME has entered M0 state.

4.6.2.4 Basic Intel ME Functionality Tests

This is a test suit that verifies basic Intel ME functionality. It is executed by MESDC console, and the console must run on the SUT host system.

Figure 4-7. MESDC Console Running on the SUT Host System



When you start this test suite in the MESDC console the console will guide you with the steps to do for each of the tests described in this chapter.

A detailed set of commands can be found in Appendix_D_MESDC commands.

4.6.2.4.1 Test 4.1: Dynamic CPU Core Allocation Control

Note: This set of tests is applicable to the following Intel ME firmware variants: Intel Node Manager.

This test verifies BIOS support for controlling CPU core allocation. It is done automatically by MESDC console without any user interaction.

Note that this test does not check whether the operating system supports dynamic changes of the number of CPU cores running. This test only checks whether Intel Node Manager notifications reach OSPM via ACPI tables provided by BIOS.

Procedure:

1. Disable Intel Node Manager Policy Control to enable CPU core allocation control commands.
2. Run core allocation set command.
3. Check for OSPM acknowledge.
4. Restore former Intel Node Manager Policy Control state.

Success Criteria:

- MESDC reports that the OSPM acknowledge was received.

4.6.2.4.2 Test 4.2: PSU Revision verification

Note: This set of tests is applicable to the following Intel ME firmware variants: Intel NM.

This test checks for supported PSUs and revisions. It is done automatically by MESDC console without any user interaction.

Note that this test communicates with Intel ME only to check for supported PSUs.

Procedure:

1. Send MESDC command Get PSU Discovery Data for all PSUs to Intel ME.
2. Check revisions reported by Intel ME for all detected PSUs.

Success Criteria:

1. At least one PSU with revision 1.2 is detected.
2. Warning is reported if any PSU with revision 1.1. is detected.

4.6.2.4.3 Test 4.3: PSU Capabilities verification (slow)

Note: This set of tests is applicable to the following Intel ME firmware variants: Intel NM, DNM.

This test checks PSUs capabilities using a slow command exchange. It is done automatically by MESDC console without any user interaction.

Note that this test communicates with PSUs over Intel ME to check PSUs capabilities. It checks for communication errors, support for a few basic PMBus commands, correctness of returned capabilities and power measurements. All of the commands are sent by Intel ME to PSUs synchronously.

Procedure:

1. Send several PMBus commands to all PSUs using MESDC command Access SMBus.
2. Check for communications errors with PSUs.
3. Check support for a few basic PMBus commands.
4. Check capabilities returned by PMBus commands.
5. Check power measurements returned by PSUs.

Success Criteria:

1. No communication errors with PSUs occurred.
2. Several basic commands are supported by PSUs.
3. Capabilities returned by commands are reasonable.
4. Power measurements returned by PSUs are non-zero.

4.6.2.4.4 Test 4.4: PSU Capabilities verification (fast)

Note: This set of tests is applicable to the following Intel ME firmware variants: Intel Node Manager.

This test checks PSUs capabilities using a fast command exchange. It is done automatically by MESDC console without any user interaction.

Note that this test communicates with PSUs over Intel ME to check PSUs capabilities. It checks for communication errors, support for a few basic PMBus commands, correctness of returned capabilities and power measurements. All of the commands are sent by Intel ME to PSUs asynchronously (in a chain).

Procedure:

1. Send several PMBus commands to all PSUs using MESDC command Access SMBus.
2. Check for communications errors with PSUs.
3. Check support for a few basic PMBus commands.
4. Check capabilities returned by PMBus commands.
5. Check power measurements returned by PSUs.

Success Criteria:

1. No communication errors with PSUs occurred.
2. Several basic commands are supported by PSUs.
3. Capabilities returned by commands are reasonable.

4. Power measurements returned by PSUs are non-zero.

4.6.2.4.5 Test 4.5: Platform Power Readings Precision

Note: This set of tests is applicable to the following Intel ME firmware variants: Intel NM.

This test checks platform power readings precision for low and high power values. The power readings provided to Intel ME FW may not differ more than 5% from the power consumption measured with external power analyzer. Otherwise power limiting algorithms may work unsteadily.

To run this test you need to configure MESDC with command line parameters that can be used to obtain power readings from the external power analyzer.

CPU power load application and External power analyzer application textboxes appear only if this test is selected.

Note that this test is able to perform complete verification for input power readings only, provided either by BMC or PMBus PSU. If output (DC) power, or per-rail readings are used, this test only checks whether none zero values are provided to Intel ME.

Procedure:

1. Read power consuming by platform with external tool.
2. Read power (total) using “NM Get Current Reading” diagnostic command (0x4B).
3. Compare these measurements.
4. Test passes if accuracy is better than 5%.

Success Criteria:

- For both readings values from external analyzer and Intel ME power statistics may differ no more than 10%. The accuracy and other requirements for PSUs are defined in [PMBus AC/DC Profile].

SUT Load	Intel ME Statistics [W]	External Reading [W]	Difference [%]
Idle			
Busy			

4.6.2.4.6 Test 4.6: P/T State Limit Control

Note: This set of tests is applicable to the following Intel ME firmware variants: Intel NM.

This test verifies BIOS support for setting P-state and T-state limit. It is done automatically by MESDC console without any user interaction.

Note that this test does not check whether the operating system respects the P-state and T-state limit set by Intel ME. It will be tested in 4.6.2.4.7. This test only checks whether Intel Node Manager notifications reach OSPM and are properly handled in ACPI tables provided by BIOS.

Procedure:

1. Set max allowed CPU P-state and T-state using "Set Max Allowed CPU P-State/T-State" diagnostic command (0x56)
   a. P-State number to be set = 0.
   b. T-State number to be set = 0.
2. If above command succeed – test passed.

Success Criteria:

- MESDC reports that the OSPM acknowledge was received.

4.6.2.4.7 Test 4.7: Power Limiting

Note: This set of tests is applicable to the following ME firmware variants: Intel Node Manager.

This test verifies generic power limiting functionality in a simple scenario.

To run this test you need to configure MESDC with command line parameters that can be used to generate host load at 100%.

Procedure:

1. Read power (total) using "NM Get Current Reading" diagnostic command (0x4B).
2. Apply load using external application.
3. Read power (total) using "NM Get Current Reading" diagnostic command (0x4B).
4. Create Intel NM Policy with power limit in the middle between previous readings.
5. Read power (total) using "NM Get Current Reading" diagnostic command (0x4B).
6. Compare last readings with power limit set in created Intel NM Policy.
7. Remove created Intel NM Policy.
8. Test passes if accuracy is better than 2.5%.

Success Criteria:

- MESDC reports that the average power was within +2.5%. -5% range of the requested power limit.

4.6.2.4.8 Test 5.1: Verify MCTP discovery process with ME acting as a Bus Owner

Note: This test is applicable to the following ME firmware variants: Silicon Enabling, Intel MN.

This is a test for verify MCTP Endpoints discovery process with ME acting as a Bus Owner.

It uses diagnostic command “Get MCTP Statistic”.

The test performs one time run of above command, parsing response data (MCTP statistics) and report adequate message(s) with test status.

Also the test may print additional information into log file named ‘report-<date>.txt’ which is not displayed in the GUI.

If additional information is available status messages contain “**Please look to compliance test log for more details”**.

Procedure:

- Same as above

The test checks followed conditions:

Result	Additional info	Condition
FAIL	MCTP Get Statistics command not supported!	MCTP is not supported by ME
FAIL	MCTP statistics not ready yet!	MCTP statistics are not yet collected by ME
FAIL	Illegal Bus Owner EID, allowed range: [0x00 - 0xFE].	Bus Owner EID is out of allowed range
FAIL	Illegal number of supported Endpoints, allowed range: [0 - 16].	Number of supported Endpoints is out of allowed range.
FAIL	Number of discovered EPs exceeds number of supported EPs.	Number of discovered Endpoints is out of number supported Endpoints
FAIL	Empty EP discovery statistics (possible statistics reset).	All statistics are zeros
FAIL	Some Endpoints discovered, but no control messages captured (possible statistics reset).	All followed conditions are met: any Endpoint is discovered, Intel ME doesn’t collect any statistics for “Number of Prepare for Endpoints Discovery responses sent”, Intel ME does not collect any statistics for “Number of Endpoints Discovery responses received”. Such scenario may suggest MCTP statistics was cleared while discovery process.
FAIL	Inconsistent statistics.	One of followed conditions is met: Number of Get EID received responses is different than number of Discovered Endpoints. Number of Set EID received responses is less than number of Discovered Endpoints.

		Number of Set EID received responses is two times bigger than number of Discovered Endpoints.
WARNING*	Probably more EPs present than discovered.	Number of "Prepare for Endpoints Discovery responses sent" is bigger than for number of Discovered Endpoints
WARNING*	Possible timeouts while waiting for control messages.	One of followed conditions is met: Number of "Get EID Requests Sent" is different than number of "Get EID Responses Received", Number of "Set EID Requests Sent" is different than number of "Set EID Responses Received".

* WARNINGS are not displayed in GUI

Success Criteria:

1. If test not failed but met one of warnings.

   Test will **PASS** with message: **Discovery process performed successfully. However there are warnings…**

2. If test not failed and:
   a. Number of Endpoints is zero, test will **PASS** with message: **No Endpoints discovered.**,
   b. Endpoints are discovered test will **PASS** with message: **Discovery process performed successfully.**

4.6.2.4.9 Test 6.1: BMC verification

Note: This test is applicable to the following ME firmware variants: Intel Node Manager. Test needs RMCP+ connection.

This is test for verification of communication between MESDC (RMCP+) and ME via BMC.

The test sends group of IPMI commands to ME via BMC using RMCPP interface and analyzes their responses to determine that BMC Proxy works correct.

Procedure:

1. Send group of commands to ME via BMC
2. Check their responses

Success Criteria:

1. All commands should return appropriate completion codes.

4.6.2.4.10 Test 7.1: Platform stability after Reset To Defaults

Note: This set of tests is applicable to the following Intel ME firmware variants: SiEn, Intel Node Manager.

Test needs IPMB or RMCP+ interface.

This test verifies Restore Factory Defaults command.

Procedure:

1. Check that platform works by sending Get Device Id.
2. Check that platform works and is in operational by sending Read Aux Register.
3. Start to iterate 5 times.
    - 3.1. Send Restore Factory Default.
    - 3.2. Init interface.
    - 3.3. Check that platform works by sending Get Device Id.
    - 3.4. Check that platform works and is in operational mode by sending Read Aux Register.
4. End of iteration.

Success Criteria:

1. After sending Restore Factory Defaults command, platform returned to working state.

4.6.2.4.11 Test 7.2: Policy storage parameter

Note: This set of tests is applicable to the following Intel ME firmware variants: Intel Node Manager.

Test needs IPMB or RMCP+ interface.

This test verifies that storage parameter can be changed in defined policy.

Procedure:

1. Start to iterate through all available domains repeatedly for 8 seconds.
    - 1.1. Get data limitations for current domain.
    - 1.2. Enable policy control for all power domains.
    - 1.3. Enable policy control for current domain.
    - 1.4. Using obtained data limitations create policy in current domain with storage parameter set to 1 (volatile) and state set to enabled.
    - 1.5. Check if policy was created properly.

    1.6. Disable policy.
    1.7. Change policy storage parameter to 0 (persistent).
    1.8. Enable policy.
    1.9. Check if policy was changed correctly.
    1.10. Disable and remove policy.
    1.11. Disable policy control for current domain.
    1.12. Disable policy control for all power domains.
2. End of iteration.

Success Criteria:

1. Policy was created properly.
2. Changing storage parameter succeeded.

4.6.2.4.12 Test 7.3: Volatile Policy – Cold Reset

Note: This set of tests is applicable to the following Intel ME firmware variants: Intel Node Manager.

Test needs IPMB or RMCP+ interface.

This test verifies that volatile policies do not exist after reset.

Procedure:

1. Enable policy control for all power domains.
2. Enable policy control for each available domain.
3. Create volatile policy in each available domain.
    3.1. Get data limitations for current domain.
    3.2. Using obtained data limitations create policy in current domain with storage parameter set to 1 (volatile) and state set to enabled.
    3.3. Check if policy was created properly.
4. Reset ME by sending Cold Reset IPMI command.
5. Wait 3 seconds for ME.
6. Check that all volatile policies do not exist.

Success Criteria:

1. All volatile policies disappeared after reset.

4.6.2.4.13 Test 7.6: Adding and removing policies for multiple domains

Note: This set of tests is applicable to the following Intel ME firmware variants: Intel Node Manager.

Test needs IPMB or RMCP+ interface.

This test verifies firmware ability to limit power for multiple domains at the same time.

Procedure:

1. Detect minimal and maximal power consumption for all domains
2. Enable the Intel® Node Manager Policy Control globally
3. Disable the Intel® Node Manager Policy Control per domain: Entire platform
4. Repeat the previous task for domains: CPU subsystem and Memory subsystem
5. Set the policies for domain: Entire platform
6. Enable the Intel® Node Manager Policy Control per domain: Entire platform
7. Start generic load
8. Wait for policy to start limiting
9. Check if policy is limiting correctly
10. Validate which policy is actively limiting
11. Disable domain: Entire platform
12. Check if policy is still limiting
13. Repeat tasks 5 - 12 for domains: CPU subsystem and Memory subsystem

Success Criteria:

1. All sent IPMI commands shall return completion code 0x00.
2. The limiting quality should be as expected (2.5%).

4.6.2.4.14 Test 7.7: Set PSU Configuration

Note: This set of tests is applicable to the following Intel ME firmware variants: Intel Node Manager.

Test needs IPMB or RMCP+ interface.

This test verifies that PSU Configuration is saved properly.

Procedure:

1. Get PSU configuration.

2. Set different PSU configuration.
3. Check if PSU configuration was changed correctly.
4. Restore initial PSU configuration.

Success Criteria:

1. Set and get values are the same.

4.6.2.4.15 Test 7.8: Set PSU Configuration w/Power Cycle

Note: This set of tests is applicable to the following Intel ME firmware variants: Intel Node Manager.

Test needs IPMB or RMCP+ interface.

This test verifies that PSU Configuration is saved properly after Power Cycle.

Procedure:

1. Get PSU configuration.
2. Set different PSU configuration using 0x00 and 0xFF values too.
3. Power Cycle DUT.
4. Check if PSU configuration was changed correctly.
5. Restore initial PSU configuration.

Success Criteria:

1. Set and get values are the same.

4.6.2.4.16 Test 7.10: Predictive Power Limit enable/disable

Note: This set of tests is applicable to the following Intel ME firmware variants: Intel Node Manager.

Test needs IPMB or RMCP+ interface.

Test also requires an application (.exe) or script (.bat) that will start generic load on DUT on 100% (load must last for at least 3 seconds).

This test verifies that Predictive Power Limit policy can be enabled/disabled.

Procedure:

1. Start generic load on DUT on 100% using given application/script.
2. Create Predictive Power Limit policy.
   - 2.1. Get current power value

2.2. Create policy with these values:
    - Domain = Entire Platform
    - Policy state = Enabled
    - Power limit = 0.8 * current power value
    - Power correction = Aggressive
    - Storage parameter = 1 (volatile)

3. Check if policy was created properly.
4. Disable policy.
5. Check if policy is not limiting.
6. Enable policy.
7. Wait a second for policy to start limiting.
8. Check if policy is limiting.
9. Remove policy.

Success Criteria:

1. Policy was created properly.
2. Policy was not limiting when disabled.
3. Policy was limiting when enabled.

4.6.2.4.17 Test 7.11: PSU Current Total Input Power Reading

Note: This set of tests is applicable to the following Intel ME firmware variants: Intel Node Manager.

Test needs IPMB or RMCP+ interface.

Test also requires an application (.exe) or script (.bat) that will start generic load on DUT on 100% (load must last for at least 1 min).

This test will verify that Intel® NM FW is able to properly read Power values from PSU. Test will use Diagnostic Console.

Procedure:

1. Start generic load on DUT on 100% using given application/script.
2. Create Predictive Power Limit to set stable platform power consumption.
3. Using power meter read current platform power consumption
4. Send Diagnostic Console command "NM Get Stats" and verify that returned current power value is similar to platform power consumption read from power meter.

5. Disable and remove policy

9.

Success Criteria:

1. FW properly read Power values from PSU.

4.6.2.4.18 Test 7.13: HW Protection Policy is present when no PSUs are configured(TBD)

Note: This set of tests is applicable to the following Intel ME firmware variants: Intel Node Manager.

Test needs IPMB or RMCP+ interface.

This test will verify that HW Protection Policy is present in Intel® NM Policy Control when no PSUs are configured.

Procedure:

1. Send IPMI command "Get Node Manager Policy" with the following parameters:
   a. Byte 4 [0:3] = 03h - Domain Id - HW Protection
   b. Byte 5 = 00h - Policy Id
2. Verify that response completion code is 00h
3. Using IPMI command "Set PMBus Device Configuration" to remove all configured PSUs (set PMBus address to 0x00)
4. Send IPMI command "Get Node Manager Policy" with the following parameters:
   a. Byte 4 [0:3] = 03h - Domain Id - HW Protection
   b. Byte 5 = 00h - Policy Id
5. Verify that response completion code is still 00h - Policy should not be removed.

Success Criteria:

1. HW Protection Policy is present in Intel® NM Policy Control when no PSUs are configured.

4.6.2.4.19 Test 7.14: HW Protection Policy Statistics (TBD)

Note: This set of tests is applicable to the following Intel ME firmware variants: Intel Node Manager.

Test needs IPMB or RMCP+ interface.

This test will verify functionality of HW Protection Domain. IPMI commands depicted below can return Global and Per Policy Statistics for power. IPMI commands Reset and Get have different Modes filed that identify type of returned statistics.

Procedure:

1. Execute command ´Get Node Manager capabilities´ to discover Intel® Node Manager initial knowledge about whole platform
2. Create an Intel Node Manager policy
3. Use IPMI command ´Get Node Manager Statistics´ for reading Global and Per Policy Statistics
4. Execute resetting of Global and Per Policy Stats, Use IPMI command ´Reset Node Manager Statistics´
5. Verify proper Completion Code
6. Use IPMI command ´Get Node Manager Statistics´ for reading Global and Per Policy Stats.

Success Criteria:

1. All commands are executed properly.

4.6.2.4.20 Test 7.15: Simultaneous Parameters change on single Policy (TBD)

Note: This set of tests is applicable to the following Intel ME firmware variants: Intel Node Manager.

Test needs IPMB or RMCP+ interface.

Check that continuous and simultaneous change of many Policy parameters on single enabled Policy works properly. If at least one parameter is invalid, all transaction shall be rejected with proper Completion Code.

Procedure:

1. Assure that Intel® Node Manager is Enabled.
2. Execute ´Set Nm Policy´ with Policy State bit set to 1 (Policy Enabled)
3. Using ´Get Nm Capabilities´ prepare valid and invalid values for policy parameters change scenario
4. In loop change all policy parameters that are allowed for modifications without disabling policy
5. Check ´Set Nm Policy´ IPMI command Completion Code Correctness
6. For valid values check that new parameters values are really applied to policy.

Success Criteria:

1. The following Policy parameters do not change:
   - Domain ID
   - Policy Trigger Type.

4.6.2.4.21 Test 7.16: Alert Thresholds modifications

Note: This set of tests is applicable to the following Intel ME firmware variants: Intel Node Manager.

Test needs IPMB or RMCP+ interface.

This test will verify that it is possible to perform continuous Alert Thresholds modifications on enabled Policy.

Procedure:

7. Create Policy with valid parameters according to current Domain and Policy Trigger.
8. In loop set Alert Thresholds with different valid and invalid parameters (Thresholds Number and Threshold Limits)
9. Each step check Completion Code correctness
    9.1. For valid parameters, Completion Code 0x00 is expected
    9.2. Additionally check that Alert Thresholds modifications has been applied
    9.3. For invalid parameters Completion Code != 0x00 is expected
    9.4. Additionally check that Alert Thresholds modifications has been not applied
10. Remove Policy

Success Criteria:

2. All returned Completion Codes should be 0x00

10.

4.6.2.4.22 Test 7.17: Alert Thresholds and Suspend Periods add modify and remove

Note: This set of tests is applicable to the following Intel ME firmware variants: Intel Node Manager.

Test needs IPMB or RMCP+ interface.

This test will check that Intel® Node Manager allows us to change and remove Alert Thresholds and Suspend Periods on enabled Policy.

