System Tools - Intel® Management Engine Firmware 12.0

User Guide

February 2019

Revision 1.51

Intel Confidential

Contents

Figures

Tables

Revision History

Revision Number	Description	Date
0.5	• Initial release	April 2016
0.6	• Updated MEInfo output examples	October 2016
0.61	• Updated FPT command line option information.	November 2016
0.7	• Removed ISH Functionalities from MEInfo and MEManuf • Removed NFC References	March 2017
0.8	• Updated Build Settings Image • Replaced MEU tool usage with reference to Manifesting and Signing Guide • Small Fixes • Updates to FIT section • Updates to MEInfo section	May 2017
0.81	• Updates to Supported OS in various sections • Updates to FPT section • MeManuf – BIST runs regardless of power source • MeInfo –feat supports column name • Update to FWUpdLcl.exe requirements	June 2017
0.9	• Updated OS Table • Removed redundant tool usage information	June 2017
0.92	• Improved documentation for FPT -IN and -MASTERACCESSGEN • Updated example for FPT -cfggen • Added details for SPI software binding (PCH replacement)	August 2017
0.93	• Updated source for LZMA	September 2017
1.0	• Updated error codes appendix • Updated MeInfo section	December 2017
1.1	• Added -ALL command to MEManuf/MEManuf win table • Added a note for PKI DNS Suffix to indicate dots location within the sting along with an example	January 2018
1.2	• Add details for -ALL command under Chapter 5.3, "Usage" • Added a note clarifying Privacy/Security Level Default Setting under Appendix A, "Intel® ME NVARs" • Update Table 6-2, "List of Components that Intel® MEINFO Displays" with Touch relevant information • Added a new section under Chapter 3, "Setting the Intel® PMC Binary File" with information about adding the Intel® PMC binary file. • Added details about SVN ARB in relevant MEInfo, MEManuf, and FPT tools' sections	March 2018
1.3	• Add new tool Chapter 8, "UEFI Sample Application Leveraging FWUpdate API Library" • Updated Appendix with Appendix B.3	May 2018

Revision Number	Description	Date
1.4	• Removed VSCCCOMMN.bin reference from MEManuf chapter	November 2018
1.5	• Updated Appendix B with new system level error codes. • Added 2 new EOL tests in Chapter 5 “Intel® MEManuf” • Boot Guard Status • FW Status • Updated Chapter 8 “UEFI Sample Application Leveraging FWUpdate API Library” with new APIs and with RS mark on relevant Reduced Size APIs • Removed -ErrList command from Chapter 5 “Intel® MEManuf” • Update Appendix A – Intel® NVARs. eDP Port and LSPCON Port Config NVARs need only ME reset type. • Removed MEBx password protection requirement from Chapter 7 Intel® ME Firmware Update.	November 2018
1.51	• removed “FWUpdLcl -generic” command from FW Update Tool.	February 2019

§ §

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
AMT	Intel® AMT
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
BIST	Built In Self-Test
CCM	Client Control Mode (Host Based Setup and Configuration)
CLI	Command Line Interface
CM0	Intel® ME power state where all HW power planes are activated. Host power state is S0.
CM1	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 PCI-E* interface is unavailable to the host SW. This power state is not available in Cougar Point.
CM3	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 PCI-E* interface is unavailable to the host SW. The main memory is not available for Intel® ME use.
CM-Off	No power is applied to the management processor subsystem. Intel® ME is shut down.
CRB	Customer Reference Board
DHCP	Dynamic Host Configuration Protocol

Acronym/Term	Definition
DIMM	Dual In-line Memory Module
DLL	Dynamic Link Library
DNS	Domain Naming System
EC	Embedded Controller
EEPROM	Electrically Erasable Programmable Read Only Memory
EFI	Extensible Firmware Interface
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
FLOCKDN	Flash Configuration Lock-Down
FMBA	Flash Master Base Address
FOV	Fixed Offset Variable
FPSBA	Flash PCH Strap Base Address
FPT	Flash Programming Table
FQDN	Fully Qualified Domain Name
FRBA	Flash Region Base Address
FTP	Fault Tolerant Partition
Full Image	A full image starts with an FPT and contains FTP and NFTP partitions
Full Update	Updates all the regions
FW	Firmware
FWUpdate	Firmware Update
FWUpdateLib	Firmware Update Library
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
GPIO	General Purpose Input/output
GUI	Graphical User Interface

Acronym/Term	Definition
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	Hyper Text 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 and Play feature. When installing a driver, this file provides the OS with the necessary information about driver filenames, driver components, and supported hardware.
Intel® AMT	The Intel® AMT Firmware running on the embedded processor
Intel® DAL	Intel® Dynamic Application Loader (Intel® DAL)
Intel® FIT	Intel® Flash Image Tool
Intel® FPT	Intel® Flash Programming Tool
Intel® ME	Intel® Management Engine. The embedded processor residing in the chipset PCH.
Intel® MEBx	Intel® Management Engine BIOS Extensions
Intel® MEI driver	Intel® AMT host driver that runs on the host and interfaces between ISV Agent and the Intel® AMT HW.
Intel® MEINFO	Intel® Manageability Engine Information Tool to check whether ME is alive or not.
Intel® MEInfoWin	Windows® version of Intel® Manageability Engine Information Tool
Intel® MEManuf	Intel® Manageability Engine Manufacturing Tool validates Intel® ME functionality on the manufacturing line
Intel® MEManufWin	Windows® version of Intel® Manageability Engine Manufacturing Tool
ISV	Independent Software Vendor

Acronym/Term	Definition
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
LOCL	Localization Language
LMS	Local Management Service. An SW application which runs on the host machine and provides a secured communication between the ISV agent and the Intel® Management Engine Firmware.
LPC	Low Pin Count Bus
MAC address	Media Access Control address
MCP	Multi-Chip Package (Central Processing Unit / Platform Controller Hub)
NFTP	Non-Fault Tolerant Partition
NM	Number of Masters
NVAR	Named Variable
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
DIMM	Dual In-line Memory Module
DLL	Dynamic Link Library
DNS	Domain Naming System
EC	Embedded Controller
EEPROM	Electrically Erasable Programmable Read Only Memory
EFI	Extensible Firmware Interface
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
FLOCKDN	Flash Configuration Lock-Down
FMBA	Flash Master Base Address
FOV	Fixed Offset Variable
FPSBA	Flash PCH Strap Base Address
FPT	Flash Programming Table
FQDN	Fully Qualified Domain Name
FRBA	Flash Region Base Address
FTP	Fault Tolerant Partition
Full Image	A full image starts with an FPT and contains FTP and NFTP partitions
Full Update	Updates all the regions
FW	Firmware
FWUpdate	Firmware Update
FWUpdateLib	Firmware Update Library
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
GPIO	General Purpose Input/output
GUI	Graphical User Interface

Acronym/Term	Definition
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	Hyper Text 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 and Play feature. When installing a driver, this file provides the OS with the necessary information about driver filenames, driver components, and supported hardware.
Intel® AMT	The Intel® AMT Firmware running on the embedded processor
Intel® DAL	Intel® Dynamic Application Loader (Intel® DAL)
Intel® FIT	Intel® Flash Image Tool
Intel® FPT	Intel® Flash Programming Tool
Intel® ME	Intel® Management Engine. The embedded processor residing in the chipset PCH.
Intel® MEBx	Intel® Management Engine BIOS Extensions
Intel® MEI driver	Intel® AMT host driver that runs on the host and interfaces between ISV Agent and the Intel® AMT HW.
Intel® MEINFO	Intel® Manageability Engine Information Tool to check whether ME is alive or not.
Intel® MEInfoWin	Windows® version of Intel® Manageability Engine Information Tool
Intel® MEManuf	Intel® Manageability Engine Manufacturing Tool validates Intel® ME functionality on the manufacturing line
Intel® MEManufWin	Windows® version of Intel® Manageability Engine Manufacturing Tool
ISV	Independent Software Vendor

Acronym/Term	**Definition**
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
LOCL	Localization Language
LMS	Local Management Service. An SW application which runs on the host machine and provides a secured communication between the ISV agent and the Intel® Management Engine Firmware.
LPC	Low Pin Count Bus
MAC address	Media Access Control address
MCP	Multi-Chip Package (Central Processing Unit / Platform Controller Hub)
NFTP	Non-Fault Tolerant Partition
NM	Number of Masters
NVAR	Named Variable
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
PAVP	Protected Video and Audio Path
Partial Image	A partial image starts with either WCOD or LOCL partitions. No FPT, FTO, and NFTP in the file
Partial Update	Only updates regions that require an Update such as WCOD or LOCL
PC	Personal Computer
PCH	Peripheral 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
RCFG	Remote Configuration
RCS	Remote Connectivity Service
RNG	Random Number Generator
ROM	Read Only Memory
RPAS	Remote Connectivity Service
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 states 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.
SEBP	Single Ended Buffer Parameters
SHA	Secure Hash Algorithm
SMB	Small Medium Business mode

Acronym/Term	Definition
SMBus	System Management Bus
Snooze mode	Intel® ME activities are mostly suspended to save power. Intel® ME monitors HW activities and can restore its activities depending on the HW event.
SOAP	Simple Object Access Protocol
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.
SW	Software
Sx	All S states which are different than S0
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
UEP	Unified Emulation Partition
UI	User Interface
UIM	User Identifiable Mark
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
UPDPARAM	Update Parameter Tool
USB	Universal Serial Bus
USBr	Universal Serial Bus Redirection
UUID	Universally Unique Identifier
VLAN	Virtual Local Area Network
VSCC	Vendor Specific Component Capabilities
WCOD	Wireless Card Device
Windows® PE	Windows® Pre installation Environment
WIP	Work in Progress
WLAN	Wireless Local Area Network

Acronym/Term	Definition
XML	Extensible Markup Language. Intel® AMT's XML-based protocol has 3 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.
ZTC	Zero Touch Configuration
ARB SVN	Anti Rollback Security Version Number

1.2 Reference Documents

Document	Document No./Location
FW Bring Up Guide	Included in released Kits
Firmware Variable Structures for Intel® Management Engine and Intel® Active Management Technology 12.0	CDI document
Cannon Lake PCH External Design Specification - EDS	CNL-H Volume 1: CDI# 571182 CNL-H Volume 2: CDI# 572235 CNL-LP Volume 1: CDI# 566439 CNL-LP Volume 2: CDI# 565870
Cannon Lake PCH-LP SPI Programming Guide	Included in released Kits

§ §

2 Preface

2.1 Overview

This document covers the system tools used for creating, modifying, and writing binary image files, manufacturing testing, Intel® ME setting information gathering, and Intel® ME FW updating. The tools are located in **Kit directory\Tools\System tools**. For information about other tools, refer Tool's user guides in the other directories in the FW release.

The system tools described in this document are platform specific in the following ways:

- Cannon LakePCH platform – All of the tools in the Cannon Lake PCH FW release kit are designed for 8th Generation Intel® Core™ Processors and Cannon Lake PCH platforms only. These tools do not work properly on any other legacy platforms (prior Generations of Intel® Core™ Processors). Tools designed for other platforms also do not work properly on the 8th Generation Intel® Core™ Processors or the Cannon Lake PCH platform.
- Intel® vPro™ platform – All features listed in this document are available for Intel® vPro™ platforms with Intel® ME FW 12.0. There are some features that are specifically designed for the Intel® vPro™ platform and only work on it.
- Intel® ME Firmware 12.0 SKU – A common set of tools are provided for the following Intel® ME FW 12.0 SKUs: Consumer Intel® ME FW SKU and Corporate Intel® ME FW SKU. The following features are only available for Corporate Intel® ME FW SKUs and Consumer Intel® ME FW SKU users should generally ignore them:

  Intel® AMT

  Intel® ME BIOS Extension (Intel® MEBx)

  The description of each tool command or option that is not available for Consumer Intel® ME FW SKU contains a note indicating this.

  - Note: For LBG, Non-POR features are WLAN and PTT.

2.2 Image Editing Tools

The following tools create and write flash images:

- Intel® FIT
  Combines the Descriptor, GbE, BIOS, PDR, ISH and Intel® ME FW binaries into one image.
  Configures soft straps and NVARs for Intel® ME settings and another for outputs

that can be programmed by a flash programming device or the FPT Tool.

- FPT:
  Programs the SPI flash memory of individual regions or the entire flash device. Modifies some Intel® ME settings (NVAR), FPFs after Intel® ME is flashed on the SPI part.
- FWUpdate – updates the Intel® ME FW code region on a flash device that has already been programmed with a complete image.

Note: The firmware update tool provided by Intel only works on the platforms that support the FWUpdate feature.

2.3 Manufacturing Line Validation Tool

The manufacturing line validation tool (Intel® MEManuf) allows the Intel® ME and Intel® AMT functionality to be tested immediately after the PCH chipset is generated. This tools is designed to be able to run quickly and is generally run on the manufacturing line to do manufacturing testing.

2.4 Intel® Management Engine Setting Checker Tool

The Intel® ME setting checker tool (Intel® MEInfo) retrieves and displays information about some of the Intel® ME settings, the Intel® ME FW version, and the FW capability on the platform.

2.5 Operating System Support

Table 2-1. OS Support for Tools

Intel® ME and Manufacturing Tools	Free DOS	UEFI (64 bit)	Windows® 10 DT 64 bit	OSX® (El Capitan / Yosemite)	Windows PE for Windows 10	Ubuntu 16.04.3 LTS (64 Bit)
Intel® Flash Programing Tool	X	X	X		x	X
Intel® MEManuf Tool	X	X	X		x	X
Intel® ME Info Tool	X	X	X		x	X
Intel® Firmware Update Tool	X	X	X		x	X
Intel® Manifest Extension Utility Tool			X	x		
Intel® Flash Image Tool			X	x		
ICC CCT Tool	X	X			x	

Notes:

1. 64 bit support may NOT mean that a tool is compiled as a 64 bit application – but that it can run as a 32 bit application on a 64 bit platform.

2. The Windows® 64 bit tools will not function when the OS is configured to use EFI / GPT boot capabilities.

3. ISH is not supported on MEInfo/ MEManuf for Linux or UEFI. Also, a separate ISH tool must be used where functionalities are ported from MEInfo and MEManuf tool.

4. Currently the System Tools use the EDK II Development Kit exclusively.

2.6 Generic System Requirements

The installation of the following services is required by integration validation tools that run locally on the system under test with the Intel® Manageability Engine:

- Intel® MEI driver.
- Intel® AMT LMS – not applicable to Consumer Intel® ME FW SKU.

Refer the description of each tool for its exact requirements.

Table 2-2. Tools Summary

ToolName	Feature Tested	Runs on Intel® ME device
Intel® MEManuf and Intel® MEManufWin	Connectivity between Intel® ME Devices	X
Intel® MEInfo and Intel® MEInfoWin	Firmware Aliveness – outputs certain Intel® ME parameters	X
Intel® FPT	Programs the image onto the flash memory and Programming NVARs / FPFPs	X
Intel® FWUpdate	Updates the FW code while maintaining the previously set values	X

2.7 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. (Refer to Appendix B for a list of these error codes.)
For Intel® MEManuf tool, there is error level 2 which indicates Success with Warnings.

2.8 Usage of Double-Quote Character (")

The EFI version of the tools handle multi-word argument differently than the DOS/ Windows® version. If there is a single argument that consists of multiple words delimitated by spaces, the argument needs to be entered as following:

FPT.efi –f “” Wlan well power config ””.

The command shell used to invoke the tools in EFI, DOS and Windows® has a built-in CLI.

The command shell was 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, the user 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 the user wants these words to be input – input"string – the command line is: input\"string.

2.9 PMX Driver Limitation

Several tools (Intel® MEInfo and Intel® FPT) 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 (and higher), 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 (and higher) driver verifier turned on causes the OS to crash. Do not include PMX as part of the verifier driver list if the user is running Windows® 7 (and higher) with the driver verifier turned on.

2.10 Control Handler Support

Intel® MEInfo and Intel® FPT and Intel® MEManuf support control handlers (Ctrl + C, Ctrl + Break, Ctrl + Close, etc.) for supported Microsoft Windows versions. When the control handlers are invoked, upon the following execution of the tools (after the 1st execution was aborted by the above control handlers), the tools will execute their regular flows.

§ §

3 Intel® Flash Image Tool

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

1. FIT creates and allows configuration of the Flash Descriptor Region, which contains configuration information for platform hardware and FW.
2. FIT assembles the following into a single image:

   Binary files of the following regions:

   - BIOS
   - Intel integrated LAN (GbE)
   - IFWI: Intel® ME and PMC
   - EC
   - Platform Descriptor Region
   - ISH

   The Flash Descriptor Region created by FIT
3. The user can manipulate the completed image via a GUI and change the various chipset parameters to match the target hardware. Various configurations can be saved to independent files, so the user does not have to recreate a new image each time.

FIT supports a set of command line parameters that can be used to build an image from the CLI or from a makefile. When a previously stored configuration is used to define the image layout, the user does not have to interact with the GUI.

Note: FIT just generates a complete image file; it does not program the flash device. This complete image must be programmed into the flash with FPT any third-party flash burning tool, or some other flash burner device.

3.1 System Requirements

Intel® FIT runs on Microsoft Windows® 10. The tool does not have to run on an Intel® ME-enabled system.

3.2 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 3-1. SPI Flash Image Regions

Descriptor	IFWI: Intel® ME amd PMC Intel® ME Applications	EC	GbE	PDR	BIOS

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. Refer to Section 3.4.10 below for more information. Failure to lock the Descriptor Region leaves the Intel® ME device vulnerable to security attacks.
Ifwi: Intel® ME and PMC	This region contains code and configuration data for Intel® ME applications, such as Intel® AMT technology. It takes up a variable amount of space at the end of the Descriptor.
EC	This contains the Embedded Controller binary used for eSPI.
GbE	This region contains code and configuration data for an Intel Integrated LAN (Gigabit Ethernet). It takes up a variable amount of space at the end of the Intel® ME region.
BIOS	This region contains code and configuration data for the entire computer.
PDR	This region lets system manufacturers describe custom features for the platform.

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

- vsccommn.bin
- .xml file

3.4 Intel® Flash Image Tool

Refer following for further information:

- General configuration information – Refer FW Bring Up Guide from the appropriate Intel® ME FW kit.
- Detailed information on how to configure PCH Soft Straps and VSCC information – Refer to the Cannon Lake PCH SPI Programming Guide and to the C620 Lewisburg platforms refer LBG SPI Programming Guide within the kit.

3.4.1 Configuration Files

The flash image can be configured in many different ways, depending on the target hardware and the required FW options. FIT lets the user 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.

3.4.2 Creating New Configuration

FIT provides a XML configuration file template that will help the user create their own configuration XML. This template configuration XML file can be created by clicking **File** > **New and then save**. It can also be created from the command line using –save option.

3.4.3 Opening Existing Configuration

To open an existing configuration file:

1. Choose File → **Open**; **Open File** dialog appears.
2. Select the XML file to load.
3. Click Open.

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

3.4.4 Saving Configuration

To save the current configuration in an XML file:

Choose File → **Save** 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.

1. Select the path and enter the file name for the configuration.

2. Click Save.

3.4.5 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. The user can set the environment variables appropriate for the platform being used, or override the variables with command line options.

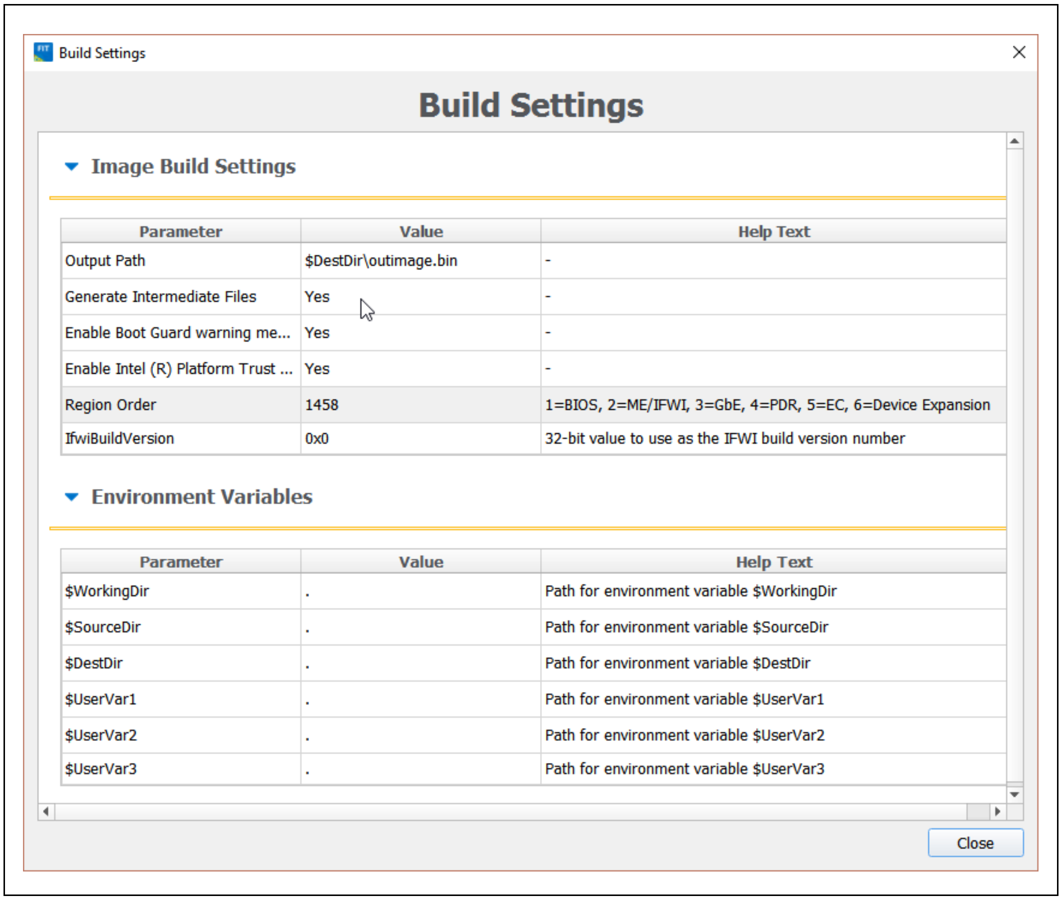
It is recommended that the environment variables be the first thing that the user sets when working with a new configuration. This ensures that FIT can properly substitute environment variables into paths to keep them relative. Doing this also speeds up configuration because many of the **Open File**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

To modify the environment variables:

1. Choose Build → **Build Settings**; a dialog appears displaying the current working directory on top, followed by the current values of all the environment variables:

- $WorkingDir – the directory functions as a basic path variable when modified in the GUI. If $WorkingDir CLI flag is used when launching FIT GUI, then the fit.log will be created in $WorkingDir directory.
- $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 intermediate files generated during the build. Also the directory where the components of an image are stored when an image is decomposed.
- $UserVar1-3 – used when the above variables are not populated.

Figure 3-2. Environment Variables Dialog



Build Settings

Image Build Settings

Parameter	Value	Help Text
Output Path	$DestDir\outimage.bin	-
Generate Intermediate Files	Yes	-
Enable Boot Guard warning me...	Yes	-
Enable Intel (R) Platform Trust ...	Yes	-
Region Order	1458	1=BIOS, 2=ME/IFWI, 3=GbE, 4=PDR, 5=EC, 6=Device Expansion
IfwiBuildVersion	0x0	32-bit value to use as the IFWI build version number

Environment Variables

Parameter	Value	Help Text
$WorkingDir	.	Path for environment variable $WorkingDir
$SourceDir	.	Path for environment variable $SourceDir
$DestDir	.	Path for environment variable $DestDir
$UserVar1	.	Path for environment variable $UserVar1
$UserVar2	.	Path for environment variable $UserVar2
$UserVar3	.	Path for environment variable $UserVar3

---

2. Press 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
5. **OK**.

Note: The environment variables are saved in the XML file. They can be overridden on the command line if using the XML file on multiple systems.

Note: Build Settings

FIT lets the user set several options that control how the image is built. The options that can be modified are described in Build Settings Dialog Options.

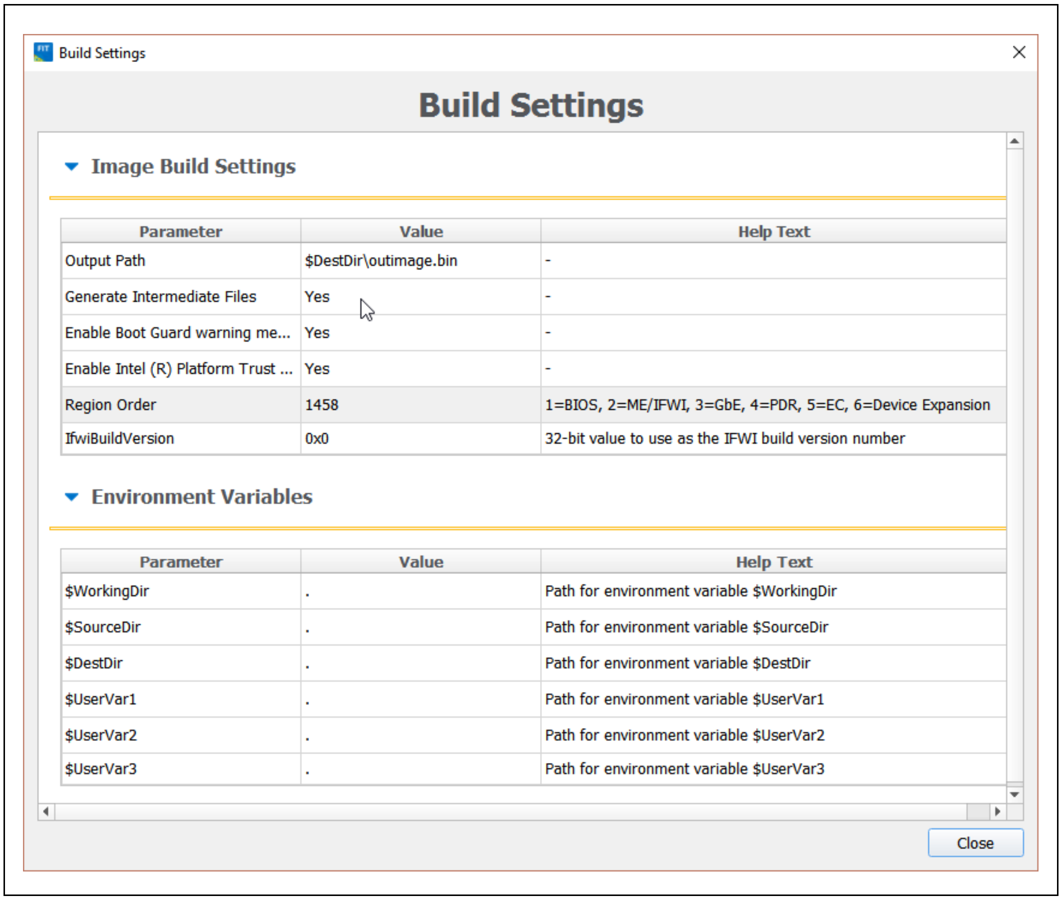
To modify the build setting:

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 (Refer Figure 3). These files are located in the specified output folder's INT subfolder. These image files can be programmed individually with the FPT.
Enable Boot Guard Warning message at build time.	Allows to enable boot guard warning messages at the build time.
Enable Intel® Platform Trust Technology messages at build time.	Allows to enable Intel® Platform Trust Technology warning messages at the build time
CPU Stepping	Which CPU stepping to use.
Environment Variables	

Figure 3-3. Build Settings Dialog



Build Settings

Image Build Settings

Parameter	Value	Help Text
Output Path	$DestDir\outimage.bin	-
Generate Intermediate Files	Yes	-
Enable Boot Guard warning me...	Yes	-
Enable Intel (R) Platform Trust ...	Yes	-
Region Order	1458	1=BIOS, 2=ME/IFWI, 3=GbE, 4=PDR, 5=EC, 6=Device Expansion
IfwiBuildVersion	0x0	32-bit value to use as the IFWI build version number

Environment Variables

Parameter	Value	Help Text
$WorkingDir	.	Path for environment variable $WorkingDir
$SourceDir	.	Path for environment variable $SourceDir
$DestDir	.	Path for environment variable $DestDir
$UserVar1	.	Path for environment variable $UserVar1
$UserVar2	.	Path for environment variable $UserVar2
$UserVar3	.	Path for environment variable $UserVar3

3.4.6 Modifying the Flash Descriptor Region

The Flash Descriptor Region 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 computer may not function as expected. This region also needs to be configured correctly in order to ensure that the system is secure.