Procedure:

1. Check that Intel® Node Manager Policy Control is enabled
2. Define one policy with policyId = 0x95 and make sure that it is in enabled state
3. Set Alert Thresholds and Suspend Periods on enabled Policy

4. Change Thresholds and Suspend Periods with new valid parameters
5. Remove Alert Thresholds and check that they are removed
6. Remove Suspend Periods and check that they are removed
7. Remove Intel Node Manager Policy

Success Criteria:

11. Alert Thresholds and Suspend Periods should be set and removed correctly.

4.6.2.4.23 Test 7.19: Aux Read over IPMB

Test needs IPMB or RMCP+ interface.

12. This test will check that Aux Read command works properly.

Procedure:

1. Take each value from the defined list.
2. Turn that into an address and add it to the command prefix.
3. Send the request via the IPMB.
4. Check the return completion code and data length.
5. If the completion code and length match the values found in the defined list, the passes. If not the test fails.
6. Continue until the list is exhausted.

Success Criteria:

1. Completion code and length match the values found in the defined list.

4.6.2.4.24 Test 7.20: MESDC Memory read command over IPMB

Test needs IPMB or RMCP+ interface.

13. This test will check that Memory Read command works properly.

Procedure:

1. Take each value from the defined list.
2. Turn that into an address and add it to the command prefix.
3. Send the request via the IPMB.
4. Check the return completion code and data length.

5. If the completion code and length match the values found in the defined list, the passes. If not the test fails.
6. Continue until the list is exhausted.

Success Criteria:

1. Completion code and length match the values found in the defined list.

4.6.2.4.25 Test 7.21: Get PID command over IPMB

Test needs IPMB or RMCP+ interface.

14. This test will check that Get PID command works properly.

Procedure:

1. Send Get PID command
2. Check response status
3. Check response length

Success Criteria:

1. Response status is 0x00
2. Response Length is 20 bytes

4.6.2.4.26 Test 7.22: Get, Reset Statistics over HECI2

Note: This set of tests is applicable to the following Intel ME firmware variants: Intel Node Manager.

Verify that MESDC is able to send commands over HECI2

Procedure:

1. Send Reset HECI 2 Statistics and check completion code.
2. Generate traffic on HECI 2 Interface(using set P/T-state command)
3. Check ME to Host Message Counter value
4. Check Control Status Register value
5. Check Control Status Register Host Access value
6. Check Host to ME Message Counter value
7. Do points from 2 to 6 ten times

Success Criteria:

1. Host to ME Message Counter value is greater than value in previous step.
2. ME to Host Message Counter value is greater than value in previous step.
3. Control Status Register, Control Status Register Host Access value have proper values.

4.6.2.4.27 Test 7.23: Get, Reset Statistics over HECI1

Test needs HECI or Agent interface.

Verify that MESDC is able to send commands over HECI1

Procedure:

1. Send Reset HECI 1 Statistics and check completion code.
2. Generate traffic on HECI 1 Interface
3. Check ME to Host Message Counter value
4. Check Control Status Register value
5. Check Control Status Register Host Access value
6. Check Host to ME Message Counter value
7. Do points from 2 to 6 ten times

Success Criteria:

1. Host to ME Message Counter value is greater than value in previous step.
2. ME to Host Message Counter value is greater than value in previous step.
3. Control Status Register, Control Status Register Host Access value have proper values.

4.6.2.4.28 Test 7.24: Basic functionality of OEMDiag Command

Note: This set of tests is applicable to the following Intel ME firmware variants: Intel Node Manager

Test needs IPMB or RMCP+ interface.

Test will verify if MESDC commands function over IPMB interface.

Procedure:

1. Send commands listed below and check their length and completion code

Commands List :

Thread Statistics

Get Scalability Factors

Get SMBus Address

IPMB Statistics

Get Me Boot State

Data Power Read

Get Version

Memory Status

PBC Get Statistics

Get Ptam Statistics

Success Criteria:

1. All responses completion codes are equal to 0x00 and their length are correct.

4.6.2.4.29 Test 7.25: MESDC over IPMB commands functional in Recovery Mode

Note: This set of tests is applicable to the following Intel ME firmware variants: Intel Node Manager.

Test needs IPMB or RMCP+ interface.

Test will verify if MESDC commands function over IPMB interface in Recovery mode.

Procedure:

1. Put FW into Recovery mode.
2. Send commands listed below and check their length and completion code.
3. Put FW into Operational mode.

Commands List :

Thread Statistics

Get Scalability Factors

Get SMBus Address

IPMB Statistics

Get Me Boot State

Data Power Read

Get Version

Memory Status

PBC Get Statistics

Get Ptam Statistics

Success Criteria:

2. All responses completion codes are equal to 0x00 and their length are correct.

4.6.2.4.30 CUPS_001: CUPS Status Test

Note: This set of tests is applicable to the following Intel ME firmware variants: CUPS.

Test needs IPMB or RMCP+ interface.

Test will verify CUPS feature is enabled in NM/SiEn.

Procedure:

1. Send Get CUPS Capabilities command.
2. Check command completion code.
3. Check CUPS Capabilities field in response.

Success Criteria:

1. Completion code should be 0x00.
2. CUPS Capabilities should be 0x01.

4.6.2.4.31 CUPS_002: IN-Band PECI test

Note: This set of tests is applicable to the following Intel ME firmware variants: `PECI_PROXY`.

Test needs IPMB or RMCP+ interface.

Test will verify that PECI over DMI is functional.

Procedure:

1. Send Raw PECI command GetTemp()
2. Check completion code

Success Criteria:

1. Completion code should be 0x00.

4.6.2.4.32 CUPS_004: CUPS function test

Note: This set of tests is applicable to the following Intel ME firmware variants: `CUPS`.

Test needs IPMB or RMCP+ interface.

Test will verify the CUPS feature is functional for all 3 domains(core, mem, io).

Procedure:

1. Send Get Sensor Reading command for core sensor, mem sensor, io sensor
2. Check completion codes and response length.

Success Criteria:

1. Setting field(byte3) is equal to 0xC0 and Byte 1 field(byte4) is equal to 0x00.

4.6.2.4.33 CUPS_005: CUPS reading test

Note: This set of tests is applicable to the following Intel ME firmware variants: `CUPS`.

Test needs IPMB or RMCP+ interface.

Test will verify CUPS data provided by ME is valid.

Procedure:

1. Send Get Cups Data with Parameter Selector field set as 0x04.
2. Check response completion code and length
3. Verify response

Success Criteria:

1. Byte5 – byte12 != 0 (some of the bits in byte 5 -12 should be set).
2. Byte13 – byte20 != 0(some of the bits in byte 13 – 20 should be set).
3. Byte21 - byte28 == 0(all bits should be zero).

4.6.2.4.34 CUPS_006: CUPS parameter test

Note: This set of tests is applicable to the following Intel ME firmware variants: `CUPS`.

Test needs IPMB or RMCP+ interface.

Test will verify that ME FW has initialized all CUPS parameters and cleared all irrelevant values.

Procedure:

1. Send Get Cups Data with Parameter Selector field set as 0x03.
2. Check response completion code and length
3. Verify response

Success Criteria:

4. Byte5 – byte12 != 0 (some of the bits in byte 5 -12 should be set).
5. Byte13 – byte20 != 0(some of the bits in byte 13 – 20 should be set).
6. Byte21 - byte28 != 0(all bits should be zero).

4.6.2.4.35 MCTP_001: MCTP BO HECI message test

Note: This set of tests is applicable to the following Intel ME firmware variants: Intel Node Manager.

Test needs SMBus interface.

This test verifies that B0 is registered and that HECI message has been sent.

Procedure:

1. Check schematics SMLink data, and clock signals termination.
2. Check SMLink are not connected together.
3. Use Aardvark and MESDC tool to check MCTP statistics.
4. Plug MCTP PCIe Device (endpoint)
5. Power on the platform.
6. Verify statistics.

Success Criteria:

One of statistics: 1, 2, 3, 4, 5, 6, 9 is nonzero.

4.6.2.4.36 MCTP_002: BO – ME MCTP Proxy communication

Note: This set of tests is applicable to the following Intel ME firmware variants: Intel Node Manager.

Test needs SMBus interface.

This test verifies B0 – ME communication and that PROXY functions properly.

Procedure:

1. Check schematics SMLink data, and clock signals termination.
2. Check SMLink are not connected together.
3. Use Aardvark and MESDC tool to check MCTP statistics.
4. Plug MCTP PCIe Device (endpoint)
5. Power on the platform.
6. Verify statistics.

Success Criteria:

1. One of statistics: 4, 5, 6, 9 is nonzero.
2. If statistics 4 is different from statistics 5 then B0 – EP communication did not work entirely properly.

4.6.2.4.37 MCTP_003: MCTP Endpoint - BO communication

Note: This set of tests is applicable to the following Intel ME firmware variants:
Intel Node Manager.

Test needs SMBus interface.

This test verifies EP and B0 communication.

Procedure:

1. Check schematics SMLink data, and clock signals termination.
2. Check SMLink are not connected together.
3. Use Aardvark and MESDC tool to check MCTP statistics.
4. Plug MCTP PCIe Device (endpoint)
5. Power on the platform.
6. Verify statistics.

Success Criteria:

Statistics 9 is nonzero.

4.6.2.4.38 MIC_001: MIC detection test.

Note: This set of tests is applicable to the following Intel ME firmware variants:
Intel Node Manager.

Test needs RMCP+ or IPMB interface.

This test verify MIC card (Intel Xeon Phi) has been integrated successfully on the platform.

Procedure:

1. Send Get PCIe SMBus Slot Card Info command using Intel ME MIC Proxy
2. Check Response completion code.
3. If completion code is equal 0xC1, send this command using BMC MIC Proxy.

Success Criteria:

3. Field Total Number Cards Detected in response should be equal to real number of these cards on target.

4.6.2.4.39 MIC_002: MIC proxy test.

Note: This set of tests is applicable to the following Intel ME firmware variants:
Intel Node Manager.

Test needs RMCP+ or IPMB interface.

This test verify MIC proxy is functional after a MIC card has been successfully integrated and connection has been established with Intel ME.

Procedure:

1. Send Get PCIe SMBus Slot Card Info command with CardInstance filed set as zero using Intel ME MIC Proxy.
2. Check Response completion code and TotalNumberCardsDetected field.
3. Send Get PCIe SMBus Slot Card Info command 8 times for values from 1 to 8 for CardInstance filed.
4. If command in previous step had competition code equals to 0x00 send Slot IPMB command with Address Type field set as Address Protocol Bus field from response, Slot Number field set as Slot Number field form response Id or Slave Address field set as Id Slave Address from response, NetFun set as 0x30, Cmd set as 0x06, and cmd data set as 0x01.

Success Criteria:

1. Field Total Number Cards Detected in response should be equal to real number of these cards on target.
2. Number responses for SlotIPMB command which returns completion code equals to 0x00 should be equal to Total Numbers Cards Detects field in Get PCIe SMBus Slot Card Info command response with CardInstance filed set as zero.

4.6.2.4.40 MIC_003: Reverse MIC proxy test.

Note: This set of tests is applicable to the following Intel ME firmware variants:
Intel Node Manager.

Test needs RMCP+ or IPMB interface.

This test verify MIC proxy is functional after a MIC card has been successfully integrated and connection has been established with Intel ME.

Procedure:

1. Send Get PCIe SMBus Slot Card Info command with CardInstance filed set as zero using BMC MIC Proxy.
2. Check Response completion code and TotalNumberCardsDetected field.
3. Send Get PCIe SMBus Slot Card Info command 8 times for values from 1 to 8 for CardInstance filed.
4. If command in previous step had competition code equals to 0x00 send Slot IPMB command with Address Type field set as Address Protocol Bus field from response, Slot Number field set as Slot Number field form response Id or Slave Address field set as Id Slave Address from response, NetFun set as 0x30, Cmd set as 0x06, and cmd data set as 0x01.

Success Criteria:

1. Field Total Number Cards Detected in response should be equal to real number of these cards on target.
2. Number responses for SlotIPMB command which returns completion code equals to 0x00 should be equal to Total Numbers Cards Detects field in Get PCIe SMBus Slot Card Info command response with CardInstance filed set as zero.

4.6.2.4.41 MIC_004: MIC power reading test.

Note: This set of tests is applicable to the following Intel ME firmware variants: Intel Node Manager.

Test needs RMCP+ or IPMB interface.

This test verify that Intel ME is able to obtain the total power statistics including MIC power. Intel ME communicates to MIC through BMC.

Procedure:

1. Send Get Intel Node Manager Statistics for HPIO Domain

Success Criteria:

1. Check that :
    a. Completion code == 0.
    b. Byte[2-4] == Intel Manufacturer Id
    c. Bytes[5-12] != 0

4.6.2.4.42 NM_001: NM BIOS support test

Note: This set of tests is applicable to the following Intel ME firmware variants: Intel Node Manager.

Test needs SMBus interface.

This test verifies whether Intel Node Manager running in Intel ME has received correct host configuration info from BIOS. It is done automatically by MESDC console without any user interaction.

Procedure:

1. Read max allowed CPU P/T-states using "Get Max Allowed CPU P-State/T-State" diagnostic command (0x57).
2. Verify that "Current maximum P-State" and "Current maximum T-State" are greater or equal 0.
3. Read max allowed CPU cores using "Get Max Allowed CPU P-State/T-State" diagnostic command (0x57).
4. Verify that "Total requested by Intel ME number of allowed cores on a system" is greater or equal 0.
5. If all above conditions passed test passes.

Success Criteria:

1. MESDC reports that the host configuration info looks reasonable.

4.6.2.4.43 NM_002: NM platform power reading test

Note: This set of tests is applicable to the following Intel ME firmware variants: Intel Node Manager.

Test needs RMCP+ or IPMB interface.

This test verify that power consumption readings are correct in platform power domain.

Test also requires an application (.exe) or script (.bat) that will start memory and CPU load on DUT on 100% (load must last for at least 2 min) and another one to read from external power meter.

Procedure:

1. With IPMI command "Get NM Statistics" get global power statistics for platform power domain.
2. Check with external power meter platform power consumption is matching the current value (Byte 5:6) reported by global power statistics.
3. Run load on host system with PTU
4. With IPMI command "Get NM Statistics" get global power statistics for platform power domain.
5. Check with external power meter platform power consumption is matching the current value (Byte 5:6) reported by global power statistics.

Success Criteria:

1. External power meter platform power consumption is matching the current value (Byte 5:6) reported by global power statistics.

4.6.2.4.44 NM_003: NM CPU power reading test

Note: This set of tests is applicable to the following Intel ME firmware variants: Intel Node Manager.

Test needs RMCP+ or IPMB interface.

This test verify that power consumption readings are correct in CPU power domain.

Test also requires an application (.exe) or script (.bat) that will start CPU load on DUT on 100% (load must last for at least 2 min)

Procedure:

1. With IPMI command "Get NM Capabilities" get CPU domain power range Min Power (Byte 8:9) and Max Power (Byte 6:7).
2. With IPMI command "Get NM Statistics" get global power statistics for CPU power domain -current value (Byte 5:6).
3. Run load on host system.
4. With IPMI command "Get NM Statistics" get global power statistics for CPU power domain - current value (Byte 5:6).

Success Criteria:

1. IPMI command "Get NM Statistics" global power statistics for CPU power domain - current value (Byte 5:6) is greater than zero and lower than Max Power (Byte 6:7) reported by IPMI command "Get NM Capabilities" for CPU domain.

4.6.2.4.45 NM_004: NM memory power reading test

Note: This set of tests is applicable to the following Intel ME firmware variants: Intel Node manager.

Test needs RMCP+ or IPMB interface.

This test verify that power consumption readings are correct in Memory power domain.

Test also requires an application (.exe) or script (.bat) that will start memory load on DUT on 100% (load must last for at least 2 min)

Procedure:

1. With IPMI command "Get NM Capabilities" get memory domain power range Min Power (Byte 8:9) and Max Power (Byte 6:7).
2. With IPMI command "Get NM Statistics" get global power statistics for memory power domain -current value (Byte 5:6).
3. Run load on host system.
4. With IPMI command "Get NM Statistics" get global power statistics for memory power domain - current value (Byte 5:6).

Success Criteria:

1. IPMI command "Get NM Statistics" global power statistics for memory power domain - current value (Byte 5:6) is greater than zero and lower than Max Power (Byte 6:7) reported by IPMI command "Get NM Capabilities" for memory domain.

4.6.2.4.46 NM_005: NM HPIO power reading test

Note: This set of tests is applicable to the following Intel ME firmware variants: Intel Node Manager.

Test needs RMCP+ or IPMB interface.

This test verify that power consumption readings are correct in HPIO power domain.

Test also requires an application (.exe) or script (.bat) that will start HPIO domain load on DUT on 100% (load must last for at least 2 min).

Procedure:

1. With IPMI command "Get NM Capabilities" get HPIO domain power range Min Power (Byte 8:9) and Max Power (Byte 6:7).
2. With IPMI command "Get NM Statistics" get global power statistics for HPIO power domain -current value (Byte 5:6).
3. Run load on host system.
4. With IPMI command "Get NM Statistics" get global power statistics for HPIO power domain - current value (Byte 5:6).

Success Criteria:

1. IPMI command "Get NM Statistics" global power statistics for HPIO power domain - current value (Byte 5:6) is greater than zero and lower than Max Power (Byte 6:7) reported by IPMI command "Get NM Capabilities" for HPIO domain.

4.6.2.4.47 NM_006: NM RTC time test

Note: This set of tests is applicable to the following Intel ME firmware variants: Intel Node manager.

Test needs RMCP+ or IPMB interface.

This test verify that valid RTC time is passed to NM.

Procedure:

1. Get internal Intel NM clock value with IPMI command "Get SEL Time".
2. Check that reported time value is valid

Success Criteria:

IPMI command "Get SEL Time" response Present Timestamp value (Bytes 2:5) is different from 0xFFFFFFFF.

4.6.2.4.48 NM_009: NM platform power limiting test

Note: This set of tests is applicable to the following Intel ME firmware variants: Intel Node Manager.

Test needs RMCP+ or IPMB interface.

This test verify that power limiting is working correctly in platform power domain.

Test also requires an application (.exe) or script (.bat) that will start memory and CPU load on DUT on 100% (load must last for at least 2 min).

Procedure:

1. Run load on host system.
2. With IPMI command “Get NM Statistics” get global power statistics for platform power domain - current value (Byte 5:6).
3. With IPMI command “Set NM Policy” set NM policy for platform power domain with power limit set to 80% of the power statistics current value collected in previous step.
4. Wait for set NM policy correction time.
5. With IPMI command “Get NM Statistics” get global power statistics for platform power domain - current value (Byte 5:6) and verify if it is matching set NM policy power limit with 5% tolerance.

Success Criteria:

After setting Intel NM policy power consumption in platform power domain is equal to set Intel NM policy power limit with 5% tolerance.

4.6.2.4.49 NM_010: NM CPU power limiting test

Note: This set of tests is applicable to the following Intel ME firmware variants: Intel Node Manager.

Test needs RMCP+ or IPMB interface.

This test verify that power limiting is working correctly in CPU power domain.

Test also requires an application (.exe) or script (.bat) that will start CPU load on DUT on 100% (load must last for at least 2 min).

Procedure:

1. Run load on host system.

2. With IPMI command “Get NM Statistics” get global power statistics for CPU power domain - current value (Byte 5:6).
3. With IPMI command “Set NM Policy” set NM policy for CPU power domain with power limit set to 80% of the power statistics current value collected in previous step.
4. Wait for set NM policy correction time.
5. With IPMI command “Get NM Statistics” get global power statistics for CPU power domain - current value (Byte 5:6) and verify if it is matching set NM policy power limit with 5% tolerance.

Success Criteria:

After setting Intel NM policy power consumption in CPU power domain is equal to set Intel NM policy power limit with 5% tolerance.

4.6.2.4.50 NM_011: NM memory power limiting test

Note: This set of tests is applicable to the following Intel ME firmware variants: Intel Node manager.

Test needs RMCP+ or IPMB interface.

This test verify that power limiting is working correctly in memory power domain.

Test also requires an application (.exe) or script (.bat) that will start memory load on DUT on 100% (load must last for at least 2 min).

Procedure:

1. Run load on host system.
2. With IPMI command “Get NM Statistics” get global power statistics for memory power domain - current value (Byte 5:6).
3. With IPMI command “Set NM Policy” set Intel NM policy for memory power domain with power limit set to 80% of the power statistics current value collected in previous step.
4. Wait for set Intel NM policy correction time.
5. With IPMI command “Get NM Statistics” get global power statistics for memory power domain - current value (Byte 5:6) and verify if it is matching set NM policy power limit with 5% tolerance.

Success Criteria:

After setting Intel NM policy power consumption in memory power domain is equal to set Intel NM policy power limit with 5% tolerance.

4.6.2.4.51 PECI_001: PECI proxy test

Note: This set of tests is applicable to the following Intel ME firmware variants: Intel Node manager, PECI_PROXY.

Test needs RMCP+ or IPMB interface.

This test verify PECI proxy interface functionality.

Procedure:

Run PECI Ping command using IPMI CMD to verify if PECI Proxy communication is available

Success Criteria:

Returned completion code should be 0x00.

4.6.2.4.52 PTAS_001: BMC sensor readings test

Note: This set of tests is applicable to the following Intel ME firmware variants: Intel Node manager.

Test needs RMCP+ or IPMB interface.

This test verify thermals integrated successfully as part of Intel NM.

Procedure:

Send Get sensor readings command for: inlet temp PIA sensor, PIA Outlet Temp, PIA volumetric airflow sensor.

Success Criteria:

For all responses : Byte1 == 00; Byte3 == c0

4.6.2.4.53 PTAS_002: Volumetric airflow test

Note: This set of tests is applicable to the following Intel ME firmware variants: Intel Node Manager.

Test needs RMCP+ or IPMB interface.

This test verify volumetric airflow feature is functional in ME.

Procedure:

Send Get Node Manager Statistics command with Mode field set as 0x04(Global volumetric airflow statistics [1/10th of CFM])

Success Criteria:

Response bytes should be:

Byte1 == 0.

Bytes[2-4] == Intel ManufacturerId.

Bytes[5-12] != 0

4.6.2.4.54 PTAS_003: Outlet temp test.

Note: This set of tests is applicable to the following Intel ME firmware variants: Intel Node manager.

Test needs RMCP+ or IPMB interface.

This test verify outlet temp data is available from ME.

Procedure:

Send Get Node Manager Statistics command with Mode field set as 0x05(Global outlet airflow temperature statistics [Celsius])

Success Criteria:

Response bytes should be:

Byte1 == 0.

Bytes[2-4] == Intel ManufacturerId.

Bytes[5-12] != 0

4.6.2.4.55 PTU_001: Retrieve Characterization Results.

Note: This set of tests is applicable to the following Intel ME firmware variants: Intel Node manager.

Test needs RMCP+ or IPMB interface.

This test verify PTU is successfully launched on platform reset.

Procedure:

Send Get Node Manager Characterization Range commands for all power domains.

Success Criteria:

Check completion code for all responses.

4.6.2.4.56 PTU_002: Launch PTU On Next Boot.

Note: This set of tests is applicable to the following Intel ME firmware variants: Intel Node Manager.

Test needs RMCP+ or IPMB interface.

This test verify PTU is successfully launched on platform reset.

Procedure:

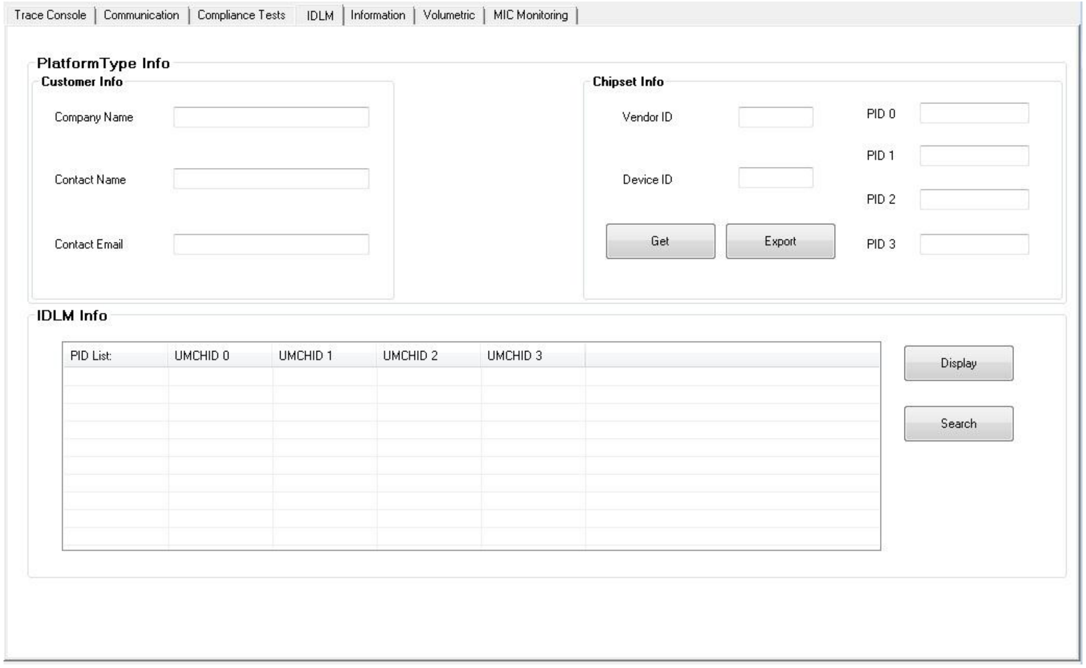
1. Send Node Manager Characterization Lunch Request command with Command Request Type filed set as 0x01.
2. Send Get Node Manager Characterization Range commands for all power domains.

Success Criteria:

1. Check completion code for all responses are 0x00.
2. Check TimeStamp field is not equal to zero or 0xFFFFFFFF.

4.7 IDLM Module

The IDLM (Intermediate Debug Load Module) functionality is used by Intel support team to enable additional debug capabilities in a particular system.



4.8 Command Line Mode Support

MESDC also has Command Line Interface to support test and reports to be run automatically. The command line will only support limited feature of MESDC. Here is the options supported in MESDC command line interface.

MESDC.exe [-ver] [-h|-?|-help] [-testsCmdLine [-xmlTestsPath <path>] [-powerReadAppPath <path>] [-cpuLoadAppPath <path>] [-memLoadAppPath <path>] [-bootTestsTimeout <ms>]] [-interface <interface>] [-aardvarkPort <port>] [-stateReport <report> [-logFile <file>]]

Table 4-1.MESDC Command Line Options

Option	Description
-h|-?|-help	Display help screen
-testsCmdLine	Run Compliance Tests from command line based on a xml file with pre-defined tests
-xmlTestsPath <path>	Path to xml file for Compliance Tests
-powerReadAppPath <path>	Path to Power Read Application for compliance test
-cpuLoadAppPath <path>	Path to CPU Load Application for compliance test
-memLoadAppPath <path>	Path to Memory Load Application for compliance test
-bootTestsTimeout <ms>	Timeout (ms) for compliance tests. Default: 240000
-interface <interface>	Select interface to use. For compliance test default interface is SMBus, for reports default is last use interface in GUI <SMBus> <IPMB> <RMCPP>

Option	Description
-aardvarkPort <port>	Aardvark port number. Default: 0
-stateReport <report>	Run report(s); <Icc> - Icc Settings; <MeFwHealth> - Intel ME FW Health Check; <Nm> - Intel Node Manager State Check; <NmExt> - Intel Node Manager State Check Extended; <CrashDump> - Me Crash Dump; <SusramDirect> - Susram Direct; <SusramMemory> - Susram Memory; <SusramParse> - Susram Parse; <MeConfBasic> - Intel ME configuration Basic Partition; <MeConfPresets> - Intel ME configuration Factory Presets; <SysInfo> - SYS Info; <Oem> - OEM Capture; <Smt> - SMT Driver Statistics; <Peci> - PECI Wire Statistics; <Heci> - HECI Statistics; <Mctp> - MCTP Statistics;
-logFile <file>	Set file log name. Default: ReportsLog.txt

4.8.1 Compliance tests

Here is an example on how to run compliance test(s) over CLI

MESDC.exe –testCmdLine –xmlTestPath test.xml –interface SMBus –*aardvarkPort 0*

Switches "– interface" and "– *aardvarkPort" are optional. MESDC.exe by default tries to connect by SMBus on port 0.*

Following is an example of test item in xml file:

```
<Test id="1.2" name="BIOS Get Interface Version message">
                <Params>enabled</Params>
 </Test>
```

Test ID should align with the test ID shown in the GUI interface.

Name is optional and for notes only

There are two valid values for Params field, "enabled" and "disabled".

All tests with enabled value in the pre-defined xml file will be run in the command line.

Test result/log will be stored at MESDC directory as the same as run test from GUI.

4.8.2 Reports

Here is an example on how to run report over CLI

MESDC.exe –stateReport SysInfo –logFile log.txt –interface SMBus –*aardvarkPort 0*

Switches "– interface" and "– *aardvarkPort" are optional. MESDC.exe by default tries to connect by interface used in GUI mode last time.*

Switch "–logFile" is optional. By default "ReportsLog.txt" name is used.

Argument for "–stateReport" switch is also optional. By default basic set of reports will be run ("Sys Info" and "Node Manager State Check" if Intel Node Manager feature enabled*).*

5 Flash Programming Tool

The spsFPT is used to program a complete SPI image into the SPI flash device(s).

spsFPT can program each region individually or it can program all of the regions with a single command. You can also use FPT to perform various functions such as:

- View the list of regions in the flash on the screen.
- Dump the contents of the flash to a file.
- Perform a binary file to flash comparison.
- Write to a specific address block.

5.1 System Requirements

The DOS version of spsFPT (**spsFPT.exe**) runs on MS DOS 6.22, DRMKDOS, and FreeDOS.

The Windows version (**spsFPTW.exe**) requires administrator privileges to run under Windows OS. You must use the **Run as Administrator** option to open the CLI in Windows* Vista 64/32-bit and Windows* 7 64/32-bit.

The Windows 64-bit version (spsFPTW64.exe) is designed for running in a 64-bit OS environment which does not have 32-bit compatible mode available, for example WinPE 64.

spsFPT requires an operating system to run on. It is designed to deliver a custom image to a computer that is already able to boot and is not a means to get a blank system up and running. spsFPT must be run on the system with the flash memory that you are programming.

One possible workflow for using spsFPT is:

1. A pre-programmed flash with a legacy or generic BIOS image is plugged into a new computer.
2. The computer boots.
3. spsFPT is run and a custom BIOS/Intel ME/GbE/PDR/DER (optional) image is written to flash.
4. The computer powers down.
5. The computer powers up, boots, and is able to access its Intel ME/GbE capabilities as well as any new custom BIOS features.

5.2 Flash Image Details

A flash image is composed of six regions. The locations of these regions are referred to in terms of where they can be found within the overall layout of the flash memory.

Figure 5-1. Flash Image Regions

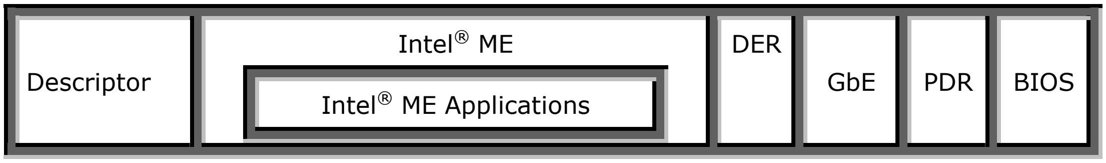


Table 5-1. Flash Image Regions–Description

Component	Description
Descriptor	Region that takes up a fixed amount of space at the beginning of the flash memory. Contains information such as: Space allocated for each region of the flash image. Read/write permissions for each region. A space that can be used for vendor-specific data.
Intel ME	Region that takes up a variable amount of space at the end of the Descriptor. Contains code and configuration data for Intel Server Platform Services firmware.
DER	Device extension region for Intel Node Manager-PTU feature
GbE	Optional region that takes up a variable amount of space at the end of the Intel ME region. Contains code and configuration data for GbE.
BIOS	Region that takes up a variable amount of space at the end of the flash memory. Contains code and configuration data for the entire platform.
PDR	Region that allows system manufacturers to define custom features for the platform.

5.3 Microsoft Windows* Required Files

The Microsoft Windows version of the spsFPT executable is spsFPTW**.exe**. The following files must be in the same directory:

- fparts.txt – contains a comma-separated list of attributes for supported flash devices. The text in the file explains each field. An additional entry may be required in this file to describe the flash part which is on the target system. Examine the target board before adding the appropriate attribute values. The supplied file is already populated with default values for SPI devices used with Intel CRBs.
- spsFPTW.exe – the executable used to program the final image file into the flash.
- Pmxdll.dll
- Idrvdll.dll

In order for tools to work under the Windows* PE environment, you must manually load the driver with the .inf file in the Intel® ME interface driver installation files. Once you locate the .inf file you must use the Windows* PE cmd drvload *.inf to load it into the running system each time Windows* PE reboots. Failure to do so causes errors for some features.

5.4 DOS Required Files

The DOS version of the spsFPT main executable is **spsFPT.exe**. The following files must be in the same directory:

- spsFPT.exe – the executable used to program the final image file into the flash.
- fparts.txt – contains a comma-separated list of attributes for supported flash devices. The text in the file explains each field. An additional entry may be required in this file to describe the flash part which is on the target system. Examine the target board before adding in the appropriate attribute values. The supplied file is already populated with default values for SPI devices used with CRBs.

5.5 EFI Required Files

The EFI version of the spsFPT main executable is **spsFPT.efi**. The following files must be in the same directory:

- spsFPT.efi – the executable used to program the final image file into the flash.
- fparts.txt – contains a comma-separated list of attributes for supported flash devices. The text in the file explains each field. An additional entry may be required in this file to describe the flash part which is on the target system. Examine the target board before adding in the appropriate attribute values. The supplied file is already populated with default values for SPI devices used with CRBs.

5.6 Programming the Flash Device

Once the Intel ME is programmed, it runs at all times. Intel ME is capable of writing to the flash device at any time, even when the management mode is set to none and it may appear that no writing would occur.

Note: Programming the flash device while Intel ME is running may cause the flash device to become corrupted. Intel ME SPI accessing should be stopped for any flash accessing before programming the full flash device. This should be done to force Intel ME into recovery mode.

5.7 Usage

Windows, DOS and EFI versions of the spsFPT can run with command line options.

To view all of the supported commands: Run the application with the `-?` option.

The commands in DOS, Windows and EFI versions have the same syntax. The command line syntax for **spsFPT.exe**, **spsFPTW.exe and spsFPT.efi** is:

```
spsFPT.exe [-H|?] [-VER] [-EXP] [-VERBOSE] [-Y] [-P] [-LIST]
[-I] [-F] [-ERASE] [-VERIFY] [-D] [-DESC] [-BIOS] [-ME]
[-GBE] [-PDR] [-DER] [-SAVEMAC] [-C] [-B] [-E] [-REWRITE] [-HARDERASE] [-
ADDRESS|A] [-LENGTH|L] [-PAGE]
```

Table 5-2. Command Line Options for spsFPT.exe and spsFPTW.exe

Option	Description
-H|?:	Displays help screen.
-VER:	Shows the version of the tools.
-EXP:	Displays example usage of the tool.
-VERBOSE <file>	Displays the tool's debug information or stores it in a log file.
-Y:	Prevents the tool from prompting when a warning occurs and assumes YES as the default answer.
-P:	Specifies a flash part definition file to use.
-LIST:	Supported Flash Parts. Displays all supported flash parts. This option reads the contents of the flash parts definition file and displays the contents on the screen.
-I:	Info. Displays information about the image currently used in the flash.
-F <file> [NoVerify]:	Flash. Programs a binary file into an SPI flash. You must specify the binary file to be flashed. spsFPT reads the binary, erases the flash, and then programs the binary into the flash. After a successful flash, spsFPT verifies that the SPI flash matches the provided image. Without specify the length with –L option, spsFPT will use the total SPI size instead of an image size. NoVerify flag prevents the tool from verifying the flash content after programming it.
-ERASE:	Block Erase. Erases all the blocks in a flash. If a block is already empty then the tool skips it. This option does not use the chip erase command but instead erases the SPI flash block by block. This option can be used with a specific region argument to erase that region. This option cannot be used with the `-f`, `-b`, `-c`, `-d` or `-verify` options.
-VERIFY <file>:	Compare a content of a binary file with the content of the flash.
-D <file> :	Dump. Reads the SPI flash and dumps the flash contents to a file or to the screen using the STDOUT option. The flash device must be written in 4 KB sections. The total size of the flash device must also be in increments of 4 KB.
-DESC:	Read/Write/Verify Descriptor region. Specifies that the Descriptor region is to be read, written, or verified. The start address is the beginning of the region.
-BIOS:	Read/Write/Verify BIOS region. Specifies that the Intel ME region is to be read, written, or verified. The start address is the beginning of the region.
-ME:	Read/Write/Verify Intel ME region. Specifies that the Intel ME region is to be read, written, or verified. The start address is the beginning of the region.

Option	Description
-GBE:	Read/Write/Verify GbE region. Specifies that the GbE region is to be read, written, or verified. The start address is the beginning of the region.
-PDR:	Read/Write/Verify PDR region. Specifies that the PDR region is to be read, written, or verified. The start address is the beginning of the region.
-DER:	Read/Write/Verify DER region. Specifies that the DER region is to be read, written, or verified. The start address is the beginning of the region.
-SAVEMAC:	Saves the GbE MAC when GbE is being reflashed.
-C:	Chip erase. Erases the contents of SPI flash device(s). This function does NOT erase block by block.
-B:	Blank Check. Checks whether the SPI flash is erased. If the SPI flash is not empty, the application halts as soon as contents are detected. The tool reports the address at which data was found.
-REWRITE:	Rewrites the SPI flash with data from a file even if the content of the file is identical to the content of the flash.
-HARDERSE:	Block Erase. Erases all the blocks in a flash without checking firstly if each block is empty. This option does not use the chip erase command but instead erases the SPI flash block by block. This option can be used with a specific region argument to erase that region. This option cannot be used with the `-f`, `-b`, `-c`, `-d` or `-verify` options.
-ADDRESS | A <address>:	Specifies the address from which spsFPT will start reading/writing/verifying.
-LENGTH | L <length>	Specify the length of data which will be read/written/verified.
-page	Pauses at screen / page / window boundaries. Hit any key to continue.

Note: Please be aware that –rewrite option used without any region option will first try to erase entire flash from the beginning.

Table 5-3. Intel Recommended Access Settings

	Intel® ME	GbE	BIOS
Read	0b 0000 1101 = 0x0d	0b 0000 1000 = 0x08	0b 0000 0011 = 0x0B
Write	0b 0000 1100 = 0x0c	0b 0000 1000 = 0x08	0b 0000 0010 = 0x0A

5.8 fparts.txt File

The **fparts.txt** file contains a list of all flash devices that are supported by spsFPT. The flash devices listed in this file must contain a 4 KB erase block size. If the flash device is not listed, you receive the following error:

```
Intel (R) Flash Programming Tool for Server Platform Services.
Version:  X.X.XX.XX
Copyright (c) 2007 - 2014, Intel Corporation. All rights reserved.

Number of LPC Devices supported: 225
LPC Device Id: 8D40.
Platform: Intel(R) 9 Super SKU 0x8D40
Initializing SPI utilities
Reading HSFSTS register... Flash Descriptor: Valid

Region Limits as programmed into the SPI Registers
    FREG0 -  DESC Region:Base Address: 0x000000   Limit :  0x000FFF
    FREG1 -  BIOS Region:Base Address: 0x800000   Limit :  0xFFFFFF
    FREG2 -  ME   Region:Base Address: 0x013000   Limit :  0x7FFFFF
    FREG3 -  GbE  Region:Base Address: 0x001000   Limit :  0x002FFF
    FREG4 -  PDR  Region:Base Address: 0x003000   Limit :  0x012FFF
    FREG5 -  DER  Region:Base Address: 0x1FFF000   Limit :  0x000FFF
Address Limit 0x1000000    Maximum Memory 16384kB


    --- Flash Devices Found ---
```

Error 103: There are no supported SPI flash devices installed. Please check connectivity and orientation of SPI flash device. If the device is not located in **fparts.txt**, you are expected to provide information about the device, inserting the values into **fparts.txt** in same format as is used for the rest of the devices. Detailed information on how to derive the values in **fparts.txt** is found in the Intel® 6 Series Chipset SPI Programming Guide. The device must have a 4 KB erase sector and the total size of the SPI Flash device must be a multiple of
4 KB. The values are listed in columns in the following order:

- Display name
- Device ID (2 or 3 bytes)
- Device Size (in bits)
- Block Erase Size (in bytes - 256, 4K, 64K)
- Block Erase Command
- Write Granularity (1 or 64)
- Enable Write Status Register Command (1- True, 0- False) Chip Erase Command.