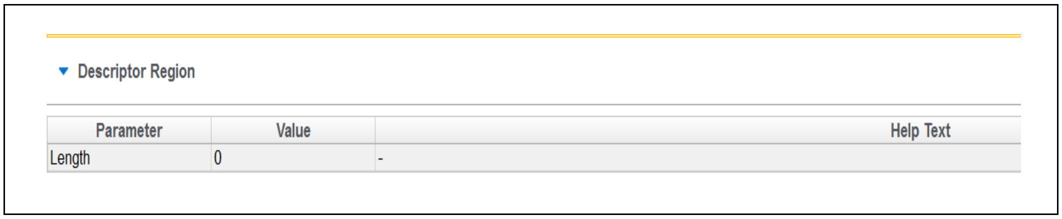
3.4.7 Descriptor Region Length

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

To set the value of the Descriptor Region Length parameter:

1. Select **Flash Layout** in the left pane; the **Length** parameter appears in the right pane.
2. 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



Descriptor Region

Parameter	Value	Help Text
Length	0	-

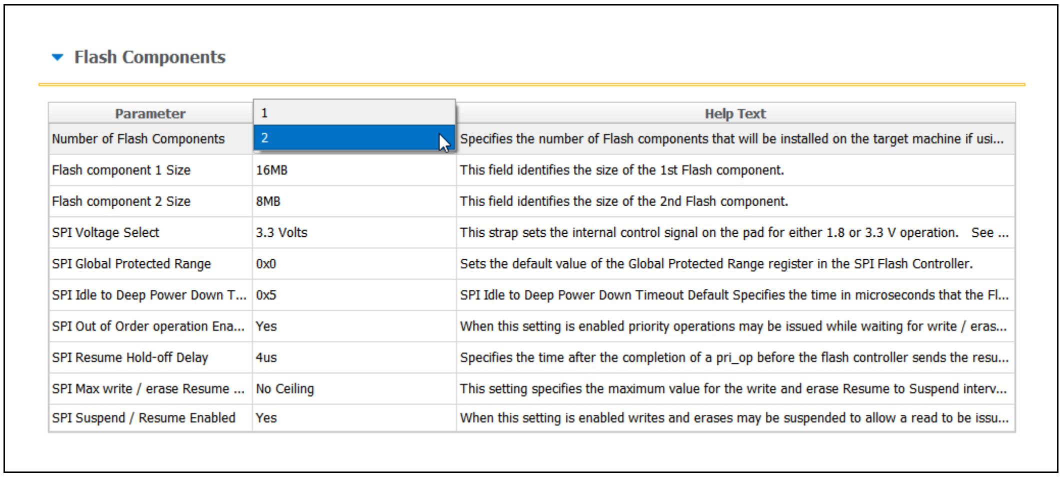
3.4.8 Setting the Number and Size of the Flash Components

To set the number of flash components:

1. Select **Flash Settings** in the left pane; expand the Flash Component node in the right pane.

   Refer to Figure 3-5, the parameters in the Flash Component section are listed in the right pane.

Figure 3-5. Flash Settings > Flash Components



Flash Components

Parameter	1	Help Text
Number of Flash Components	2	Specifies the number of Flash components that will be installed on the target machine if usi...
Flash component 1 Size	16MB	This field identifies the size of the 1st Flash component.
Flash component 2 Size	8MB	This field identifies the size of the 2nd Flash component.
SPI Voltage Select	3.3 Volts	This strap sets the internal control signal on the pad for either 1.8 or 3.3 V operation. See ...
SPI Global Protected Range	0x0	Sets the default value of the Global Protected Range register in the SPI Flash Controller.
SPI Idle to Deep Power Down T...	0x5	SPI Idle to Deep Power Down Timeout Default Specifies the time in microseconds that the Fl...
SPI Out of Order operation Ena...	Yes	When this setting is enabled priority operations may be issued while waiting for write / eras...
SPI Resume Hold-off Delay	4us	Specifies the time after the completion of a pri_op before the flash controller sends the resu...
SPI Max write / erase Resume ...	No Ceiling	This setting specifies the maximum value for the write and erase Resume to Suspend interv...
SPI Suspend / Resume Enabled	Yes	When this setting is enabled writes and erases may be suspended to allow a read to be issu...

2. Double-click the value of **Number of Flash Components** in the right pane (Figure 3-5)
3. Select the number of flash components (valid values are 1 or 2) from the dropdown.

To set the size of each flash component:

1. Double-click on the value of one of these parameters Flash Component 1 Size / Flash Component 2 Size.
2. Select the correct component size from the drop-down list; that parameter is updated.
3. Repeat steps 2-3 for the other parameter.

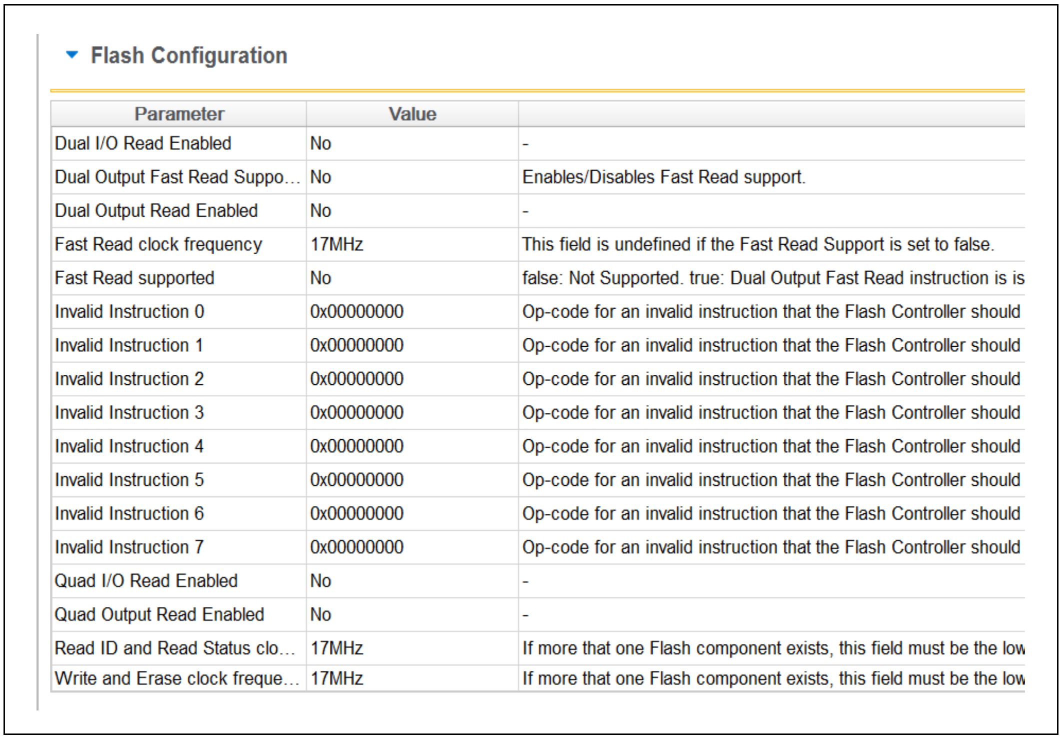
Note: The size of the second flash component is only editable if the number of flash components is set to 2.

3.4.9 SPI Software Binding (PCH Replacement)

When enabled, the Flash Componenets “SPI Software Binding Enabled” parameter will allow for SPI re-binding to a new PCH during manufacturing and remanufacturing flows prior to platform EOM.

Note: Note: Re-binding to a replacement PCH can only be done a maximum of 5 times before the SPI part needs to be re-flashed. The replacement counter is exposed in the PCH section of MEInfo.

Figure 3-6. Flash Settings > Flash Configuration



Flash Configuration

Parameter	Value	
Dual I/O Read Enabled	No	-
Dual Output Fast Read Suppo…	No	Enables/Disables Fast Read support.
Dual Output Read Enabled	No	-
Fast Read clock frequency	17MHz	This field is undefined if the Fast Read Support is set to false.
Fast Read supported	No	false: Not Supported. true: Dual Output Fast Read instruction is is
Invalid Instruction 0	0x00000000	Op-code for an invalid instruction that the Flash Controller should
Invalid Instruction 1	0x00000000	Op-code for an invalid instruction that the Flash Controller should
Invalid Instruction 2	0x00000000	Op-code for an invalid instruction that the Flash Controller should
Invalid Instruction 3	0x00000000	Op-code for an invalid instruction that the Flash Controller should
Invalid Instruction 4	0x00000000	Op-code for an invalid instruction that the Flash Controller should
Invalid Instruction 5	0x00000000	Op-code for an invalid instruction that the Flash Controller should
Invalid Instruction 6	0x00000000	Op-code for an invalid instruction that the Flash Controller should
Invalid Instruction 7	0x00000000	Op-code for an invalid instruction that the Flash Controller should
Quad I/O Read Enabled	No	-
Quad Output Read Enabled	No	-
Read ID and Read Status clo…	17MHz	If more that one Flash component exists, this field must be the low
Write and Erase clock freque…	17MHz	If more that one Flash component exists, this field must be the low

3.4.10 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

	Master Read/Write Access			
Region (#)	**CPU and BIOS**	**ME/PCH**	**GbE Controller**	**EC**
Descriptor (0)	Not Accessible	Not Accessible	Not Accessible	Not Accessible
BIOS (1)	CPU and BIOS can always read from and write to BIOS region	Read / Write	Read / Write	Read / Write
ME (2)	Read / Write	ME can always read from and write to ME region	Read / Write	Read / Write
GbE (3)	Read / Write	Read / Write	GbE software can always read from and write to GbE region	Read / Write
PDR (4)	Not Accessible	Not Accessible	Not Accessible	Not Accessible
EC - Embedded Controller (Optional) (8)	Read / Write	Read / Write	Read / Write	EC can always read from and write to EC region

NOTES:

1. Descriptor and PDR region is not a master, so they will not have Master R/W access.
2. Descriptor should NOT have write access by any master in production systems.
3. PDR region should only have read and/or write access by CPU/Host. GbE and ME should NOT have access to PDR region.

		Regions That Can Be Accessed					
		PDR	**Intel® ME**	**GbE**	**BIOS**	**IOSF Sideband Privileged Master**	**Descriptor**
Region to Grant Access	**Intel® ME**	None/Read/Write	None/Read/Write	Write only. 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	Write only. GbE can always read from and write to GbE Region.	None/Read/Write	None/Read/Write	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	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 FIT:

1. Select **Flash Settings Tab → Host CPU/BIOS Master Access, Intel ME Master Access, Gbe Master Access and EC Master Access** in the right 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:

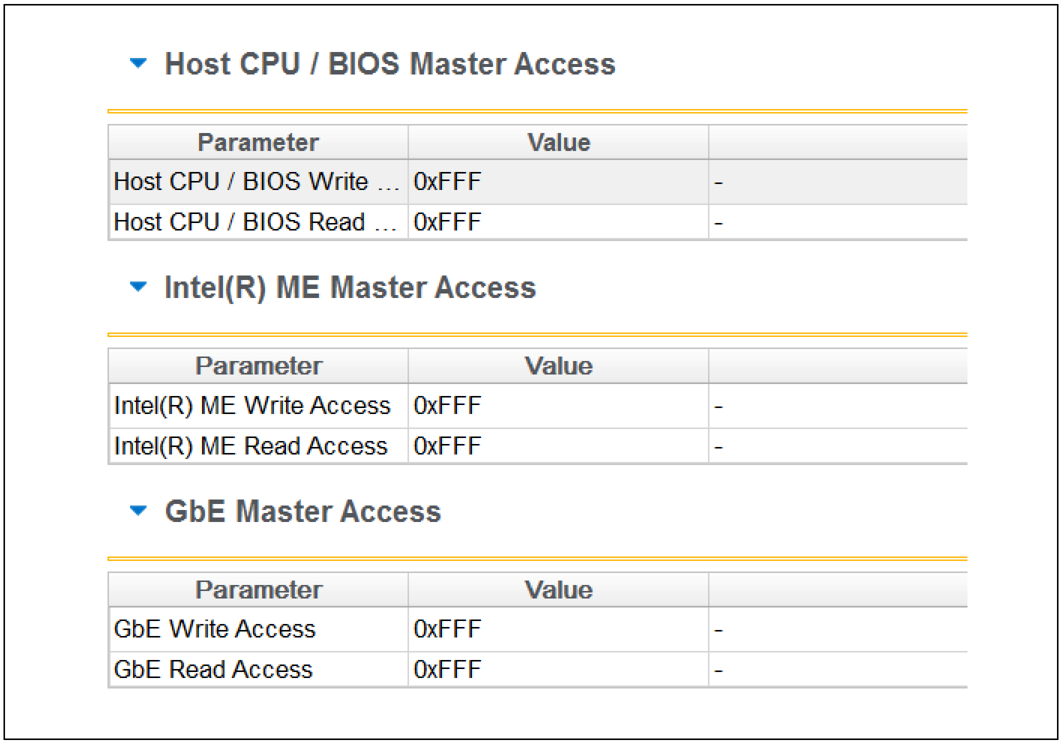
   To generate an image for debug purposes or to leave the SPI region open:
   select 0xFF for both read and write access in all the sections.

   To generate a production image with BIOS access to the PDR region select
   read access 0x00B / 0x01B and write access 0x00A / 0x01A.

Note: These settings should only be used if the PDR region is implemented.

To lock the SPI in the image creation phase: select the recommended settings for production (e.g., select 0x0C for Intel® ME read access and 0x0D for Intel® ME write access).

Figure 3-7. Descriptor Region Master Access Section



▼ Host CPU / BIOS Master Access

Parameter	Value	
Host CPU / BIOS Write ...	0xFFF	-
Host CPU / BIOS Read ...	0xFFF	-

▼ Intel(R) ME Master Access

Parameter	Value	
Intel(R) ME Write Access	0xFFF	-
Intel(R) ME Read Access	0xFFF	-

▼ GbE Master Access

Parameter	Value	
GbE Write Access	0xFFF	-
GbE Read Access	0xFFF	-

3.4.11 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 FPT. **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, refer to the Cannon Lake PCH-LPSPI Programming Guide, Section 6.4.) and For Lewisburg C620 family platform, refer LBG SPI Programming Guide, Section 4.4.)

VSCC Table can be accessed:

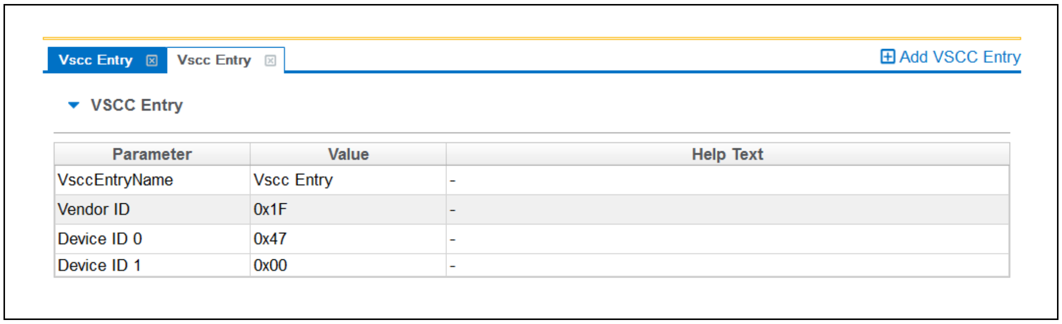
1. Select Flash Settings Tab on the left pan
2. Expand VSCC Entries on the right pan as shown below in Figure 3-7:

3.4.12 Adding New Table

To add a new table:

1. Choose ⊞ Add VSCC Entry **on top left → VSCC Entry**.

Figure 3-8. Add VSCC Table Entry Dialog



Parameter	Value	Help Text
VsccEntryName	Vscc Entry	-
Vendor ID	0x1F	-
Device ID 0	0x47	-
Device ID 1	0x00	-

1. 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 FIT to prevent table entries that have the same name and no error message is displayed in such cases.

2. User can enter into the values for the flash device. (Figure 3-7), which shows the parameters of a new VSCC table.)

Note: The VSCC register value will be automatically populated by FIT using the vsccommn.bin file the appropriate information for the Vendor and Device ID.

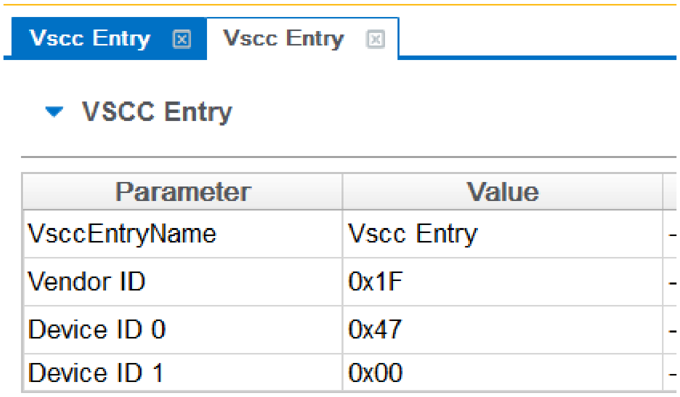
Note: If the descriptor region is being built manually the user will need to reference the VSCC table information for the parts being supported from the manufacturers' serial flash data sheet. The Cannon Lake PCH-LP SPI Programming Guide should be used to calculate the VSSC values. For C620 family of workstation systems, use the LBG SPI Programming Guide for further reference concerning the VSCC table definitions.

3.4.13 Removing Existing VSCC Table

To remove an existing table:

1. Click on the name of the table in the top tab that the user wants to remove.

Figure 3-9. Deleting VSCC Table Entry Dialog



Parameter	Value	
VsccEntryName	Vscc Entry	-
Vendor ID	0x1F	-
Device ID 0	0x47	-
Device ID 1	0x00	-

2. Click close, the table and all of the information will be removed.

3.4.14 FPF Configuration

The “FPF Hardware Binding Enabled” setting configures the FPF hardware binding behavior for the platform image.

For non-revenue parts:

If the “FPF Hardware Binding Enabled” setting is enabled
Hardware binding will occur when the close manufacturing flow is executed.

If the “FPF Hardware Binding Enabled” setting is disabled
Hardware binding will not occur when the close manufacturing flow is executed.

Note: ***For Revenue parts this setting will be ignored and FPF Hardware binding will take place when close manufacturing flow is executed.***

3.4.15 Modifying the Intel® Management Engine Region

The Intel® ME Region contains all of the FW data for the Intel® ME (including the Intel® ME FW Kernel).

Note:Changing the Intel® ME Region will prompt the user and require the users to reset parameters in Intel® FIT.

3.4.16 Setting the Intel® Management Engine Region Binary File

To select the Intel® ME region binary file:

1. Select the Intel® ME and PMC Region available under Flash Layout tab on the left pane.

2. Double-click on the **Intel® ME Binary file parameter** in the list; select the Intel® ME 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.

3.4.17 Setting the Intel® PMC Binary File

To select the Intel® PMC binary file:

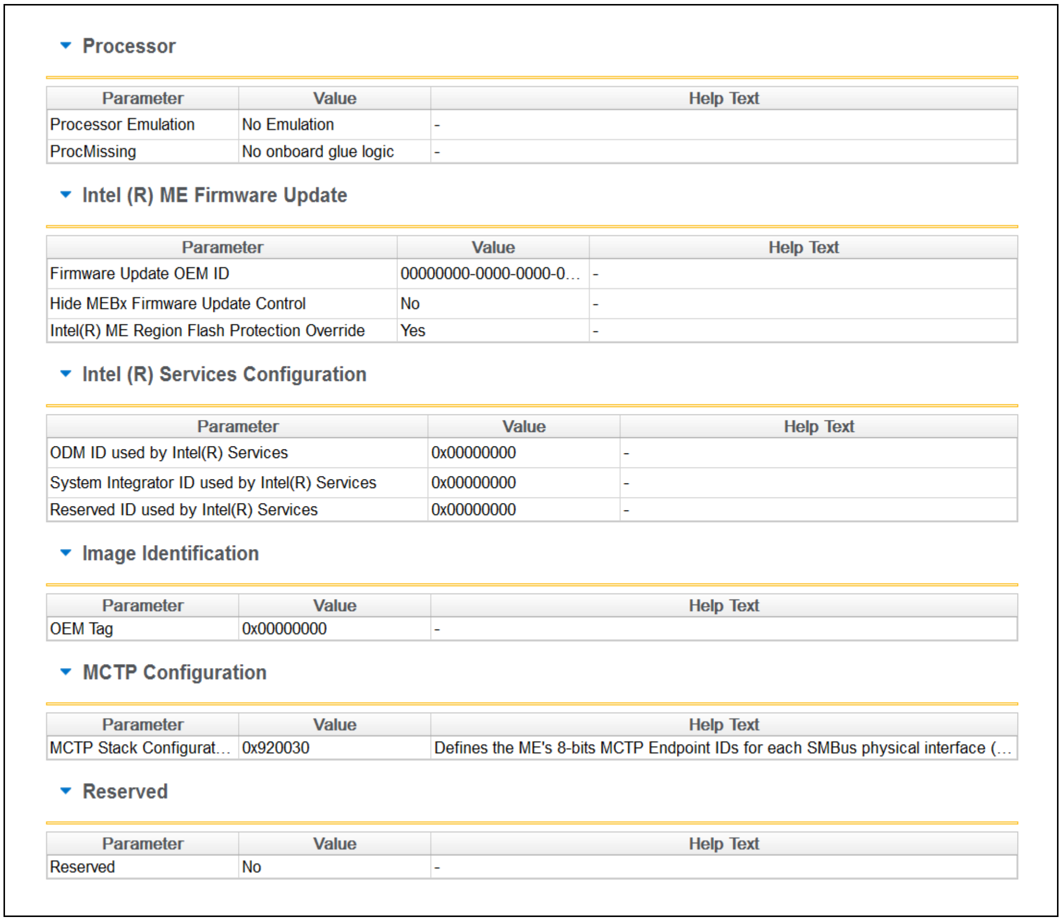
1. Select the Intel® ME & PMC Region available under Flash Layout tab on the left pane.
2. Double-click on the **PMC Binary file parameter** in the list; select the Intel® PMC file to be used.
3. Click **OK** to update the parameter; when the flash image is built, the contents of this file will be merged into the output image generate by the Intel® FIT tool.

Note: Intel FIT tool would return a build error in case wrong PMC binary is selected for stitching.

3.4.18 Intel® Management Engine Section

This section describes Intel® ME FW Kernel parameters. (Refer FW Bringup guide for general information and refer Appendix for more details.)
Click on the Intel® ME Kernel Tab on the left pane to configure the Intel® ME parameters. The parameter values can be found in the Help Text next to the parameter value as shown in Figure 3-9.

Figure 3-10. Intel® ME Kernel



Processor

Parameter	Value	Help Text
Processor Emulation	No Emulation	-
ProcMissing	No onboard glue logic	-

Intel (R) ME Firmware Update

Parameter	Value	Help Text
Firmware Update OEM ID	00000000-0000-0000-0…	-
Hide MEBx Firmware Update Control	No	-
Intel(R) ME Region Flash Protection Override	Yes	-

Intel (R) Services Configuration

Parameter	Value	Help Text
ODM ID used by Intel(R) Services	0x00000000	-
System Integrator ID used by Intel(R) Services	0x00000000	-
Reserved ID used by Intel(R) Services	0x00000000	-

Image Identification

Parameter	Value	Help Text
OEM Tag	0x00000000	-

MCTP Configuration

Parameter	Value	Help Text
MCTP Stack Configurat…	0x920030	Defines the ME's 8-bits MCTP Endpoint IDs for each SMBus physical interface (…

Reserved

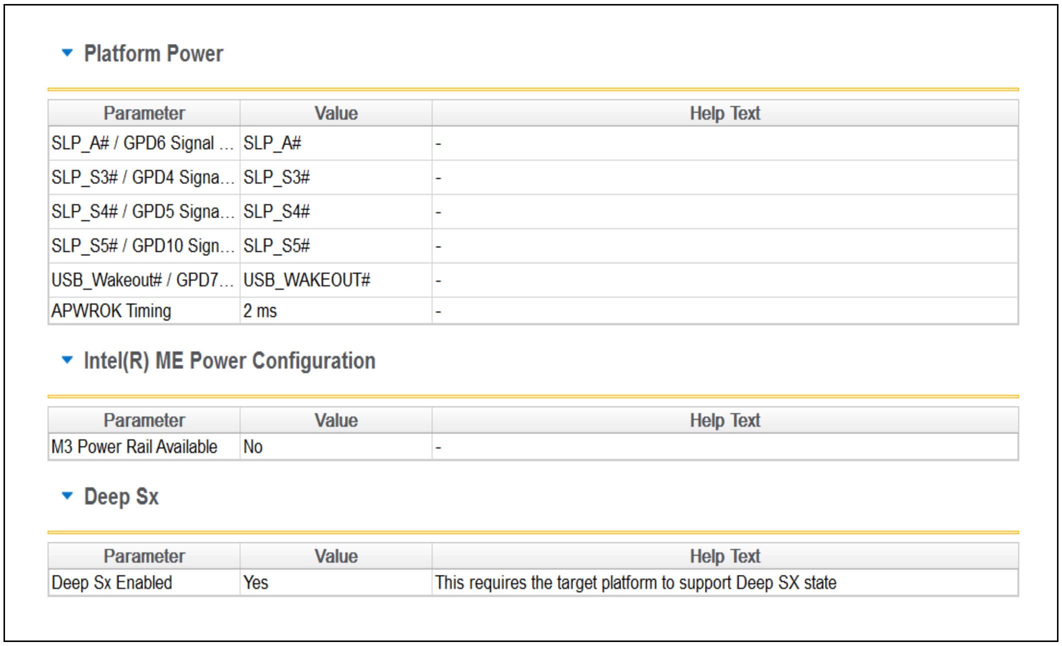
Parameter	Value	Help Text
Reserved	No	-

3.4.19 Power

This section describes the platform power configuration settings.

Click on the Power tab on the left pane to configure power parameters. (Figure 3-10)

Figure 3-11. Power



Platform Power

Parameter	Value	Help Text
SLP_A# / GPD6 Signal ...	SLP_A#	-
SLP_S3# / GPD4 Signa...	SLP_S3#	-
SLP_S4# / GPD5 Signa...	SLP_S4#	-
SLP_S5# / GPD10 Sign...	SLP_S5#	-
USB_Wakeout# / GPD7...	USB_WAKEOUT#	-
APWROK Timing	2 ms	-

Intel(R) ME Power Configuration

Parameter	Value	Help Text
M3 Power Rail Available	No	-

Deep Sx

Parameter	Value	Help Text
Deep Sx Enabled	Yes	This requires the target platform to support Deep SX state

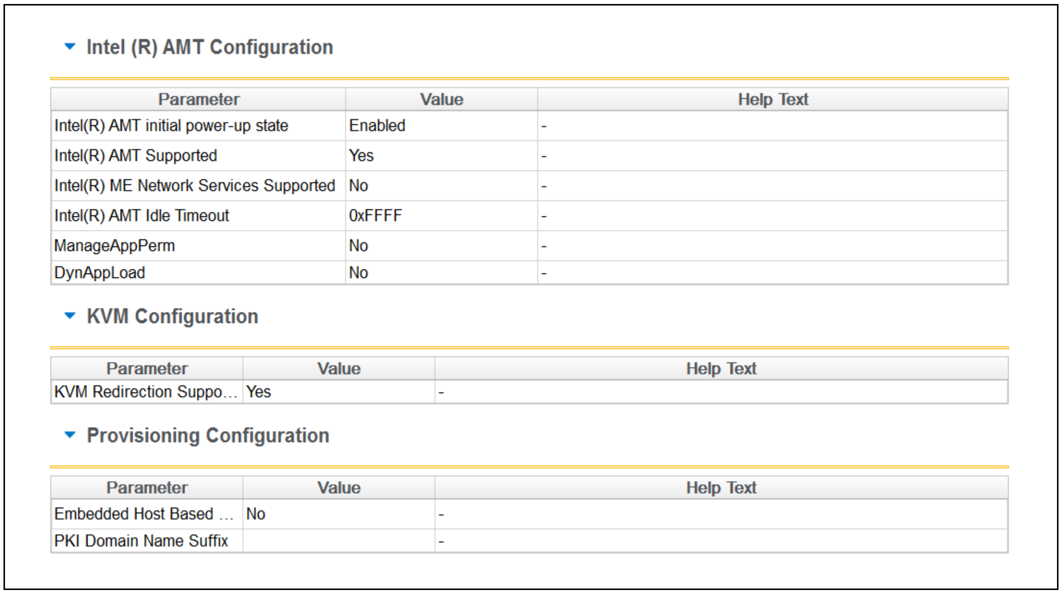
3.4.20 Manageability Application Section

Note:This section is not applicable to Consumer Intel® ME FW SKU.

This section describes the Manageability Application parameters. (Refer FW Bring up guide for general information.)

The Manageability section lets the user define the default Intel® AMT parameters. The values specified in this section are used after the Intel® AMT device is un-provisioned (full or partial). Click Intel® AMT Tab on the left tab to configure Intel® AMT parameters.

Figure 3-12. Manageability Application Section



Intel (R) AMT Configuration

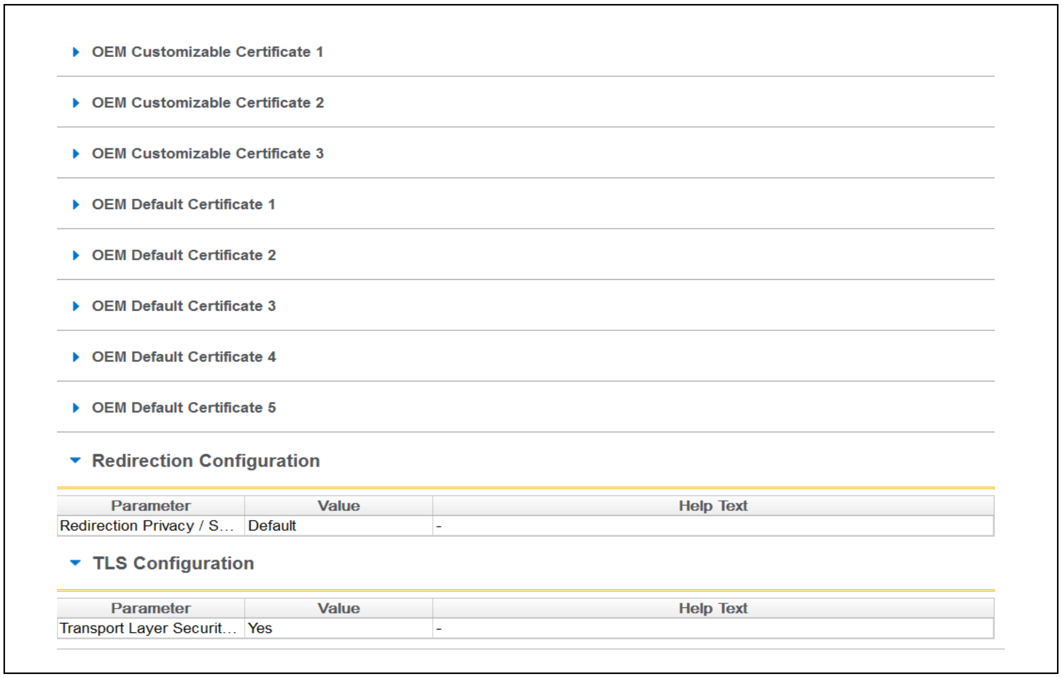
Parameter	Value	Help Text
Intel(R) AMT initial power-up state	Enabled	-
Intel(R) AMT Supported	Yes	-
Intel(R) ME Network Services Supported	No	-
Intel(R) AMT Idle Timeout	0xFFFF	-
ManageAppPerm	No	-
DynAppLoad	No	-

KVM Configuration

Parameter	Value	Help Text
KVM Redirection Suppo…	Yes	-

Provisioning Configuration

Parameter	Value	Help Text
Embedded Host Based …	No	-
PKI Domain Name Suffix		-



OEM Customizable Certificate 1

OEM Customizable Certificate 2

OEM Customizable Certificate 3

OEM Default Certificate 1

OEM Default Certificate 2

OEM Default Certificate 3

OEM Default Certificate 4

OEM Default Certificate 5

Redirection Configuration

Parameter	Value	Help Text
Redirection Privacy / S…	Default	-

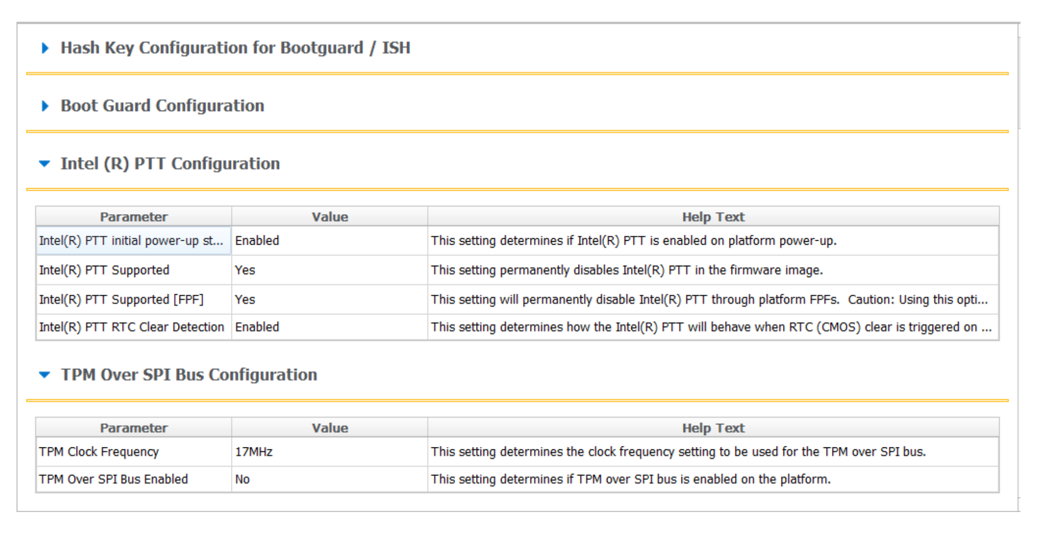
TLS Configuration

Parameter	Value	Help Text
Transport Layer Securit…	Yes	-

3.4.21 Platform Protection

The Platform Protection section determines which features are supported by the system. If a system does not meet the minimum hardware requirements, no error message is given when programming the image. (Refer to the FW Bringup guide for general information).

Figure 3-13. Platform Protection Section



Intel (R) PTT Configuration

Parameter	Value	Help Text
Intel(R) PTT initial power-up st...	Enabled	This setting determines if Intel(R) PTT is enabled on platform power-up.
Intel(R) PTT Supported	Yes	This setting permanently disables Intel(R) PTT in the firmware image.
Intel(R) PTT Supported [FPF]	Yes	This setting will permanently disable Intel(R) PTT through platform FPFs. Caution: Using this opti...
Intel(R) PTT RTC Clear Detection	Enabled	This setting determines how the Intel(R) PTT will behave when RTC (CMOS) clear is triggered on ...

TPM Over SPI Bus Configuration

Parameter	Value	Help Text
TPM Clock Frequency	17MHz	This setting determines the clock frequency setting to be used for the TPM over SPI bus.
TPM Over SPI Bus Enabled	No	This setting determines if TPM over SPI bus is enabled on the platform.

These options control the availability and visibility of FW features.

The ability to change certain options is SKU-dependent and – depending on the SKU selected – some of default values will be disabled and cannot be changed.

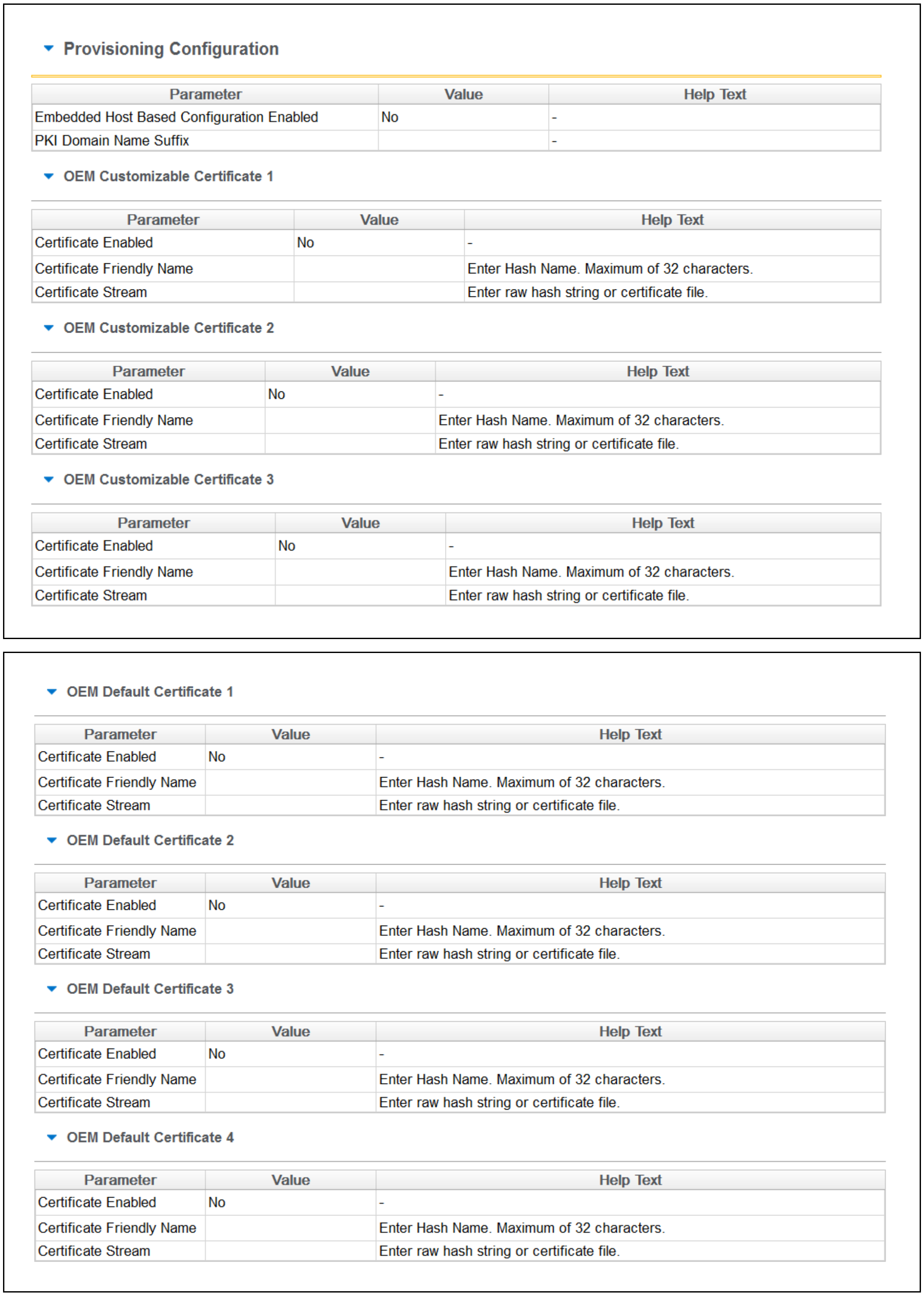
Note: PCH SKU and FW SKU selection is not within the tool. It is based on the PCH SKU part that customer chooses and the FW SKU they load on that platform.

- Intel® Platform Trusted Technology
- Intel® Content Protection

3.4.22 Provisioning Section

The Provisioning section allows the end user to specify the configuration settings, Intel® Upgrade Service, and Intel® DAL. (See the FW Bring up guide for general information).
Click the Intel® AMT tab on the left pane to specify the OEM settings.

Figure 3-14. Provisioning Configuration Section



Provisioning Configuration

Parameter	Value	Help Text
Embedded Host Based Configuration Enabled	No	-
PKI Domain Name Suffix		-

OEM Customizable Certificate 1

Parameter	Value	Help Text
Certificate Enabled	No	-
Certificate Friendly Name		Enter Hash Name. Maximum of 32 characters.
Certificate Stream		Enter raw hash string or certificate file.

OEM Customizable Certificate 2

Parameter	Value	Help Text
Certificate Enabled	No	-
Certificate Friendly Name		Enter Hash Name. Maximum of 32 characters.
Certificate Stream		Enter raw hash string or certificate file.

OEM Customizable Certificate 3

Parameter	Value	Help Text
Certificate Enabled	No	-
Certificate Friendly Name		Enter Hash Name. Maximum of 32 characters.
Certificate Stream		Enter raw hash string or certificate file.

OEM Default Certificate 1

Parameter	Value	Help Text
Certificate Enabled	No	-
Certificate Friendly Name		Enter Hash Name. Maximum of 32 characters.
Certificate Stream		Enter raw hash string or certificate file.

OEM Default Certificate 2

Parameter	Value	Help Text
Certificate Enabled	No	-
Certificate Friendly Name		Enter Hash Name. Maximum of 32 characters.
Certificate Stream		Enter raw hash string or certificate file.

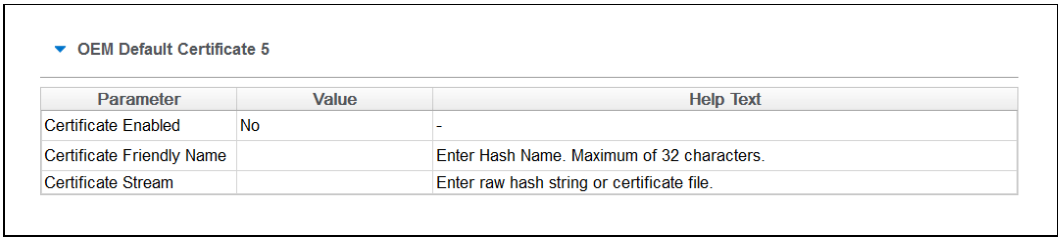
OEM Default Certificate 3

Parameter	Value	Help Text
Certificate Enabled	No	-
Certificate Friendly Name		Enter Hash Name. Maximum of 32 characters.
Certificate Stream		Enter raw hash string or certificate file.

OEM Default Certificate 4

Parameter	Value	Help Text
Certificate Enabled	No	-
Certificate Friendly Name		Enter Hash Name. Maximum of 32 characters.
Certificate Stream		Enter raw hash string or certificate file.

Figure 3-15. Provisioning Configuration Section (Cont...)



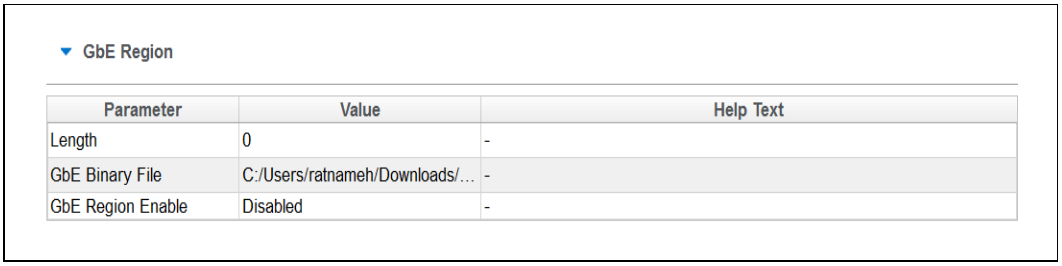
OEM Default Certificate 5

Parameter	Value	Help Text
Certificate Enabled	No	-
Certificate Friendly Name		Enter Hash Name. Maximum of 32 characters.
Certificate Stream		Enter raw hash string or certificate file.

3.4.23 Gbe (LAN) Region Settings

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

Figure 3-16. GbE Region Options



GbE Region

Parameter	Value	Help Text
Length	0	-
GbE Binary File	C:/Users/ratnameh/Downloads/...	-
GbE Region Enable	Disabled	-

3.4.24 Setting 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.

3.4.25 Setting Gbe Region Binary File

To select the Gbe Region binary file:

1. Click on Flash Layout tab on the left pane to load the binary file for Gbe region.
2. Select a file. When the flash image is built, the contents of this file are copied into the Gbe Region.

3.4.26 Enabling/Disabling GbE Region

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

To disable the GbE Region:

1. Click on Flash Layout tab on the left pane to load the binary file for Gbe region.
5. Choose **Disable Region** from the drop down. When the flash image is built it will not contain a GbE Region.

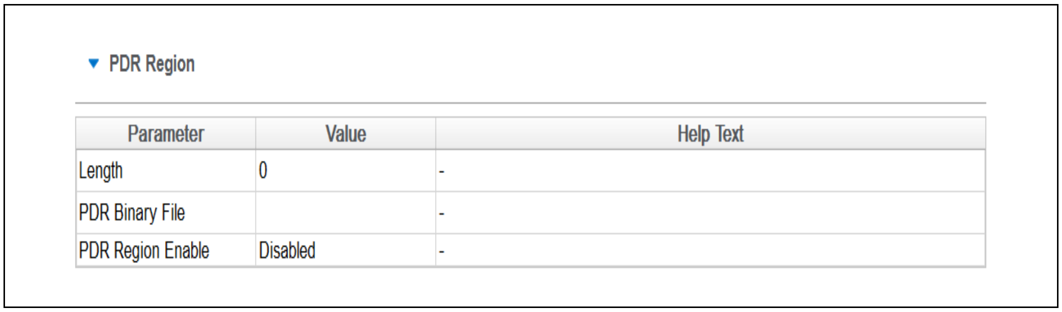
To enable the GbE Region:

1. Click on Flash Layout tab on the left pane to load the binary file for Gbe region
2. Choose **Enable Region** from the drop down menu.

3.4.27 Modifying PDR Region

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

Figure 3-17. PDR Region Options



Parameter	Value	Help Text
Length	0	-
PDR Binary File		-
PDR Region Enable	Disabled	-

3.4.28 Setting 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 Section 3.2.1.

3.4.29 Setting PDR Region Binary File

To select the PDR region binary file:

1. Click on Flash Layout tab on the left pane to load the binary file for PDR region
2. Click **OK** to update the parameter; when the flash image is built, the contents of this file is copied into the BIOS region.

3.4.30 Enabling/Disabling PDR Region

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

To disable the PDR Region:

1. Click Flash Layout tab on the left pane to load the binary file for Gbe region.
2. Choose **Disable Region** from the drop down menu; when the flash image is built, there is no PDR Region in it.

Note: This region is disabled by default.

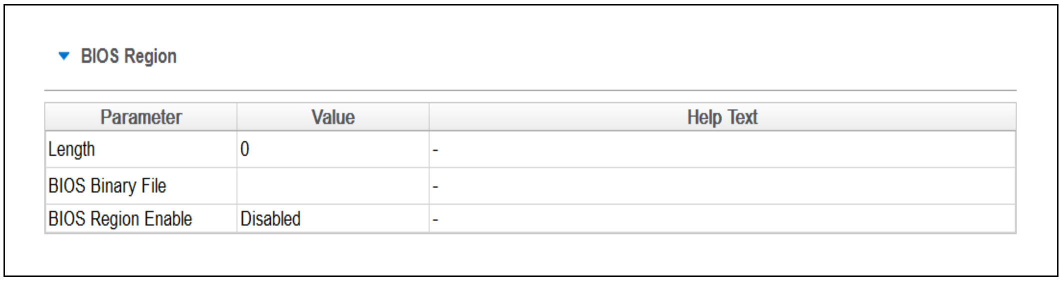
To enable the PDR Region:

1. Click on Flash Layout tab on the left pane to load the binary file for Gbe region
2. Choose **Enable Region** from the drop down menu.

3.4.31 Modifying BIOS Region

The BIOS Region contains the BIOS code run by the host processor. 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 3-18. BIOS Region Parameters



BIOS Region

Parameter	Value	Help Text
Length	0	-
BIOS Binary File		-
BIOS Region Enable	Disabled	-

3.4.32 Setting 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 as described in Section 3.2.1.

3.4.33 Setting the BIOS Region Binary File

To select the BIOS region binary file:

1. Click on Flash Layout tab on the left pane to load the binary file for BIOS region
2. Click **OK** to update the parameter; when the flash image is built, the contents of this file are copied into the BIOS region.

3.4.34 Enabling/Disabling the BIOS Region

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

To disable the BIOS Region:

1. Click on Flash Layout tab on the left pane to load the binary file for BIOS region
2. Choose **Disable Region** from the drop down menu; when the flash image is built, there is no BIOS Region in it.

To enable the BIOS Region:

1. Click on Flash Layout tab on the left pane to load the binary file for BIOS region
2. Select **Enable Region** from the drop down menu.

3.4.35 Building Flash Image

The flash image can be built with the FIT 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.

FIT 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, the user should select the single larger binary file when using FPT.

3.4.36 Decomposing Existing Flash Image

FIT 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 (refer below). A new image can be built from this configuration that is almost identical to the original, except for the changes 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 NVARs will not be able to be decomposed by FIT. FIT will use Intel default value instead. User might want to check the log file to find out which NVARs were not parsed.

Note: The ME region binary contained in INT folder after image generation only contains the firmware default base settings for ME region no FIT customization is applied.

3.4.37 Command Line Interface

FIT supports command line options.

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

The command line syntax for FIT is:

```
FIT [/h] [/?][/b] [/o <file>] [/rombypass <true|false>] [/sku <value>]
      [/me <file>] [/gbe <file>] [/bios <file>] [/pdr <file>] [/w <path>]
      [/s <path>] [/d <path>] [/u1 <value>] [/u2 <value>] [/u3 <value>]
      [/i <enable|disable>] [/flashcount <1|2>] [/flashsize1 <size>]
      [/flashsize2 <size>] [/save <file>] [XML or BIN file]
```

Table 3-5. FIT Command Line Options

Option	Description
<XML_file>	Used when generating a flash image file. A sample xml file is provided along with the FIT. When an xml file is used with the `/b` option, the flash image file is built automatically.
<Bin File>	Decomposes the BIN file. The individual regions are separated and placed in a folder with the same name as the BIN file.
-H or -?	Displays the command line options.
-B	Automatically builds the flash image. The GUI does not appear if this flag is specified. This option causes the program to run in auto-build mode. If there is an error, a valid message is displayed and the image is not built. If a BIN file is included in the command line, this option decomposes it.
-O <file>	Path and filename where the image is saved. This command overrides the output file path in the XML file.
-ROMBYPASS	Overrides rombypass settings in the XML file.
-ME <file>	Overrides the binary source file for the Intel® ME Region with the specified binary file.
-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.
-PDR <file>	Overrides the binary source file for the PDR Region with the specified binary file.
-I <enable|disable>	Enables or disables intermediate file generation.
-W <path>	Overrides the working directory environment variable $WorkingDir. It is recommended that the user 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 the user set these environmental variables before starting a project.

Option	Description
-D <path>	Overrides the destination directory environment variable $DestDir. It is recommended that the user 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.
-FLASHCOUNT <0, 1 or 2>	Overrides the number of flash components in the Descriptor Region. If this value is zero, only the Intel® ME Region is built.
-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 = 512KB 1 = 1MB 2 = 2MB 3 = 4MB 4 = 8MB 5 = 16MB.
-FLASHSIZE2 <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 = 512KB 1 = 1MB 2 = 2MB 3 = 4MB 4 = 8MB 5 = 16MB.
-Save <file>	Saves the XML file.
-SKU <value>	This option is used to change the SKU configuration being built. Use the words Q77, QM77, etc. as a reference to a SKU from the drop-down menu.

3.4.38 Example – Decomposing Image and Extracting Parameters

The NVARS variables and 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:

```
FIT.exe /f output.bin /b
```

This command would create a folder named "output". The folder contains the individual region binaries (Descriptor, GBE, Intel® ME, and BIOS) and the Map file.

The xml file contains the current Intel® ME parameters.

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

3.4.39 More Examples of FIT 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:

```
FIT.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:

```
FIT.exe /b /me ".\..\..\Image Components\Firmware\ME12.0_5M_PreProduction.BIN"
<file.bin or file.xml>
```

Note: The ME override option changes the ME base used on command line but still uses the values from the xml or binary passed in.

Take an existing image and put in a different GbE region:

```
FIT.exe /b /gbe ".\..\..\Image
Components\GbE\NAHUM6_CLARKSVILLE_DESKTOP_11.bin" <file.bin or file.xml>
```

§ §

4 Flash Programming Tool

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

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

- View the contents of the flash on the screen.
- Write the contents of the flash to a log file.
- Perform a binary file to flash comparison.
- Write to a specific address block.
- Program Named variables.
- Provision HDCP
- Provided FPF’s Access
- Helps doing Closemnf

Note: For proper function in a Multi-SPI configuration the Block Erase, Block Erase Command and Chip Erase must all match.

4.1 System Requirements

The DOS version of FPT (**fpt.exe**) runs on FreeDOS.

The EFI version of FPT (**fpt.efi**) runs on a 64-bit EFI environment.

The Windows® version (**fptw.exe**) requires administrator privileges to run under Windows® OS. The user needs to use the **Run as Administrator** option to open the CLI in Windows® 10.

The Windows® 64 bit version (fptw64.exe) is designed for running in native 64 bit OS environment which does not have 32 bit compatible mode available for example Windows®PE 64.

FPT requires that the platform is bootable (i.e. working BIOS) and has an operating system available 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. FPT must be run on the system with the flash memory to be programmed.

One possible workflow for using FPT is:

1. A pre-programmed flash with a bootable BIOS image is plugged into a new computer.
2. The computer boots.
3. FPT is run and a new BIOS/Intel® ME/GbE 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.

4.2 Flash Image Details

See the flash image details as described in the FIT Chapter 3.

4.3 Microsoft Windows® Required Files

The Microsoft Windows® version of the FPT executable is **fptw.exe**. The following files must be in the same directory as **fptw.exe**:

- 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.
- fptw.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® MEI driver installation files. Once you locate the .inf file you must use the Windows® PE cmd `drvload HECI.inf` to load it into the running system each time Windows® PE reboots. Failure to do so causes errors for some features.

Table 4-1. FPT OS Requirements

FPT Version	Target OS	Support Drivers
FPT.EXE	DOS	None
FPTw.EXE	Windows® 32 / 64 bit w/WOW64	idrvdll.dll, pmxdll.dll
FPTW64.EXE	Windows® Native 64 bit	idrvdll32e.dll, pmxdll32e.dll

Note:In the Windows® environment for operations involving global reset you should add a pause or delay when running FPTW using a batch or script file.

4.4 EFI Required Files

The EFI version of the FPT executable is **fpt.efi**. The following files must be placed in **the root directory** as **fpt.efi**:

- 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.

- fpt.efi – the executable used to program the final image file into the flash. Before running fpt.efi, all the required files must be placed at root directory of the disk otherwise error like "FPT is unable to find FPARTS.TXT "might be displayed.

4.5 DOS Required Files

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

- fpt.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.

4.6 Programming 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.

4.6.1 Stopping Intel® ME SPI Operations

FPT will automatically halt Intel® ME SPI access prior to erasing or writing data in the ME region. Customers do not have use either of the following steps listed below when updating platforms unless the descriptor has been locked.

Intel® ME SPI Operations can be stopped in the following ways:

·Assert HDA_SDO (known as GPIO 33 or Flash descriptor override/Intel® ME manufacturing jumper) to high while powering on the system. This is not a valid method if the parameters are configured to ignore this jumper.

·Send the HMRFPO ENABLE Intel® MEI command to Intel® ME (for more information refer PCH Intel® ME BIOS writer's guide).

Note: Pulling out DIMM from slot 0 or leaving the Intel® ME region empty to stop Intel® ME are not valid options for current generation platforms.

4.7 Programming NVARS

FPT can program the NVARS and change the default values of the parameters. The modified parameters are used by the Intel® ME FW after a global reset (Intel® ME +

HOST reset) or upon returning from a G3 state. NVARS can be programmed using getfile/setfile/CommitFiles APIs.

SetFile API will allow for the host to change the values in UEP (Unified Emulation Partition). Note: Intel® ME Firmware does NOT require commit File after a UEP SetFile. Attempting to execute Commit file when not necessary will result in firmware returning an error.

The variables can be modified individually or all at once via a text file.

Note: Files output when using the Intel® FPT -CFGGEN command line option in EFI environments do not contain the Carriage Return code 0x0D ('\r') as part of the EOL (end-of-line) sequence. As a result, when opened in Windows® or DOS environments, some applications may show all lines of text on a single line. If the output configuration files are intended to be edited in Windows® or DOS environments, it is recommended to use the Windows® or DOS version of Intel® FPT accordingly to collect the configuration data. Otherwise, they may be opened using a text editor which can process files which contain only Line Feed 0x0A ('\n') EOL sequences.

Table 4-2. Named Variables Options

Option	Description
fpt.exe –CVARS	Displays a list of the supported manufacturing configurable named variables (NVARs).
fpt.exe –cfggen	Creates a list of blank NVARs in a text file that lets the user update multiple line configurable NVARS. The variables have the following format in the text file: NVAR name = value which will be used by setfile.
fpt.exe –U –N <NVAR name>	Accept for updating UEP values using SetFile API
fpt.exe –U –IN <Text file>	Accepts cfggen file with values set and will use setfile to update

Refer to Appendix A for a description of all the NVAR parameters.

4.7.1 Programming GPIO NVAR

FPT tool will support configuring the GPIO via string inputted by the user on command line. The string inputted should be in defined format which FPT tool will parse and turn into a binary.

In this method, customer will specify the string which includes configuration data required by the GPIO NVAR (Feature ID, Usage, Owner and Attributes).

Format of command line will look like:

FPT –u CSE_GPIO GPIO [(FID, Usage, Owner, Attributes),…].

Each GPIO entry will include the FID, Usage, Owner, Attributes

4.8 Usage

The EFI, DOS and Windows® versions of the FPT can run with command line options.

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

The commands in the EFI, DOS and Windows® versions have the same syntax. The command line syntax for fpt.efi, fpt.exe and fptw.exe is:

```
FPT.exe [-H|?] [-VER] [-EXP] [-VERBOSE] [-Y] [-P] [-LIST] [-I]
        [-F] [-ERASE] [-VERIFY] [-NOVERIFY] [-D] [-DESC] [-BIOS]
        [-ME] [-GBE] [-PDR] [-EC] [-SAVEMAC] [-SAVESXID] [-B] [-E]
        [-REWRITE] [-ADDRESS|A] [-LENGTH|L] [-CVARS] [-MASTERACCESSGEN]
        [-CFGGEN] [-U] [-CLEAR][-O] [-IN] [-N] [-V] [-CLOSEMNF] [-GRESET]
        [-PAGE][-SPIBAR] [-R] [-VARS] [-COMMIT] [-HASHED] [-DISABLEME]
        [-COMPAREFPF] [-FPFS] [-COMMITFPF] [-PROVHDCP] [-READHDCP]
        [-GETPID] [-WRITETOKEN] [-ERASETOKEN] [-PROVKB] [-COMMITARBSVN]
```

Table 4-3. Command Line Options for fpt.efi, fpt.exe and fptw.exe

Option	Description
Help (-H, -?)	Displays the list of command line options supported by FPT tool. **Note:** Use -H for help when running in the EFI Shell.
-VER	Shows the version of the tools.
-EXP	Shows examples of how to use the tools.
-VERBOSE [<file>]	Displays the tool's debug information or stores it in a log file.
-Y	Bypasses Prompt. FPT does not prompt user for input. This confirmation will automatically be answered with "y".
-P <file>	Flash parts file. Specifies the alternate flash definition file which contains the flash parts description that FPT has to read. By default, FPT reads the flash parts definitions from fparts.txt.
-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.

Option	Description
-F <file> [NOVERIFY]	Flash. Programs a binary file into an SPI flash. The user needs to specify the binary file to be flashed. FPT reads the binary, and then programs the binary into the flash. After a successful flash, FPT verifies that the SPI flash matches the provided image. Without specify the length with –L option, FPT will use the total SPI size instead of an image size. The NOVERFY sub-option *must* follow the file name. This will allow flashing the SPI without verifying the programming was done correctly. The user will be prompted before proceeding unless '-y' is used.
-ERASE	Block Erase. Erases all the blocks in a flash. 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>	Verify. Compares a binary to the SPI flash. The image file name has to be passed as a command line argument if this flag is specified.
-NOVERIFY	Suboption of –F, see –F for details.
-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 4KB sections. The total size of the flash device must also be in increments of 4KB.
-DESC	Read/Write 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 BIOS region. Specifies that the BIOS region is to be read, written, or verified. Start address is the beginning of the region.
-ME	Read/Write 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.
-EC	Read/Write EC region. Specifies that the EC region is to be read, written, or verified. The start address is the beginning of the region.
-GBE	Read/Write 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 PDR region. Specifies that the PDR region is to be read, written, or verified. The start address is the beginning of the region.

Option	Description
-SAVEMAC	This is used to save the GbE MAC Address. It is appropriate only when GbE Firmware is being over written. It also saves the GbE SSID and SVID.
-SAVESXID	Saves the GbE SSID and SVID when GbE is being reflashed.
-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.
-E	Skip Erase. Does not erase blocks before writing. This option skips the erase operation before writing and should be used if the part being flashed is a blank SPI flash device.
-A<value>, -ADDRESS <value>	Write/Read Address. Specifies the start address at which a read, verify, or write operation must be performed. The user needs to provide an address. This option is not used when providing a region since the region dictates the start address.
-L <value>, -LENGTH <value>	Write/Read Length. Specifies the length of data to be read, written, or verified. The user needs to provide the length. This option is not used when providing a region since the region/file length determines this.
-CVARS	Lists all the current manufacturing line configurable variables.
-MASTERACCESSGEN	Generates a Manufacturing Line Configurable Master Access Input File.
-CFGGEN	NVAR Input file generation option. This creates a file which can be used to update the line configurable NVARS.
-U -CLEAR	Update. Updates variables in the UEP. The user can update the multiple FOVs by specifying their names and values in the parameter file. The parameter file must be in an INI file format (the same format generated by the `–cfggen` command). The `-in <file>` option is used to specify the input file. Using the -CLEAR flag will clear the variable in the UEP. This flag is only supported for a single variable.
-O <file>	Output File. The file used by FPT to output NVAR information.
-IN <file>	Input File. This option flag must be followed by a text file The text file may be either: A parameter file such as the one generated with the -cfggen option (used with the -u option) or: A Configurable Master Access file such as the one generated with the -masteraccessgen option (used with the -closemanuf option)

Option	Description
-N <value>	Name. Specifies the name of the NVAR that the user wants to update in the image file or flash. The name flag must be used with Value (`-v`).
-V <value>	Value. Specifies the value for the NVAR variable. The name of variable is specified in the Name flag. The Value flag must follow the Name flag.
-CLOSEMNF <NO> <PDR>	End of Manufacturing. This option is executed at the end of manufacturing phase. This option does the following: Sets the Intel® ME manufacturing mode done bit (Global Locked bit). Verifies that the Intel® ME manufacturing mode done bit (Global Locked) is set. Sets the master region access permission in the Descriptor region to its Intel-recommended value (see the -MASTERACCESSGEN and -IN options) Verifies that flash regions are locked. If the image was properly set before running this option, FPT skips all of the above and reports PASS. If anything was changed, FPT automatically forces a global reset through the CF9GR mechanism. The user can use the no reset option to bypass the reset. If nothing was changed, based on the current setting, the tool reports PASS without any reset. The "NO" addition will prevent the system from doing a global reset following a successful update of the ME Manufacturing Mode Done, the Region Access permissions, or both. The "PDR" addition will allow CPU\BIOS Read and Write access to the PDR region of flash. Note: Running `FPT-closemnf` also sets the default value for any unprovisioning process. Run `FPT -closemnf` first if the user wants to test any unprovisioning related process. In order to allow FPT to perform a global reset, BIOS should not lock CF9GR when Intel® ME is in manufacturing mode. This step is highly recommended to the manufacturing process. Without doing proper end of manufacturing process would lead to ship platform with potential security/privacy risk. Important: Before using this option with Production MCP / FW verify that the values for the PTT and Anchor Cove are correct in your image. Once this setting is used it will permanently commit values into the Field Programmable Fuses and cannot be undone.
-GRESET	Global Reset. FPT performs a global reset.

Option	Description
-PAGE	Pauses the screen when a page of text has been reached. Hit any key to continue.
-SPIBAR	Display SPI BAR. FPT uses this option to display the SPI Base Address Register.
-R <name>	NVAR Read. FPT uses this option to retrieve NVAR value for a specific NVAR file name. The value of the variable is displayed. By default, all non- secure variables are displayed in clear-text and secure NVAR will be displayed in HASH. The `-hashed` option can be used to display the hash of a value instead of the clear-text value.
-VARS	Display Supported Variables. FPT uses this option to display all variables supported for the `-R` and `-COMPARE` commands. Note: This will no longer display UEP based values which are tied to configuring FPFs.
-COMMIT	Commit. FPT uses this option to commit all setfile commands NVARs changes to NVAR and cause relevant reset accordingly If no pending variable changes are present, Intel® ME does not reset and the tool displays the status of the commit operation.
-HASHED	Hash Variable Output. FPT uses this option to distinguish whether the displayed output is hashed by the FW. For variables that can only be returned in hashed form (such as the Intel® MEBx password), this option has no effect – the data displayed is hashed regardless.
-DISABLEME	Disable the Management Engine.
-COMPAREFPF<name>	Compare the FPF with a value passed in by the user.
-FPFS	Displays a list of the FPFs.
-COMMITFPF <name>	Commits NVAR values to FPF via firmware and prevents further modification of FPFs.
-PROVHDCP <file><file>	Provision platform with the key and cert provided.
-READHDCP	Displays the HDCP Rx provisioning status.
-GETPID <file>	Retrieve the part id.
-REWRITE	Allows to rewrite the SPI with file data even if flash is identical.
-WRITETOKEN <file>	Write the token where the file name is the token name.
-ERASETOKEN	Delete the token.
–PROVKB <iv_and_keybox.bin>	Provision Widevine using IV (Initialization Vector) and encrypted KeyBox file.
-COMMITARBSVN	Commits ARB SVN to FPFs. This would commit the Anti Rollback SVN to the FPFs

Table 4-4. FPT–closemnf Behavior

Condition before FPT -closemnf			Condition after FPT -closemnf			Other FPT Action	
Intel ME Mfg Done bit set	Flash Access set to Intel rec values	Intel ME Mfg Mode	Intel ME Mfg Done bit set	Flash Access set to Intel rec values?	Intel ME Mfg Mode	FPT return value **	Global Reset
No	No	Enabled	**Yes**	**Yes**	**Disabled**	0	Yes
No	Yes	Enabled	No	Yes	Enabled	1	No
Yes	No	Enabled	Yes	**Yes**	**Disabled**	0	Yes
Yes	Yes	Disabled	Yes	Yes	Disabled	0	No

** Return value 0 indicates successful completion. In the second case, FPT –closemnf returns 1 (= error) because it is unable to set the Intel ME Mfg Done bit, because flash permissions are already set to Intel recommended values (host cannot access Intel ME Region).

Table 4-5. Intel-Recommend Access Settings

	Intel® ME	GbE	BIOS
Read	0b 0000 1101 = 0x0d	0b 0000 1000 = 0x08	0b 0000 0011 = 0x0B 0b 0001 1011 = 0x1B – BIOS access to PDR
Write	0b 0000 1100 = 0x0c	0b 0000 1000 = 0x08	0b 0000 0010 = 0x0A 0b 0001 1010 = 0x1A – BIOS access to PDR

4.9 Updating Hash Certificate through NVAR

Note:This section is not applicable for Consumer Intel® ME FW SKU.

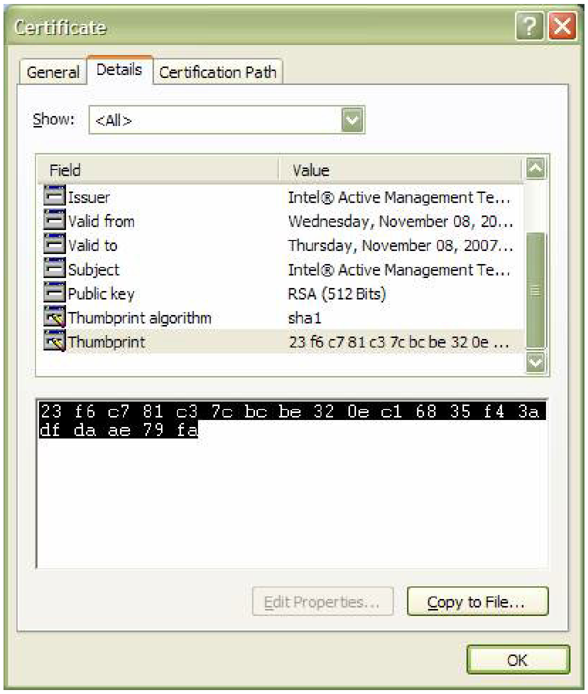
There are 3 OEM Customizable certificate hash values that can be stored in the Intel® ME region:

- The OEM Customizable Certificates 1-3 are not default certificates and are deleted after a full un-provisioning.
- The OEM Customizable Certificates 1-3 are configurable by NVAR (with FPT or other flash programming methods) or FIT.

To store certificate hash values in the Intel® ME region:

1. Copy the raw hash values from a valid certificate file.

Figure 4-1. Raw Hash Values from Certificate File

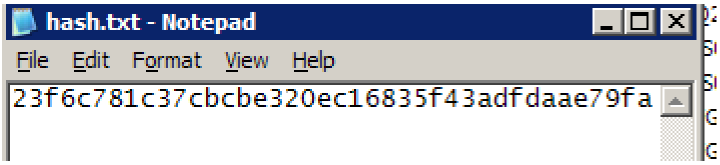


Field	Value
Issuer	Intel® Active Management Te...
Valid from	Wednesday, November 08, 20...
Valid to	Thursday, November 08, 2007...
Subject	Intel® Active Management Te...
Public key	RSA (512 Bits)
Thumbprint algorithm	sha1
Thumbprint	23 f6 c7 81 c3 7c bc be 32 0e ...

```
23 f6 c7 81 c3 7c bc be 32 0e c1 68 35 f4 3a
df da ae 79 fa
```

2. Paste the raw hash values into a text file
3. Remove all the spaces from the text file.

Figure 4-2. Sample Hash.txt File



```
23f6c781c37cbcbe320ec16835f43adfdaae79fa
```

4. Save the text file as **hash.txt**.
5. Copy and paste the text saved from hash.txt and add it to **FPT.CFG file** in order to update the NVAR:
   EXAMPLE:

```
;  OEMCustomCert1 Certificate
;  All data is required to update the certificate.
;  See the Tools Users Guide for detailed explanation
;  of required data and format.
OEMCustomCert1 IsActive      = 0x01
OEMCustomCert1 FriendlyName  = MyCert
OEMCustomCert1 RawHashFile   = 23f6c781c37cbcbe320ec16835f43adfdaae79fa
```

6. Flash Hash NVAR with FPT's `-u` –in option (e.g., fpt –u –in fpt.cfg).

Note: **FTP.CFG** is the file that is used to update NVAR

4.10 Fparts.txt File

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

```
Intel (R) Flash Programming Tool. Version:  x.x.x.xxxx
Copyright (c) 2007-2014, Intel Corporation. All rights reserved.
Platform: Intel(R) Qxx Express Chipset
Error 75: “fparts.txt” file not found.
```

If the device is not located in **fparts.txt**, the user is 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 Cannon LakePCH-LP SPI Programming Guide. The device must have a **4KB erase sector** and the total size of the SPI Flash device must be a multiple of 4KB. 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)
- Unused

4.11 Examples

The following examples illustrate the usage of the EFI and DOS versions of the tool (fpt.efi and fpt.exe respectively). The Windows® version of the tool (Fptw.exe) behaves in the same manner apart from running in a Windows® environment.

4.11.1 Complete SPI Flash Device with Binary File

In order to use FPT Tool for Flashing the SPI Image the following BIOS settings need to be done manually otherwise errors may be seen related to BIOS Region Protected while executing fpt.exe –f spi.bin.

1.BIOS MENU INTEL ADVANCED → CPU CONFIGURATION → BIOS GUARD : Disabled

2.BIOS MENU → INTEL ADVANCED → PCH I/O CONFIGURATION → SECURITY CONFIGURATION → BIOS LOCK : Disabled

3.BIOS MENU -> INTEL ADVANCED ->PCH I/O CONFIGURATION -> Flash Protection Range: Disabled..

4.BIOS MENU -> INTEL ADVANCED ->PCH I/O CONFIGURATION -> Flash Protection Range: Disabled..

In order to use FPT Tool with Lewisburg C620 series, the following BIOS settings are recommended (to avoid errors when running fpt.exe –f spi.bin):

1.EDKII Menu → Platform Configuration →PCH Configuration → Security Configuration → SMM BIOS Write Protect = Disabled

2.EDKII Menu

3.→ Platform Configuration → PCH Configuration → PCH DFX Configuration → Show SPI device = EnableKII Menu

4.→ Platform Configuration →PCH Configuration → PCH DFX Configuration → BIOS Lock = Disable

5.EDKII Menu

6.→ Platform Configuration → Miscellaneous Configuration → BIOS Guard = unchecked

7.EDKII Menu

8.→ Platform Configuration → Server ME Configuration → Manageability Application Configuration → Manageability State = Enable

9.EDKII Menu

10.→ Platform Configuration → PCH Configuration → PCH Devices → Dirty Warm Reset = Disable

```
C:\ fpt.exe –f spi.bin

EFI:
>fpt.efi –f spi.bin or fs0:\>fpt.efi –f spi.bin
```

This command writes the data in the **spi.bin** file into a whole SPI flash from address 0x0.

4.11.2 Program Specific Region

```
fpt.exe –f bios.rom –BIOS

EFI:
fpt.efi –f bios.rom –BIOS

--------------------------------------------
Intel (R) Flash Programming Tool. Version:  x.x.x.xxxx
Copyright (c) 2007-2014, Intel Corporation. All rights reserved.
Platform: Intel(R) Qxx Express Chipset
Reading HSFSTS register... Flash Descriptor: Valid
--- Flash Devices Found ---
    W25Q64BV    ID:0xEF4017    Size: 8192KB (65536Kb)
- Erasing Flash Block [0x800000]... - 100% complete.
- Programming Flash [0x800000]2560KB or 2560KB - 100% complete.
- Verifying Flash [0x800000]2560KB or 2560KB - 100% complete.
```

```
RESULT: The Data is identical.
FPT Operation Passed
```

This command writes the data in **bios.bin** into the BIOS region of the SPI flash and verifies that the operation ran successfully.

4.11.3 Program SPI Flash from Specific Address

```
fpt.exe -F image.bin -A 0x100 -L 0x800

EFI:
fpt.efi -F image.bin -A 0x100 -L 0x800
```

This command loads 0x800 of the binary file **image.bin** starting at address 0x0100. The starting address and the length needs to be a multiple of 4KB.

4.11.4 Dump Full Image

```
fpt.exe –d imagedump.bin

EFI:
fpt.efi –d imagedump.bin

--------------------------------------------
Intel (R) Flash Programming Tool. Version:  x.x.x.xxxx
Copyright (c) 2007-2014, Intel Corporation. All rights reserved.
Platform: Intel(R) Qxx Express Chipset
Reading HSFSTS register... Flash Descriptor: Valid
--- Flash Devices Found ---
    W25Q64BV    ID:0xEF4017    Size: 8192KB (65536Kb)
- Reading Flash [0x00800000]... 8192KB of 8192KB - 100% complete.
Writing flash contents to file "imagedump.bin"...
Memory Dump Complete
FPT Operation Passed
```

4.11.5 Dump Specific Region

```
fpt.exe –d descdump.bin –desc
EFI:
fpt.efi –d descdump.bin –desc

--------------------------------------------
Intel (R) Flash Programming Tool. Version:  x.x.x.xxxx
Copyright (c) 2007-2014, Intel Corporation. All rights reserved.
Platform: Intel(R) Qxx Express Chipset
Reading HSFSTS register... Flash Descriptor: Valid
--- Flash Devices Found ---
    W25Q64BV    ID:0xEF4017    Size: 8192KB (65536Kb)
- Reading Flash [0x000040]... 4KB of 4KB - 100% complete.
Writing flash contents to file "descdump.bin"...
Memory Dump Complete
```

```
FPT Operation Passed
```

This command writes the contents of the Descriptor region to the file **descdump.bin**.

4.11.6 Display SPI Information

```
fptw.exe –I
--------------------------------------------
Intel (R) Flash Programming Tool. Version:  XX.X.X.XXXX
Copyright (c) 2007 - 2017, Intel Corporation. All rights reserved.

Reading HSFSTS register... Flash Descriptor: Valid

   --- Flash Devices Found ---
   W25Q256FVID:0xEF4019Size: 32768KB (262144Kb)


Warning: There are some addresses that are not defined in any regions.
Read/Write/Erase operations are not possible on those addresses.

   --- Flash Image Information --
   Signature: VALID
   Number of Flash Components: 1
      Component 1 - 32768KB (262144Kb)
   Regions:
      DESC     - Base: 0x00000000, Limit: 0x00000FFF
      BIOS     - Base: 0x01183000, Limit: 0x01B82FFF
      CSME     - Base: 0x00083000, Limit: 0x01082FFF
      GbE      - Base: 0x00081000, Limit: 0x00082FFF
      PDR      - Not present
      EC       - Base: 0x00001000, Limit: 0x00080FFF
   Master Region Access:
      BIOS     - ID: Read: 0xFFFF, Write: 0xFFFF
      CSME     - ID: Read: 0xFFFF, Write: 0xFFFF
      GbE      - ID: Read: 0xFFFF, Write: 0xFFFF
      EC       - ID: Read: 0xFFFF, Write: 0xFFFF

Total Accessable SPI Memory: 28172KB, Total Installed SPI Memory : 32768KB

FPT Operation Successful.
```

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**, FPT returns the error message "There is no supported SPI flash device installed".

4.11.7 Verify Image with Errors

```
fpt.exe -verify outimage.bin

EFI:
fpt.efi -verify outimage.bin

--------------------------------------------
```

```
Intel(R) Flash Programming Tool. Version:  x.x.x.xxxx
Copyright (c) 2007-2014, Intel Corporation. All rights reserved.
Platform: Intel(R) Qxx Express Chipset
Reading HSFSTS register... Flash Descriptor: Valid
--- Flash Devices Found ---
    W25Q64BV    ID:0xEF4017    Size: 8192KB (65536Kb)
RESULT: Data does not match!
[0x00000000] Expected 0x5A, Found: 0x5A
[0x00000001] Expected 0xA5, Found: 0xA5
Total mismatches found in 64 byte block: 2
Error 204: Data verify mismatch found at address 0x000
```

This command compares the Intel® ME region programmed on the flash with the specified FW image file **outimage.bin**. If the `-y` option is not used; the user is notified that the file is smaller than the binary image. This is due to extra padding that is added during the program process. The padding can be ignored when performing a comparison. The `-y` option proceeds with the comparison without warning.

4.11.8 Verify Image Successfully

```
fpt.exe -verify outimage.bin

EFI:
fpt.efi -verify outimage.bin

--------------------------------------------
Intel (R) Flash Programming Tool. Version:  x.x.x.xxxx
Copyright (c) 2007-2014, Intel Corporation. All rights reserved.
Platform: Intel(R) Qxx Express Chipset
Reading HSFSTS register... Flash Descriptor: Valid
--- Flash Devices Found ---
    W25Q64BV    ID:0xEF4017    Size: 8192KB (65536Kb)
-Verifying Flash [0x800000] 8192KB of 8192KB – 100% complete.
RESULT: The data is identical.
FPT Operation Passed
```

This command compares **image.bin** with the contents of the flash. Comparing an image should be done immediately after programming the flash device. Verifying the contents of the flash device after a system reset results in a mismatch because Intel® ME changes some data in the flash after a reset.

4.11.9 Get Intel® ME settings

```
fpt.exe –r “Privacy/SecurityLevel”
fpt.efi –r “^”Privacy/SecurityLevel”^”
---------------------------------------------
Intel (R) Flash Programming Tool. Version:  x.x.x.xxxx
Copyright (c) 2007-2014, Intel Corporation. All rights reserved.
Platform: Intel(R) Qxx Express Chipset
Reading HSFSTS register... Flash Descriptor: Valid
--- Flash Devices Found ---
    W25Q64BV    ID:0xEF4017    Size: 8192KB (65536Kb)
Variable: "Privacy/SecurityLevel"
```

```
Value: True / 01
Retrieve Operation: Successful
```

Note:Only –r (get command) supports the –hashed optional command argument. When –hashed is used, variable value will be returned in hashed format, otherwise it will be returned in clear txt. There are a few exceptions in the case of variables MEBxPassword, PID and PPS, their value will be always returned in hashed format regardless –hashed is used or not. This is primarily because of security concern.

4.11.10 CVAR Configuration File Generation (-cfggen)

It creates an input file which can be used to update CVARs. The file includes all the current CVAR. When creating the file, it extracts the fixed offset variables from flash. Note, the file generated will change every time the list of CVAR changes.

```
fpt.exe –cfggen [ -o <Output Text File> ][ options ]
```

-o <Output File Name>	The desired name of the file generated. If none is provided the default, fpt.cfg, will be used.
-p < file name >	Alternate SPI Flash Parts list file.
-page	Pauses at screen / page / window boundaries. Hit any key to continue.
-Verbose [<file name>]	Displays more information.
-y	Will not pause to user input to continue

Example FPT.CFG output:

```
;
;   Flash Programming Tool FOV Programming File
;
;   Any entry that is not included, or does not have a value
;   following the label will not be updated.
;
;   Comments can be added by using a ‘;’ as the first entry
;   on the line.
;
;   For further explanation  of the required inputs see the
;   System Tools User Guide.doc
;
;   Any entries, FOVs, that are displayed with values
;   indicates that the FOV has already been given a value,
;   but has not yet been committed.  Entries without values
;   indicates that the FOV has not been written, at least
;   since the system reset or use of the ‘-commit’ command.
```

```
GpioNvar = 0x3035303030303030303430303030303031373030

DAM =

;   EDP_PORT_CFG NVAR value is not displayed because it is stored
encrypted.
EDP_PORT_CFG =

;   LSPCON_PORT NVAR value is not displayed because it is stored
encrypted.
LSPCON_PORT =

;  OEM Customizable Certificate 1 Certificate
;  All data is required to update the certificate.
;  See the Tools Users Guide for detailed explanation
;  of required data and format.
OEMCustomCert1 IsActive      =
OEMCustomCert1 FriendlyName  =
OEMCustomCert1 RawHashFile   =

;  OEM Customizable Certificate 2 Certificate
;  All data is required to update the certificate.
;  See the Tools Users Guide for detailed explanation
;  of required data and format.
OEMCustomCert2 IsActive      =
OEMCustomCert2 FriendlyName  =
OEMCustomCert2 RawHashFile   =

;  OEM Customizable Certificate 3 Certificate
;  All data is required to update the certificate.
;  See the Tools Users Guide for detailed explanation
;  of required data and format.
OEMCustomCert3 IsActive      =
OEMCustomCert3 FriendlyName  =
OEMCustomCert3 RawHashFile   =

;   CfgSrvFqdn NVAR value is not displayed because it is stored encrypted.
CfgSrvFqdn =

Rcfg = 0x01

StorageState = 0x01

SOL = 0x01

KVM = 0x01

OptInPolicy = 0x11

HostName =

DomainName =
CfgSrvAdr =

CfgSrvPort = 0x26F3

Privacy/SecurityLevel = 0x01

IdleTO = 0xFFFF

ScreenBlankingEn = 0x00

AmtWdAutoReset = 0x00

;   PkiDns NVAR value is not displayed because it is stored encrypted.
PkiDns =

EhbcState = 0x00

;   MEBxPassword NVAR value is not displayed because it is stored
encrypted.
MEBxPassword =

;   ODM_ID NVAR value is not displayed because it is stored encrypted.
ODM_ID =

;   SystemIntegratorID NVAR value is not displayed because it is stored
encrypted.
SystemIntegratorID =

;   ReservedID NVAR value is not displayed because it is stored encrypted.
ReservedID =

Intel(R) AMT Supported = 0x01

Manageability Application Supported = 0x01

Transport Layer Security Supported = 0x01

iTouch = 0x00

PTTEnable = 0x00

URTC = 0x00

SetWLANPowerWell = 0x86

OEM_TAG = 0x00000000

FWUpdLcl = 0x01

PTT = 0x01

ENF0 = 0x00

ENF1 = 0x00
```

```
OEM_DID =

OEM_PID =

NCC = 0x00

TxtSupp = 0x00

BootGuard = 0x0040

CPU Debugging = 0x00

BSP Initialization = 0x00

Protect BIOS Environment Enabled = 0x00

Measured Boot Enabled = 0x00

Verified Boot Enabled = 0x00

Key Manifest ID = 0x01

Force Boot Guard ACM Enabled = 0x00

S3 Optimization Disabled = 0x00

;   OEM_CRD NVAR value is not displayed because it is stored encrypted.
OEM_CRD =
```

§ §

5 Intel® MEManuf and MEManufWin

Intel® MEManuf validates Intel® ME functionality on the manufacturing line. It does not check for LAN functionality as it assumes that all Intel® ME components on the test board have been validated by their respective vendors. It does verify that these components have been assembled together correctly.

The Windows® version of Intel® MEManufWin (Intel® MEManufWin) requires administrator privileges to run under Windows® OS. The user needs to use the **Run as Administrator** option to open the CLI in Windows® 10.

Intel® MEManuf validates all components and flows that need to be tested according to the FW installed on the platform in order to ensure the functionality of Intel® ME applications: BIOS-FW, Flash, SMBus, M-Link, KVM, etc. This tool is meant to be run on the manufacturing line.

5.1 Windows® PE Requirements

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

5.2 How to Use Intel® MEManuf

Intel® MEManuf checks the FW SKU and runs the proper tests accordingly unless an option to select tests is specified. If Intel® AMT is enabled on the platform; it automatically causes a reboot to test system hardware connections when the system is in sleep state.

Intel® MEManuf is intelligent enough to know if it should run the test or report a result. If there is no test result available for an Intel® ME enabled platform, MEManuf calls the test. Otherwise, it reports the result or the failure message from the previous test.

Intel® MEManuf tools report the result or cause a reboot. If there is a reboot, Intel® MEManuf should be run again.

5.3 Usage

The DOS 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:

```
MEManuf[-EXP] [-H|?] [-VER] [-BLOCKNET] [-ALLOWNET]
   [-TEST] [-S0] [-BISTRESULT] [-NEXTREBOOT] [-EOL]
   [-CFGGEN] [-F] [-VERBOSE] [-PAGE] [-ERRLIST] [-ALL]
   [-NOWLAN] [-WLAN] [-NOGFX] [-GFX] [-NOLAN] [-LAN]

Tool might returning following values for BIST to indicate either SUCCESS/ ERROR/
SUCCESS WITH WARNING.