- Chip Erase Timeout (in milliseconds)

5.9 Examples

The following examples illustrate the usage of the DOS version of the tool (**spsFPT.exe)**. The Windows version of the tool (**spsFPTW.exe**) and EFI version of the tool (**spsFPT.efi**) behave in the same manner apart from running in a Windows/EFI environment.

5.9.1 Example 1 – Flash SPI Flash Device with Binary File

```
C:\ spsFPT.exe –f spi.bin
```

This command writes the data in the **spi.bin** file into a whole SPI flash from address 0x00.

5.9.2 Example 2 – Program a Specific Region

```
spsFPT.exe –f ME.rom –ME
--------------------------------------------
Intel (R) Flash Programming Tool for Server Platform Services.
Version:  1.1.25.43
Copyright (c) 2007 - 2014, Intel Corporation. All rights reserved.

Platform: Intel(R) 9 Super SKU 0x8D40
Reading HSFSTS register... Flash Descriptor: Valid

    --- Flash Devices Found ---
    W25Q128BV    ID:0xEF4018    Size: 16384KB (131072Kb)


- Reading Flash [0x800000] 8116KB of 8116KB - 100% complete.
- Erasing Flash Block [0x019000] - 100% complete.
- Programming Flash [0x019000]   24KB of   24KB - 100% complete.
- Erasing Flash Block [0x025000] - 100% complete.
- Programming Flash [0x025000]    8KB of    8KB - 100% complete.
- Erasing Flash Block [0x04D000] - 100% complete.
- Programming Flash [0x04D000]  100KB of  100KB - 100% complete.
- Erasing Flash Block [0x055000] - 100% complete.
- Programming Flash [0x055000]    4KB of    4KB - 100% complete.
- Erasing Flash Block [0x062000] - 100% complete.
- Programming Flash [0x062000]   40KB of   40KB - 100% complete.
- Erasing Flash Block [0x065000] - 100% complete.
- Programming Flash [0x065000]    4KB of    4KB - 100% complete.
- Erasing Flash Block [0x069000] - 100% complete.
- Programming Flash [0x069000]    4KB of    4KB - 100% complete.
- Erasing Flash Block [0x06D000] - 100% complete.
- Programming Flash [0x06D000]    4KB of    4KB - 100% complete.
- Erasing Flash Block [0x071000] - 100% complete.
- Programming Flash [0x071000]    4KB of    4KB - 100% complete.
- Erasing Flash Block [0x075000] - 100% complete.
- Programming Flash [0x075000]    4KB of    4KB - 100% complete.
- Erasing Flash Block [0x079000] - 100% complete.
- Programming Flash [0x079000]    4KB of    4KB - 100% complete.
- Erasing Flash Block [0x07D000] - 100% complete.
- Programming Flash [0x07D000]    4KB of    4KB - 100% complete.
- Erasing Flash Block [0x081000] - 100% complete.
- Programming Flash [0x081000]    4KB of    4KB - 100% complete.
- Erasing Flash Block [0x085000] - 100% complete.
- Programming Flash [0x085000]    4KB of    4KB - 100% complete.
- Erasing Flash Block [0x089000] - 100% complete.
- Programming Flash [0x089000]    4KB of    4KB - 100% complete.
- Erasing Flash Block [0x08D000] - 100% complete.
- Programming Flash [0x08D000]    4KB of    4KB - 100% complete.
- Erasing Flash Block [0x091000] - 100% complete.
- Programming Flash [0x091000]    4KB of    4KB - 100% complete.
- Erasing Flash Block [0x1EB000] - 100% complete.
- Programming Flash [0x1EB000] 1372KB of 1372KB - 100% complete.
- Erasing Flash Block [0x3EB000] - 100% complete.
- Programming Flash [0x3EB000] 1372KB of 1372KB - 100% complete.
- Erasing Flash Block [0x497000] - 100% complete.
- Programming Flash [0x497000]   12KB of   12KB - 100% complete.
- Verifying Flash [0x800000] 8116KB of 8116KB - 100% complete.
RESULT: The data is identical.
```

spsFPT Operation Passed - This command writes the data in **ME.bin** into the Intel ME region of the SPI flash and verifies that the operation ran successfully.

5.9.3 Example 3 – Display SPI Information

```
spsFPTW.exe –I
--------------------------------------------
Intel (R) Flash Programming Tool for Server Platform Services.
```

```
Version:  X.X.XX.XX
Copyright (c) 2007 - 2014, Intel Corporation. All rights reserved.

Platform: Intel(R) 9 Super SKU 0x8D40
Reading HSFSTS register... Flash Descriptor: Valid

    --- Flash Devices Found ---
    W25Q128BV    ID:0xEF4018    Size: 16384KB (131072Kb)

    --- Flash Image Information --
    Signature: VALID
    Number of Flash Components: 1
        Component 1 - 16384KB (131072Kb)
    Regions:
        Descriptor - Base: 0x000000, Limit: 0x000FFF
        BIOS       - Base: 0x800000, Limit: 0xFFFFFF
        ME         - Base: 0x013000, Limit: 0x7FFFFF
        GbE        - Base: 0x001000, Limit: 0x002FFF
        PDR        - Base: 0x003000, Limit: 0x012FFF
        DER        - Not present
    Master Region Access:
        CPU/BIOS - ID: 0x0000, Read: 0x1B, Write: 0x3A
        ME       - ID: 0x0000, Read: 0x25, Write: 0x04
        GbE      - ID: 0x0118, Read: 0x09, Write: 0x08

Total Accessible SPI Memory: 16384KB, Total Installed SPI Memory : 16384KB
```

spsFPT Operation Passed - This command displays information about the flash devices present in the computer. The base address refers to the start location of that region and the limit address refers to the end of the region. If the flash device is not specified in **fparts.txt**, spsFPT returns the error message "There is no supported SPI flash device installed."

6 spsManuf and spsManufWin

spsManuf validates Intel ME functionality (verifies that all its components have been assembled together correctly) on the manufacturing line.

The Windows version of spsManuf requires administrator privilege to run under windows OS. You need to explicitly click on the context menu in Windows “Run as Administrator“ under Windows Server 2008 R2 64 bit.

spsManuf does not check for LAN functionality. The tool assumes that all Intel ME components on the test board have been validated by their respective vendors. The tool verifies that these components have been assembled together correctly.

6.1 How to use spsManuf

Functionality of spsManuf consist of two test groups:

- default tests
- optional tests

Default tests are run every time user starts spsManuf and there is no possibility to turn it off.

By configuration file user can specify optional tests which should be executed. It is necessary because spsManuf have to know what value is correct for particular test in user opinion.

VSCCOMMN.bin file is required to verify the VSCC entry on the platform. You need to have this file at the location you run spsManuf, otherwise spsManuf will report error.

6.2 Tests Description

Table 6-1 List of Default and Optional tests

Test Group	Subtest	Runs when	Purpose
Default	Intel ME Hardware and Firmware Status	Always	Confirms that Intel ME HW and FW are alive and operating in Normal Mode.
	Intel ME VSCC		Confirms that VSCC in Intel ME include the Intel-recommended value for the installed SPI device(s).
Optional	Runtime Image FW Version	If subtest is not commented out in config file and spsManuf.cfg exists or –F<file> option was set.	Compares FW version of the running OpImage.
	Backup Image FW Version		Compares FW version of the backup OpImage.
	Recovery Image FW Version		Compares FW version of the Recovery Image.
	Intel Node Manager		Checks if Intel Node Manager is enabled or disabled
	Factory Default Configuration		Confirms that Factory Default Configuration matches the intended design.
	EOP Status		Checks End-Of-Post reception by Intel ME Firmware.
	Dengate		Checks if Dengate is enabled or disabled
	MCTP Proxy		Checks if MCTP Proxy is enabled or disabled
	MCTP Infrastructure		Checks if MCTP Infrastructure is enabled or disabled
	CUPS		Checks if CUPS is enabled or disabled
	Thermal Reporting		Checks if Thermal Reporting is enabled or disabled
	PTU Payload		Checks if PTU Payload is enabled or disabled
	Hotham		Checks if Hotham is enabled or disabled
	PECI Proxy		Checks if PECI Proxy is enabled or disabled

Test Group	Subtest	Runs when	Purpose
	MIC Proxy		Checks if MIC Proxy is enabled or disabled
	PM Bus Proxy		Checks if PM Bus Proxy is enabled or disabled
	Turbo State Limiting		Checks if Turbo State Limiting is enabled or disabled
	Intel ME Integrity Check		Performs checksum-style integrity check of Intel ME firmware image
	PTU Option ROM Version Check		Performs comparison between expected and obtained PTU ROM version

6.3 Usage

The DOS, EFI and Linux version of the tool can be operated using the same syntax as the Windows version. The Windows version of the tool can be executed by:

```
spsManufWin64.exe [-EXP] [-H|?] [-VER] [-F] [-CFGGEN] [-VERBOSE] [-PAGE]
[-PCHBUSID]
```

It is possible to use "/" instead of "–" in command line.

Table 6-2. Command Line Options for spsManuf

Option	Description
No option	Runs all hardcoded default subtests. In addition, if a file named spsManuf.cfg exists in the spsManuf directory, all optional subtests found in it will run.
–F <file>	Runs all hardcoded default subtests. In addition, this option will run several checks according to configuration file. The checks can be configured by customer to select which test items he is expecting to run and what is the proper value. Sub option "file" is mandatory.
–CFGGEN <-F [file]>	This option generates default spsManuf.cfg configuration file with complete help and comments included. User can specify name of generated file by <– F[file]> sub option.
–VERBOSE <file>	Displays the tool's debug information or stores it in a log file.
–PAGE	When more than one full screen (80 x 25 under DOS, various under Windows depending on console windows setting for the visible windows size) of information is displayed, this option allows user to pause the output and press any key before continuing on to the next screen.
–VER	Show the version of the tool.
–H or –?	Display help screen.
–EXP	Show the examples on how to use the tool.

Option	Description
–PCHBUSID <pchBusId>	Select PCH by PCI Bus Id Note: This option applies only for multi-PCH system. Without this option by default PCI Bus Id is 0. To select PCH connected to another PCI bus you need to know to which PCI bus Id the PCH is attached. **Note**: Tool doesn't provide scan functionality.

6.4 spsManuf.cfg File

Configuration file (by default: spsManuf.cfg) includes all the test's configurations for spsManuf –F check. It needs to be at the same folder as you run spsManuf from.
If there is no configuration file existing in that folder you can generate it by –CFGGEN <-F[file]> command.

Here is an example of configuration file:

```
// If one of these check fails, by default spsManuf will report error and
// continue on to the next check. If a user doesn't wish to continue
// when an error is found, ErrAction field can be used. Please see
// the examples here for detailed explanation:
//
//  SubTestName="Runtime Image FW Version", ReqVal="1.2.3.4", ErrAction="ErrorStop"
//
// If the above test fails, spsManuf will report error and stop. There
// are total of three different error actions user can choose from:
//
// ErrorContinue - report error and continue on to the next check
// ErrorStop - report error and stop any check after the current one
// WarnContinue - report warning and continue on to the next check
//
// To add comment or take out a specific test, leave // at the start
// of a line. This file is processed by spsManuf line by line as text
// file. Duplication of the same sub-tests are allowed, but spsManuf
// will always perform the first test to the last test from the file.

// All string comparisons given in this file are case insensitive
// compare. There might be multiple field name/value pairs in one
// entry, but each field needs to be specified in the following
// format where <field name> can be replaced by SubTestName, ReqVal
// or ErrAction, <field value> can be replaced by any string including
// dash and/or spaces surrounded by double quotation marks.
// No line Wrapping is supported:
//
//    <field name>="<field value>", such as ReqVal="<value>"
// Some default checks run every time user starts spsManuf
// There is no possibility to turn it off
//
// SubTestName="ME Hardware and Firmware Status"
// SubTestName="ME VSCC"

////////////////////////////////////////////////////////////////////////////////
// The following Configuration Check requires a user to enter an expected
// value after ReqVal=
////////////////////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////////////////////
// Runtime/Recovery Image FW version is a string as
//"<major ver>.<minor ver>.<hotfix ver>.<build num>"
////////////////////////////////////////////////////////////////////////////////

// SubTestName="Runtime Image FW version", ReqVal=
// SubTestName="Backup Image FW version", ReqVal=
// SubTestName="Recovery Image FW version", ReqVal=

/////////////////////////////////////////////////////////////////////////////////
// Factory Default Configuration is a string as
//"XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
XX XX XX XX XX"
// where X is a hex value
/////////////////////////////////////////////////////////////////////////////////

// SubTestName="Factory Default Configuration", ReqVal=

/////////////////////////////////////////////////////////////////////////////////
// ME Integrity Check is a string as
//"XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
XX XX XX XX XX"
// where X is a hex value
// NOTE: Execution time may take up to 3 seconds
/////////////////////////////////////////////////////////////////////////////////

// SubTestName="ME Integrity Check", ReqVal=

/////////////////////////////////////////////////////////////////////////////////
// ReqVal for each feature below is a string as "enabled" or "disabled"
/////////////////////////////////////////////////////////////////////////////////

// SubTestName="Node Manager", ReqVal=
// SubTestName="Dengate", ReqVal=
// SubTestName="MCTP Proxy", ReqVal=
// SubTestName="MCTP Infrastructure", ReqVal=
// SubTestName="CUPS", ReqVal=
// SubTestName="Thermal Reporting", ReqVal=
// SubTestName="PTU Payload", ReqVal=
// SubTestName="Hotham", ReqVal=
// SubTestName="PECI Proxy", ReqVal=
// SubTestName="MIC Proxy", ReqVal=
// SubTestName="PM Bus Proxy", ReqVal=
// SubTestName="Turbo State Limiting", ReqVal=

/////////////////////////////////////////////////////////////////////////////////
// PTU Option ROM version is a string of format X.Y
// where X is major and Y is minor version decimal value
/////////////////////////////////////////////////////////////////////////////////

// SubTestName="PTU Option ROM Version Check", ReqVal=

/////////////////////////////////////////////////////////////////////////////////
// Tests without ReqVal needed
/////////////////////////////////////////////////////////////////////////////////

// SubTestName="EOP Status"
```

Please note that lines start with // are for comment, and they also used for the purpose to inform users the available test group names and specific checks names included in each test that spsManuf recognizes. To select which test items to run, user can create a line begins with `SubTestName=` with a specific sub test name. Here are some additional examples that explain how to use this feature:

User wants to run Intel ME FW version check and a valid Intel ME FW version should be equal to string 1.2.3.4:

```
SubTestName="Runtime Image FW version", ReqVal=”1.2.3.4”
```

6.5 Output/Result

There are 3 possible results displayed in verbose mode at the optional tests checking:

- Pass – meaning all tests passed
- Pass with warning – meaning only tests with error action set as “`WarnContinue`” failed.
- Fail - meaning any error occurs in the test as customer defined at error items.

6.6 Examples

```
>spsManufWin64.exe -f spsManufAllPositive.cfg
```

```
spsManuf Test Passed
```

```
>spsManufWin64.exe -f spsManufAllPositive.cfg -verbose
```

```
Intel(R) spsManuf Version: 1.1. 30.25
Copyright(C) 2005 - 2014, Intel Corporation. All rights reserved.

Number of LPC Devices supported: 226
LPC Device Id: 8D44.
Platform: Intel(R) 9 -G SKU 0x8D44

Checking ME Hardware and Firmware Status...passed
vsccommn.bin was created on 18:47:14 06/03/2013 GMT
SPI Flash ID #1 ME VSCC value is 0x2025, device supports SFDP capability
SPI Flash ID #1 (ID: 0xEF4018) SFDP BES is 4kB, ME VSCC comparison to
Intel recommended value not needed. Checking ME VSCC status...passed
Checking default spsManuf tests...passed

./spsManufAllPositive.cfg is found with 19 valid test entries

Checking Runtime Image FW Version status...passed

Checking Backup Image FW Version status...passed

Checking Recovery Image FW Version status...passed

Checking Factory Default Configuration status...passed

Checking ME Integrity status...passed

Checking End-Of-Post status...passed

Checking Node Manager status...passed

Checking Dengate status...passed

Checking MCTP Proxy status...passed

Checking MCTP Infrastructure status...passed

Checking CUPS status...passed

Checking Thermal Reporting status...passed

Checking PTU Payload status...passed
Checking Hotham status...passed

Checking PECI Proxy status...passed

Checking MIC Proxy status...passed

Checking PM Bus Proxy status...passed

Checking Turbo State Limiting status...passed

Checking PTU OROM version status...passed

Number of optional tests executed: 19
Passed: 19
Failed: 0
Checking optional spsManuf tests...passed

spsManuf Test Passed
```

7 spsInfo and spsInfoWin

spsInfoWin and spsInfo provide a simple test to check whether the Intel ME FW is alive or not. Both tools perform the same test, query the Intel ME FW

The Windows version of spsInfo (spsInfoWin) requires administrator privileges to run under Windows OS. You must use the Run as Administrator option to open the CLI in Windows* Vista 64/32 bit and Windows* 7 64/32 bit.

7.1 Usage

The DOS, EFI and Linux version of the tool can be operated using the same syntax as the Windows version. The Windows version of the tool can be executed by:

```
spsInfoWin64.exe [-EXP] [-H|?] [-FWSTS] [-VER] [-VERBOSE] [-PAGE] [-
PCHBUSID]
```

It is possible to use "/" instead of "–" in command line.

Table 7-1. Command Line Options for spsInfo

Option	Description
No option	Display all information about Intel ME FW.
–VERBOSE <file>	Display the debug information of the tool or store it in a log file.
–PAGE	When more than one full screen (80 x 25 under DOS, various under Windows depending on console windows setting for the visible windows size) of information is displayed, this option allows user to pause the output and press any key before continuing on to the next screen.
–VER	Show the version of the tool.
–H or –?	Display help screen.
–EXP	Show the examples on how to use the tool.
–PCHBUSID <pchBusId>	Select PCH by PCI Bus Id Note: This option applies only for multi-PCH system. Without this option by default PCI Bus Id is 0. To select PCH connected to another PCI bus you need to know to which PCI bus Id the PCH is attached. ***Note***: Tool doesn't provide scan functionality.
–FWSTS 0x... [0x...]	Decode given hex like ME Firmware status register. It is acceptable to type only first register to decode, second one is optional.