0 means SUCCESS
1 means ERROR
2 means SUCCESS (With Warnings)
```

Table 5-1. Options for MEManuf

Option	Description
No option	There are differences depending on the firmware SKU type the system is running on: If BIST is disabled in the Intel® ME Boot: The first time running Intel® MEManuf, since there is no CM3 test result stored in SPI, the tool will request the FW to run a complete BIST which includes a power reset at the end of the test for the DOS version and a Hibernation for the Windows® version. This power reset is only host side power cycle that triggered by Intel® ME. When host resets, Intel® ME FW will transition from CM0 to CM3, and then attempt automatically transition back from CM3 to CM0 along bringing host back to S0. Once host is booted back into OS, user needs to run the tool again in order to run runtime BIST and retrieve the test result. If BIST is enabled in the Intel® ME Boot: If there is no CM3 test result, the tool will report error and request user to use –test to run a full BIST. If there is CM3 test result, the tool will execute the runtime BIST and report the result. If running on a Consumer SKU image, the tool will request the FW to run a complete BIST which does not involve any power transition at the end of the test. Test result will be reported back right after the test is done and cleared. If BIST test result is not displayed after BIST test is done, the tool needs to be run again (with or without any BIST related argument combinations) to retrieve the result, once test result is displayed, it will be cleared. Tool is capable of remembering whether/what tests (including host based tests) have been run from previous invocation. Host based tests will be run for all cases (whether it’s retrieving test result or run the actual BIST). Currently there are two host based tests; they are VSCC Table validation check and ICC data check.
-EXP	Shows examples of how to use the tools.

Option	Description
-H or -?	Displays the help screen. **Note:** Use -H for help when running in the EFI Shell.
-VER	Shows the version of the tools.
-S0	The same as No option, except that there is no power reset/hibernation performed at the end of the BIST test including Intel® AMT SKU. The test result is reported back right after the test is done and cleared.
-F <filename>	Load customer defined .cfg file
-TEST	Run full test
-NOWLAN	Note: This option is not applicable for Consumer Intel® ME FW SKU. This option only applies to the AMT test so that the user can skip the wireless LAN NIC test if there is no wireless LAN NIC attached to the hardware. When `–nowlan` switch is not used, Intel® MEManuf also checks for the HW presence of Intel WLAN card based on a pre-defined list. If Intel® MEManuf detects an Intel WLAN card present on the platform, Intel® MEManuf runs the WLAN BIST test and reports pass/fail accordingly. If Intel® MEManuf cannot find any known WLAN card, Intel® MEManuf skips the WLAN BIST test and does not report errors. With the `–verbose` option, it displays "No Intel wireless LAN card detected" Note: `–S0` can only be used on the platform which Intel® AMT is present and can be enabled in the field.
-WLAN	Force wireless LAN test
-BLOCKNET	**Note**: This option is not applicable for Consumer Intel® ME FW SKU. This option blocks any network traffic that goes in/out of the integrated GbE wired/wireless LAN interface. If Intel® AMT is disabled, "Error 9257: Cannot run the command since Intel® AMT is not available" is returned.
-ALLOWNET	**Note**: This option is not applicable for Consumer Intel® ME FW SKU. This option allows any network traffic that goes in/out of the integrated GbE wired/wireless LAN interface. If Intel® AMT is disabled, "Error 9257: Cannot run the command since Intel® AMT is not available" is returned.
-BISTRESULT	Returns last BIST results.
-ERRLIST <test name>	Return a list of available codes.

Option	Description
-EOL <Var|Config> -F <filename>	This option runs several checks for the use of OEMs to ensure that all settings and configurations have been made according to Intel requirements before the system leaves the manufacturing process. The check can be configured by the customer to select which test items to run and their expected value (only applicable for Variable Values, FW Version, BIOS Version, and Gbe Version). The sub option `config` or `var` is optional. Using `–EOL` without a sub option is equivalent to the `–EOL config`. ICC data check is performed for all options. The Full BIST test for ME12.0 is a combination of M0_HW, Live_HW and M0_Config. The Runtime BIST is a combination of M0_HW and M0_Config. Intel® MEManuf Sx test will require system is capable to enter sleep state, keep pinging the platform with network package and keep the system up will make the test failed. Host based Tests ME/BIOS VSCC validation, Intel® MEManuf verifies that flash SPI ID on the system is described in VSCC table. If found, VSCC entry for relevant SPI part should match the known good values that pre-populated in the file. Intel® ME state check, Intel® MEManuf verifies Intel® ME is in normal state. This is done by checking the value of 4 fields (initialization state, mode of operation, current operation state, and error state) in FW status register1. If any of these fields indicates Intel® ME is in abnormal state, Intel® MEManuf will report error without running BIST test. ICC data check, Intel® MEManuf verifies that valid OEM ICC data is present and programmed accordingly. This is done by checking FW status register2 ICC bits (which are bit 1 and 2 equal to 3). When –f flag is used along with a file name (<filename>), the tool will load the file as the configuration file, instead of using MEManuf.xml.
-NEXTREBOOT	Upon successful platform reboot CM3 Autotest will be performed. **Note:** This is a standalone command and will only work if CM3 Autotest has been enabled in the firmware image. CM3 Autotest will be executed on the next CMoff – CM0 transition (example: Cold Reset), Global Reset or G3. The option itself will not trigger any platform reboots.
-CFGGEN <filename>	Use this option along with a filename to generate a default configuration file. This file (with or without modification) can be used for the `–EOL` option. Rename it MEManuf.xml before using it. It is highly recommended to use this option to generate a new MEManuf.xml with an up-to-date variable names list before using the Intel® MEManuf End-Of-Line check feature.
-ALL	Use this option to generate all possible tests for configuration file. All BIST, EOLConfig, and EOLVAR types of tests will be included in the generated XML. ***Note:*** Intel recommended tests will be enabled regardless of *-all* parameter to meet corresponding dependencies
-VERBOSE <file>	Displays the debug information of the tool or stores it in a log file.

Option	Description
–PAGE	When it takes more than one screen to display all the information, this option lets the user pause the display and then press any key to continue on to the next screen.
-NOGFX	This option will skip KVM related test.
-GFX	This option will force KVM related test.
-NOLAN	**Note:** This option is not applicable for Consumer Intel® ME FW SKU. This option only applies to the Intel® AMT test so that the user can skip the wired LAN NIC test if there is no wired LAN NIC attached to the hardware. **Note:** `–S0` can only be used on the platform which Intel® AMT is present and can be enabled in the field.
-LAN	This option will force LAN test

Note: The KVM test will be skipped if the platform being tested contains both internal and external GFX and BIOS has disabled internal GFX.

Table 5-2. Intel® MEManuf Test Matrix

		CM3 Supported SKU	Consumer SKU
BIST Disabled in the ME BOOT	No option	-1st time: Run full BIST test (with ME triggered reset under DOS, host triggered hibernation under Windows®), and save the CM3 test result in SPI - After: Run Runtime BIST and query CM3 test result from SPI without reset.	Run runtime BIST test (with no reset)
	-Test	-Run full BIST test with Intel ME triggered reset in DOS and host triggered hibernation in Windows® - Save the CM3 test result in SPI.	Run runtime BIST test (with no reset)
	-S0	Run runtime BIST test (with no reset).	Same as CM3 Supported SKU
BIST Enabled in the ME BOOT	No option	Run the Runtime BIST and query M3 test result from SPI without reset, if not CM3 test result retrieved, return error.	Run runtime BIST test (with no reset)
	-Test	-Run full BIST test with Intel ME triggered reset in DOS and host triggered hibernation in Windows® - Save the CM3 test result in SPI .	Run runtime BIST test (with no reset)
	-S0	Run runtime BIST test (with no reset)	Same as CM3 Supported SKU

Note: ICC data check is performed for all options.

Note: The Full BIST test for ME12.0 is a combination of M0_HW, Live_HW and M0_Config. The Runtime BIST is a combination of M0_HW and M0_Config.

Intel® MEManuf Sx test will require system is capable to enter sleep state, keep pinging the platform with network package and keep the system up will make the test failed.

5.3.1 Host based Tests

1. ME/BIOS VSCC validation, Intel® MEManuf verifies that flash SPI ID on the system is described in VSCC table. If found, VSCC entry for relevant SPI part should match the known good values that pre-populated in the file.

2. Intel® ME state check, Intel® MEManuf verifies Intel® ME is in normal state. This is done by checking the value of 4 fields (initialization state, mode of operation, current operation state, and error state) in FW status register1. If any of these fields indicates Intel® ME is in abnormal state, Intel® MEManuf will report error without running BIST test.
3. ICC data check, Intel® MEManuf verifies that valid OEM ICC data is present and programmed accordingly. This is done by checking FW status register2 ICC bits (which are bit 1 and 2 equal to 3).

5.4 Intel® MEManuf –EOL Check

MEManuf `-EOL` check is used to give customers the ability to check Intel® ME-related configuration before shipping. There are two sets of tests that can be run: variable check and configuration check. Variable check is very similar as FPT –compare option. Refer that section.

5.4.1 ErrorAction Field

The end_of_line (-EOL) check is split into two categories; *Variable Check*, and *Configuration Check*. If any of these checks fails, by default Intel® MEManuf will report the error and continue to the next check.

If it is desired to change this default behavior, ‘ErrorAction’ field can be used. In other words, ErrorAction is used to define the importance of a test. It can be defined with one of the following values:

- **ErrorContinue**: this is the default value, it reports the error and continue to the next check.
- **ErrorStop**: When an error is encountered, it’s reported and the testing process stops.
- **WarnContinue**: reports a warning regarding the error and continues to the next check.

5.4.2 MEManuf.xml File

The MEManuf**.xml** file includes all the test configurations for `MEManuf -EOL` check. It needs to be at the same folder that MEManuf is run. If there is no MEManuf**.xml** file on that folder, MEManuf `-EOL config` runs the Intel recommended default check only.

Note: Only MAC address, Wireless MAC address and System UUID tests allow the user to set the ReqVal option.

```
   <?xml version="1.0" encoding="utf-8"?>
<!-- This is the configuration file for the csmemanuf test tool. -->
<!-- This file is divided into the different test types (csmebist, eolconfig,
eolvar). -->
<!-- Any line in this file that is marked with "<!" to start with is NOT editable by
the user and is strictly informational. Any changes to these lines will be ignored -
->
<!-- Generally the user may change enabled(true/false), errorlevel(error,warning),
and in some cases required value -->
<!-- It is recommended that you edit this document with an XML specific/capable
editor -->

<!-- A missing field or bad value will fail validation and result in an error -->
<!-- State PossibleValues="Enabled/Disabled" -->
<!-- ErrAction PossibleValues="ErrorContinue/ErrorStop/WarningContinue" -->
<memanuf_config>
   <!-- CSME BIST TESTS -->
   <csmebist name="Policy Kernel - Power Package : Live Heap Test">
      <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
      <!-- Description>Allocate memory in live heap in M0, write in M3, read back in
M0.</Description -->
      <!-- IntelRequired>True</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- TestType>LIVE_HW</TestType -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
   </csmebist>
   <csmebist name="Common Services - General : WLAN enabled only on mobile or
desktop">
      <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
      <!-- Description>If platform is not desktop or mobile (server) there should be
no WLAN (value 0x80).</Description -->
      <!-- IntelRequired>True</IntelRequired -->
      <!-- Dependencies>WLAN</Dependencies -->
      <!-- TestType>M0_CONFIG</TestType -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
   </csmebist>
   <csmebist name="Common Services - Wireless LAN : Connectivity to NIC">
      <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
      <!-- Description>Runs the Wlan test WLAN access through Clink 1.</Description
-->
      <!-- IntelRequired>True</IntelRequired -->
      <!-- Dependencies>WLAN</Dependencies -->
      <!-- TestType>M0_HW</TestType -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
   </csmebist>
   <csmebist name="Policy Kernel - ME Configuration : Wlan Power Well">
      <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
      <!-- Description>WLAN power well setting.</Description -->
      <!-- IntelRequired>True</IntelRequired -->
      <!-- Dependencies>WLAN</Dependencies -->
```

```
        <!-- TestType>M0_CONFIG</TestType -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
    </csmebist>
    <csmebist name="Policy Kernel - ME Password : Validate MEBx password">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Verify password is acceptable.</Description -->
        <!-- IntelRequired>True</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- TestType>M0_CONFIG</TestType -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
    </csmebist>
    <csmebist name="Policy Kernel - Boot Guard : Self Test">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Get test result from NVAR SECURE_BOOT_SELF_TEST_RESULT.</
Description -->
        <!-- IntelRequired>True</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- TestType>M0_HW</TestType -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
    </csmebist>
    <csmebist name="Policy Kernel - ME Configuration : PROC_MISSING">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Only on mobile. Test fails if rule is not set to
MEFWCAPS_NO_ONBOARD_GLUE_LOGIC.</Description -->
        <!-- IntelRequired>True</IntelRequired -->
        <!-- Dependencies>MOBILE</Dependencies -->
        <!-- TestType>M0_CONFIG</TestType -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
    </csmebist>
    <csmebist name="SMBus - SMBus : Read byte">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Read one byte from SmBus ICH device (offset 0x44), if fails,
read DIMM0 (offset 0xA0 >> 1), if fails, read DIMM1 (0xA2 >> 1) and so on (0xA4 >> 1,
0xA6 >> 1, 0xA8 >> 1, 0xAA >> 1). Test fails if all trials failed.</Description -->
        <!-- IntelRequired>True</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- TestType>M0_HW</TestType -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
```

```
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
   </csmebist>
   <csmebist name="VDM - General : VDM engine">
      <!-- The commented fields below CANNOT be edited. Any edits will be ignored by 
the tool -->
      <!-- Description>Test VDM.</Description -->
      <!-- IntelRequired>True</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- TestType>M0_HW</TestType -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
   </csmebist>
   <csmebist name="GFX - General : Sampling engine">
      <!-- The commented fields below CANNOT be edited. Any edits will be ignored by 
the tool -->
      <!-- Description>Test KVM sampling engine..</Description -->
      <!-- IntelRequired>True</IntelRequired -->
      <!-- Dependencies>IPV6_LAN_ADDR</Dependencies -->
      <!-- TestType>M0_HW</TestType -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
   </csmebist>
   <csmebist name="USBr - General : Storage">
      <!-- The commented fields below CANNOT be edited. Any edits will be ignored by 
the tool -->
      <!-- Description>Test USBr Storage.</Description -->
      <!-- IntelRequired>True</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- TestType>M0_HW</TestType -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
   </csmebist>
   <csmebist name="USBr - General : KVM">
      <!-- The commented fields below CANNOT be edited. Any edits will be ignored by 
the tool -->
      <!-- Description>Test USBr KVM.</Description -->
      <!-- IntelRequired>True</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- TestType>M0_HW</TestType -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
   </csmebist>
   <csmebist name="Common Services - LAN : Connectivity to NIC in M3">
      <!-- The commented fields below CANNOT be edited. Any edits will be ignored by 
the tool -->
      <!-- Description>LAN test runs only if AMT is not permanently disabled and 
```

```
mDNSProxy is not disabled.</Description -->
        <!-- IntelRequired>True</IntelRequired -->
        <!-- Dependencies>LAN</Dependencies -->
        <!-- TestType>LIVE_HW</TestType -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
    </csmebist>
    <csmebist name="Common Services - LAN : Connectivity to NIC in M0">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>LAN test runs only if AMT is not permanently disabled and
mDNSProxy is not disabled.</Description -->
        <!-- IntelRequired>True</IntelRequired -->
        <!-- Dependencies>LAN</Dependencies -->
        <!-- TestType>M0_HW</TestType -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
    </csmebist>
    <csmebist name="Common Services - EHBC State : EHBC and Privacy Level states
compatibility">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Check while both EHBC and privacy level are available,
(PrivLevel != Default) && (EHBCState == EHBC_STATE_ENABLE).</Description -->
        <!-- IntelRequired>True</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- TestType>M0_CONFIG</TestType -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
    </csmebist>
    <csmebist name="Common Services - EHBC State : Valid Embedded Host Based
Configuration (EHBC) state">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Check if EHBC state is available.</Description -->
        <!-- IntelRequired>True</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- TestType>M0_CONFIG</TestType -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
    </csmebist>
    <csmebist name="Common Services - Privacy Level : Valid Privacy Level settings">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Check if privacy level is available.</Description -->
        <!-- IntelRequired>True</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- TestType>M0_CONFIG</TestType -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
    </csmebist>
    <csmebist name="AMT - KVM : Compression engine">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Test KVM compression engine.</Description -->
        <!-- IntelRequired>True</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- TestType>M0_HW</TestType -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
    </csmebist>
    <csmebist name="AMT - KVM : Compare engine">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Test KVM compare engine.</Description -->
        <!-- IntelRequired>True</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- TestType>M0_HW</TestType -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
    </csmebist>
    <csmebist name="AMT - EC : Basic connectivity">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Only on mobile, if power source is DC, test fails.</
Description -->
        <!-- IntelRequired>True</IntelRequired -->
        <!-- Dependencies>MOBILE</Dependencies -->
        <!-- TestType>M0_HW</TestType -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
    </csmebist>
    <csmebist name="AMT - Power : Valid WLAN power well (Mobile)">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Run the tests verifying the internal variables.</Description
-->
        <!-- IntelRequired>True</IntelRequired -->
        <!-- Dependencies>WLAN|MOBILE</Dependencies -->
        <!-- TestType>M0_CONFIG</TestType -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
    </csmebist>
    <csmebist name="AMT - Power : Valid LAN power well">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Run the tests verifying the internal variables.</Description
-->
        <!-- IntelRequired>True</IntelRequired -->
        <!-- Dependencies>LAN</Dependencies -->
        <!-- TestType>M0_CONFIG</TestType -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
    </csmebist>
    <csmebist name="PAVP - General : Verify Edp and Lspcon Configurations">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Check if LSPCON and 5K ports are overlapped</Description -->
        <!-- IntelRequired>True</IntelRequired -->
        <!-- Dependencies>VPRO|STD|IPV6_WLAN_ADDR</Dependencies -->
        <!-- TestType>M0_HW</TestType -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
    </csmebist>
    <csmebist name="PAVP - General : Set Lspcon Port">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Test the validility of the 5K port configuration</
Description -->
        <!-- IntelRequired>True</IntelRequired -->
        <!-- Dependencies>VPRO|STD|IPV6_WLAN_ADDR</Dependencies -->
        <!-- TestType>M0_HW</TestType -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
    </csmebist>
    <csmebist name="PAVP - General : Set Edp Port">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Test the validility of the LSPCON port configuration</
Description -->
        <!-- IntelRequired>True</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- TestType>M0_HW</TestType -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
    </csmebist>
    <csmebist name="Touch - General : Reset Panel">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
```

```
        <!-- Description>Hard Reset the sensor</Description -->
        <!-- IntelRequired>True</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- TestType>M0_HW</TestType -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
    </csmebist>
    <csmebist name="Touch - General : Generate Test Packets">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Generate Packets from sensor</Description -->
        <!-- IntelRequired>True</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- TestType>M0_HW</TestType -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
    </csmebist>
    <csmebist name="Touch - General : Panel Detect">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Check if sensor is detected</Description -->
        <!-- IntelRequired>True</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- TestType>M0_HW</TestType -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
    </csmebist>
    <!-- END OF CSME BIST TESTS -->
    <!-- EOL CONFIG TESTS -->
    <eolconfig name="TXT Supported FPF">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Check fpf TXT Supported FPF against expected value</
Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="No/00/Yes/01" example="No"> </RequiredValue>
    </eolconfig>
    <eolconfig name="SPI Boot Source FPF">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Check fpf SPI Boot Source FPF against expected value</
Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Enabled/00/Disabled/01" example="Enabled"> </
RequiredValue>
    </eolconfig>
    <eolconfig name="SoC Config Lock FPF">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Check fpf SoC Config Lock FPF against expected value</
Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Disabled/00/Enabled/01" example="Disabled"> </
RequiredValue>
    </eolconfig>
    <eolconfig name="RPMB Migration Done FPF">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Check fpf RPMB Migration Done FPF against expected value</
Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Disabled/00/Enabled/01" example="Disabled"> </
RequiredValue>
    </eolconfig>
    <eolconfig name="Persistent PRTC Backup Power FPF">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Check fpf Persistent PRTC Backup Power FPF against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Exists/00/None/01" example="Exists"> </RequiredValue>
    </eolconfig>
    <eolconfig name="Key Manifest ID FPF">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Check fpf Key Manifest ID FPF against expected value</
Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Hex number with 0x prefix." example="0x00"> </
RequiredValue>
    </eolconfig>
    <eolconfig name="OEM Secure Boot Policy FPF">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Check fpf OEM Secure Boot Policy FPF against expected value</
Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Hex number with 0x prefix." example="0x0000"> </
RequiredValue>
    </eolconfig>
    <eolconfig name="OEM Public Key Hash FPF">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Check fpf OEM Public Key Hash FPF against expected value</
Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="32 hex pairs with space between pairs" example="04 AB
F3 45 03 1D EF A2 B7 E8 98 79 10 45 AB DE F2 35 49 A0 01 35 78 29 37 AB DE EF FA 10
EF 33"> </RequiredValue>
    </eolconfig>
    <eolconfig name="OEM Platform ID FPF">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Check fpf OEM Platform ID FPF against expected value</
Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Hex number with 0x prefix." example="0x0000"> </
RequiredValue>
    </eolconfig>
    <eolconfig name="OEM KM Present">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Check fpf OEM KM Present against expected value</Description
-->
```

```
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="No/00/Yes/01" example="No"> </RequiredValue>
   </eolconfig>
   <eolconfig name="OEM ID FPF">
      <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
      <!-- Description>Check fpf OEM ID FPF against expected value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Hex number with 0x prefix." example="0x0000"> </
RequiredValue>
   </eolconfig>
   <eolconfig name="KM SVN FPF">
      <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
      <!-- Description>Check fpf KM SVN FPF against expected value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Hex number with 0x prefix." example="0x00"> </
RequiredValue>
   </eolconfig>
   <eolconfig name="PTT FPF">
      <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
      <!-- Description>Check fpf PTT FPF against expected value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Disabled/00/Enabled/01" example="Disabled"> </
RequiredValue>
   </eolconfig>
   <eolconfig name="BSMM SVN FPF">
      <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
      <!-- Description>Check fpf BSMM SVN FPF against expected value</Description -
->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Hex number with 0x prefix." example="0x00"> </
RequiredValue>
   </eolconfig>
   <eolconfig name="ACM SVN FPF">
      <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
      <!-- Description>Check fpf ACM SVN FPF against expected value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Hex number with 0x prefix." example="0x00"> </
RequiredValue>
   </eolconfig>
   <eolconfig name="Enforcement Policy FPF">
      <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
      <!-- Description>Check fpf enf against expected value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="2 digit hex number with 0x prefix" example="0x00"> </
RequiredValue>
   </eolconfig>
   <eolconfig name="Confirm ARB SVN value">
      <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
      <!-- Description>Confirms that the minimum ARB SVN saved in the PCH fuses
matches the ARB SVN of the FW image</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
   </eolconfig>
   <eolconfig name="PCH Unlocked state">
      <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
      <!-- Description>Verifies that PCH is locked</Description -->
      <!-- IntelRequired>True</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
   </eolconfig>
```

```
    <eolconfig name="HW Binding enabled">
       <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
       <!-- Description>Verifies that HW binding is disabled</Description -->
       <!-- IntelRequired>True</IntelRequired -->
       <!-- Dependencies></Dependencies -->
       <!-- End of uneditable fields -->
       <!-- Please edit the fields below ONLY with the State or ErrAction -->
       <State>Enabled</State>
       <ErrAction>ErrorContinue</ErrAction>
    </eolconfig>
    <eolconfig name="SOC Config Lock">
       <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
       <!-- Description>Check that SOC Config Lock FPF is set.</Description -->
       <!-- IntelRequired>True</IntelRequired -->
       <!-- Dependencies></Dependencies -->
       <!-- End of uneditable fields -->
       <!-- Please edit the fields below ONLY with the State or ErrAction -->
       <State>Enabled</State>
       <ErrAction>ErrorContinue</ErrAction>
    </eolconfig>
    <eolconfig name="FPFs in UEP Committed">
       <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
       <!-- Description>Check that FPFs in UEP are committed to Hardware.</
Description -->
       <!-- IntelRequired>True</IntelRequired -->
       <!-- Dependencies></Dependencies -->
       <!-- End of uneditable fields -->
       <!-- Please edit the fields below ONLY with the State or ErrAction -->
       <State>Enabled</State>
       <ErrAction>ErrorContinue</ErrAction>
    </eolconfig>
    <eolconfig name="Validate Keybox Provisioning">
       <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
       <!-- Description>Check to see if Keybox is provisioned</Description -->
       <!-- IntelRequired>False</IntelRequired -->
       <!-- Dependencies></Dependencies -->
       <!-- End of uneditable fields -->
       <!-- Please edit the fields below ONLY with the State or ErrAction -->
       <State>Enabled</State>
       <ErrAction>ErrorContinue</ErrAction>
    </eolconfig>
    <eolconfig name="Touch - Vendor ID">
       <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
       <!-- Description>Check Vendor ID (Touch) against expected value.</Description
-->
       <!-- IntelRequired>False</IntelRequired -->
       <!-- Dependencies></Dependencies -->
       <!-- End of uneditable fields -->
       <!-- Please edit the fields below ONLY with the State or ErrAction -->
       <State>Enabled</State>
```

```
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Hex number with 0x prefix." example="0x8086"> </
RequiredValue>
    </eolconfig>
    <eolconfig name="Firmware Update OEM ID">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Check Firmware Update OEM ID value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Hex" example="00000000-0000-0000-0000-000000000000">
</RequiredValue>
    </eolconfig>
    <eolconfig name="Wireless LAN micro-code mismatch">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Check ucode WLAN against programmed ucode</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>VPRO|CORP|IPV4_WLAN_HW</Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Yes/No -OR- 1/0" example="1"> </RequiredValue>
    </eolconfig>
    <eolconfig name="GBE version">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Check Gbe Version against expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>LAN|SPI_DEP</Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="major_ver.minor_ver" example="0.6"> </RequiredValue>
    </eolconfig>
    <eolconfig name="BIOS version">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Check BIOS Version against expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Customer specific"
example="HSWLPTU1.86C.0117.R00.1303102001"> </RequiredValue>
    </eolconfig>
    <eolconfig name="ME FW version">
```

```
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Check Firmware Version against expected value</Description -
->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="major_ver.minor_ver.hotfix_ver.build_num H|LP|ULT
Corporate|Consumer|Slim" example="12.0.0.1040 LP Consumer"> </RequiredValue>
    </eolconfig>
    <eolconfig name="System UUID">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Check System UUID against programmed value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>VPRO</Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="See example" example="550e8400-e29b-41d4-a716-
446655440000"> </RequiredValue>
    </eolconfig>
    <eolconfig name="Wireless MAC address">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Check Wireless MAC address</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>VPRO|IPV4_WLAN_HW|WLAN_MAC_ADDR_AVAIL</Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="6 hex pairs separated by ':'"
example="00:01:12:A2:3B:45"> </RequiredValue>
    </eolconfig>
    <eolconfig name="MAC address">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Check MAC address</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>VPRO|IPV4_LAN_HW</Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="6 hex pairs separated by ':'"
example="00:01:12:A2:3B:45"> </RequiredValue>
    </eolconfig>
    <eolconfig name="Security Descriptor Override (SDO) check">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
```

```
       <!-- Description>Check SDO pin</Description -->
       <!-- IntelRequired>True</IntelRequired -->
       <!-- Dependencies>SPI_DEP</Dependencies -->
       <!-- End of uneditable fields -->
       <!-- Please edit the fields below ONLY with the State or ErrAction -->
       <State>Enabled</State>
       <ErrAction>ErrorContinue</ErrAction>
   </eolconfig>
   <eolconfig name="EC Write Access Permissions">
       <!-- The commented fields below CANNOT be edited. Any edits will be ignored by 
the tool -->
       <!-- Description>Check EC write access</Description -->
       <!-- IntelRequired>True</IntelRequired -->
       <!-- Dependencies>SPI_DEP</Dependencies -->
       <!-- End of uneditable fields -->
       <!-- Please edit the fields below ONLY with the State or ErrAction -->
       <State>Enabled</State>
       <ErrAction>ErrorContinue</ErrAction>
       <RequiredValue format="Hex number with 0x prefix." example="0x0101. Value 
left empty will result in checking against Intel recommended values."> </
RequiredValue>
   </eolconfig>
   <eolconfig name="EC Read Access Permissions">
       <!-- The commented fields below CANNOT be edited. Any edits will be ignored by 
the tool -->
       <!-- Description>Check EC read access</Description -->
       <!-- IntelRequired>True</IntelRequired -->
       <!-- Dependencies>SPI_DEP</Dependencies -->
       <!-- End of uneditable fields -->
       <!-- Please edit the fields below ONLY with the State or ErrAction -->
       <State>Enabled</State>
       <ErrAction>ErrorContinue</ErrAction>
       <RequiredValue format="Hex number with 0x prefix." example="0x0101. Value 
left empty will result in checking against Intel recommended values."> </
RequiredValue>
   </eolconfig>
   <eolconfig name="BIOS Write Access Permissions">
       <!-- The commented fields below CANNOT be edited. Any edits will be ignored by 
the tool -->
       <!-- Description>Check BIOS write access</Description -->
       <!-- IntelRequired>True</IntelRequired -->
       <!-- Dependencies>SPI_DEP</Dependencies -->
       <!-- End of uneditable fields -->
       <!-- Please edit the fields below ONLY with the State or ErrAction -->
       <State>Enabled</State>
       <ErrAction>ErrorContinue</ErrAction>
       <RequiredValue format="Hex number with 0x prefix." example="0x0101. Value 
left empty will result in checking against Intel recommended values."> </
RequiredValue>
   </eolconfig>
   <eolconfig name="BIOS Read Access Permissions">
       <!-- The commented fields below CANNOT be edited. Any edits will be ignored by 
the tool -->
       <!-- Description>Check BIOS read access</Description -->
       <!-- IntelRequired>True</IntelRequired -->
```