7.2 Examples

```
>spsInfoWin64.exe
```

```
Intel(R) spsInfo Version: 1.1.30.4
```

```
Copyright(C) 2005 - 2014, Intel Corporation. All rights reserved.


FW Status Register 1: 0x700F0345
  CurrentState (3:0):                      Normal (5)
  ManufacturingMode (4):                   Disabled (0)
  FlashPartition (5):                      Valid (0)
  OperationalState (8:6):                  M0 with no UMA (5)
  InitComplete (9):                        Complete (1)
  BUPLoadState (10):                       Success (0)
  FwUpdateInProgress (11):                 No (0)
  ErrorCode (15:12):                       No Error (0)
  ModeOfOperation (19:16):                 Server Platform Services (15)

FW Status Register 2: 0xB8000000
  BiosControlBootMode (0):                 Power (0)
  MfsFailure (6):                          No Mfs failure (0)
  WarmReset (7):                           No warm reset request (0)
  TargetImageBoot (12):                    Success (0)
  Heartbeat (15:13):                       0
  ExtendedStatusData (27:16):              2048
  Phase (30:28):                           POLICY (3)
  EndOfPOST (31):                          Received (1)

Server Platform Service firmware is detected on the system.
SPS Image FW version: 3.0.4.106 (Recovery), 3.0.4.106 (Operational)
Feature list:
  Interface Version:                       1.0
  Node Manager:                            Enabled (1)
  Platform Instrumentation:                Enabled (1)
  Silicon Enabling:                        Enabled (1)
  Datacenter Manager:                      Disabled (0)
  IDE Redirection:                         Disabled (0)
  Serial over LAN:                         Disabled (0)
  Networking:                              Disabled (0)
  KVM:                                     Disabled (0)
  Dengate:                                 Disabled (0)
  MCTP Proxy:                              Disabled (0)
  MCTP Infrastructure:                     Enabled (1)
  CUPS:                                    Enabled (1)
  Thermal Reporting:                       Enabled (1)
  PTU Payload:                             Enabled (1)
  Hotham:                                  Enabled (1)
  PECI Proxy:                              Enabled (1)
  MIC Proxy:                               Enabled (1)
  PM Bus Proxy:                            Enabled (1)
  Turbo State Limiting:                    Enabled (1)

PTU Option ROM version: 0.3
```

```
>spsInfoWin64.exe -FWSTS 0x001F0347 0xB9006101
```

```
Intel(R) spsInfo Version: 1.1.30.4
Copyright(C) 2005 - 2014, Intel Corporation. All rights reserved.

FW Status Register 1: 0x001F0347
  CurrentState (3:0):                      State transition (7)
  ManufacturingMode (4):                   Disabled (0)
  FlashPartition (5):                      Valid (0)
  OperationalState (8:6):                  M0 with no UMA (5)
  InitComplete (9):                      Complete (1)
  BUPLoadState (10):                     Success (0)
  FwUpdateInProgress (11):               No (0)
  ErrorCode (15:12):                     No Error (0)
  ModeOfOperation (19:16):               Server Platform Services (15)

FW Status Register 2: 0xB9006101
  BiosControlBootMode (0):               Performance (1)
  MfsFailure (6):                        No Mfs failure (0)
  WarmReset (7):                         No warm reset request (0)
  TargetImageBoot (12):                  Success (0)
  Heartbeat (15:13):                     3
  ExtendedStatusData (27:16):            2304
  Phase (30:28):                         POLICY (3)
  EndOfPOST (31):                        Received (1)
```

>spsInfoWin64.exe -FWSTS 0x001F0345

```
Intel(R) spsInfo Version: 1.1.30.4
Copyright(C) 2005 - 2014, Intel Corporation. All rights reserved.

FW Status Register 1: 0x001F0345
  CurrentState (3:0):                    Normal (5)
  ManufacturingMode (4):                 Disabled (0)
  FlashPartition (5):                    Valid (0)
  OperationalState (8:6):                M0 with no UMA (5)
  InitComplete (9):                      Complete (1)
  BUPLoadState (10):                     Success (0)
  FwUpdateInProgress (11):               No (0)
  ErrorCode (15:12):                     No Error (0)
  ModeOfOperation (19:16):               Server Platform Services (15)
```

A *Tool Detail Error Code*

A.1 Common Error Code for all Tools

Error Code	Error Message	Response
0	Success	
1	Memory allocation error occurred	Make sure there is enough memory in the system
2	Invalid descriptor region	Check descriptor region
3	Region does not exist	Check region to be programmed
4	Failure. Unexpected error occurred	Contact Intel
5	Invalid data for Read ID command	Contact Intel
6	Error occurred while communicating with SPI device	Check SPI device
7	Hardware sequencing failed. Make sure that you have access to target flash area	Check descriptor region access settings
8	Software sequencing failed. Make sure that you have access to target flash area	Check descriptor region access settings
9	Unrecognized value in the HSFSTS register	Unrecognized value in the HSFSTS register
10	Hardware Timeout occurred in SPI device	Hardware Timeout occurred in SPI device
11	AEL is not equal to zero	AEL is not equal to zero
12	FCERR is not equal to zero	FCERR is not equal to zero
25	The host CPU does not have write access to the target flash area. To enable write access for this operation you must modify the descriptor settings to give host access to this region.	Check descriptor region access settings
26	The host CPU does not have read access to the target flash area. To enable read access for this operation you must modify the descriptor settings to give host access to this region.	Check descriptor region access settings
27	The host CPU does not have erase access to the target flash area. To enable erase access for this operation you must modify the descriptor settings to give host access to this region.	Check descriptor region access settings
28	Protected Range Registers are currently set by BIOS, preventing flash access. Contact the target system BIOS vendor for an option to disable Protected Range Registers.	Assert Flash Descriptor Override Strap (GPIO33) to Low, Power Cycle, and Retry. If Protected Range Registers (memory location: SPIBAR + 74h -> 8Fh) are still set, contact the target BIOS vendor.
50	General Erase failure	Attempt the command again. If it fails again, contact Intel.

Error Code	Error Message	Response
51	An attempt was made to read beyond the end of flash memory	Check address
52	An attempt was made to write beyond the end of flash memory	Check address
53	An attempt was made to erase beyond the end of flash memory	Check address
54	The address <address> of the block to erase is not aligned correctly	Check address
55	Internal Error	Contact Intel
56	The supplied zero-based index of the SPI Device is out of range.	The supplied zero-based index of the SPI Device is out of range.
57	AEL or FCERR is not equal to zero for Software Sequencing	AEL or FCERR is not equal to zero for Software Sequencing
75	Common VSCC file not found	Check file location
76	Access was denied opening the file	Check file location
77	An unknown error occurred while opening the file	Verify the file is not corrupt
78	Failed to allocate memory for the flash part definition file	Check system memory Verify the file is not corrupt
79	Failed to read the entire file into memory	Check system memory Verify the file is not corrupt
80	Parsing of file failed	Check system memory Verify the file is not corrupt
100	The SPI Flash configuration registers are write protected by the Flash Configuration Lock-Down bit (FLOCKDN). Cannot access the SPI flash. Contact your BIOS vendor to unlock this bit or enable hardware sequencing in descriptor mode.	Check with BIOS vendor or SPI programming Guide
101	No SPI flash device could be identified. Please verify if Fparts.txt has support for this part	Verify **Fparts.txt** contains device supported.
102	Failed to read the device ID from the SPI flash part	Verify **Fparts.txt** has correct values
103	There are no supported SPI flash devices installed. Check connectivity and orientation of SPI flash device	Verify **Fparts.txt** has correct values. Check SPI Device
104	The two SPI flash devices do not have compatible command sets	Verify both SPI devices on the system are compatible
105	An error occurred while writing to the write status register of the SPI flash device. This program will not be able to modify the SPI flash	Check SPI Device
8196	HECI message receive buffer memory allocation failed	
8193	Intel® ME Interface: Cannot locate Intel® ME device driver	

Error Code	Error Message	Response
8199	Could not issue %s command message Where %s can be the following: Get FWU Version Get FWU Info Get FWU Feature State Intel ME Kernel Test	Contact Intel
8203	Unexpected result in %s command response Where %s can be the following: Get FWU Version Get FWU Info Get FWU Feature State Intel ME Kernel Test	Contact Intel
8204	Intel ME Interface: Unsupported message type	
8213	Requesting HECI receive buffer size is too small	
9489	Couldn't receive Intel(R) MEI get FW features message response	
9507	Fail to load driver (PCI access for Windows).Tool needs to run with an administrator privilege account	

A.2 spsManuf Errors

Error Codes	Error Messages
9251	Failed to create verbose log file %s Where %s is the log file name user specified
9257	Internal error (Could not determine FW features information)
9267	Failed to establish communication with SPI flash interface
9268	Failed to load vsccommn.bin
9269	Zero flash device found for VSCC check
9270	Failed to load driver (PCI access for Windows) Tool needs to run with an administrator privilege account.
9271	Flash ID 0x%06X Intel® ME VSCC mismatch Programmed value of 0x%X doesn't match the recommended value of 0x%X See PCH SPI programming Guide for more details
9272	Flash ID 0x%06X Intel® ME VSCC value didn't find recommended value
9276	Failed to read FW Status Register value 0x%X
9278	Cannot locate hardware platform identification. This program cannot be run on the current platform.

Error Codes	Error Messages
	Unknown or unsupported hardware platform. or A %s hardware platform is detected. This program cannot be run on the current platform. Unknown or unsupported hardware platform. Where %s is the official name of the hardware platform
9279	SPI flash Intel® ME region is not locked
9282	BIOS has granted Intel® Gbe and/or Intel® ME access to its region
9283	Region access permissions don't match Intel recommended values
9284	Read firmware flash master region permission failure
9296	Intel® ME Manuf Test Failed Use <VERBOSE> option for more details
9299	Single flash part found, Flash Partition Boundary Address must be zero
9300	Flash Partition Boundary Address should be in between flash parts
9301	The two flash parts on this platform require different BIOS VSCC values
9303	Memory allocation failed for checking variable "<Variable Name>"
9304	Variable "<Variable Name>" mismatch, actual value is - <Variable Value>
9305	Intel ME firmware version mismatch, actual value is - <Version String> Gbe version mismatch, actual value is - <Version String> BIOS version mismatch, actual value is - <Version String>
9308	Flash Descriptor Override Strap is enabled
9314	Intel ME test result reports error(s)
9317	No valid OEM ICC data programmed
9458	Communication error between application and Intel(R) ME module (FW Update client)
9459	Internal error (Could not determine FW features information)
9487	Couldn't issue Intel(R) MEI get FW version message (0x%X)
9488	Couldn't receive Intel(R) MEI get FW version message response (0x%X)
9489	Couldn't receive Intel(R) MEI get FW features message response
9500	spsManuf Test Failed
9501	Unsupported command line option(s)
9502	Unknown or unsupported hardware platform
9503	Configuration file %s is missing
9504	spsManuf config file generation failed
9505	Intel(R) Fail to read FW Status Register value 0x%X

Error Codes	Error Messages
9506	Fail to create verbose log file %s
9507	Fail to load driver (PCI access for Windows).Tool needs to run with an administrator priviledge account.
9508	Configuration file syntax corrupted
9510	Intel(R) ME FW invalid status
9511	Intel(R) Bad checksum of Flash Partition Table or broken factory defaults
9512	Intel(R) Failure in starting desired ME FW image
9520	Failure getting SPI address and/or loading VSCC file
9521	Single flash part found, Flash Partition Boundary Address must be zero
9522	Flash Partition Boundary Address should be on the boundary between flash parts
9523	The two flash parts on this platform require different BIOS VSCC values
9524	Access flash device failure
9525	Fail to establish a communication with SPI flash interface
9526	Fail to load vsccommn.bin
9527	Flash ID 0x%06X Intel(R) %s VSCC value mismatch
9528	No recommended %s VSSCC value found for Flash ID 0x%06X
9530	ME FW version is incorrect
9531	ME recovery version is incorrect
9532	Backup ME FW version is incorrect
9533	No backup image or single image configuration
9540	Intel(R) ME-BIOS Interface Versions mismatch
9541	Intel(R) Node Manager error - fail or disabled
9542	Intel(R) Datacenter Manager - fail or unsupported
9543	Intel(R) Node Manager status mismatch
9544	Intel(R) Datacenter Manager status mismatch
9550	Intel(R) ME internal communication error (FW)
9551	Error: %s Factory Default Configuration status failed
9552	Intel(R) ME Integrity Check mismatch, actual value is - <Version String>
9553	PTU OROM version mismatch, actual value is - <Version String>
9554	No PTU Option ROM detected in DER region.
9555	Invalid PTU OROM version length, actual value is - <Version String>
9560	Error: %s BMC Connection status failed
9561	Intel(R) Incorrect ME Address for BMC Connection test

Error Codes	Error Messages
9562	Intel(R) BMC Connection test cannot be run under this configuration
9570	Intel(R) Read flash master region permission failure
9571	Intel(R) Incorrect format of expected access permission
9572	Intel(R) Incorrect Access Rights
9573	Intel(R) Correct vsccommn.bin file was not found
9580	Intel(R) Not existing or invalid region
9581	Intel(R) ME Region Definition address mismatch
9582	Intel(R) ME Region Definition length mismatch
9590	Error: %s End-Of-Post status failed
9591	Error: %s BIOS VSCC failed
9600	Intel(R) Dengate error - fail or unsupported
9601	Intel(R) Dengate status mismatch
9602	Intel(R) MCTP Proxy error - fail or unsupported
9603	Intel(R) MCTP Proxy status mismatch
9604	Intel(R) MCTP Infrastructure error - fail or unsupported
9605	Intel(R) MCTP Infrastructure status mismatch
9606	Intel(R) CUPS error - fail or unsupported
9607	Intel(R) CUPS status mismatch
9608	Intel(R) Thermal Reporting error - fail or unsupported
9609	Intel(R) Thermal Reporting status mismatch
9610	Intel(R) Cloud Scheduling Agent error - fail or unsupported
9611	Intel(R) Cloud Scheduling Agent status mismatch
9612	Intel(R) PTU Payload error - fail or unsupported
9613	Intel(R) PTU Payload status mismatch
9614	Intel(R) Hotham error - fail or unsupported
9615	Intel(R) Hotham status mismatch
9616	Intel(R) PECI Proxy error - fail or unsupported
9617	Intel(R) PECI Proxy status mismatch
9618	Intel(R) MIC Proxy error - fail or unsupported
9619	Intel(R) MIC Proxy status mismatch
9620	Intel(R) PM Bus Proxy error - fail or unsupported
9621	Intel(R) PM Bus Proxy status mismatch
9622	Intel(R) Turbo State Limiting error - fail or unsupported

Error Codes	Error Messages
9623	Intel(R) Turbo State Limiting status mismatch

A.3 spsInfo Errors

Error Code	Error Messages
0218	Internal error
0219	Version feature was not available
0220	Feature was not available
9253	Firmware did not return a valid value for iTPM full self-test
9258	TPM parsing response problem, response is less than minimum required
9259	TPM parsing response problem, bad tag value
9260	TPM parsing response problem, bad param size
9269	Zero flash device found for VSCC check
9271	Incorrect VSCC table entry mismatch
9272	No VSCC table entry found
9279	SPI flash Intel(R) ME region is not locked
9280	Intel(R) Gbe/ME has read or write access to BIOS region
9281	SPI flash descriptor region is not locked
9282	BIOS has granted Intel(R) Gbe and/or ME access to its region
9283	Region access permissions don't match Intel recommended values
9284	Tool fails to retrieve setting information
9289	Couldn't issue Intel(R) MEI get event log message
9290	Couldn't receive Intel(R) MEI get event log message response
9293	Create Context in Vista OS failed
9299	Single flash part found, Flash Partition Boundary Address isn't zero
9300	Flash Partition Boundary Address should be in between flash parts
9301	Two flash parts require different BIOS VSCC values
9303	Checking variable "%s" memory allocation failed
9304	Getting variable "%s" failed or not found
9306	System UUID status failed
9307	MAC address status failed
9308	Security Descriptor Override status failed
9310	ME Manufacturing Mode status failed

Error Code	Error Messages
9311	CF9GR locking status failed
9451	Communication error between application and Intel(R) AMT module (PTHI client)
9452	Communication error between application and Intel(R) ME module (iCLS client)
9455	Failed to read FW Status Register value 0x%X
9457	Failed to create verbose log file %s: Where %s is the log file name user specified
9458	Communication error between application and Intel® ME module (FW Update client)
9459	Internal error (Could not determine FW features information)
9460	Cannot locate hardware platform identification This program cannot be run on the current platform. Unknown or unsupported hardware platform Or A %s hardware platform is detected This program cannot be run on the current platform. Unknown or unsupported hardware platform Where %s is the official name of the hardware platform
9467	Cannot use zero as SPI Flash ID index number
9468	Couldn't find a matching SPI Flash ID
9469	Access to SPI Flash device(s) failed
9471	%s feature was not found
9472	Parameter invalid
9473	Parameter not equal
9487	Couldn't issue Intel(R) MEI get FW version message (0x%X)
9488	Couldn't receive Intel(R) MEI get FW version message response (0x%X)
9489	Couldn't receive Intel(R) MEI get FW features message response
9502	Unknown or unsupported hardware platform
9505	Intel(R) Fail to read FW Status Register value 0x%X
9506	Fail to create verbose log file %s

A.4 spsFPT Errors

Error Code	Error	Response
1	Memory allocation error occurred	Make sure there is enough memory in the system

Error Code	Error	Response
200	Invalid parameter value specified by the user. Use -? Option to see help.	Check the command line arguments supported by using the "-?"
201	spsFPT.exe cannot be run on the current platform. Please contact your vendor.	Contact your vendor.
202	Confirmation is not received from the user who performed the operation.	User input required
203	Flash is not blank. Data <data> found at address <address>.	Attempt to erase the device again
204	Data verify mismatch found at address <address>.	Reprogram the device
205	Failure. Unexpected error occurred	File a sighting
206		PDR region exists
207	Invalid parameter value specified by user. The option specified cannot be run on a platform with Intel (R) ME Ignition FW.	
210	The Intel ME Failed to reset.	
211	There was a communications error between spsFPT and the Intel ME	
212	The request to disable the Intel ME failed.	
215	The attempt to commit the FOVs has failed.	
216	The Close Manufacturing process failed.	
217	Setting Global Reset Failed	
240	Access was denied while opening the file <file>	Check the permissions for the file
241	Access was denied while creating the file <file>	Check the permissions for the file
242	An unknown error occurred while opening the file <file>	Verify the file is not corrupt
243	An unknown error occurred while creating <file>	Verify the file is not corrupt
244	<name> is not a valid file name.	Check the filename
245	<file> file not found	Check file location
246	Failed to read the entire file into memory. File: <file>	Check system memory. Verify the file is not corrupt
247	Failed to write the entire flash contents to file	Check system memory
248	<file> file already exists	Delete the file that already exists
249	The file is longer than the flash area to write	Check file size
250	The file is smaller than the flash area to write	Check file size
251	Length of image file extends past the flash area	Check file size
252	Image file <file> not found	Check filename
253	<file> file does not exist	Check filename

Error Code	Error	Response
254	Not able to open the file <file>	Check filename
255	Error occurred while reading the file <file>.	Check filename
256	Error occurred while writing to the file <file>	Check filename
280	Failed to disable write protection for the BIOS space!	Verify BIOS does not have write protection enabled
281	The Enable bit in the LPC RCBA register is not set. The value of this register cannot be used as the SPI BIOS base address	
282	Failed to get information about the installed flash devices	Check descriptor region access settings
283	Unable to write data to flash. Address <address>.	Check descriptor region access settings
284	Failed to load driver (PCI access for Windows). Tool needs to run with an administrator privilege account.	
320	General Read failure	Attempt the command again. If symptom persists file a sighting
321	The address <address> is outside the boundaries of flash area	Check address
360	Invalid Block Erase Size value in <file>.	Check **fparts.txt** or its equivalent file
361	Invalid Write Granularity value in <file>	Check **fparts.txt** or its equivalent file
362	Invalid Enable Write Status Register Command value in <file>	Check fparts.txt or its equivalent file
363	Invalid Chip Erase Timeout value in <file>	Check fparts.txt or its equivalent file
400	Flash descriptor does not have correct signature	Verify file is not corrupt
401	An error occurred reading the flash mapping data	Check SPI device
402	An error occurred while reading the flash components data	Check SPI device
403	An error occurred while reading the flash region base/limit data	Check SPI device
404	An error occurred while reading the flash master access data	Check SPI device
405	An error occurred while reading the flash descriptor signature	Check SPI device
406	System booted in Non-Descriptor mode, but the flash appears to contain a valid signature	Check SPI device
407	User-provided Chip Erase Timeout has been reached. If the timeout value was set incorrectly the chip erase may still occur.	Check fparts.txt or its equivalent file
440	Invalid Fixed Offset variable name	
441	Invalid Fixed Offset variable Id	
442	Param file is already opened.	