```
        <!-- Dependencies>SPI_DEP</Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Hex number with 0x prefix." example="0x0101. Value 
left empty will result in checking against Intel recommended values."> </
RequiredValue>
    </eolconfig>
    <eolconfig name="GBE Write Access Permissions">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by 
the tool -->
        <!-- Description>Check GBE write access</Description -->
        <!-- IntelRequired>True</IntelRequired -->
        <!-- Dependencies>SPI_DEP</Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Hex number with 0x prefix." example="0x0101. Value 
left empty will result in checking against Intel recommended values."> </
RequiredValue>
    </eolconfig>
    <eolconfig name="GBE Read Access Permissions">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by 
the tool -->
        <!-- Description>Check GBE read access</Description -->
        <!-- IntelRequired>True</IntelRequired -->
        <!-- Dependencies>SPI_DEP</Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Hex number with 0x prefix." example="0x0101. Value 
left empty will result in checking against Intel recommended values."> </
RequiredValue>
    </eolconfig>
    <eolconfig name="ME Write Access Permissions">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by 
the tool -->
        <!-- Description>Check ME write access</Description -->
        <!-- IntelRequired>True</IntelRequired -->
        <!-- Dependencies>SPI_DEP</Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Hex number with 0x prefix." example="0x0101. Value 
left empty will result in checking against Intel recommended values."> </
RequiredValue>
    </eolconfig>
    <eolconfig name="ME Read Access Permissions">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by 
the tool -->
        <!-- Description>Check ME read access</Description -->
```

```
      <!-- IntelRequired>True</IntelRequired -->
      <!-- Dependencies>SPI_DEP</Dependencies -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Hex number with 0x prefix." example="0x0101. Value
left empty will result in checking against Intel recommended values."> </
RequiredValue>
   </eolconfig>
   <eolconfig name="ME Manufacturing Mode status">
      <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
      <!-- Description>Check End of Manufacturing Mode against Intel recommended
value</Description -->
      <!-- IntelRequired>True</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
   </eolconfig>
   <eolconfig name="EOP status check">
      <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
      <!-- Description>Check that EOP was sent/recieved</Description -->
      <!-- IntelRequired>True</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
   </eolconfig>
   <!-- END OF EOL CONFIG TESTS -->
   <!-- EOL VAR TESTS -->
   <eolvar name="eDP Port Configuration">
      <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
      <!-- Description>Test variable against expected value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="32 hex pairs with space between pairs" example="04 AB
F3 45 03 1D EF A2 B7 E8 98 79 10 45 AB DE F2 35 49 A0 01 35 78 29 37 AB DE EF FA 10
EF 33"> </RequiredValue>
   </eolvar>
   <eolvar name="WLAN Power Well">
      <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
      <!-- Description>Test variable against expected value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
```

```
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Disabled/80/CoreWell/81/SusWell/82/MEWell/83/WLAN
Sleep via SLP_WLAN#/86" example="Disabled"> </RequiredValue>
    </eolvar>
    <eolvar name="Unconfigure On RTC">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Test variable against expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Enabled/00/Disabled/01" example="Enabled"> </
RequiredValue>
    </eolvar>
    <eolvar name="Transport Layer Security Supported">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Test variable against expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="No/00/Yes/01" example="No"> </RequiredValue>
    </eolvar>
    <eolvar name="System Integrator ID used by Intel (R) Service">
        <!-- The commented fields bellow CANNOT be edited. Any edits will be ignored
by the tool -->
        <!-- Description>Test variable against expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Hex" example="0x0"> </RequiredValue>
    </eolvar>
    <eolvar name="StorageState">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Test variable against expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Disabled/00/Enabled/01" example="Disabled"> </
```

```
RequiredValue>
   </eolvar>
   <eolvar name="SOL">
      <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
      <!-- Description>Test variable against expected value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Disabled/00/Enabled/01" example="Disabled"> </
RequiredValue>
   </eolvar>
   <eolvar name="Reserved ID used by Intel(R) Services">
      <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
      <!-- Description>Test variable against expected value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="32 hex pairs with space between pairs" example="04 AB
F3 45 03 1D EF A2 B7 E8 98 79 10 45 AB DE F2 35 49 A0 01 35 78 29 37 AB DE EF FA 10
EF 33"> </RequiredValue>
   </eolvar>
   <eolvar name="Redirection Privacy / Security Level">
      <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
      <!-- Description>Test variable against expected value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Default/01/Enhanced/02/Extreme/03" example="Default">
</RequiredValue>
   </eolvar>
   <eolvar name="RCFG/ZTC">
      <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
      <!-- Description>Test variable against expected value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Disabled/00/Enabled/01" example="Disabled"> </
RequiredValue>
   </eolvar>
```

```
    <eolvar name="Processor Emulation">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Test variable against expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="No Emulation/00/vPro/01/Core/02/Celerno/03/Pentium/04/
Xeon/05/Xeon Manageability Capable/06" example="No Emulation"> </RequiredValue>
    </eolvar>
    <eolvar name="PROC_MISSING">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Test variable against expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="No onboard glue logic/ff" example="No onboard glue
logic"> </RequiredValue>
    </eolvar>
    <eolvar name="PKI Domain Name Suffix">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Test variable against expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="String" example="Any"> </RequiredValue>
    </eolvar>
    <eolvar name="PAVP Supported">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Test variable against expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="No/00/Yes/01" example="No"> </RequiredValue>
    </eolvar>
    <eolvar name="Opt-in Policy">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Test variable against expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
```

```
      <!-- Dependencies>CORP</Dependencies -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Hex number with 0x prefix." example="0x00"> </
RequiredValue>
   </eolvar>
   <eolvar name="OEM Tag">
      <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
      <!-- Description>Test variable against expected value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Hex number with 0x prefix." example="0x00000000"> </
RequiredValue>
   </eolvar>
   <eolvar name="OEM Default Certificate Stream">
      <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
      <!-- Description>Test variable against expected value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Hex number with 0x prefix."
example="0x000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000"> </RequiredValue>
   </eolvar>
   <eolvar name="OEM Default Certificate Friendly Name">
      <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
      <!-- Description>Test variable against expected value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="String" example="Any"> </RequiredValue>
   </eolvar>
   <eolvar name="OEM Default Certificate Active">
      <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
      <!-- Description>Test variable against expected value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or ErrAction -->
```

```
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="False/00/Not Active/00/True/01/Active/01"
example="False"> </RequiredValue>
    </eolvar>
    <eolvar name="OEM Default Certificate 5 Stream">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Test variable against expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Hex number with 0x prefix."
example="0x000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000"> </RequiredValue>
    </eolvar>
    <eolvar name="OEM Default Certificate 5 Friendly Name">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Test variable against expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="String" example="Any"> </RequiredValue>
    </eolvar>
    <eolvar name="OEM Default Certificate 5 Active">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Test variable against expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="False/00/Not Active/00/True/01/Active/01"
example="False"> </RequiredValue>
    </eolvar>
    <eolvar name="OEM Default Certificate 4 Stream">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Test variable against expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Hex number with 0x prefix."
```

```
example="0x000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000"> </RequiredValue>
    </eolvar>
    <eolvar name="OEM Default Certificate 4 Friendly Name">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Test variable against expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="String" example="Any"> </RequiredValue>
    </eolvar>
    <eolvar name="OEM Default Certificate 4 Active">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Test variable against expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="False/00/Not Active/00/True/01/Active/01"
example="False"> </RequiredValue>
    </eolvar>
    <eolvar name="OEM Default Certificate 3 Stream">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Test variable against expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Hex number with 0x prefix."
example="0x000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000"> </RequiredValue>
    </eolvar>
    <eolvar name="OEM Default Certificate 3 Friendly Name">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Test variable against expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="String" example="Any"> </RequiredValue>
    </eolvar>
    <eolvar name="OEM Default Certificate 3 Active">
```

```
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Test variable against expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="False/00/Not Active/00/True/01/Active/01"
example="False"> </RequiredValue>
    </eolvar>
    <eolvar name="OEM Default Certificate 2 Stream">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Test variable against expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Hex number with 0x prefix."
example="0x000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000"> </RequiredValue>
    </eolvar>
    <eolvar name="OEM Default Certificate 2 Friendly Name">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Test variable against expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="String" example="Any"> </RequiredValue>
    </eolvar>
    <eolvar name="OEM Default Certificate 2 Active">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Test variable against expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="False/00/Not Active/00/True/01/Active/01"
example="False"> </RequiredValue>
    </eolvar>
    <eolvar name="OEM Customizable Certificate 3 Stream">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Test variable against expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Hex number with 0x prefix."
example="0x000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000"> </RequiredValue>
    </eolvar>
    <eolvar name="OEM Customizable Certificate 3 Friendly Name">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Test variable against expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="String" example="Any"> </RequiredValue>
    </eolvar>
    <eolvar name="OEM Customizable Certificate 3 Active">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Test variable against expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="False/00/Not Active/00/True/01/Active/01"
example="False"> </RequiredValue>
    </eolvar>
    <eolvar name="OEM Customizable Certificate 2 Stream">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Test variable against expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Hex number with 0x prefix."
example="0x000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000"> </RequiredValue>
    </eolvar>
    <eolvar name="OEM Customizable Certificate 2 Friendly Name">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Test variable against expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="String" example="Any"> </RequiredValue>
    </eolvar>
    <eolvar name="OEM Customizable Certificate 2 Active">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Test variable against expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="False/00/Not Active/00/True/01/Active/01"
example="False"> </RequiredValue>
    </eolvar>
    <eolvar name="OEM Customizable Certificate 1 Stream">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Test variable against expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Hex number with 0x prefix."
example="0x000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000"> </RequiredValue>
    </eolvar>
    <eolvar name="OEM Customizable Certificate 1 Friendly Name">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Test variable against expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="String" example="Any"> </RequiredValue>
    </eolvar>
    <eolvar name="OEM Customizable Certificate 1 Active">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Test variable against expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
```

```
       <RequiredValue format="False/00/Not Active/00/True/01/Active/01" 
example="False"> </RequiredValue>
   </eolvar>
   <eolvar name="ODM ID used by Intel(R) Services">
       <!-- The commented fields below CANNOT be edited. Any edits will be ignored by 
the tool -->
       <!-- Description>Test variable against expected value</Description -->
       <!-- IntelRequired>False</IntelRequired -->
       <!-- Dependencies></Dependencies -->
       <!-- End of uneditable fields -->
       <!-- Please edit the fields below ONLY with the State or ErrAction -->
       <State>Disabled</State>
       <ErrAction>ErrorContinue</ErrAction>
       <RequiredValue format="32 hex pairs with space between pairs" example="04 AB 
F3 45 03 1D EF A2 B7 E8 98 79 10 45 AB DE F2 35 49 A0 01 35 78 29 37 AB DE EF FA 10 
EF 33"> </RequiredValue>
   </eolvar>
   <eolvar name="Manageability Application initial power-up state">
       <!-- The commented fields below CANNOT be edited. Any edits will be ignored by 
the tool -->
       <!-- Description>Test variable against expected value</Description -->
       <!-- IntelRequired>False</IntelRequired -->
       <!-- Dependencies>CORP</Dependencies -->
       <!-- End of uneditable fields -->
       <!-- Please edit the fields below ONLY with the State or ErrAction -->
       <State>Disabled</State>
       <ErrAction>ErrorContinue</ErrAction>
       <RequiredValue format="Disabled/00/Enabled/01" example="Disabled"> </
RequiredValue>
   </eolvar>
   <eolvar name="Manageability Application Supported">
       <!-- The commented fields below CANNOT be edited. Any edits will be ignored by 
the tool -->
       <!-- Description>Test variable against expected value</Description -->
       <!-- IntelRequired>False</IntelRequired -->
       <!-- Dependencies>CORP</Dependencies -->
       <!-- End of uneditable fields -->
       <!-- Please edit the fields below ONLY with the State or ErrAction -->
       <State>Disabled</State>
       <ErrAction>ErrorContinue</ErrAction>
       <RequiredValue format="No/00/Yes/01" example="No"> </RequiredValue>
   </eolvar>
   <eolvar name="MEBxPassword">
       <!-- The commented fields below CANNOT be edited. Any edits will be ignored by 
the tool -->
       <!-- Description>Test variable against expected value</Description -->
       <!-- IntelRequired>False</IntelRequired -->
       <!-- Dependencies>CORP</Dependencies -->
       <!-- End of uneditable fields -->
       <!-- Please edit the fields below ONLY with the State or ErrAction -->
       <State>Disabled</State>
       <ErrAction>ErrorContinue</ErrAction>
       <RequiredValue format="32 hex pairs with space between pairs" example="04 AB 
F3 45 03 1D EF A2 B7 E8 98 79 10 45 AB DE F2 35 49 A0 01 35 78 29 37 AB DE EF FA 10 
EF 33"> </RequiredValue>
```

```
    </eolvar>
    <eolvar name="MCTP Device Ports">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by 
the tool -->
        <!-- Description>Test variable against expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Hex number with 0x prefix." example="0x00000000"> </
RequiredValue>
    </eolvar>
    <eolvar name="LSPCON Port Configuration">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by 
the tool -->
        <!-- Description>Test variable against expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="32 hex pairs with space between pairs" example="04 AB 
F3 45 03 1D EF A2 B7 E8 98 79 10 45 AB DE F2 35 49 A0 01 35 78 29 37 AB DE EF FA 10 
EF 33"> </RequiredValue>
    </eolvar>
    <eolvar name="LAN Power Well">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by 
the tool -->
        <!-- Description>Test variable against expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Core Well/00/Sus Well/01/ME Well/02/SLP_LAN#(MGPIO3)/
03" example="Core Well"> </RequiredValue>
    </eolvar>
    <eolvar name="KVM Redirection Supported">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by 
the tool -->
        <!-- Description>Test variable against expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="No/00/Yes/01" example="No"> </RequiredValue>
    </eolvar>
    <eolvar name="KVM">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
```

```
the tool -->
      <!-- Description>Test variable against expected value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Disabled/00/Enabled/01" example="Disabled"> </
RequiredValue>
   </eolvar>
   <eolvar name="Intel(R) Precise Touch Technology">
      <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
      <!-- Description>Test variable against expected value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="No/00/Yes/01" example="No"> </RequiredValue>
   </eolvar>
   <eolvar name="Intel(R) PTT initial power-up state">
      <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
      <!-- Description>Test variable against expected value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Disabled/00/Enabled/01" example="Disabled"> </
RequiredValue>
   </eolvar>
   <eolvar name="Intel(R) PTT Supported">
      <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
      <!-- Description>Test variable against expected value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Disabled/00/Enabled/01" example="Disabled"> </
RequiredValue>
   </eolvar>
   <eolvar name="Intel(R) ME Region Flash Protection Override">
      <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
      <!-- Description>Test variable against expected value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
```

```
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="False/00/True/01" example="False"> </RequiredValue>
    </eolvar>
    <eolvar name="Intel(R) ME Network Services Supported">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by the tool -->
        <!-- Description>Test variable against expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Yes/00/No/01" example="Yes"> </RequiredValue>
    </eolvar>
    <eolvar name="Intel(R) ME CLINK Signal Enabled">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by the tool -->
        <!-- Description>Test variable against expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Disabled/00/Enabled/01" example="Disabled"> </RequiredValue>
    </eolvar>
    <eolvar name="Intel(R) AMT Watchdog Automatic Reset Enabled">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by the tool -->
        <!-- Description>Test variable against expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="No/00/Yes/01" example="No"> </RequiredValue>
    </eolvar>
    <eolvar name="Intel(R) AMT Supported">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by the tool -->
        <!-- Description>Test variable against expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="No/00/Yes/01" example="No"> </RequiredValue>
    </eolvar>
    <eolvar name="Intel(R) AMT Idle Timeout">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Test variable against expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Hex number with 0x prefix." example="0x0000"> </
RequiredValue>
    </eolvar>
    <eolvar name="Integrated Sensor Hub Supported">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Test variable against expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Disabled/00/Enabled/01" example="Disabled"> </
RequiredValue>
    </eolvar>
    <eolvar name="Host Name">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Test variable against expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="String" example="Any"> </RequiredValue>
    </eolvar>
    <eolvar name="Firmware Update OEM ID">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Test variable against expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Hex" example="00000000-0000-0000-0000-000000000000">
</RequiredValue>
    </eolvar>
    <eolvar name="Firmware KVM Screen Blanking">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Test variable against expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="No/00/Yes/01" example="No"> </RequiredValue>
    </eolvar>
    <eolvar name="Firmware Dynamic Application Loader">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by the tool -->
        <!-- Description>Test variable against expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="No/00/Yes/01" example="No"> </RequiredValue>
    </eolvar>
    <eolvar name="FWUpdLcl">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by the tool -->
        <!-- Description>Test variable against expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Disabled/00/Enabled/01" example="Disabled"> </RequiredValue>
    </eolvar>
    <eolvar name="Embedded Host Based Configuration Enabled">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by the tool -->
        <!-- Description>Test variable against expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Disabled/00/Enabled/01" example="Disabled"> </RequiredValue>
    </eolvar>
    <eolvar name="Domain Name">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by the tool -->
        <!-- Description>Test variable against expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="String" example="Any"> </RequiredValue>
    </eolvar>
    <eolvar name="Delayed Authentication Mode Configuration">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Test variable against expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Disabled/00/Enabled/01" example="Disabled"> </
RequiredValue>
    </eolvar>
    <eolvar name="Debug Override Production Silicon">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Test variable against expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Hex number with 0x prefix." example="0x00000000"> </
RequiredValue>
    </eolvar>
    <eolvar name="Debug Override Pre-Production Silicon">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Test variable against expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Hex number with 0x prefix." example="0x00000000"> </
RequiredValue>
    </eolvar>
    <eolvar name="Config Server IPv6/IPv4 Port">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Test variable against expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Hex number with 0x prefix." example="0x0000"> </
RequiredValue>
    </eolvar>
    <eolvar name="Config Server IPv6/IPv4 Address">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Test variable against expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="String" example="Any"> </RequiredValue>
    </eolvar>
    <eolvar name="Config Server FQDN">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Test variable against expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="String" example="Any"> </RequiredValue>
    </eolvar>
    <eolvar name="Automatic Built in Self Test">
        <!-- The commented fields below CANNOT be edited. Any edits will be ignored by
the tool -->
        <!-- Description>Test variable against expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Disabled/00/Enabled/01" example="Disabled"> </
RequiredValue>
    </eolvar>
    <!-- END OF EOL VAR TESTS -->
</memanuf_config>
```

Lines which start with <! -- -- > are comments. They are also used to inform users of the available test group names and the names of specific checks that are included in each test that Intel® MEManuf recognizes.

To select which test items to run: Modify the State item as <State> Enabled </ State>to enable the subtest
Wherever there is a section for Required Value, Example: <RequiredValue format="major_ver.minor_ver" example="0.6"> </RequiredValue>, Please enter the required values in the xml file which will be used by MEManuf for testing.

Here is the example that explain how to use this feature:
<eolconfig name="PTT FPF">

```
                <!-- The commented fields bellow CANNOT be edited. Any edits will be ignored by the tool -->
                <!-- Description>Check ptt against expected value</Description -->
                <!-- IntelRequired>False</IntelRequired -->
                <!-- Dependencies>PlatformTrust</Dependencies -->
                <!-- End of uneditable fields -->
                <!-- Please edit the fields below ONLY with the State or ErrAction -->
                <State>Enabled</State>
                <ErrAction>ErrorContinue</ErrAction>
                <RequiredValue format="Not set/Enabled/Disabled" example="Not set"> </RequiredValue>
        </eolconfig>
```

5.4.3 MEManuf –EOL Variable Check

MEManuf `-EOL variable` check is designed to check the Intel® ME settings on the platform before shipping. To minimize the security risk in exposing this in an end-user environment, this test is only available in Intel® ME manufacturing mode or No EOP Message Sent.

Note: -EOL Variable check. The system must be in Intel® ME manufacturing mode when -EOL Variable check is run or No EOP Message Sent.

5.4.4 MEManuf –EOL Config Check

MEManuf `-EOL Config` check is designed to check the Intel® ME-related configuration before shipping. Running Intel-recommended tests before shipping is highly recommended.

Table 5-3. MEManuf - EOL Config Tests

Test	Expected Configuration
EOP status check	Enabled
Intel® ME VSCC check	Set according to the Intel-recommended value.
BIOS VSCC check	Set according to the Intel-recommended value.
Intel® ME Manufacturing Mode status	Disabled.
Flash Region Access Permissions	Set according to the Intel-recommended value.
Flash Descriptor Override Strap (HDA_SDO)	Disabled.
MAC address	None, all 0, or f
Wireless MAC address	None, all 0, or f
System UUID	None, all 0.

Note: `–EOL Config` check. If the system is in Intel® ME manufacturing mode when `–EOL Config` check is run there will be an error report or No EOP Message Sent.

5.4.5 Output/Result

The following test results can be displayed at the end-of-line checking:

- Pass – all tests passed.
- Pass with warning – all tests passed except the tests that were modified by the customer to give a warning on failure. (This modification does not apply to Intel-recommended tests.
- Fail with warning - all tests passed except some Intel-recommended tests that were modified by the customer to give a warning on failure.
- Fail - any customer-defined error occurred in the test.

5.5 Examples

5.5.1 Example for Consumer Intel® ME FW SKU

```
MEManuf –verbose

Intel(R) MEManuf Version: XX.XX.XX.xxxx
Copyright(C) 2005 - 2014, Intel Corporation. All rights reserved.

   FW   Status   Register1:   0x86000255
          FW   Status   Register2:   0x6085012E
          FW   Status   Register3:   0x00000000
          FW   Status   Register4:   0x00004000
          FW   Status   Register5:   0x00000000
          FW   Status   Register6:   0x00000000

  CurrentState:                         Normal
  ManufacturingMode:                    Enabled
  FlashPartition:                       Valid
  OperationalState:                     CM0 with UMA
  InitComplete:                         Complete
  BUPLoadState:                         Success
  ErrorCode:                            No Error
  ModeOfOperation:                      Normal
  ICC:                                  Valid OEM data, ICC programmed

Get FWU info command...done

Get FWU version command...done

Get FWU feature state command...done

Get ME FWU platform type command...done

Get ME FWU feature capability command...done
Feature enablement is 0x1001C60
```

```
gFeatureAvailability value is 0x1
System is running on consumer/4M image, start Intel(R) ME Runtime Test
OEM ICC data valid and programmed correctly

Request Intel(R) ME test result command...done
vsccommn.bin was created on 23:32:28 05/05/2010 GMT
SPI Flash ID #1 ME VSCC value is 0x2005
SPI Flash ID #1 (ID: 0xEF4017) ME VSCC value checked
SPI Flash ID #1 BIOS VSCC value is 0x2005
SPI Flash ID #1 (ID: 0xEF4017) BIOS VSCC value checked
SPI Flash ID #2 ME VSCC value is 0x2005
SPI Flash ID #2 (ID: 0xEF4017) ME VSCC value checked
SPI Flash ID #2 BIOS VSCC value is 0x2005
SPI Flash ID #2 (ID: 0xEF4017) BIOS VSCC value checked
FPBA value is 0x0
No Intel Wireless device was found

Request Intel(R) ME Runtime BIST test command...done

Get Intel(R) ME test data command...done
Total of 22 Intel(R) ME test result retrieved
Micro Kernel - Blob Manager: Set - Passed
Micro Kernel - Blob Manager: Get - Passed
Micro Kernel - Blob Manager: Remove - Passed
Policy Kernel - SMBus: Read byte - Passed
Policy Kernel - ME Password: Valid MEBx password - Passed
Policy Kernel - ME Configuration: Wlan Power Well - Passed
Policy Kernel - ME Configuration: CPU Missing Logic - Passed
Policy Kernel - ME Configuration: CM3 Power Rails Available - Passed
Policy Kernel - Embedded Controller: Get power source - Passed
Common Services - General: Low power idle timeout - Passed
Common Services - Provisioning: Valid MEBX password change policy - Passed
Common Services - Provisioning: Zero-Touch configuration enabled - Passed
Common Services - Provisioning: Client Config mode is valid - Passed
Common Services - General: Vlan not enabled on mobile - Passed
Common Services - Provisioning: Both PID and PPS are set - Passed
Common Services - Provisioning: MEBX password set when PID and PPS set - Passed
Common Services - Wireless LAN: Connectivity to NIC - Skipped
AMT - Privacy Level: Valid Privacy Level settings - Passed

Clear Intel(R) ME test data command...done

MEManuf Test Passed
```

5.5.2 Example for Corporate Intel® ME FW SKU

```
MEManuf –verbose

Intel(R) MEManuf Version: XX.XX.XX.xxxx
Copyright(C) 2005 - 2014, Intel Corporation. All rights reserved.