Error Code	Error	Response
444	Invalid name or Id of FOV.	
445	Invalid length of FOV value. Check FOV configuration file for correct length.	
446	Password does not match the criteria.	
447	Error occurred while reading FOV configuration file	
448	Invalid hash certificate file	
449	Valid PID/PPS/Password records are not found	
450	Invalid ME Manufacturing Mode Done value entered.	
451	Unable to get master base address from the descriptor	Check file integrity
452	Verification of End Of Manufacturing settings failed	
453	End Of Manufacturing Operation failure - Verification failure on ME Manufacturing Mode Done settings.	
454	The Global Lock Bit has already been set.	
455	End Of Manufacturing Operation failure - Verification failure on Intel ME Manuf counter.	
456	End Of Manufacturing Operation failure - Verification failure on Descriptor Lock set	
457	Parsing of file <file> failed	
459	There is a problem with the GbE binary which prevents saving the data	
480	The setup file header has an illegal UUID	
481	The setup file version is unsupported	Check setup file integrity
482	A record encountered that does not contain an entry with the Current MEBx password	
483	The given buffer length is invalid	Check buffer length value
484	The record chunk count cannot contain all of the setup file record data	Setup file number exceeded
485	The setup file header indicates that there are no valid records	Setup file has no valid records. Check setup file integrity
486	The given buffer is invalid	Check buffer value
487	A record entry with an invalid Module ID was encountered	Check record values. Check Setup file integrity
488	A record was encountered with an invalid record number	Check record values. Check Setup file integrity
489	The setup file header contains an invalid module ID list	Check record values. Check Setup file integrity
490	The setup file header contains an invalid byte count	Check record values. Check Setup file integrity

Error Code	Error	Response
491	The setup file record ID is not RECORD_IDENTIFIER_DATA_RECORD	Check record values. Check Setup file integrity
492	The list of data record entries is invalid	Check record values. Check Setup file integrity
493	The CurrentMEBx password is invalid	
494	The NewMEBx password is invalid	
495	The PID is invalid	
496	The PPS is invalid	
497	The PID checksum failed	
498	The PPS checksum failed	
499	The data record is missing a CurrentMEBx password entry	
500	The data record is missing a NewMEBx password entry	
501	The data record is missing a PID entry.	
502	The data record is missing a PPS entry.	
503	The file <file> has an invalid entry	
504	The requested index is invalid	
505	Failed to write to the given file	
506	Failed to read from the given file	
507	Failed to create random numbers	
508	The data record is missing a PKI DNS Suffix entry	
509	The data record is missing a Config Server FQDN entry	
510	The data record is missing a ZTC entry	
511	The data record is missing a Pre-Installed Certificate enabled entry	
512	The data record is missing a User defined certificate config entry	
513	The data record is missing a User defined certificate Add entry	
514	The data record is missing a SOL/IDER enable entry	
515	OEM Firmware Update Qualifier data missing in USB file	
516	The file "%s" has an invalid entry	
517	User selected to cancel the operation	
522	Failed getting variable "%s" value	
523	Failed comparing variable "%s" value	
525	Failed to perform ME Reset	

Error Code	Error	Response
1000	Invalid command line option(s)	
1001	Unsupported OS	
1002	Failed to retrieve Intel (R) ME FW Version	

B MESDC Commands

Recovery	**SiEn**	**Intel Node Manager**	**DM**	**DNM**	**Fields**	**Length [Bytes]**	**Value**
00h					**Get Version**		
Y	Y	Y	Y	Y	**Request:**		
					Protocol Version Major	1	Major Version of protocol used by Console Application. Currently there is only one version supported – 1. Versions with different major version are not compatible.
					Protocol Version Minor	1	Minor Version of protocol used by Console Application. If versions are different it is assumed that only a subset of commands recognized by party with lower version minor can be used.
					Respond:		
					Protocol Version Major	1	Major Version of protocol used by Console Application. Currently there is only one version supported – 1. Versions with different major version are not compatible.
					Protocol Version Minor	1	Minor Version of protocol used by Console Application. If versions are different it is assumed that only a subset of commands recognized by party with lower version minor can be used.
					Firmware Status	4	Current FW status
					Extended Firmware Status	4	Additional information about FW Status
					Uptime	4	System uptime
					Statuses: STATUS_SUCCESS. This command should always return status STATUS_SUCCESS		
03h					**Access SMBus**		
N	Y	Y	Y	Y	**Request:**		
					SMB Command	1	SMBus Command
					Flags	1	SMBus Command Flags
					Slave Address	1	Slave address
					Data	N	Writing data.
					Response:		
					Data	N	Reading data.
					Statuses: : STATUS_SUCCESS, STATUS_NOT_FOUND, STATUS_FAILURE or other error code		

Recovery	**SiEn**	**Intel Node Manager**	**DM**	**DNM**	**Fields**	**Length [Bytes]**	**Value**
0Fh					**MDES Frame**		
					Request		
					None		This command transfer diagnostic information from Intel ME.
					Response		
					MDES SMBus header		
					Event Id	4	
					Fragment Length	1	
					Fragment Flags		[0] - Defines if packet is part of a fragmented packet stream. [1] - Last packet of fragment stream. [2..7] - Reserved.
					MDES Event header		
					MDES Event		[0..7] - Event Group of MDES Event [8..9] - Value from MDES_SEVERITY_LEVEL [10..11] - Value from MDES_SENSITIVITY_LEVEL [12] - Indicates payload is a MDES_CHECKPOINT [13..15] - Reserved
					TimeStamp	4	
					File Line id	4	
					MDES Payload		
					CheckPoint Event/FW status	4/8	Checkpoint /FW Status 1 + FW Status 2
25h					**Read AUX Register**		
Y	Y	Y	Y	Y	**Request:**		
					Register Address		Address of AUX register as seen by Intel ME
					Respond:		
					Register Value	4	Value of AUX register
					Statuses: STATUS_SUCCESS. This command should always return status STATUS_SUCCESS		
26h					**Write AUX Register**		
Y	Y	Y	Y	Y	**Request:**		

Recovery	SiEn	Intel Node Manager	DM	DNM	Fields	Length [Bytes]	Value
					Address		
					Value		
					Respond:		
					AUX Value[hex]		
					Statuses: STATUS_SUCCESS, STATUS_FAILURE or other error code.		
27h					**Memory Read**		
Y	Y	Y	Y	Y	**Request:**		
					Address		
					Respond:		
					MEM Value[hex]		
					Statuses: STATUS_SUCCESS, STATUS_FAILURE or other error code.		
29h					**Read PCI Register**		
Y	Y	Y	Y	Y	**Request:**		
					PCI Device		
					Register		
					Respond:		
					PCI Reg Value[hex]		
					Statuses: STATUS_SUCCESS, STATUS_FAILURE or other error code. F -> bits 0-2; D -> bits 3-7; B -> bits 8-15 inline with the PCI device addressing used by the RdPCIConfig PECI function. This cmd applies only to **devices connected to the PCI-M bus**		
2Ah					**Write PCI Register**		
					Request:		
					PCI Device		
					Register		

Recovery	SiEn	Intel Node Manager	DM	DNM	Fields	Length [Bytes]	Value
					Value		
					Respond:		
					PCI Reg Value[hex]		
					Statuses: STATUS_SUCCESS, STATUS_FAILURE or other error code.		
30h					**Send Raw PECI**		
N	Y	Y	Y	Y	**Request:**		
					CPU Index	1	Index of CPU
					Write Length	1	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
					Read Length	1	Read Length (part of PECI standard header); this field shall be set to the proper value for this PECI command
					PECI Write Data	N	The remaining part of PECI command following the Read Length field (if any – this field does not exist for PECI Ping command); onlywrite 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, 0xC1, 0xC5, 0xE1, 0xE5; note that only Domain 0 codes are supported)
					Response:		
					PECI Read Data	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
					Statuses: = 00h – PECI response successfully retuned (see PECI response completion code for detailed response from PECI client) = A4h – Bad Read FSC in the response (even after the retry) = A5h – Bad Write FCS field in the response (even after the retry); this error code is also returned in case of Abort FSC in the response (as defined in PECI spec) and no response from PECI client (client device is not responding at all) = A6h – bad Write Length in the request = A7h – bad Read Length in the request = ABh – command code in the request not understood by Intel ME FW		

Recovery	**SiEn**	**Intel Node Manager**	**DM**	**DNM**	**Fields**	**Length [Bytes]**	**Value**
31h					**CRYPTO Test**		
N	Y	Y	Y	Y	**Request:**		
					Test ID	1	
					Respond:		
					Context	4	
32h					**MFS Test**		
N	Y	Y	Y	Y	**Request:**		
					Test ID	1	This parameter defines test scope. 0 - full tests for all encryption methods (algorithms). In current Intel ME FW version only full scope is supported. Other Test IDs are reserved for the future releases. It will be possible to run selected tests.
					Respond:		
					Context	4	Test Status. 0x00 - test passed successfully, 0x9E - test failed, other values are reserved for the future.
33h					**Get Image Version**		
Y	Y	Y	Y	Y	**Request:**		
					Image	1	
					Respond:		
					Major Version	2	
					Minor Version	2	
					Hotfix	2	
					Build	2	
					Statuses: STATUS_SUCCESS, STATUS_FAILURE or other error code.		
34h					**Switch to Image**		
Y	Y	Y	Y	Y	**Request:**		
					Image	1	
					Respond:		
					Statuses: STATUS_SUCCESS,		

Recovery	SiEn	Intel Node Manager	DM	DNM	Fields	Length [Bytes]	Value
					STATUS_FAILURE or other error code.		
35h					**Get Current Image**		
Y	Y	Y	Y	Y	**Request:**		
					None		
					Respond:		
					Image Number	1	
					Statuses: STATUS_SUCCESS, STATUS_FAILURE or other error code.		
36h					**Send Raw SST**		
N	Y	Y	Y	Y	**Request:**		
					Cpu Index	1	Index of CPU
					Write Length	1	Write Length
					Read Length	1	Read Length
					PECI Write Data	N	The remaining part of SST command following the Read Length field (if any); onlywrite data bytes shall be put here
					Response:		
					PECI Read Data	N	SST response data (if any); 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
					Statuses: = 00h – SST response successfully retuned = A2h – Bad Write FCS field in the response (even after the retry); this error code is also returned in case of Abort FSC in the response (as defined in PECI spec) and no response from PECI client (client device is not responding at all) = A4h – Bad read FSC in the response (even after the retry)		
37h					**File Directory Get First**		
N	Y	Y	Y	Y	**Request:**		
					Image	1	
					Respond:		

Recovery	SiEn	Intel Node Manager	DM	DNM	Fields	Length [Bytes]	Value
					Logical partition	1	
					Name	3	
					Attributes	2	
					Enumeration Context	4	
					Length	2	
					Statuses: STATUS_SUCCESS, STATUS_FAILURE or other error code.		
38h					**File Directory Get Next**		
N	Y	Y	Y	Y	**Request:**		
					Image	1	
					Logical partition	1	
					Enumeration context	4	
					Respond:		
					Logical partition	1	
					Name	3	
					Attributes	2	
					Enumeration Context	4	
					Length	2	
					Statuses: STATUS_SUCCESS, STATUS_FAILURE or other error code.		
39h					**File Contents Get**		
N	Y	Y	Y	Y	**Request:**		
					Image	1	
					Logical partition	1	
					File name	3	
					File offset	2	

Recovery	SiEn	Intel Node Manager	DM	DNM	Fields	Length [Bytes]	Value
					Read length	2	
					Respond:		
					Logical partition	1	
					Name	3	
					File attributes	2	
					Actual bytes returned of file contents	2	
					File contents	n	
					Statuses: STATUS_SUCCESS, STATUS_FAILURE or other error code.		
3Ah					**Intel Node Manager Get PSU Discovery Data**		
N	N	Y	Y	Y	**Request:**		
					PSU Index	1	Index of PSU (0…7)
					Response:		
					Power Reading Source	1	Same for all PSUs: 0 - IPMB BMC Total power via using the OEM command, 1 - IPMB BMC per-rail CPU via OEM command, 2 - PMBUS on the SMBUS PSU total output power reading, 3 - PMBUS on the SMLINK0 PSU total output power reading, 4 - PMBUS on the SMLINK1 PSU total output power reading, 5 - PMBUS on the SMBUS PSU total input power reading, 6 - PMBUS on the SMLINK0 PSU total input power reading, 7 - PMBUS on the SMLINK1 PSU total input power reading, 8 - PMBUS on the SMBUS PSU per-rail output power reading, 9 - PMBUS on the SMLINK0 PSU per-rail output power reading, 10 - PMBUS on the SMLINK1 PSU per-rail output power reading.
					Power Sensor Address	1	Address for the power readings
					Flags	1	BIT0: 0 - for PMBUS 1.2 PSU, 1 - for PMBUS 1.1 PSU; BIT1 - 1- Power read command (READ_PIN or READ_POUT) supported, 0 - Power read command not supported; BIT2 - 1 - PEC supported, 0 - PEC not supported
					Coefficient m	2	Coefficient m value obtained during discovery process
					Coefficient b	2	Coefficient b value obtained during discovery process
					Coefficient R	1	Coefficient R value obtained during discovery process
					Discovery Timestamp	4	Timestamp when the PSU was discovered
					Statuses: : STATUS_SUCCESS, STATUS_NOT_FOUND, STATUS_FAILURE or other error code		

Recovery	**SiEn**	**Intel Node Manager**	**DM**	**DNM**	**Fields**	**Length [Bytes]**	**Value**
3Bh					**Intel Node Manager Get LM75 Discovery Data**		
N	N	Y	Y	Y	**Request:**		
					Respond:		
3Ch					**Intel Node Manager Get SST Discovery Data**		
N	N	Y	Y	Y	**Request:**		
					Respond:		
4Ah					**Intel Node Manager Get CPU Discovery Data**		
N	N	Y	Y	Y	**Request:**		
					CPU Index	1	Index of CPU (0…7)
					Response:		
					PECI Address	1	PECI address supported by CPU protocol (0x30…0x37)
					CPU ID	4	CPUID family / model / stepping
					Platform ID	4	
					Uncore Device ID	4	
					CPU Patch revision	4	
					Tcontrol Value	1	
					Tjmax Value	1	
					Package Accumulated Energy status support	1	1 if Accumulated Energy status package is supported, 0 - otherwise
					Power Plane Accumulated Energy status support	1	BIT x = 1 if Accumulated Energy status for power plane x is supported, 0 – otherwise (Power planes 0 and 1 supported)
					Accumulated DRAM Energy status support	1	1 if Accumulated total DRAM Energy status package is supported, 0 - otherwise

Recovery	SiEn	Intel Node Manager	DM	DNM	Fields	Length [Bytes]	Value
					Accumulated DRAM Channel Energy status support	1	BIT x = 1 if Accumulated Energy status for DIMM x is supported, 0 – otherwise (DIMMs 0..3 are supported)
					Discovery Timestamp	4	Timestamp when the CPU was discovered
					Statuses: : STATUS_SUCCESS, STATUS_NOT_FOUND, STATUS_FAILURE or other error code		
4Bh					**Intel Node Manager Get Current Reading**		
N	N	Y	Y	Y	**Request:**		
					Reading Type	1	0x10 - Power domain 0 (total) reading in Watts, 0x11 - Power domain 1 (CPU) reading in Watts, 0x12 - Power domain 2 (memory) reading in Watts, 0x20 - Inlet temperature in deg. C
					Response:		
					Reading Value	4	The value of recent reading
					Statuses: STATUS_SUCCESS, STATUS_FAILURE or other error code.		
4Ch					**Intel Node Manager Reset Stats**		
N	N	Y	Y	Y	**Request:**		
					Mode	1	As defined in IPMI command "Reset Node Manager Statistics"
					Domain ID	1	
					Policy ID	1	
					Statuses: STATUS_SUCCESS, STATUS_FAILURE or other error code.		
4Dh					**Intel Node Manager Get Stats**		
N	N	Y	Y	Y	**Request:**		
					Mode	1	As defined in IPMI command "Get Node Manager Statistics"
					Domain ID	1	
					Policy ID	1	
					Response:		
					Current Reading	2	
					Min	2	Min reading value
					Max	2	Max reading value
					Average	2	Average reading value

Recovery	**SiEn**	**Intel Node Manager**	**DM**	**DNM**	**Fields**	**Length [Bytes]**	**Value**
					Timestamp	4	As defined in IPMI command "Get Node Manager Statistics"
					Statistics Reporting Period	4	As defined in IPMI command "Get Node Manager Statistics"
					DomainId / Policy State	1	As defined in IPMI command "Get Node Manager Statistics"
					Statuses: STATUS_SUCCESS, STATUS_FAILURE or other error code.		
50h					**Intel Node Manager Set Power/Throttling Target**		
N	N	Y	D	Y	**Request:**		
					DomainId	1	0x00 - platform; 0x01 - CPU; 0x02 - Memory
					ActionType	1	0x00 - non-aggressive power limit; 0x01 - aggressive power limit; 0x02 - throttling level
					Limit	2	Power limit in Watts; throttling level in percentage
					Statuses: = 00h – P/T Limit was set successfully = 81h – Invalid Domain ID = 83h – Invalid Action Type = 84h – Invalid P/T Limit		
51h					**Get Current PMC Patch Info**		
Y	Y	Y	Y	Y	**Request:**		
					No additional parameters		
					Response:		
					PMC Patching Status	2	Status containing bits like ERROR, STS_LOCK, DIS etc.
					PMC Patch Prod ID	1	ID of the motherboard for which patch was prepared.
					PMC Patch Rev ID Min	1	Minimum version of the PMC revision for which patch was prepared.
					PMC Patch Rev ID Max	1	Maximum version of the PMC revision for which patch was prepared.
					PMC Patch ROM ID Min	1	Minimum version of the ROM for which patch was prepared.
					PMC Patch ROM ID Max	1	Maximum version of the ROM for which patch was prepared.
					Prod ID	1	Actual ID of the motherboard of the DUT.

Recovery	**SiEn**	**Intel Node Manager**	**DM**	**DNM**	**Fields**	**Length [Bytes]**	**Value**
					Rev ID	1	Actual PMC Revision of the DUT.
					ROM ID	1	Actual ROM of the DUT.
					PMC Patching Time	4	Amount of tics from the start of platform booting to the end of PMC Patch applying
					Statuses: STATUS_SUCCESS, STATUS_FAILURE or other error code.		
52h					**Get First PMC Patch Info**		
Y	Y	Y	Y	Y	**Request:**		
					No additional parameters		
					Response:		
					PMC Patch Existence	1	If patch at this enumeration exists.
					PMC Patch Prod ID	1	ID of the motherboard for which patch was prepared.
					PMC Patch Rev ID Min	1	Minimum version of the PMC revision for which patch was prepared.
					PMC Patch Rev ID Max	1	Maximum version of the PMC revision for which patch was prepared.
					PMC Patch ROM ID Min	1	Minimum version of the ROM for which patch was prepared.
					PMC Patch ROM ID Max	1	Maximum version of the ROM for which patch was prepared.
					Enumeration Context	1	Opaque number (handler) that needs to be returned to the Intel ME FW to get a next PMC Patch
					Statuses: STATUS_SUCCESS, STATUS_FAILURE or other error code.		
53h					**Get Next PMC Patch Info**		
Y	Y	Y	Y	Y	**Request:**		
					Enumeration Context	1	From the response to Get First PMC Patch Info/Get Next PMC Patch Info
					Response:		
					PMC Patch Existence	1	If patch at this enumeration exists.
					PMC Patch Prod ID	1	ID of the motherboard for which patch was prepared.
					PMC Patch Rev	1	Minimum version of the PMC revision for which patch was prepared.

Recovery	SiEn	Intel Node Manager	DM	DNM	Fields	Length [Bytes]	Value
					ID Min		
					PMC Patch Rev ID Max	1	Maximum version of the PMC revision for which patch was prepared.
					PMC Patch ROM ID Min	1	Minimum version of the ROM for which patch was prepared.
					PMC Patch ROM ID Max	1	Maximum version of the ROM for which patch was prepared.
					Enumeration Context	1	Opaque number (handler) that needs to be returned to the Intel ME FW to get a next PMC Patch
					Statuses: STATUS_SUCCESS, STATUS_FAILURE or other error code.		
54h					**Get PBC statistics**		
N	N	Y	Y	Y	**Request:**		
					Domain Id	1	0x00 - platform; 0x01 - CPU; 0x02 - Memory
					Mode Type	1	0x00 - Limiting quality statistics; 0x01 - processing statistics; Other - Reserved
					Response:		
					Mode	1	0x00 - Limiting quality statistics; 0x01 - processing statistics;
					For Mode 0x00 or without Mode Type byte:		
					Above Limit Time	1	[0.1s] Time above limit
					Readings Error Time	1	[0.1s]
					Statistics Time	1	[0.1s] Total time for which statistics are reported. Max 20s.
					For Mode 0x01		
					Reserved	1	
					Max reading processing time in domain	2	[10 us] Time of reading processing in domain
					Max reading processing time in all domains	2	[10 us] Time of reading processing in all domains
					Statuses: STATUS_SUCCESS, STATUS_FAILURE or other error code.		
55h					**Clear PBC statistics**		
N	N	Y	Y	Y	**Request:**		

Recovery	**SiEn**	**Intel Node Manager**	**DM**	**DNM**	**Fields**	**Length [Bytes]**	**Value**
					Domain Id	1	0x00 - platform; 0x01 - CPU; 0x02 - Memory
					Statuses: STATUS_SUCCESS, STATUS_FAILURE or other error code.		
56h					**Set Max Allowed CPU P-State/T-State**		
					Request:		
					Domain Id	1	0 – Entire platform – for compatibility with previous Intel Node Manager versions P/T state settings are applied to CPU subsystem, others reserved

					Control Knob	1	1 – set max allowed CPU P-state and T-state;
							2 – set max allowed CPU cores;
							Others – reserved

					– For Control Knob set to 1:		
					P-State	1	P-State number to be set
					T-State	1	T-State number to be set
					– For Control Knob set to 2:		
					Cores	1	Max allowed cores
					Package	1	Processor package number
					Response:		
					Knob Sequence No	1	Sequence number of the request sent to host OSPM
					Statuses: STATUS_SUCCESS, STATUS_FAILURE or other error code.		
57h					**Get Max Allowed CPU P-State/T-State**		
N	N	Y	Y	Y	**Request:**		
					Domain Id	1	0x00 – Entire platform – for compatibility with previous NM versions P/T state settings are applied to CPU subsystem, others – Reserved
					Control Knob	1	1 – get max allowed CPU P-state/T-state 2 – get max allowed CPU cores Others – Reserved
					Package	1	1..8 - get number of cores available at given package 0xFF - get total number of allowed cores on a system Doesn't apply (dummy param) for Control Knob == 1
					Response:		

Recovery	SiEn	Intel Node Manager	DM	DNM	Fields	Length [Bytes]	Value
					– For Control Knob set to 1:		
					P-State	1	Current maximum P-State
					T-State	1	Current maximum T-State
					– For Control Knob set to 2:		
					Cores	1	Total requested by Intel ME number of allowed cores on a system. This is a number requested by Intel ME and OSPM is not required to fulfill this request
					Knob Sequence No	1	Sequence number of the request recently confirmed by host OSPM
					Statuses: STATUS_SUCCESS, STATUS_FAILURE or other error code.		
58h					**Read All Fuses**		
N	Y	Y	Y	Y	**Request:**		
					Response:		
					HW fuses	4	Settings of all HW fuses of the chipset responsible for disabling/enabling capabilities (aka Hard Staps)
					FW straps	4	Values of fuse overrides saved in Intel ME FW responsible for disabling/enabling capabilities (aka Soft Staps)
					Final capability states	4	The result of an AND operation for capabilities enabled/disabled by the HW and FW straps
					Statuses: STATUS_SUCCESS, STATUS_FAILURE or other error code.		
59h					**Read CPU Complex ID "CPUID"**		
N	Y	Y	Y	Y	**Request:**		
					Response:		
					CCID	4	Currently installed CPU's Complex ID
					CCID in Intel ME FW	4	Saved in Intel ME FW CPU's Complex ID
					Statuses: STATUS_SUCCESS, STATUS_FAILURE or other error code.		