FW   Status   Register1:   0x86000255
FW   Status   Register2:   0x6085012E
FW   Status   Register3:   0x00000000
FW   Status   Register4:   0x00004000
FW   Status   Register5:   0x00000000
FW   Status   Register6:   0x00000000

  CurrentState:                        Normal
  ManufacturingMode:                   Enabled
  FlashPartition:                      Valid
  OperationalState:                    CM0 with UMA
  InitComplete:                        Complete
  BUPLoadState:                        Success
  ErrorCode:                           No Error
  ModeOfOperation:                     Normal
  ICC:                                 Valid OEM data, ICC programmed

Get FWU info command...done

Get FWU version command...done

Get FWU feature state command...done

Get ME FWU platform type command...done

Get ME FWU feature capability command...done
Feature enablement is 0xDF65C65
gFeatureAvailability value is 0x1

Request Intel(R) ME test result command...done

ME initialization state valid
ME operation mode valid
Current operation state valid
ME error state valid
Verifying FW Status Register1...done
OEM ICC data valid and programmed correctly

Request Intel(R) ME test result command...done
vsccommn.bin was created on 03:08:01 01/25/2011 GMT
SPI Flash ID #1 ME VSCC value is 0x2005
SPI Flash ID #1 (ID: 0xEF4017) ME VSCC value checked
SPI Flash ID #1 BIOS VSCC value is 0x2005
SPI Flash ID #1 (ID: 0xEF4017) BIOS VSCC value checked
FPBA value is 0x0
No Intel Wireless device was found

Request Intel(R) ME Full BIST test command...done

Get Intel(R) ME test data command...done
Total of 31 Intel(R) ME test result retrieved

Common Services - LAN: Connectivity to NIC in CM3 - Passed

MicroKernel - Internal Hardware Tests: Internal Hardware Tests - Passed
```

```
Policy Kernel - SMBus: Read byte - Passed
Policy Kernel - ME Password: Validate MEBx password - Passed

MicroKernel - Blob Manager: Set - Passed
MicroKernel - Blob Manager: Get - Passed
MicroKernel - Blob Manager: Remove - Passed

Policy Kernel - ME Configuration: Wlan Power Well - Passed
Policy Kernel - ME Configuration: PROC_MISSING - Passed
Policy Kernel - ME Configuration: CM3 Power Rails Available - Passed
Policy Kernel - Embedded Controller: Power source type - Passed

Common Services - General: Low power idle timeout - Passed
Common Services - Privacy Level: Valid Privacy Level settings - Passed
Common Services - General: Vlan not enabled on mobile - Passed
Common Services - Provisioning: Both PID and PPS are set - Passed
Common Services - Provisioning: MEBX password set when PID and PPS set - Passed
Common Services - LAN: Connectivity to NIC in CM0 - Passed

AMT - Power: Valid LAN power well - Passed
AMT - Power: Valid WLAN power well (Mobile) - Failed
Error 9357: WLAN power well setting is set incorrectly
AMT - KVM: USBr is enabled when KVM is enabled - Passed
AMT - EC: Basic connectivity - Passed
AMT - Hardware Inventory: BIOS tables - Passed
AMT - KVM: Compare engine - Passed
AMT - KVM: Compression engine - Passed
AMT - KVM: Sampling engine - Skipped
AMT - KVM: VDM engine - Passed
AMT - USBr: Hardware - Passed

Clear Intel(R) ME test data command...done

Error 9296: MEManuf Test Failed
```

§ §

6 Intel® MEInfo

MEInfoWin and Intel® MEInfo provide a simple test to check whether the Intel® ME FW is alive. Both tools perform the same test; query the Intel® ME FW including Intel® AMT – and retrieve data.

Table 18 contains a list of the data that each tool returns.

The Windows® version of MEInfo (MEInfoWin) requires administrator privileges to run under Windows® OS. The user needs to use the Run as Administrator option to open the CLI in Windows® 10.

6.1 Windows® PE Requirements

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

Meinfo reports an LMS error. This behavior is expected as the LMS driver cannot be installed on Windows® PE.

6.2 Usage

The executable can be invoked by:

```
MEInfo.exe [-EXP] [-H|?] [-VER] [-FITVER] [-FEAT] [-VALUE] [-FWSTS]
[-VERBOSE] [-PAGE]

MEInfo.efi [-EXP] [-H] [-VER] [-FITVER] [-FEAT] [-VALUE] [-FWSTS]
[-VERBOSE] [-PAGE]
```

Table 6-1. Intel® MEInfo Command Line Options

Option	Description
-FEAT <name> <column> -VALUE <value>	Compares the value of the given feature name (and optional column name for features displayed in a table) with the value in the command line. If the feature name or value is more than one word, the entire name or value must be enclosed in quotation marks (together with the optional column name). For example –feat "PTT FPF". If the values are identical, a message indicating success appears. If the values are not identical, the actual value of the feature is returned. Only one feature may be requested in a command line.
-FITVER	Displays FIT version information

Option	Description
-FEAT <name> <column>	Retrieves the current value for the specified feature (and optional column name for features displayed in a table). If the feature name is more than one word, the entire feature name (and optional column name) must be enclosed in quotation marks. For example –feat "PTT FPF". The feature name entered must be the same as the feature name displayed by Intel® MEINFO. Intel® MEINFO can retrieve all of the information detailed below. However, depending on the SKU selected, some information may not appear. **Note:** For the EFI shell version you need to add additional "^" to enclose the text string in order for it to be properly parsed. **Example:** MEINFO.efi –feat "^"BIOS boot state"^"
–FWSTS	Decodes the Intel® ME FW status register value field and breaks it down into the following bit definitions for easy readability:

```
FW Status Register1: 0x90000255
FW Status Register2: 0x00F10506
FW Status Register3: 0x00000020
FW Status Register4: 0x00004004
FW Status Register5: 0x00000000
FW Status Register6: 0x00400000

    CurrentState:                             Normal
    ManufacturingMode:                        Enabled
    FlashPartition:                           Valid
    OperationalState:                         CM0 with UMA
    InitComplete:                             Complete
    BUPLoadState:                             Success
    ErrorCode:                                No Error
    ModeOfOperation:                          Normal
    SPI Flash Log:                            Present
    Phase:                                    ROM/Preboot
    ME File System Corrupted:                 No
    PhaseStatus:                              PROTECTED_START
    FPF and ME Config Status:                 Not committed
```

Option	Description
-VERBOSE <filename>	Turns on additional information about the operation for debugging purposes. This option has to be used together with the above mentioned option(s). Failure to do so generates the error: "Error 9254: Invalid command line option". This option works with no option and `–feat`.
-H or -?:	Displays the list of command line options supported by the Intel® MEINFO tool. **Note:** Use -H for help when running in the EFI Shell.
-VER	Shows the version of the tools.
- PAGE	When it takes more than one screen to display all the information, this option lets the user pause the display and then press any key to continue on to the next screen.
-EXP	Shows examples about how to use the tools.
No option:	If the tool is invoked without parameters, it reports information for all components listed in Table 6-2 below for full SKU FW.

Table 6-2. List of Components that Intel® MEINFO Displays

Feature Name	Feature Data Source (Intel® ME Kernel/ Intel® AMT/ SW/ Other)	Consumer SKU	Corporate SKU	Specific Feature Dependency	Field Value
Tools Version	SW (Intel® MEInfo)	X	X	N/A	Version string Example: 12.x.y.ZZZZ; where x=minor, y = HF/MR, ZZZZ = Build Number.
BIOS Version	Intel® ME Kernel	X	X	MEBx needs to be present. Not available on Corporate Sku	Version string
MEBx Version	Intel® ME Kernel	X	X	MEBx needs to be present. Not available on Corporate Sku	Version string 12.x.y.ZZZZ; where x=minor, y = HF/MR, ZZZZ = Build Number.
GbE Version	Other (Directly reading from SPI)	X	X	GbE Region to be present in the image	A version string
PMC Firmware Viersion	Other (Directly reading from SPI)	X	X	PMC Region to be present in the image	A version string Unknown if partition does not exist. O if empty
Descriptor Version	Other (Directly reading from SPI)	X	X	SPI Image	A version string
VendorID	Intel® ME Kernel	X	X	N/A	A number (in Hex)

Feature Name	Feature Data Source (Intel® ME Kernel/ Intel® AMT/ SW/ Other)	Consumer SKU	Corporate SKU	Specific Feature Dependency	Field Value
FW Version	Intel® ME Kernel	X	X	N/A	Version string XX.x.y.ZZZZ A B; where XX=major, x=minor, y = HF/MR, ZZZZ = Build Number, A=LP/H, B=SKU type [Consumer/ Corporate].
Security Version (SVN)	Intel® ME Kernel	X	X	N/A	Version Number
LMS version*	Other (Reading Windows® registry entries)	X	X	Only when Windows® LMS driver is installed	A version string
Intel® MEI Driver version*	Other (Reading Windows® registry entries	X	X	Only when Windows® Intel® MEI driver is installed	A version string
Wireless Driver/ Hardware Version*	Other (Reading Windows® registry entries)	X	X	Only when wireless HW is present, and wireless Windows® driver is installed	A version string
PCH Information	Intel® ME Kernel	X	X	N/A	Display of PCH Information including: • Version • Device ID • Step Data • SKU Type • PCH Replacement Counter • PCH Replacement Counter State • PCH Unlocked State

Feature Name	Feature Data Source (Intel® ME Kernel/ Intel® AMT/ SW/ Other)	Consumer SKU	Corporate SKU	Specific Feature Dependency	Field Value
FW Capabilities	Intel® ME Kernel	X	X	N/A	Combination of feature name list breakdown (with a Hexadecimal value) *This is a display of the Feature State for the Intel® ME. Is enabled / disabled on the system. Each bit in the value represents a feature state. Intel® ME features including Full manageability, standard manageability, Anti-theft technology etc. Information Includes: • Intel(R) Active Management Technology • Protect Audio Video Path • Intel(R) Dynamic Application Loader • Service Advertisement & Discovery • Intel(R) Platfrom Trust Technology • Persistent RTC and Memory • Intel(R) Precise Touch and Stylus
FW Type	Intel® ME Kernel	X	X	N/A	Pre-Production/Production
Intel® AMT State	Intel® ME Kernel		X	Both Full Manageability and Manageability Application have to be PRESENT (Capable)	Enabled/Disabled

Feature Name	Feature Data Source (Intel® ME Kernel/ Intel® AMT/ SW/ Other)	Consumer SKU	Corporate SKU	Specific Feature Dependency	Field Value
TLS	Intel® ME Kernel	X	X	N/A	Enabled/Disabled
Last Intel® ME Reset Reason	Intel® ME Kernel	X	X	N/A	Power up/ Firmware reset/ Global system reset/ Unknown
Local FWUpdate	Intel® ME Kernel	X	X	N/A	Enabled/Disabled/ Password Protected
BIO	Other (Directly reading from SPI)	X	X	N/A	Enabled/Disabled/ Unknown
GbE Config Lock	Other (Directly reading from SPI)	X	X	N/A	Enabled/Disabled/ Unknown
Host Read Access to Intel® ME	Other (Directly reading from SPI)	X	X	N/A	Enabled/Disabled/ Unknown
Host Write Access to Intel® ME	Other (Directly reading from SPI)	X	X	N/A	Enabled/Disabled/ Unknown
Host Read Access to EC/Host Write Access to EC	Other (Directly reading from SPI)	X	X	N/A	Enabled/Disabled/ Unknown
SPI Flash ID	Other (Directly reading from SPI)	X	X	Only when there are flash parts HW installed	A JEDEC ID number (in Hex)

Feature Name	Feature Data Source (Intel® ME Kernel/ Intel® AMT/ SW/ Other)	Consumer SKU	Corporate SKU	Specific Feature Dependency	Field Value
ME/BIOS VSCC register values	Other (Directly reading from SPI)	X	X	Only when there are flash parts HW installed	A 32bit VSCC number (in Hex)
BIOS Boot State	Intel® ME Kernel	X	X	N/A	Pre Boot/ In Boot/ Post Boot
OEM Id	Intel® ME Kernel	X	X	Only if fw image supports OEM Id	UUID for OEM to check during FW Update
Capability Licensing Service	Intel® ME Kernel	X	X	Not available on Corporate Sku. Not shown unless Fw feature capability supports it	Enabled/Disabled
OEM Tag	Intel® ME Kernel	X	X	N/A	A 32bit Hexadecimal number
Report on Revenue Sharing ID Fields	Intel® ME Kernel Firmware Host Interface	Both	All	N/A	3 slot of 32-bit integer values (in Hex)
M3 Autotest	Intel® ME Kernel		X	FIT CM3 Autotest Enabled set to 'true'	Enabled/Disabled
C-Link Status	Intel® ME Kernel		X	Intel® Wireless LAN	Enabled/Disabled

Feature Name	Feature Data Source (Intel® ME Kernel/ Intel® AMT/ SW/ Other)	Consumer SKU	Corporate SKU	Specific Feature Dependency	Field Value
Link Status	Intel® AMT	X	X	Intel® AMT CEM (a.k.a. Common Service) is used. Not available on Corporate Sku	Link up/down
System UUID	Intel® AMT	N/A	X	AMT CEM (a.k.a. Common Service) is used. Not available on Corporate Sku	UUID of the system
Configuration State	Intel® AMT	N/A	X	AMT CEM (a.k.a. Common Service) is used. Not available on Consumer Sku	Not started/ In process/ Completed/ Unknown
MAC Address	Intel® AMT	N/A	X	AMT CEM (a.k.a. Common Service) is used only when wired Hw is present. Not available on Consumer Sku	A MAC address (in Hex separated by "=")

Feature Name	Feature Data Source (Intel® ME Kernel/ Intel® AMT/ SW/ Other)	Consumer SKU	Corporate SKU	Specific Feature Dependency	Field Value
Wireless MAC Address	Intel® AMT	N/A	X	AMT CEM (a.k.a. Common Service) is used only when wireless HW is present. Not available on Consumer Sku	A MAC address (in Hex separated by "=")
IPv4 Address (Wired and Wireless)	Intel® AMT	N/A	X	Intel® AMT CEM (a.k.a. Common Service) is used only when wired/ wireless Hw is present. Not available on Consumer Sku	IPv4 IP address (in decimal separated by ".")
IPv6 Address (Wired and Wireless)	Intel® AMT	N/A	X	Intel® AMT CEM (a.k.a. Common Service) is used only when wired/ wireless Hw is present. Not available on Consumer Sku	All IPv6 IP addresses

Feature Name	Feature Data Source (Intel® ME Kernel/ Intel® AMT/ SW/ Other)	Consumer SKU	Corporate SKU	Specific Feature Dependency	Field Value
IPv6 enabled (Wired and Wireless)	Intel® AMT	N/A	X	Intel® AMT CEM (a.k.a. Common Service) is used only when wired/ wireless Hw is present. Not available on Consumer Sku	Enabled/Disabled
Privacy / Security Level	Intel® AMT	N/A	X	Not available on Consumer SKU. Only shown when AMT is enabled	Default/Enhanced/ Extreme/Unknown
Provisioning Mode	Intel® AMT	N/A	X	Intel® AMT CEM (a.k.a. Common Service) is used only when wired/ wireless Hw is present. Not available on Consumer Sku	
FWSTS	Intel® ME Kernel	X	X	N/A	Firmware status, 32bit Hexadecimal numbers and their bit definition breakdown. Available when -fwsts or -verbose are specified.
Wireless Micro-code Mismatch	FWU	Corporate	All	N/A	Yes: FW has detected a ucode mismatch, and partial FWUpdate needs to be performed

Feature Name	Feature Data Source (Intel® ME Kernel/ Intel® AMT/ SW/ Other)	Consumer SKU	Corporate SKU	Specific Feature Dependency	Field Value
Wireless LAN in Firmware	FWU	Corporate	All	N/A	The "friendly name" matching the WLAN ucode in FW
Wireless Micro-code ID in Firmware	FWU	Corporate	All	N/A	The current WLAN ucode in FW
Wireless LAN Hardware	PCI address	Corporate	All	N/A	The "friendly name" of the Wireless LAN hardware installed on the system
Wireless Hardware ID	PCI address	Corporate	All	N/A	The WLAN DeviceID read from PCI space of the installed WLAN on the system
Localized Language	FWU	All	All	N/A	Displaying the language installed in the flash in English
Keybox	Intel® ME Kernel	All	All	N/A	Enabled/Disabled
Intel® PTT Supported	Intel® ME Kernel	All	All	N/A	Yes/No
Intel® PTT Initial Power State	Intel® ME Kernel	All	All	N/A	Enabled/Disabled
PAVP Supported	Intel® ME Kernel	All	All	Platform Protection	Yes/No
Integrated Sensor Hub Initial Power State	Intel® ME Kernel	All	All		Enabled/Disabled
End of Manufacturing Enable	Intel® ME Kernel	All	All		Yes/No

Feature Name	Feature Data Source (Intel® ME Kernel/ Intel® AMT/ SW/ Other)	Consumer SKU	Corporate SKU	Specific Feature Dependency	Field Value
Post Manufacturing NVAR Config Enabled	Intel® ME Kernel	All	All		Yes/No
Minimum Allowed Anti Rollback SVN	Intel® ME Kernel	All	All	BIOS	
Image Anti Rollback SVN	Intel® ME Kernel	All	All	BIOS	
Trusted Computing Base SVN	Intel® ME Kernel	All	All	BIOS	
ACM SVN FPF	Intel® ME Kernel	All	All	BIOS	
KM SVN FPF	Intel® ME Kernel	All	All	BIOS	
BSMM SVN FPF	Intel® ME Kernel	All	All	BIOS	
OEM Public Key Hash FPF	Intel® ME Kernel	All	All	BIOS	SHA-256bit Hash entry (Set once fuses are burned)
OEM Public Key Hash UEP	Intel® ME Kernel	All	All	BIOS	SHA-256bit Hash entry (Value prior to burning fuses)
OEM Public Key Hash ME FW	Intel® ME Kernel	All	All	BIOS	SHA-256bit Hash entry (Value currently in use by FW)
HW Binding	Intel® ME Kernel	All	All	N/A	Enabled/Disabled

Feature Name	**Feature Data Source (Intel® ME Kernel/ Intel® AMT/ SW/ Other)**	**Consumer SKU**	**Corporate SKU**	**Specific Feature Dependency**	**Field Value**
GuC Encryption Key ME	Intel® ME Kernel	All	All	BIOS	256-bit string
Force Boot Guard ACM	Intel® ME Kernel	All	All	BIOS	Yes / No
Key Manifest ID	Intel® ME Kernel	All	All	BIOS	Hash of Public Key to verify Boot Policy Manifest
PTT	Intel® ME Kernel	All	All	BIOS	Enabled / Disabled
SPI Boot Source	Intel® ME Kernel	All	All	BIOS	Enabled / Disabled
Enforcement Policy	Intel® ME Kernel	All	All	BIOS	Unrestricted / Remediation / Restricted
OEM ID	Intel® ME Kernel	All	All	BIOS	Hex Value
TXT Supported	Intel® ME Kernel	All	All	BIOS	Enabled/Disabled
OEM Key Manifest Present	Intel® ME Kernel	All	All	BIOS	Present / Not Present
OEM Platform ID	Intel® ME Kernel	All	All	BIOS	Hex Value
SOC Config Lock	Intel® ME Kernel	All	All	BIOS	Done / Not Done
Persistent PRTC Backup Power	Intel® ME Kernel	All	All	BIOS	Enabled / Disabled
EK Revoke State	Intel® ME Kernel	All	All	BIOS	Revoked / Not Revoked
CPU Debugging	Intel® ME Kernel	All	All	BIOS	Enabled / Disabled

Feature Name	Feature Data Source (Intel® ME Kernel/ Intel® AMT/ SW/ Other)	Consumer SKU	Corporate SKU	Specific Feature Dependency	Field Value
BSP Initialization	Intel® ME Kernel	All	All	BIOS	Enabled / Disabled
Measured Boot	Intel® ME Kernel	All	All	BIOS	Yes / No
Verified Boot	Intel® ME Kernel	All	All	BIOS	Yes / No
Protect BIOS Environment	Intel® ME Kernel	All	All	BIOS	Yes / No
iTouch	SW (Intel® MEInfo)	All	All	iTouch	iTouch information includes: • Device ID • HW Revision ID • FW Revision ID • Frame Size • Feedback Size • Sensor Mode • Maximum Number of Touch Point • SPI Frequency • SPI I/O Mode

6.3 Examples

This is a simple test that indicates whether the FW is alive. If the FW is alive, the test returns device-specific parameters. The output is from the Windows® version. The DOS version does not display the UNS version, Intel® Management Engine Interface, or LMS version numbers.

Note: **If EOM is set, for FPF's the FPF and ME column values both will be displayed**

6.3.1 Consumer Intel® ME FW SKU Sample Output

```
ÿþ
Intel(R) MEInfo Version: 12.0.0.XXXX
Copyright(C) 2005 - 2017, Intel Corporation. All rights reserved.
```

```
Intel(R) ME code versions:

BIOS Version                                 CNLSFWR1.R00.X100.A01.1708151220
MEBx Version                                 12.0.0.XXXX
GbE Version                                  0.2
PMC FW Version                               10.0.2.XXXX
Descriptor Version                           1.0
Vendor ID                                    8086
FW Version                                   12.0.0.XXXX LP Consumer
Security Version (SVN)                       1
LMS Version                                  1726.12.0.XXXX
MEI Driver Version                           1726.12.0.XXXX
Wireless Hardware Version                    Not Available
Wireless Driver Version                      Not Available

PCH Information
    PCH Version                              11
    PCH Device ID                            9D84
    PCH Step Data                            B1
    PCH SKU Type                             Pre-Production ES
    PCH Replacement Counter                  0
    PCH Replacement State                    Disabled
    PCH Unlocked State                       Disabled

FW Capabilities                              0x31109650

    Protect Audio Video Path - PRESENT/ENABLED
    Intel(R) Dynamic Application Loader - PRESENT/ENABLED
    Intel(R) Platform Trust Technology - PRESENT/ENABLED
    Persistent RTC and Memory - PRESENT/ENABLED
    Intel(R) Precise Touch and Stylus - PRESENT/ENABLED

FW Type                                      Pre-Production
TLS                                          Disabled
Last ME reset reason                         Global system reset
Local FWUpdate                               Enabled
BIOS Config Lock                             Enabled
GbE Config Lock                              Enabled
Host Read Access to ME                       Enabled
Host Write Access to ME                      Enabled
Host Read Access to EC                       Enabled
Host Write Access to EC                      Enabled
SPI Flash ID 1                               EF4019
SPI Flash ID 2                               Not Available
BIOS boot State                              Post Boot
OEM ID                                       00000000-0000-0000-0000-000000000000
Capability Licensing Service                 Enabled
OEM Tag                                      0x00000000
Slot 1 Board Manufacturer                    0x00000000
Slot 2 System Assembler                      0x00000000
Slot 3 Reserved                              0x00000000
M3 Autotest                                  Disabled
```

```
C-link Status                               Disabled
EPID Group ID                               0x4DC
Keybox                                      Not Provisioned
Intel(R) PTT Supported                      Yes
Intel(R) PTT initial power-up state         Enabled
PAVP Supported                              Yes
Integrated Sensor Hub Initial Power State   Enabled
End of Manufacturing Enable                 No
Post Manufacturing NVAR Config Enabled      Yes
ACM SVN FPF                                 0x0
KM SVN FPF                                  0x0
BSMM SVN FPF                                0x0
OEM Public Key Hash FPF                     Not set
OEM Public Key Hash UEP
4D19B4F23FF9170C2C46B3D76BF05919A7FA8B6B113DF53C86C0E8003C23A8DC
OEM Public Key Hash ME FW
4D19B4F23FF9170C2C46B3D76BF05919A7FA8B6B113DF53C86C0E8003C23A8DC
HW Binding                                  Disabled


                                            FPF         UEP         ME FW
                                                      *In Use
                                            ---         ---         -----
Key Manifest ID                             Not set     0x1         0x1
PTT                                         Not set     Enabled     Enabled
SPI Boot Source                             Not set     Enabled     Enabled
Enforcement Policy                          Not set     0x0         0x0
OEM ID                                      Not set     0x0         0x0
TXT Supported                               Disabled    Disabled    Disabled
OEM Key Manifest Present                    Not Present Present     Present
OEM Platform ID                             Not set     0x0         0x0
SOC Config Lock                             Not set     Not Done    Not Done
Persistent PRTC Backup Power                Enabled     Enabled     Enabled
EK Revoke State                             Not Revoked Not Revoked Not Revoked
CPU Debugging                               Not set     Enabled     Enabled
BSP Initialization                          Not set     Enabled     Enabled
Measured Boot                               Not set     Disabled    Disabled
Verified Boot                               Not set     Disabled    Disabled
Protect BIOS Environment                    Not set     Disabled    Disabled

Touch - Vendor ID                           Not Available
Touch - Device ID                           Not Available
Touch - HW Revision ID                      Not Available
Touch - FW Revision ID                      Not Available
Touch - Frame Size                          Not Available
Touch - Feedback Size                       Not Available
Touch - Sensor Mode                         Not Available
Touch - Maximum Number of Touch Point       Not Available
Touch - SPI Frequency                       Not Available
Touch - SPI I/O Mode                        Not Available
```

6.3.2 Corporate Intel® ME FW SKU Sample Output

```
Intel (R) MEInfo Version: 12.x.xx.xxxx
Copyright (C) 2005 - 2018, Intel Corporation. All rights reserved.

Intel(R) Manageability and Security Application code versions:

BIOS Version                                 CNLSFWR1.R00.X174.B00.1810301956
MEBx Version                                 12.x.x.xxxx
GbE Version                                  0.2
Descriptor Version                           1.0
Vendor ID                                    8086
FW Version                                   12.x.xx.xxxx LP Corporate
LMS Version                                  1846.xx.x.xxxx
MEI Driver Version                           1828.xx.x.xxxx
Wireless Hardware Version                    2.1.77
Wireless Driver Version                      20.60.2.2

PMC FW Version                               300.x.xx.xxxx
OEM FW Version                               12.x.xx.xxxx
ISHC FW Version                              5.x.x.xxxx
LOCL FW Version                              12.x.xx.xxxx
WCOD FW Version                              12.x.xx.xxxx

PCH Information
    PCH Version                              32
    PCH Device ID                            9D84
    PCH Step Data                            Not Available
    PCH SKU Type                             Pre-Production ES
    PCH Replacement Counter                  0
    PCH Replacement State                    Disabled
    PCH Unlocked State                       Disabled

FW Capabilities                              0x7DF6D655

    Intel(R) Active Management Technology - PRESENT/ENABLED
    Protect Audio Video Path - PRESENT/ENABLED
    Intel(R) Dynamic Application Loader - PRESENT/ENABLED
    Service Advertisement & Discovery - PRESENT/ENABLED
    Intel(R) Platform Trust Technology - PRESENT/ENABLED
    Persistent RTC and Memory - PRESENT/ENABLED
    Intel(R) Precise Touch and Stylus - PRESENT/ENABLED


AMT Global State                             Enabled
Capability Licensing Service                 Enabled
Discrete vPro NIC Enabled                    Disabled
Discrete vPro NIC on board SMBus address     0x49
End of Manufacturing Enable                  No
Local FWUpdate                               Enabled
OEM ID                                       00000000-0000-0000-0000-000000000000
Integrated Sensor Hub Initial Power State    Enabled
Intel(R) PTT Supported                       Yes
Intel(R) PTT initial power-up state          Enabled
OEM Tag                                      0x00
```

```
PAVP Supported                               Yes
Post Manufacturing NVAR Config Enabled       Yes
Privacy/Security Level                       Default
TLS                                          Enabled

FW Type                                      Pre-Production
Intel(R) AMT State                           Enabled
Last ME reset reason                         Global system reset
BIOS Config Lock                             Enabled
GbE Config Lock                              Enabled
Host Read Access to ME                       Enabled
Host Write Access to ME                      Enabled
Host Read Access to EC                       Enabled
Host Write Access to EC                      Enabled
SPI Flash ID 1                               EF4019
SPI Flash ID 2                               Not Available
BIOS boot State                              Post Boot
Link Status                                  Link Up
System UUID                                  88888888-8887-8888-8888-878888888888
MAC Address                                  00-02-01-88-88-88
IPv4 Address                                 192.168.1.0145
Wireless MAC Address                         00-02-01-34-13-e8
Wireless IPv4 Address                        192.168.1.0124
IPv6 Enablement                              Disabled
Wireless IPv6 Enablement                     Disabled
Configuration State                          Completed
Provisioning Mode                            PKI
Slot 1 Board Manufacturer                    0x00000000
Slot 2 System Assembler                      0x00000000
Slot 3 Reserved                              0x00000000
M3 Autotest                                  Disabled
C-link Status                                Enabled
Wireless Micro-code Mismatch                 No
Wireless Micro-code ID in Firmware           0x9DF0
Wireless LAN in Firmware                     Intel(R) Wireless-AC 9560
Wireless Hardware ID                         0x9DF0
Wireless LAN Hardware                        Intel(R) Wireless-AC 9560
Localized Language                           English
Minimum Allowed Anti Rollback SVN            1
Image Anti Rollback SVN                      4
Trusted Computing Base SVN                   1
Re-key needed                                False
HW Binding                                   Disabled
Intel(R) SMLink0b MCTP Address               0x00

Touch - Vendor ID                            Not Available
Touch - Device ID                            Not Available
Touch - HW Revision ID                       Not Available
Touch - FW Revision ID                       Not Available
Touch - Frame Size                           Not Available
Touch - Feedback Size                        Not Available
Touch - Sensor Mode                          Not Available
Touch - Maximum Number of Touch Point        Not Available
Touch - SPI Frequency                        Not Available
Touch - SPI I/O Mode                         Not Available
```

```
                                             FPF         UEP         ME FW
                                                       *In Use
                                             ---         ---         -----
Enforcement Policy                           Not set     0x00        0x00
EK Revoke State                              Not set     Not Revoke  Not Revoke
PTT                                          Not set     Enabled     Enabled
OEM ID                                       Not set     0x00        0x00
OEM Key Manifest Present                     Not set     Present     Present
OEM Platform ID                              Not set     0x00        0x00
OEM Secure Boot Policy                       Not set     0x78        0x78
CPU Debugging                                Not set     Enabled     Enabled
BSP Initialization                           Not set     Enabled     Enabled
Protect BIOS Environment                     Not set     Enabled     Enabled
Measured Boot                                Not set     Enabled     Enabled
Verified Boot                                Not set     Enabled     Enabled
Key Manifest ID                              Not set     0x01        0x01
Persistent PRTC Backup Power                 Not set     Enabled     Enabled
RPMB Migration Done                          Not set     Disabled    Disabled
SOC Config Lock                              Not set     Not Done    Not Done
SPI Boot Source                              Not set     Enabled     Enabled
TXT Supported                                Not set     Disabled    Disabled

ACM SVN FPF                                  Not set
BSMM SVN FPF                                 Not set
KM SVN FPF                                   Not set
OEM Public Key Hash FPF                      Not set
OEM Public Key Hash UEP
4D19B4F23FF9170C2C46B3D76BF05919A7FA8B6B113DF53C86C0E8003C23A8DC
OEM Public Key Hash ME FW
4D19B4F23FF9170C2C46B3D76BF05919A7FA8B6B113DF53C86C0E8003C23A8DC
PTT Lockout Override Counter FPF             Not set
```

6.3.3 Retrieve Current Value of Flash Version

```
C:\ MEINFO.exe -feat "BIOS boot state"
Intel(R) MEINFO Version: XX.XX.XX.xxxx
Copyright(C) 2005 - 2017, Intel Corporation. All rights reserved.

BIOS boot State: Post Boot

> MEINFO.efi -feat “^"BIOS boot state"^”
Intel(R) MEINFO Version: XX.XX.XX.xxxx
Copyright(C) 2005 - 2017, Intel Corporation. All rights reserved.
```

BIOS boot State: Post Boot

6.3.4 Checks Whether Computer Has Completed Set-up and Configuration Process

```
C:\ MEINFO.exe -feat "Setup and Configuration" –value "Not Completed"

Intel(R) MEINFO Version: XX.XX.XX.xxxx
```

```
Copyright(C) 2005 - 2017, Intel Corporation. All rights reserved.