Recovery	SiEn	Intel Node Manager	DM	DNM	Fields	Length [Bytes]	Value
5Ah					**Set PTAM State**		
N	N	Y	Y	Y	**Request:**		
					PTAM State	1	0 – Disable PTAM; 1 – Enable PTAM
					Response:		
					PTAM State	1	PTAM state before this command was executed
					Statuses: STATUS_SUCCESS, STATUS_FAILURE or other error code.		
5Bh					**Get PTAM State**		
N	N	Y	Y	Y	**Request:**		
					Response:		
					PTAM State	1	0x00 – PTAM Enabled; 0x01 – PTAM Disabled
					Statuses: STATUS_SUCCESS, STATUS_FAILURE or other error code.		
5Ch					**Get P/T State Violation Time**		
N	N	Y	Y	Y	**Request:**		
					Response:		
					ViolationTime / TotalTime [%]	1	Percentage of time when P/T State violation occurred
					TotalTime [0.1s]	4	Total time in which violation was counted
					Statuses: STATUS_SUCCESS, STATUS_FAILURE or other error code.		
5Dh					**Reset P/T State Violation Time**		
N	N	Y	Y	Y	**Request:**		
					Response:		
					Statuses: STATUS_SUCCESS, STATUS_FAILURE or other error code.		
60h					**Read ICC OCKEN**		
Y	Y	Y	Y	Y	**Request:**		

Recovery	**SiEn**	**Intel Node Manager**	**DM**	**DNM**	**Fields**	**Length [Bytes]**	**Value**
					Response:		
					OCKEN	4	Currently status of OCKEN register for ICC
					Statuses: STATUS_SUCCESS, STATUS_FAILURE or other error code.		
61h					**Read ICC SSCCTL**		
Y	Y	Y	Y	Y	**Request:**		
					Response:		
					SSCCTL	4	Currently status of SSCCTL register for ICC
					Statuses: STATUS_SUCCESS, STATUS_FAILURE or other error code.		
63h					**Flash Performance Test Erase Start**		
N	N	Y	Y	Y	**Request:**		
					Size To Erase	2	Size to erase [kB], up to the partition size (0 means whole partition).
					Statuses: STATUS_SUCCESS, STATUS_FAILURE or other error code.		
64h					**Flash Performance Test Write Start**		
N	N	Y	Y	Y	**Request:**		
					Size To Write	2	Size to write [kB], up to the partition size (0 means whole partition).
					Pattern	2	Pattern to write.
					Write Block Size	1	Size of the single write block (0..255 means 1..256).
					Statuses: STATUS_SUCCESS, STATUS_FAILURE or other error code.		
65h					**Flash Performance Test Get Stats**		
Y	Y	Y	Y	Y	**Request:**		
					Response:		
					Erase Throughput Min	2	Erase Throughput Min [kB/s] - for a block that took the longest to erase
					Erase Throughput Max	2	Erase Throughput Max [kB/s] - for a block that took the shortest to erase
					Erase	2	Erase Throughput Avg [kB/s] - average for entire erase test

Recovery	SiEn	Intel Node Manager	DM	DNM	Fields	Length [Bytes]	Value
					Throughput Avg		
					Last Erase End TimeStamp	4	Timestamp (in IPMI format) of the end of the last erase test
					Write Throughput Avg	2	Write Throughput Avg [kB/s] - average for entire write test
					Last Write End Time Stamp	4	Timestamp (in IPMI format) of the end of the last write test
					Read Throughput Avg	2	Read Throughput Avg [kB/s] - average for entire read test
					Last Read End Time Stamp	4	Timestamp (in IPMI format) of the end of the last read test
					Partition Address	4	Address (in flash) of partition used for tests (second operational)
					Partition Size	4	Size [B] of partition used for tests
					Partition Block Size	4	Size [B] of partition erase block
					Test Done	1	0 – test is ongoing; 1 – test finished
					Statuses: STATUS_SUCCESS, STATUS_FAILURE or other error code.		
66h					**Flash Performance Test Read Start**		
N	N	Y	Y	Y	**Request:**		
					Size To Read	2	Size to read [kB], up to the partition size (0 means whole partition).
					Read Block Size	1	Size of the single read block (0..255 means 1..256).
					Statuses: STATUS_SUCCESS, STATUS_FAILURE or other error code.		
67h					**NM PBC Get Regulator Settings**		
N	N	Y	Y	Y	**Request:**		
					Domain	1	Domain number
					Response:		
					Regulator 0 is ON	4	Is regulator enabled?
					Regulator 0 error	4	Current error value [FIXED INT]
					Regulator 0 Kp	4	Regulator's Kp coefficient [FIXED INT]

Recovery	SiEn	Intel Node Manager	DM	DNM	Fields	Length [Bytes]	Value
					Regulator 0 Ki	4	Regulator's Ki coefficient [FIXED INT]
					Regulator 0 Kd	4	Regulator's Kd coefficient [FIXED INT]
					Regulator 1 is ON	4	Is regulator enabled?
					Regulator 1 error	4	Current error value [FIXED INT]
					Regulator 1 Kp	4	Regulator's Kp coefficient [FIXED INT]
					Regulator 1 Ki	4	Regulator's Ki coefficient [FIXED INT]
					Regulator 1 Kd	4	Regulator's Kd coefficient [FIXED INT]
					Regulator 2 is ON	4	Is regulator enabled?2
					Regulator 2 error	4	Current error value [FIXED INT]
					Regulator 2 Kp	4	Regulator's Kp coefficient [FIXED INT]
					Regulator 2 Ki	4	Regulator's Ki coefficient [FIXED INT]
					Regulator 2 Kd	4	Regulator's Kd coefficient [FIXED INT]
					Statuses: STATUS_SUCCESS, STATUS_FAILURE or other internal firmware code that is present in me_status.h file.		
68h					**Intel Node Manager PBC Set Regulator Settings**		
N	N	Y	Y	Y	**Request:**		
					Domain	1	Domain number
					Regulator Number	1	Regulator Number
					Regulator is ON	4	Is regulator enabled?
					Regulator error	4	New error value [FIXED INT]
					Regulator Kp	4	Regulator's Kp coefficient [FIXED INT]
					Regulator Ki	4	Regulator's Ki coefficient [FIXED INT]
					Regulator Kd	4	Regulator's Kd coefficient [FIXED INT]
					Response:		
					Statuses: STATUS_SUCCESS, STATUS_FAILURE or other internal firmware code that is present in me_status.h file.		
70h					**Get SDR Internal Version**		
N	N	N	Y	Y	This command returns the SDR internal version number		

Recovery	SiEn	Intel Node Manager	DM	DNM	Fields	Length [Bytes]	Value
					Request		
					None		
					Response		
					SDR Version Number	4	SDR Version number
					Status: STATUS_SUCCESS, STATUS_FAILURE or other error code.		
71h					**Get Network Stats**		
N	N	N	Y	Y	This command retrieves the low level LOM network statistics		
					Request		
					None		
					Response		
					Frames Transmitted	2	Number of SMBus frames received.
					Frames Received	2	Number of SMBus frames transmitted.
					Packets Sent	2	Number of Packets sent
					Packets Received	2	Number of Packets received.
					Link Status Count	1	Link status change count
					SMBus failure count	2	SMBus transaction failure count
					Status: STATUS_SUCCESS, STATUS_FAILURE or other error code.		
72h					**Send NIC Test Packet**		
N	N	N	Y	Y	Sends a test broadcast ARP packet on the network. The test packet is a predefined packet within the FW. The packet format is as follows: {0xff,0xff,0xff,0xff,0xff, 0xff,<Src MAC>, ,0x08,0x06,0x00, 0x01,0x08,0x00,0x06,0x04, 0x00,0x01,<Src MAC>, 0xC0,0xA8,0x0, 0x64, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0xc0,0xa8, 0x0, 0x1, 0x00, 0x00, 0x01, 0x57};		
					Request		
					None		
					Response		

Recovery	**SiEn**	**Intel Node Manager**	**DM**	**DNM**	**Fields**	**Length [Bytes]**	**Value**
					None		
					Status: STATUS_SUCCESS, STATUS_FAILURE or other error code.		
73h					**Get DoH Interface Stats**		
N	N	N	Y	Y	This command retrieves the DCMI over HECI stats		
					Request		
					None		
					Response		
					DoH Received	4	Number of DoH packets received
					DoH Sent	4	Number of DoH packets sent
					Status: STATUS_SUCCESS, STATUS_FAILURE or other error code.		
74h					**Get BIOS DoH Interface Stats**		
					This command retrieves the DCMI over HECI stats		
					Request		
					None		
					Response		
					BIOS DoH Received	4	Number of DoH packets received over static DoH interface
					BIOS DoH Sent	4	Number of DoH packets sent over static DoH interface
					Status: STATUS_SUCCESS, STATUS_FAILURE or other error code.		
75h					**Get KT Stats**		
N	N	N	Y	Y	This command retrieves the KT Serial stats		
					Request		
					None		
					Response		
					Serial Recvd	4	Number of serial bytes received
					Serial Sent	4	Number of serial bytes sent

Recovery	SiEn	Intel Node Manager	DM	DNM	Fields	Length [Bytes]	Value
						4	Reserved
					Status: STATUS_SUCCESS, STATUS_FAILURE or other error code.		
76h					**Send KT Test Packet**		
N	N	N	Y	Y	Sends the test character string – “00000157” over the KT serial interface		
					Request		
					None		
					Response		
					None		
					Status: STATUS_SUCCESS, STATUS_FAILURE or other error code.		
77h					**Validate SDR PIA Correlation**		
N	Y	Y	Y	Y	Validates if there are any PIAs that do not have a corresponding SDRs		
					Request		
					None		
					Response		
					Num of SDR wo PIA	2	Number of Sensors that do not have SDRs (N)
					Sensor Num	1	First failed sensor number.
					Status: STATUS_SUCCESS, STATUS_FAILURE or other error code.		
78h				**Get Sensor Reading Status**			
N	Y	Y	Y	Y	**Request**		
					Sensor Number	1	Sensor Number
					Response		
					Reading Count	2	Number of readings taken by the hardware (maybe rolled over).
					Last Reading Timestamp	4	Last timestamp the sensor was monitored

Recovery	SiEn	Intel Node Manager	DM	DNM	Fields	Length [Bytes]	Value
					Reserved	4	
					Status: STATUS_SUCCESS, STATUS_FAILURE or other error code.		
79h			**Send Test Sensor Event**				
N	Y	Y	Y	Y	Log a time stamped test sensor event to SEL. Record format "<rec ID>, 0xC0, <Time stamp>, 0x0, 0x1, 0x57, 0xa5, 0xa5, 0xa5, 0xa5, 0xa5, 0xa5"		
					Request		
					None		
					Response		
					Status: STATUS_SUCCESS, STATUS_FAILURE or other error code.		
7Ah			**FSC Config Check**				
N	N	N	Y	Y	Tests the FSC configuration to make sure the PIA is set properly and the FSC sensors are available in the system		
					Request		
					Domain Num	1	Domain number that has error configuration (0-3)
					Response		
					First Error Profile ID	2	First error in profile type & profile ID, 0xFFFF is no error
					First missing sensor ID	1	Sensor number, 0xFF is no error
					Reserved	4	Internal Use only.
					Status: STATUS_SUCCESS, STATUS_FAILURE or other error code.		
7Bh			**Get Current Fan Contribution**				
N	N	N	Y	Y	Gets the current fan contribution on the profile number and domain number specified		
					Request		
					Domain Number		
					Profile Number		

Recovery	SiEn	Intel Node Manager	DM	DNM	Fields	Length [Bytes]	Value
					Response		
					PWM Contribution	1	% PWM contribution based on the profile algos and the current temp
					Status: STATUS_SUCCESS, STATUS_FAILURE or other error code.		
7Ch			**Set Fan PWM**				
N	N	N	Y	Y	Sets the PWM of the specified fan domain. This command causes FSC to stop and all domain fans to be boosted to 100% except the speed of the specified domain fan. In order to re enable FSC control behavior, an AC cycle is needed		
					Request		
					Domain number	1	
					PWM%	1	
					Response		
					None		
					Status: STATUS_SUCCESS, STATUS_FAILURE or other error code.		
7Dh			**Get IPMB Stats**				
N	Y	Y	N	N	This command retrieves the IPMB stats		
					Request		
					None		
					Response		
					IPMB Packets Recvd	2	Number of IPMB frames received
					IPMB Packets Sent	2	Number of IPMB frames sent
					Reserved	4	
					Status: STATUS_SUCCESS, STATUS_FAILURE or other error code.		
7Eh			**Get Netstack Status**				
N	N	N	Y	Y	This command retrieves the network connectivity status such as DHCP, and IP address		
					Request		

Recovery	**SiEn**	**Intel Node Manager**	**DM**	**DNM**	**Fields**	**Length [Bytes]**	**Value**
					None		
					Response		
					DHCP state	1	0 success, = 1 DHCP timeout, =0xFF for Static Cfg
					IP Address	4	IP Address, is valid if DHCP state value is 0 or is 0xFF
					Status: STATUS_SUCCESS, STATUS_FAILURE or other error code.		
80h			**Component Health Status**				
N	Y	Y	Y	Y	Status of registered components monitored by FW health monitoring component.		
					Request		
					Enumeration Context	2	Parameter used to iterate over the list of components. Shall be 0 for first query request. For the next request, it shall be copied from the previous response.
					Response		
					Total components in the system	2	The total count of items that can be obtained with a sequence of these commands. A single response may contain only a part of the full list. Multiple requests may be needed to obtain the full list.
					Enumeration Context	2	Opaque number (handler) that needs to be returned to the Intel ME FW to get a next request.
					Query Response - List Type	5 * 4	5 items from the list. Each item is a 4-byte response. Some specific response codes are as follows and all remaining ones are undocumented: 2c07XX00h - FSC stopped working for various reasons (HW or FW errors), 0316XX02h - Recovery bootloader image loaded because operational image is corrupted, 0316XX03h - Secondary operational image loaded because operational image is corrupted, 030dXX00h - Flash erase error, 0307XX01h - Flash file system error, 0307XX00h - Unable to write to flash due to wear out protection, 2c11XX00h - LOM chip is not accessible over SMBus, 2b04XX00h - PSU is not accessible over PMBus, 0310XX00h - PECI is not accessible (reported when PECI is expected to be available), 2c07xx81h - FSC configuration is invalid or file error, 2c0aXX81h - SDR configuration is invalid or file error, 2c1cXX81h - SEL configuration is invalid or file error, 2c1dXX81h - FRU configuration is invalid or file error, XXXXXX81h - other component configuration is invalid or file error.
					Status: STATUS_SUCCESS, STATUS_FAILURE or other error code.		

Recovery	SiEn	Intel Node Manager	DM	DNM	Fields	Length [Bytes]	Value
81h			**Buffer Pool Wait Log**				
N	Y	Y	Y	Y	Buffer pools ever exhausted		
					Request		
					Pool ID	1	Pool ID of the buffer pool
					Response		
					Count of exhausted events	4	Number of times the pool of buffers has been exhausted
					Status: STATUS_SUCCESS, STATUS_FAILURE or other error code.		
82h			**Memory Status**				
N	Y	Y	Y	Y	Memory pool current status		
					Request		
					Pool ID	1	Pool ID of the memory pool
					Response		
					Free bytes	4	Number of free bytes in the pool
					Status: STATUS_SUCCESS, STATUS_FAILURE or other error code.		
90h			**Mdes Set Logger On**				
N	Y	Y	Y	Y	Sets MDES Logger On		
					Request		
						0	
					Response		
						0	
					Status: STATUS_SUCCESS, STATUS_FAILURE or other error code.		
91h			**Mdes Set Logger Off**				
N	Y	Y	Y	Y	Sets MDES Logger Off		
					Request		

Recovery	SiEn	Intel Node Manager	DM	DNM	Fields	Length [Bytes]	Value
						0	
					Response		
						0	
					Status: STATUS_SUCCESS, STATUS_FAILURE or other error code.		
92h					**Mdes Get Logger State**		
N	Y	Y	Y	Y	Gets MDES Logger State		
					Request		
						0	
					Response		
					Logger State	1	Off = 0, On = 1
					Status: STATUS_SUCCESS, STATUS_FAILURE or other error code.		
93h					**Mdes Set Error Filter**		
N	Y	Y	Y	Y	Sets MDES Error Filter		
					Request		
					Error Filter	1	All = 0, Low = 1, High = 2, Critical = 3
					Response		
						0	
					Status: STATUS_SUCCESS, STATUS_FAILURE or other error code.		

Recovery	SiEn	Intel Node Manager	DM	DNM	Fields	Length [Bytes]	Value
94h			**Mdes Get Error Filter**				
N	Y	Y	Y	Y	Gets MDES Error Filter		
					Request		
						0	
					Response		
					Error Filter	1	All = 0, Low = 1, High = 2, Critical = 3
					Status: STATUS_SUCCESS, STATUS_FAILURE or other error code.		
95h			**Mdes Set Event Filter**				
N	Y	Y	Y	Y	Sets MDES Event Filter		
					Request		
					Event Group	1	0 … 127 (CP = 1, LOADMGR = 4, PWRMGMT = 5, FWSTS = 72, TMRALIVE = 73, KERNEL = 82, POLICY = 83, HOSTCOMM = 84)
					Event Filter	4	0x00000000 … 0xFFFFFFFF Recommended values: CP = 0x1, LOADMGR = 0x3F6, PWRMGMT = 0x1, FWSTS = 0x1, TMRALIVE = 0x1, KERNEL = 0xFFFFFFFF, POLICY = 0xFFFFFFFF, HOSTCOMM = 0xFFFFFFFF
					Response		
						0	
					Status: STATUS_SUCCESS, STATUS_FAILURE or other error code.		

Recovery	SiEn	Intel Node Manager	DM	DNM	Fields	Length [Bytes]	Value
96h			**Mdes Get Event Filter**				
N	Y	Y	Y	Y	Gets MDES Event Filter		
					Request		
					Event Group	1	0 … 127 (CP = 1, LOADMGR = 4, PWRMGMT = 5, FWSTS = 72, TMRALIVE = 73, KERNEL = 82, POLICY = 83, HOSTCOMM = 84)
					Response		
					Event Group	1	0 … 127 (CP = 1, LOADMGR = 4, PWRMGMT = 5, FWSTS = 72, TMRALIVE = 73, KERNEL = 82, POLICY = 83, HOSTCOMM = 84)
					Event Filter	4	0x00000000 … 0xFFFFFFFF
					Status: STATUS_SUCCESS, STATUS_FAILURE or other error code.		
97h			**Mdes Set Logging Iface**				
N	Y	Y	Y	Y	**Request**		Sets MDES Logging Interface
					Logging Iface	1	None = 0, SmBus = 2
					Response		
						0	
					Status: STATUS_SUCCESS, STATUS_FAILURE or other error code.		
98h			**Mdes Get Logging Iface**				
N	Y	Y	Y	Y	Gets MDES Logging Interface		
					Request		
						0	

Recovery	SiEn	Intel Node Manager	DM	DNM	Fields	Length [Bytes]	Value
					Response		
					Logging Iface	1	None = 0, SmBus = 2
					Status: STATUS_SUCCESS, STATUS_FAILURE or other error code.		
99h			**Mdes Set Buffer Mode**				
N	Y	Y	Y	Y	Sets MDES Buffer Mode		
					Request		
					Buffer Mode	1	Blocking = 0, Buffered = 1, Delayed Flush = 2
					Response		
						0	
					Status: STATUS_SUCCESS, STATUS_FAILURE or other error code.		
9Ah			**Mdes Get Buffer Mode**				
N	Y	Y	Y	Y	Gets MDES Buffer Mode		
					Request		
						0	
					Response		
					Buffer Mode	1	Blocking = 0, Buffered = 1, Delayed Flush = 2
					Status: STATUS_SUCCESS, STATUS_FAILURE or other error code.		
9Bh			**Mdes Set SmBus Address**				
N	Y	Y	Y	Y	Sets MDES SmBus Address		
					Request		
					SmBus Address	1	
					Response		7-bit SmBus address [0x00 … 0x7F]
						0	
					Status: STATUS_SUCCESS, STATUS_FAILURE or other error code.		
9Ch			**Mdes Get SmBus Address**				
N	Y	Y	Y	Y	Gets MDES SmBus Address		
					Request		

Recovery	SiEn	Intel Node Manager	DM	DNM	Fields	Length [Bytes]	Value
						0	
					Response		
					SmBus Address	1	7-bit SmBus address [0x00 … 0x7F]
					Status: STATUS_SUCCESS, STATUS_FAILURE or other error code.		
9Dh			**Set MGPIO Output State**				
N	Y	Y	Y	Y	Sets Intel ME as owner of the GPIO, puts MGPIO into output mode and sets the MGPIO output state		
					Request		
					MGPIO Number	1	MGPIO number [0:12]
					MGPIO State	1	0 - for low state, 1 for high state
					Response		
						0	
					Status: STATUS_SUCCESS, STATUS_INVALID_PARAMS or other error code.		
B0h			**Get IDLM PID**				
Y	N	N	N	N	Gets the device PID for IDLM		
					Request		
						0	
					Response		
					DeviceId	2	Device ID of the MBB bridge
					FuseTestFlags	2	Flags to be passed to Host
					UMCHID[0]	4	UMCHID value calculated from unique fuses
					UMCHID[1]	4	UMCHID value calculated from unique fuses
					UMCHID[2]	4	UMCHID value calculated from unique fuses
					UMCHID[3]	4	UMCHID value calculated from unique fuses
					Status: STATUS_SUCCESS, STATUS_FAILURE or other error code.		
F0h			**Force Intel ME Reset**				
N	Y	Y	Y	Y	Forces Intel ME reset		
					Request		
					Magic Number	4	Magic Number = 0x3CC3A55A

Recovery	SiEn	Intel Node Manager	DM	DNM	Fields	Length [Bytes]	Value
					Response		
					Status: STATUS_SUCCESS, STATUS_FAILURE or other error code.		
F0h GUI			**ReadCurrentFanSpeeds**				
Y	N	N	N	N	Note: Recovery Mode command only.		
					Request:		
					None		
					Response:		
					Fan1Speed	2	Fan #1 Speed Value expressed as RPM. 0FFFFh value indicates that the fan is not spinning i.e. has been stopped or the tachometer input is not connected to a valid signal. 0FFFEh value indicates the fan is unexpectedly stalled. 0FFFDh value indicates timeout reading the fan speed from HW (command can be retried).
					Fan2Speed	2	Fan #2 Speed Value in RPMs. Rest of detailed description is the same as for Fan #1.
					Fan3Speed	2	Fan #3 Speed Value in RPMs. Rest of detailed description is the same as for Fan #1.
					Fan4Speed	2	Fan #4 Speed Value in RPMs. Rest of detailed description is the same as for Fan #1.
					Fan5Speed	2	Fan #5 Speed Value in RPMs. Rest of detailed description is the same as for Fan #1.
					Fan6Speed	2	Fan #6 Speed Value in RPMs. Rest of detailed description is the same as for Fan #1.
					Fan7Speed	2	Fan #7 Speed Value in RPMs. Rest of detailed description is the same as for Fan #1.
					Fan8Speed	2	Fan #8 Speed Value in RPMs. Rest of detailed description is the same as for Fan #1.
					Status: STATUS_SUCCESS, STATUS_FAILURE or other error code.		
GUI			**ICC Set Spread Spectrum**				
Y	Y	Y	Y	Y	Use this command to enable/disable Spread Spectrum generators for particular clock outputs and to set parameters of the generators		
					Request:		

Recovery	**SiEn**	**Intel Node Manager**	**DM**	**DNM**	**Fields**	**Length [Bytes]**	**Value**
					Check the Intel ME-BIOS i/f spec for the command params		
					Respond:		
					ICC Record		ICC record describing the settings
GUI					**ICC Set Clock Enables**		
Y	Y	Y	Y	Y	Use this command to disable (gate) unpopulated clock outputs or enable them		
					Request:		
					Check the Intel ME-BIOS i/f spec for the command params		
					Respond:		
					ICC Record		ICC record describing the settings
GUI					**ICC Get Clock Enables**		
Y	Y	Y	Y	Y	**Request:**		
					Boot Clock Settings	1	If one, returns Boot Clock Settings if zero, returns Current Clock Settings
					Respond:		
					ICC Record		ICC record describing the settings
					FwLogGetInfo		
					FwLogGetEntry		
					FwLogGetNextEntry		

C MESDC Example Reports

C.1 ICC Settings

```
-= [ICC Settings] report generation start: 13/02/27 10:57:04.515 =-
ICC Register: SECURITY0                      state: 0xFFFFFFFF
ICC Register: SECURITY1                      state: 0xFFFFFFFF
ICC Register: SECURITY2                      state: 0xFFFFFFFF
ICC Register: BIAS0                          state: 0x2AB02AB0
ICC Register: BIAS1                          state: 0x000000F2
ICC Register: BIAS2                          state: 0x00000000
ICC Register: BIASMISC                       state: 0x00000088
ICC Register: CLKPATH                        state: 0x070F7F99
ICC Register: MODDIV_FB                      state: 0x00000134
ICC Register: LCPLL0                         state: 0x00000000
ICC Register: LCPLL1                         state: 0x00000000
ICC Register: LCPLL2                         state: 0x00005560
ICC Register: LCPLL3                         state: 0x00000000
ICC Register: LCPLL4                         state: 0x00000000
ICC Register: LCPLLMON                       state: 0x00000000
ICC Register: OSC0                           state: 0x0000005C
ICC Register: SFR0                           state: 0x00020301
ICC Register: MONPORT0                       state: 0xE0000000
ICC Register: MONPORT1                       state: 0x00000000
ICC Register: MUXTOP                         state: 0x00000000
ICC Register: VISACTL0                       state: 0x00000000
ICC Register: VISACTL1                       state: 0x00000000
ICC Register: VISACTL2                       state: 0x00000000
ICC Register: CBMISC                         state: 0x00000000
ICC Register: SBEPCTL                        state: 0x00020310
ICC Register: MONPORT2                       state: 0x00000000
```

```
                ICC Register: CMNRSTFSM                     state: 0x00001D4C
```

C.2 Intel ME FW Health Check

```
-= [ME FW Health Check] report generation start: 13/09/17 14:28:27.925 =-
ME FW VER: 3.0.3.681 (Operational)
No ME FW Exceptions Detected
SELF TEST: No Error
Last Global Reset Cause: Loss of SUS well power.
ME FW Reset Counter: 22
Reset Flags from ME: [ME reset has been initialized; ME reset occurred; CrdaNotDone flag set]
-= [ME FW Health Check] report generation end:   13/09/17 14:28:44.410 =-
```

C.3 Intel Node Manager State Check

```
-= [Node Manager State Check] report generation start: 13/09/17 14:31:36.946 =-
Global NM Policy Control Disabled
Power Budget Not Set for Domain: 0
Power Budget Not Set for Domain: 1
Power Budget Not Set for Domain: 2
Power Budget Not Set for Domain: 3
Power Budget Not Set for Domain: 4
PState = 0
TState = 0
Enabled Cores = 28
POWER STATS: Input Power Domain 0 Minimum = 88
POWER STATS: Input Power Domain 0 Maximum = 120
POWER STATS: Input Power Domain 0 Average = 91
POWER STATS: Input Power Domain 1 Minimum = 0
POWER STATS: Input Power Domain 1 Maximum = 63
POWER STATS: Input Power Domain 1 Average = 36
POWER STATS: Input Power Domain 2 Minimum = 0
POWER STATS: Input Power Domain 2 Maximum = 0
POWER STATS: Input Power Domain 2 Average = 0
POWER STATS: Input Power Domain 3 Minimum = 78
POWER STATS: Input Power Domain 3 Maximum = 105
POWER STATS: Input Power Domain 3 Average = 80
POWER STATS: Input Power Domain 4 Minimum = 0
POWER STATS: Input Power Domain 4 Maximum = 0
POWER STATS: Input Power Domain 4 Average = 0
NM Capabilities: Capability Policy Interface Not Available
PSU 0 Status: OK
PSU 5 Status: OK
PSU 6 Status: OK
PSU 7 Status: OK
Host Communication: Failure Not Detected
SMART/CLST not triggered
CPU Configuration Message: Received
EOP: Received
NM Health Flags: OK
CPU 0 Thermal Status: OK
CPU 1 Thermal Status: OK
Current Throttling Level: 0.
PSU 0 energy statistics.
First read:
Accumulator value: 0x5974
Rollover Count:    0xF6
Sample Count:      0x00C9EF
Second read:
Accumulator value: 0x5A82
Rollover Count:    0xF6
Sample Count:      0x00C9F2
Coefficients:
m -coefficient:    1
b -coefficient:    0
r -coefficient:    0
-= [Node Manager State Check] report generation end:   13/09/17 14:31:58.493 =-
```

C.4 SUSRAM Parse

```
-= [Susram Parse] report generation start: 13/09/17 14:48:21.300 =-
SUSRAM_ID_DESCR: id = FFFFFE, len = 04
SusRamDescriptor:
- HeaderId:                   53555324
- ReclaimSrcOffset:           0000
- ReclaimDestOffset:          0000
- TotalSize:                  00000800

SUSRAM_ID_HMRFPO_ENTER_RCVR: id = 000036, len = 05
BupSusFwRecoveryData:
- ImageOffset:                00000000
- ImageSize:                  00000000
- ImageNumber:                UNSPECIFIED_IMAGE      FF
- RecoveryModeReason:         RECOVERY_MODE_REASON_OPERATIONAL      00
- ExtendedErrorCode:          OPERATIONAL_EXT_ERROR_CODE_GENERIC      00
HothamRecoveryActions:        00000000
= ChangeHostAccessMeFlash:    0
= EnRead:                     0
= EnWrite:                    0

SUSRAM_ID_BUP_ERROR: id = 00000F, len = 03
BupErrorData:
- HaltReason:                 0000
  = MultipleException:        0
  = BadVscc:                  0
  = FtpLdFlr:                 0
  = MfgHalt:                  0
  = MprViolation:             0
  = ClockFreqViolation:       0
- ErrorFlags:                 0000
  = MultipleExceptions:       0
  = AltOperationalUsed:       0
  = EnterRecovery:            0
  = HaltMe:                   0

SUSRAM_ID_RECOVERY_ENTRY_CAUSE: id = 00003C, len = 03
RecoveryEntry:
- Cause:                      RECOVERY_ENTRY_CAUSE_NONE      00
- ExtendedErrorData:          00

SUSRAM_ID_PWR_MGMT: id = 000007, len = 03
ResetData:                    9000020A
- Reserved0:                  0
- MRSTMeInitiated:            1
- GlobalRSTMeInitiated:       0
- CurrHostStateS0:            1
- CurrHostStateS1:            0
- CurrHostStateS3:            0
- CurrHostStateS4:            0
- CurrHostStateS5:            0
- CurrMEStateM0:              0
- CurrMEStateM0NU:            1
- CurrMEStateM3:              0
- CurrMEStateMoff:            0
- PreIcvFailureGlobalRST:     0
- PreIcvCheckErrorStartup:    0
- CSTMGlobalRST:              0
- CryptoGlobalRST:            0
- ExceptionMeRST:             0
- ExceptionGlobalRST:         0
- HostColdRstMoff:            0
- PseudoGlobalRst:            0
- PchGlobalRst:               0
- S5GlobalRst:                0
- PchRstWarnTmOutGblRst:      0
- FwWDTGlobalRst:             0
- HwErrGlobalRst:             0
- ThermalTripGlobalRst:       0
- FtLoadFailureGlobalRst:     0
- T34Timeout:                 0
- MRSTMeOccurred:             1
- BiosHostResetNotified:      0
- HobitNotDone:               0
- CrdaNotDone:                1

SUSRAM_ID_MPHY_CTXT: id = 00005A, len = 05
MphyCtxt:
- Table Identifiers:          0120BA06
- Length:                     000000EC
- Max Length:                 000004BA

SUSRAM_ID_PMC_PATCH_DIAG: id = 00003A, len = 07
PmcPatchDiag:
- pmc_timeout:                0000
- pmc_problem:                0016
- pmc_success:                0001
- pmc_FuseDIS:                PMC_PATCH_FUSEDIS_DEFAULT      00
- pmc_runtime:                0000E12A
- status:                     PMC_NOT_NEEDED      00000002
- curr_id:                    00
- prod_id:                    05
- rev_id:                     04
- rom_id:                     27

SUSRAM_ID_BRINGUP_FUSES: id = 00000A, len = 05
FlashSettings:
- Vscc:                       00000000
- LVscc:                      00000000
- MePolicy:                   ME_POLICY_DISABLED      00
- VsccGetStatus:              00
- DidBiosTimeout:             00

SUSRAM_ID_RECOVERY_FW_UPDATE_NONCE: id = 000200, len = 04
Nonce:                        45C5A2592CF21F2D

SUSRAM_ID_EXCEPTION_LOG: id = 00003F, len = 1C
ExceptionDebugContext:
- Debug:
  = DebugArcHandler:          00
  = DebugBackBoneHandler:     00
  = DebugPostedCompHandler:   00
  = DebugWdHandler:           00
  = DebugPcimHandler:         00
  = DebugSupertaskHandler:    00
- Log[0]:
  = ExceptionSource:          EXCEPTION_SOURCE_NONE      0000
  = ExceptionCause:           INT_CAUSE_ARC_NONE      0000
  = TimeStamp:                00000000
  = InstPointer:              00000000
  = LastProcRetPtr:           00000000
  = FwOpState:                00
  = FwPowerState:             00
  = InOperational :           0
  = Reserved:                 00
- Log[1]:
  = ExceptionSource:          EXCEPTION_SOURCE_NONE      0000
  = ExceptionCause:           INT_CAUSE_ARC_NONE      0000
  = TimeStamp:                00000000
  = InstPointer:              00000000
  = LastProcRetPtr:           00000000
  = FwOpState:                00
  = FwPowerState:             00
  = InOperational :           0
  = Reserved:                 00
- Log[2]:
  = ExceptionSource:          EXCEPTION_SOURCE_NONE      0000
  = ExceptionCause:           INT_CAUSE_ARC_NONE      0000
  = TimeStamp:                00000000
  = InstPointer:              00000000
  = LastProcRetPtr:           00000000
  = FwOpState:                00
  = FwPowerState:             00
  = InOperational :           0
  = Reserved:                 00
- Log[3]:
  = ExceptionSource:          EXCEPTION_SOURCE_NONE      0000
  = ExceptionCause:           INT_CAUSE_ARC_NONE      0000
  = TimeStamp:                00000000
  = InstPointer:              00000000
  = LastProcRetPtr:           00000000
  = FwOpState:                00
  = FwPowerState:             00
  = InOperational :           0
  = Reserved:                 00
- Log[4]:
  = ExceptionSource:          EXCEPTION_SOURCE_NONE      0000
  = ExceptionCause:           INT_CAUSE_ARC_NONE      0000
  = TimeStamp:                00000000
  = InstPointer:              00000000
  = LastProcRetPtr:           00000000
  = FwOpState:                00
  = FwPowerState:             00
  = InOperational :           0
  = Reserved:                 00
- ExceptionLogCurrent:        00

SUSRAM_ID_HCI: id = 000018, len = 03
HciData:                      00000001
- HeciSusramEopSet:           1

SUSRAM_ID_ICC_OCKEN_TO_RESUME_FROM_S3: id = 00001F, len = 05
Value:                        00000000
Mask:                         00000000
CheckValue:                   0

SUSRAM_ID_NM_HEALTH_FLAGS: id = 000040, len = 03
NmHealthFlags:
- NoResponseBmcColdReset:     0
- InitateSystemShutdown:      0
- Reserved:                   00000000

SUSRAM_ID_RECOVERY_FW_UPDATE_NONCE_STATUS: id = 000203, len = 03
NonceStatus:
- NonceRetrieved:             1
- Reserved:                   0000

SUSRAM_ID_FW_RESET_COUNT: id = 000021, len = 03
MeRstCount:                   00000016

SUSRAM_ID_LAST: id = FFFFFD, len = 01
-= [Susram Parse] report generation end:   13/09/17 14:48:37.738 =-
```

C.5 Intel ME configuration Basic Partition

```
-= [ME configuration Basic Partition] report generation start: 13/09/17 14:51:51.629 =-
Nr     Logical Partition     File Name     File Name ASCII     Attributes      Size
\\0                 10        564C42                 VLB           0000        4
\\1                 16        46454C                 FEL           0000        3
\\2                 16        53454C                 SEL           0000        3
\\3                 18        494E46                 INF           0000       15
\\4                 21        464E46                 FNF           0000       15
\\5                 19        465231                 FR1           0000        8
\\6                 19        465232                 FR2           0000       72
\\7                 19        465233                 FR3           0000      256
\\8                 19        465234                 FR4           0000      384
\\9                 19        465235                 FR5           0000      384
\\10                17        534F4C                 SOL           0000       12
\\11               255        000100     \0x00\0x01\0x00           0000      216
\\12                10        534B55                 SKU           0001        4
```

```
\\13                14         54524C                  TRL           0001      288
\\14                10         564453                  VDS           0001      255
\\15                14         504E00              PN\0x00           0001        1
\\16                14         4E5050                  NPP           0001       66
\\17                10         4F4353                  OCS           0001      175
\\18                10         4F4350                  OCP           0001     2596
\\19                24         434647                  CFG           0001        4
\\20                11         505257                  PRW           0001        2
\\21                11         485052                  HPR           0001        1
\\22                16         4D4354                  MCT           0001        4
\\23                16         4D4842                  MHB           0001        4
\\24                14         4E4347                  NCG           0001       24
\\25                14         4D5343                  MSC           0001        8
\\26                14         505052                  PPR           0001       20
\\27                14         475043                  GPC           0001        7
\\28                14         504400              PD\0x00           0001       36
\\29                14         504401              PD\0x01           0001       36
\\30                14         504402              PD\0x02           0001       36
\\31                14         504403              PD\0x03           0001       36
\\32                14         504404              PD\0x04           0001       36
\\33                14         504405              PD\0x05           0001       36
\\34                14         504406              PD\0x06           0001       36
\\35                14         504407              PD\0x07           0001       36
\\36                25         444450                  DDP           0001        1
\\37                25         484352                  HCR           0001        8
\\38                16         4D4843                  MHC           0001        1
\\39                16         435053                  CPS           0001       28
\\40                16         43504F                  CPO           0001       13
\\41                16         534F4C                  SOL           0001        2
\\42                16         4C4446                  LDF           0001        1
\\43                15         505443                  PTC           0001       64
\\44                15         465331                  FS1           0001        8
\\45                15         465000              FP\0x00           0001        8
\\46                15         465001              FP\0x01           0001        8
\\47                15         465002              FP\0x02           0001        8
\\48                15         465003              FP\0x03           0001        8
\\49                15         465700              FW\0x00           0001       68
\\50                15         465701              FW\0x01           0001       68
\\51                15         465702              FW\0x02           0001       68
\\52                15         465703              FW\0x03           0001       68
\\53                15         465704              FW\0x04           0001       68
\\54                15         465705              FW\0x05           0001       68
\\55                15         464300              FC\0x00           0001       16
\\56                15         464301              FC\0x01           0001       16
\\57                15         464302              FC\0x02           0001       16
\\58                15         464303              FC\0x03           0001       16
\\59                15         464304              FC\0x04           0001       16
…
```