Local FWUpdate: Success - Value matches FW value.


> MEINFO.efi -feat “^"Setup and Configuration"^” –value “^"Not Completed"^”

Intel(R) MEINFO Version: XX.XX.XX.xxxx
Copyright(C) 2005 - 2017, Intel Corporation. All rights reserved.

Local FWUpdate: Success - Value matches FW value.
```

7 Intel® ME Firmware Update

FWUpdate allows an end user, such as an IT administrator, to update Intel® ME FW without having to reprogram the entire flash device. It then verifies that the update was successful.

FWUpdate does not update the BIOS, GbE, or Descriptor Regions. It updates the FW code portion along with the WCOD, LOCL, IUNP and ISH partitions. Intel® FWUpdate updates the entire Intel® ME code area. In addition FWUpdate local can perform a partial update to change / update the WCOD, LOCL, IUNP and ISH portions.

The image file that the FWUpdate tool uses is one of the image files that are generated by the FIT tool. Two images are created automatically by the FIT tool, *_base*.bin and *_full*.bin.

- The *_base*.bin file contains the ME firmware stitched together with the PMC binary only.
- The *_full.bin file contains the ME firmware stitched together with the PMC binary as well as any IUPs and the OEM Key Manifest (when provided).

FWUpdate takes approximately 1-4 minutes to complete depending on the flash device on the system.

After FWUpdate a host reset is needed to complete FW update. The user can also use the `-FORCERESET` option to do this automatically.

Note: In previous generations there were two tools: Intel® ME Local Firmware Update and Intel® ME Remote Firmware Update. Now there is just a local firmware update tool that is called Intel® ME Firmware Update (FWUpdate).

7.1 Requirements

FWUpdLcl.exe is a command line executable that can be run on an Intel® ME-enabled system that needs updated FW.

FW can only be updated when the system is in an S0 state. FW updates are NOT supported in the S3/S4/S5 state.

Intel® ME FWUpdate must be enabled in the Intel® MEBx or through BIOS.

The Intel® ME Interface driver must be installed for running this tool in a Windows® environment.

Note: FWUpdLcl.exe must be run with Administrator privilege for access to the Intel® MEI driver

7.2 Windows® PE Requirements

In order for tools to work under Windows® PE environment, the user will need to manually load a driver by using the .inf file in the Intel® MEI driver installation files. Once the .inf file located, the user will need to use Windows® PE command `drvload *.inf` to load it into the running system each time Windows® PE reboots. Failure to do so causes a tools reporting error.

7.3 Enabling and Disabling Intel® FWUpdate

In Intel® MEBx (or BIOS depending on customer implementation), there is an option to enable/disable local firmware update.

This option supports three value, enabled, disabled and Password protected.

Disabled – does not allow FW to be updated

Enabled – allows FW to be updated

Password Protected – allows the FW to be updated only if a valid Intel® Mebx password is provided using the "-pass" option. If password does not match the tool will display the appropriate error message. The user will have a maximum of three tries before being asked to reboot the system to try again.

For more details, refer Intel® MEBx user guide.

7.4 FWUpdate Flows

7.4.1 Full FWUpdate

This will help allow to update Intel® ME Firmware. If IUP's are present in the payload image along with Intel® ME Firmware, IUP's will also be updated along with Intel® ME as part of the Full FWUpdate.

Global Reset will be required to complete the FWUpdate operation.

PMC Firmware Update: This will be handled as part of the Full FWUpdate flow and cannot be updated on its own. PMC Firmware needs to be stitched with Intel® ME Firmware using Intel® FIT Tool and that image will be used as the payload to Full FWUpdate Flow for updating PMC Firmware.

Intel® ME Firmware Update: This will be handled as part of the Full FWUpdate Flow. Requirement: Only CSE Image won't be allowed as the payload to execute update. Pre-Stitched ME + PMC binary needs to be used as the payload to execute ME update**.**

7.4.2 Partial FWUpdate

This will help allow to update IUP's (Independent Updatable Partitions) only i.e. WLAN micro-code, ISH Firmware, Localization, IUnit Loader etc.

For optional IUP's like ISH Firmware Update only, ISH Firmware can be directly used as the payload to update ISH FW only using Partial FWUpdate. No stitching with Intel® ME Firmware required.

7.5 Usage

Note:In this section, <Image File> refers to an Intel-provided image file of the section of the FW to be updated, not the image file used in FIT to program the entire flash memory.

```
FWUpdLcl.exe [-H|?] [-VER] [-EXP] [-VERBOSE] [-F] [-Y] [-SAVE]

             [-FWVER] [-PARTID] [-ALLOWSV] [-FORCERESET] [-SILENT]

             [-OEMID] [-PARTVER] [-PARTVENDOR]


FWUpdLcl.efi [-H|?] [-VER] [-EXP] [-VERBOSE] [-F] [-Y] [-SAVE]

             [-FWVER] [-PARTID] [-ALLOWSV] [-FORCERESET] [-SILENT]

             [-OEMID] [-PARTVER] [-PARTVENDOR]
```

Table 7-1. Image File Update Options

Option	Description
-VERBOSE [<FILE>]	Verbose. Enables additional information about the tool's operation to be displayed for debugging purposes.
-Y	Ignore warning. If the warning asks for input "Y/N", this flag makes the tool automatically take "y" as the input.
-F <FILE>	File. Specifies the FWUpdate image file to be used for performing an update.
-SAVE <file>	Restore Point. Retrieves an update image from the FW based on the currently running FW. The update image is saved to the user-specified file.
-ALLOWSV	Allow Same Version. Allows the version of the input FW (based on the file input) to be the same as the version of the FW currently on the platform. Without this option, an attempt to perform an update on the same version will not proceed.
-FORCERESET	Force Reset. The tool automatically reboots the system after the update process with FW is complete. The system reboot is necessary for the new FW to take effect. An attempt to update the FW without this option will end with a message telling the user to reset the platform for the changes to take effect.

Option	Description
-OEMID <UUID>	OEM ID. The tool uses the specified OEM ID during the transaction of the new FW image with the Manageability Engine. The purpose of the OEM ID is for manufacturers to have an identifier for their system. Using any other OEM ID value other than what is on the FW running on the target platform results in a failure of the FWUpdate process. The full image (including all necessary flash partitions) flashed to the system can be configured with the Flash Image Tool to specify the OEM ID (this tool specifies a default of zeros for the OEM ID.) If this command line option is not used, the default OEM ID used for the update is zeros. The OEM ID is configured in the existing FW image running on the platform. The OEM ID value is specified in the UUID format (8-4-4-4-12).
–PARTID	This option is always used along with the –F option. The partition ID is requested using the "partid" option. If the requested partition is expected by the Firmware the tool will search for the expected partition in the image provided, extract it and send it to the FW to perform the update. If the expected partition is not found in the image or if the requested partition is not expected by the firmware an error will be returned to the user. Note: For partial FW update the image provided must either be a Full or Partial image. A full image starts with a FPT and contains FTP and NFTP partitions.
-FWVER	Display FW version
-H or -?	Displays the list of command line options supported by the Intel® MEINFO tool. **Note:** Use -H for help when running in the EFI Shell.
-EXP	Shows examples about how to use the tools.
-VER	Shows the version of the tools.
-PARTVER	Display flashed FW partition with its FW Version
-SILENT	Runs FWUpdate in Silent
-PARTVENDOR	Vendor ID of the partition

7.6 Examples

7.6.1 Updates Intel® ME with Firmware Binary File

Note:In order to execute FWUpdLcl in EFI, make sure all the payload files and FWUpdate executable are located in the root folder.

```
This command updates Intel® ME with FW.BIN file. If the firmware on current platform
is newer than then version in FW.BIN file, the tool will prompt a warning to let user
know there will be a firmware downgrade and let user choose Y/N to continue. User can
always use –y to skip this warning automatically. If the firmware on the platform is
the same as the version in FW.BIN, tools will return an error. User can use –allowsv
to allow same version update.

FWUpdLcl.exe –f FW.BIN

EFI:
FWUpdLcl.efi –f FW.BIN
```

7.6.2 Partial Firmware Update

This command will perform a partial update of the FW via Intel® MEI for either the IUPs.

```
FWUpdLcl.exe -f FW.bin -partid <PARTID>

EFI:
FWUpdLcl.efi -f FW.bin -partid <PARTID>
```

Non-Verbose Mode

```
C:\> FWUpdLcl.exe -f FW.bin -partid WCOD

Intel (R) Firmware Update Utility version xx.xx.xx.xxxx
Copyright (C) 2007-2017, Intel Corporation.  All rights reserved.

Communication Mode: MEI
Sending the update image to FW for verification: [COMPLETE]

FW Update: [100%(|)]
FW Update is completed successfully.
```

Verbose Mode

```
C:\> FWUpdLcl.exe -f FW.bin -partid WCOD -verbose

Intel (R) Firmware Update Utility version xx.xx.xx.xxxx
Copyright (C) 2007-2017, Intel Corporation.  All rights reserved.

Communication Mode: MEI
Sending the update image to FW for verification: [COMPLETE]

Firmware last update status = Firmware update success
Firmware last update reset type = 2

FW Update is completed successfully.
```

7.6.3 Display Supported Commands

Display a list of supported command line sequences based on the arguments provided.

The arguments relevant for this usage are any of the command line options with the prefix '-'roved. The tool will display all valid command sequences based on the options provided. Below is an example which displays valid command sequences with the –ipu option

```
C:\> FWUpdLcl.exe -exp partid

Intel (R) Firmware Update Utility version xx.xx.xx.xxxx
Copyright (C) 2007-2017, Intel Corporation.  All rights reserved.


 The parameters provided are supported in the following command-line sequences:
```

```
1. -F  <file> -PARTID [ <Partition ID>] [-FORCERESET] [-VERBOSE [ <file>]]
     [-SILENT] [-Y] [-ALLOWSV]


 Using -EXP without any additional input will display examples of
 common command-line input.


           EFI:
> FWUpdLcl.efi -exp partid

Intel (R) Firmware Update Utility version xx.xx.xx.xxxx
Copyright (C) 2007-2017, Intel Corporation.  All rights reserved.

The parameters provided are supported in the following command-line sequences:

  1. -F  <file> -PARTID [ <Partition ID>] [-FORCERESET] [-VERBOSE [ <file>]]
     [-SILENT] [-Y] [-ALLOWSV]

 Using -EXP without any additional input will display examples of
 common command-line input.
```

7.6.4 Language Codes

This is the instance ID used in the above tool’s description.

Language	Language Code
English	0x01
French	0x02
German	0x03
Chinese Traditional	0x04
Japanese	0x05
Russian	0x06
Italian	0x07
Spanish	0x08
Brazilian Portuguese	0x09
Korean	0x0A
Chinese Simplified	0x0B
Arabic	0x0C
Czech	0x0D
Danish	0x0E
Greek	0x0F
Finnish	0x10
Hebrew	0x11
Hungarian	0x12
Dutch	0x13
Norwegian	0x14
Polish	0x15
Portuguese-Portugal	0x16
Slovak	0x17
Slovenian	0x18
Swedish	0x19
Thai	0x1A
Turkish	0x1B

§ §

8 UEFI Sample Application Leveraging FWUpdate API Library

8.1 Getting Started - FWUpdate Library

8.1.1 Introduction

This chapter will describe the Firmware Update Full Library as well as the RS (reduced size) library that will be used for Intel® Management Engine (Intel® ME) update. It contains a description of the various APIs to be used.

The Firmware Update process is essential for updating WCOD and LOCL regions by utilizing the APIs provided in the Firmware Update Library.

8.1.2 Environment

The FWUpdate Library provided is compiled using the EFI toolkit V2.0 and MSDK.

8.1.3 Setup

Follow the setting of the references below to get started with using the Firmware Update (FWUpdate) library and compiling it correctly.

1. You will need to include/reference the "FWUpdateLib.h" file in your program.
2. A make file referencing the FW Update Library. Libraries to Reference:

```
LIBS = $(LIBS) \
    $(SDK_BUILD_DIR)\lib\libc\libc.lib \
    $(SDK_BUILD_DIR)\lib\libefi\libefi.lib \
    $(SDK_BUILD_DIR)\lib\libsmbios\libsmbios.lib \
    $(SDK_BUILD_DIR)\lib\libefishell\libefishell.lib \
    $(SDK_BUILD_DIR)\lib\FwUpdateEfiLib\FwUpdateEfiLib.lib
```

8.1.4 Files in the Kit

In both the FWUpdate and FWUpdate RS (reduced size) folders released within the relevant FW Kit. Users will find the following files:

Table 8-1. Image File Update Options

File Name	Description
errorlist.c & errorlist.h	Source and header files for the error generation.
fwudef.h	Header file including FWUpdate definitions.
fwupdatelib.h	Header file including all the functions that can used by customers.
FWUpdateLib.lib	Static library with dynamic links to import DLLs.
Fwupdatelibdeprecated.h	Old deprecated FWUpdate header file. Functions within this file will be deprecated in future projects.
FWUpdateSample.c	Source file including a sample code for customers who intend to incorporate the FWUpdate library with BIOS or UEFI application.
FWUpdLcl64.exe	Full FWUpdate tool. Not relevant to FWUpdate RS.

8.2 Function Description

This section describes all the functions listed in FWUpdateLib.h. It explains the purpose, Input arguments and return types.

Note: Some function titles are marked as *deprecated*, this is intended for functions that have new replacement functions and will be deprecated in future projects.

Note: Some function titles are marked with the initials *RS*. This is intended for functions that apply for the FWUpdate RS library as well as the full FWUpdate library (reduced size library)Get Interfaces

8.2.1 Full FWUpdate from Buffer (RS)

Uint32 FwuFullUpdateFromBuffer (Uint 8 *Buffer, Uint 32 BufferLength, _UUID *OemId, void *Func(Uint 32, Uint 32));

Purpose: This function starts executing a full FWUpdate using buffer as the base for the FWUpdate.

Arguments	***Buffer*** – Buffer of Update Image read from Update Image File ***BufferLength*** – Length of the buffer in bytes ***OemId*** – OEM ID to compare with OEM ID residing in the FW. Can be Null ***Func*** – Functions used for reporting the progress of the FWUpdate. Can be null
Returns	Success, otherwise failure with error code

8.2.2 Partial FWUpdate from Buffer (*RS*)

Uint32 FwuPartialUpdateFromBuffer (Uint8 *Buffer, Uint32 BufferLength, Uint32 PartitionId, void *Func(Uint32, Uint32));

Purpose: This function starts executing a partial FWUpdate using buffer as the base for the FWUpdate for the specified partition using PartitionId. Please note the not all partitions can be updated independently.

Arguments	***Buffer*** – Buffer of Update Image read from Update Image File ***BufferLength*** – Length of the buffer in bytes ***PartitionId*** – ID of the partition the partial update will be updating. Note that only specific partitions are considered IUPs and be updated solely. ***Func*** – Functions used for reporting the progress of the FWUpdate. Can be null
Returns	Success, otherwise failure with error code

8.2.3 Checking update progress (*RS*)

Uint32 FwuCheckUpdateProgress (bool *InProgress, Out Uint32 *CurrentPercent, Out Uint32 FwUpdateStatus, Out Uint32 *NeedResetType);

Purpose: This function checks and reports the progress of the update flow. If in progress, it would return the current percentage of completion, if finished, it would return the status of the update and the required reset to follow with. This function is to follow Update functions (Full or Partial)

Arguments	***FwuCheckUpdateProgress***
Returns	Success, otherwise failure with error code. A success would return the following: ***InProgress*** – True if update is in progress. False if update is finished ***CurrentPercent*** – Current percent of the update if the update is in progress ***FwUpdateStatus*** – ID of the partition the partial update will be updating. Note that only specific partitions are considered IUPs and be updated solely. ***NeedResetType*** – Calls out the needed reset type after the update has finished. ·0 = No reset is required ·1 = Hot reset is required ·2 = CSE reset is required ·3 = Global reset is required

8.2.4 Get FWUpdate ability (*RS*)

Uint32 FwuEnabledState (Out Uint16 *EnabledState);

Purpose: This function checks and reports the FW’s ability to perform a FWUpdate (Enabled, Disabled)

Arguments	***FwuEnabledState***

Returns	Success, otherwise failure with error code. A success would return the following:
	FW_UPDATE_DISABLED = 0
	FW_UPDATE_ENABLED = 1

8.2.5 Retrieve OEM ID from Flash (*RS*)

Uint32 FwuOemId (Out _UUID *OemId);

Purpose: This function retrieves the OEM ID from the flash.

Arguments	***FwuOemId***
Returns	Success, otherwise failure with error code. A success would return the following: OEMID

8.2.6 Retrieve FW Type (*RS*)

Uint32 FwuFwType (OUT Uint32 *fwType);

Purpose: This function retrieves the FW type from flash.

Arguments	***FwuFwType***
Returns	Success, otherwise failure with error code. A success would return the following: 0 = FWU_FW_TYPE_INVALID 1 = FWU_FW_TYPE_RESERVED 2 = FWU_FW_TYPE_SLIM 3 = FWU_FW_TYPE_CONSUMER 4 = FWU_FW_TYPE_CORPORATE

8.2.7 Retrieve PCH SKU (*RS*)

Uint32 FwuPchSku(OUT Uint32 *pchSku);

Purpose: This function retrieves the PCH SKU.

Arguments	***FwuPchSku***

Returns	Success, otherwise failure with error code. A success would return the following: 0 = FWU_PCH_SKU_INVALID 1 = FWU_PCH_SKU_H 2 = FWU_PCH_SKU_LP

8.2.8 Get version of specific partition from flash image (*RS*)

Uint32 FwuPartitionVersionFromFlash(Uint32 PartitionId, Uint16 *Major, Uint16 *Minor, Uint16 *Hotfix, Uint16 *Build);

Purpose: This function retrieves the version of the specified partition ID from the flash image.

Arguments	***PartitionId*** – ID of the partition the function is requested to retrieve its version.
Returns	Success, otherwise failure with error code. A success would return the following: Returns the version of the specified partition (Major, Minor, Hotfix, Build)

8.2.9 Get version of specific partition from buffer (*RS*)

Uint32 FwuPartitionVersionFromBuffer (Uint8 *Buffer, Uint32 BufferLength, Uint32 PartitionId, Uint16 *Major, Uint16*Minor, Uint16 *Hotfix, Uint16 *Build);

Purpose: This function retrieves the version of the specified partition ID from the buffer.

Arguments	***Buffer*** – Buffer of partition ***BufferLength*** – Length of the buffer in bytes ***PartitionId*** – ID of the partition the function is requested to retrieve its version.
Returns	Success, otherwise failure with error code. A success would return the following: Returns the version of the specified partition (Major, Minor, Hotfix, Build)

8.2.10 Get vendor ID for a specific partition (*RS*)

Uint32 FwuPartitionVendorIdFromFlash (Uint32 PartitionId, Out Uint32 VendorId);

Purpose: This function retrieves the vendor of the specified partition ID from the flash image.

Arguments	***PartitionId*** – ID of the partition the function is requested to retrieve its version.
Returns	Success, otherwise failure with error code. A success would return the following: ***VendorId*** – ID of the vendor of the specified IUP

8.2.11 Performing a full FWUpdate

Uint32 FwuFullUpdateFromFile(const char *fileName, _UUID *oemId, void(*func)(UINT32, UINT32));

Purpose: This function starts a full FW Update from a given file.

Arguments	***fileName*** – File name referring to the update image to be provided ***oemId*** – OEM ID to compare with OEM ID in FW. This is meant to prevent different OEMs from updating FW irrelevant to them. Can be left Null ***func*** – A callback function that reports the progress of sending the buffer to FW.
Returns	Success, otherwise failure with error code.

8.2.12 Performing a partial FWUpdate

Uint32 FwuPartialUpdateFromFile (const char *fileName, Uint32 PartitionId, void(*func)(Uint32, Uint32));

Purpose: This function starts a partial FW Update from a given file.

Arguments	***fileName*** – File name referring to the update image to be provided ***PartitionId*** – ID of the partition to update. Please refer to our list of IUPs to learn about partially updateable partitions ***func*** – A callback function that reports the progress of sending the buffer to FW.
Returns	Success, otherwise failure with error code.

8.2.13 Retrieving partition version from image file

Uint32 FwuPartitionVersionFromFile(const char *fileName, Uint32 partitionId, Out Uint16 *major, Out Uint16 *minor, Out Uint16 *hotfix, Out Uint16 *build);

Purpose: This function retrieves the partition ID from a given update image file.

Arguments	***fileName*** – File name referring to the update image to be provided ***PartitionId*** – ID of the partition to update. Please refer to our list of IUPs to learn about partially updateable partitions
Returns	Success, otherwise failure with error code. A success would return the following: Returns the version of the specified partition (Major, Minor, Hotfix, Build)

8.2.14 Retrieving instance of a partition

Uint32 FwuPartitionInstances(Uint32 partitionId, Out Uint32 *currentInstanceId, Out Uint32 *expectedInstanceId);

Purpose: This function retrieves the current and expected instance ID of an IUP partition from the FW.

Arguments	***PartitionId*** – ID of the partition
Returns	Success, otherwise failure with error code. A success would return the following: ***CurentInstanceId*** – Current instance ID ***ExpectedInstanceId*** – Expected instance ID

8.2.15 Performing a partial FWUpdate with Instance ID from buffer

Uint32 FwuPartialUpdateWithInstanceIdFromBuffer(Uint8 *buffer, Uint32 bufferLength, Uint32 PartitionId, Uint32 instanceId, void (*func)(Uint32, Uint32));

Purpose: This function performs a partial FWUpdate with the provided instance ID from a buffer

Arguments	***Buffer*** – Buffer of the update image read from the update image file ***BufferLength*** – Length of the buffer in bytes ***PartitionId*** – ID of the partition to update, only partially updateable partitions apply ***InstanceId*** – Instance ID of the partition to update ***func*** – A callback function that reports the progress of sending the buffer to FW.
Returns	Success, otherwise failure with error code.

8.2.16 Performing a partial FWUpdate with Instance ID from file

Uint32 FwuPartialUpdateWithInstanceIdFromFile(const char *fileName, Uint32 partitionId, Uint32instanceId, void(*func)(Uint32, Uint32));

Purpose: This function performs a partial FWUpdate with the provided instance ID from a file.

Arguments	***fileName*** – File name referring to the update image to be provided ***PartitionId*** – ID of the partition to update, only partially updateable partitions apply ***InstanceId*** – Instance ID of the partition to update ***func*** – A callback function that reports the progress of sending the buffer to FW.
Returns	Success, otherwise failure with error code.

8.2.17 Creating a restore point image into buffer (*RS*)

Uint32 FwuSaveRestorePointToBuffer(OUT Uint8 **buffer, OUT Uint32 *bufferLength);

Purpose: This function retrieves the image from the flash and saves it to a buffer.

Arguments	***FwuSaveRestorePointToBuffer***
Returns	Success, otherwise failure with error code. A success would return the following: ***Buffer*** – Buffer of the saved restore image read from flash ***BufferLength*** – Length of the buffer in bytes

8.2.18 Creating a restore point image into file

Uint32 FwuSaveRestorePointToFile(const char *fileName);

Purpose: This function retrieves the image from the flash and saves it to a file.

Arguments	***fileName*** – Name of the file to save the restore point image into.
Returns	Success, otherwise failure with error code.

8.2.19 Checking power source

Uint32 FwuPowerSource(OUT Uint32 *powerSource);

Purpose: This function checks the current power source (AC or DC).

Arguments	***FwuPowerSource***
Returns	Success, otherwise failure with error code. A success would return the following: ***powerSource*** = power source would show one of the below values ·0 = Unknown ·1 = AC ·2 = DC

8.2.20 Set ISH configuration file (*RS* Only)

Uint32 FwuSetIshConfig (Uint8 *Buffer, Uint32 BufferLength);

Purpose: This function sets the ISH configuration file "bios2ish".

Arguments	***Buffer*** – Buffer of IUP ***BufferLength*** – Length of the buffer in bytes
Returns	Success, otherwise failure with error code

8.2.21 Get PDT version and VDV version (*RS* Only)

Uint32 FwuGetIshPdtVersion (Unit8 *PdtVersion, Uint8 *VdvVersion);

Purpose: This function returns the PDT and VDV versions from ISH file INTC_pdt

Arguments	***FwuGetIshPdtVersion***
Returns	Success, otherwise failure with error code. A success would return the following: ***PdtVersion*** – Version of the PDT ***VdvVersion –*** Version of the VDV

8.2.22 Get Interfaces (Deprecated) (*RS*)

unsigned int GetInterfaces(unsigned short *interfaces);

Purpose: This function gets the local FW update settings from Intel® Management Engine BIOS Extension (Intel® MEBX) to determine whether Firmware can be updated.

Arguments	***Interfaces*** - whether the Local FW Update is disabled (0) or enabled (1) or password protected (2)
Returns	Gets the Interfaces from HECI 0 = Success Non-zero value = Failure

8.2.23 Get Last Status (Deprecated) (*RS*)

unsigned int GetLastStatus(unsigned int *lastStatus);

Purpose: This function will get the previous FW update status to ensure that FW update was successfully executed.

Arguments	***Laststatus*** – Last FW Update process Status (E.g. Success, Invalid OEM ID, FW Version mismatch etc) Refer "me_status.h" for specific values
Returns	Gets the last FW update status from HECI 0 = Success Non-zero value = Failure

8.2.24 Get Last Update Reset Type (Deprecated) (*RS*)

unsigned int GetLastUpdateResetType(unsigned int *lastResetType);

Purpose: This function will get the last Update Reset type to determine what type of system reset is required to load the partition into the memory.

Arguments	***LastResetType*** - The last FWUpdate reset type No reset – 0 Host reset – 1 ME – 2 Global - 3
Returns	Gets the last FW update status from HECI 0 = Success Non-zero value = Failure

8.2.25 Check Policy (Deprecated)

unsigned int CheckPolicy(char* ImageFileLib, int AllowSV, UPDATE_TYPE *Upd_Type,VersionLib *ver);

Purpose: This function determines whether it is a FW upgrade/downgrade or same version update using a file.

Arguments	***Image File*** - Binary Image file ***AllowSV*** - Allow Same Version flag (Set to 1 to execute same version flow) ***Update Type*** - Update Type Output. Can be DOWNGRADE_SUCCESS = 0, DOWNGRADE_FAILURE = 1, SAMEVERSION_SUCCESS = 2, SAMEVERSION_FAILURE = 3, UPGRADE_SUCCESS = 4, UPGRADE_PROMPT = 5, ***Ver***- FW Version (Major, Minor, Hotfix, Build)
Returns	0 = Success Non-zero value = Failure

8.2.26 Check Policy Buffer (Deprecated) (*RS*)

unsigned int CheckPolicyBuffer(char* buffer, int bufferLength, int AllowSV, UPDATE_TYPE *Upd_Type, VersionLib *ver);

Purpose: This function determines whether it is a FW upgrade/downgrade or same version update using buffer.

Arguments	***Buffer*** - buffer to access ***BufferLength*** - Length of buffer ***AllowSV*** - Allow Same Version flag ***Update Type***– Update Type Output. Can be DOWNGRADE_SUCCESS = 0, DOWNGRADE_FAILURE=1, SAMEVERSION_SUCCESS=2, SAMEVERSION_FAILURE=3, UPGRADE_SUCCESS=4, UPGRADE_PROMPT=5, ***Ver*** – FW Version (Major, Minor, Hotfix, Build)
Returns	0 = Success Non-zero value = Failure

8.2.27 Verify OEM Id (Deprecated) (*RS*)

bool VerifyOemId(_UUID id);

Purpose: This function verifies the OEM ID provided by the user with the one embedded in the FW.

Arguments	***Id*** - OEM id

Returns	True=OEMID matched
	False = OEM id mismatch

8.2.28 Get Ipu Partition Attributes (Deprecated) (*RS*)

unsigned int GetIpuPartitionAttributes(FWU_GET_IPU_PT_ATTRB_MSG_REPLY *FwuGetIpuAttrbMsgInfo);

Purpose: This function gets the number of Independent partial update partition attributes that is currently present and also the list of expected IPUs to be updated.

Arguments	Out parameter: FWU_GET_IPU_PT_ATTRB_MSG_REPLY – is a data structure with IPU related information
Returns	0 = Success 8193 = Heci Device not found 8204 = Heci message has incorrect message type 8728 = Heci Buffer Size is Small Error 8710 = Insufficient memory Error 8776 = Failure to Send or Receive the Get Partition Attribute Command Or even when FW returns an error status after receiving command

8.2.29 Get FW Update Info Status (Deprecated)

unsigned int GetFwUpdateInfoStatus(FWU_INFO_FLAGS *StatusFlags);

Purpose: This function gets the current status of the firmware.

> Note: This API is not used by the FWUpdate tool. It is being used by the UNS services.

Arguments	***StatusFlags*** - BITS 0:1 (2 bits) 0 = No recovery; 1 = Full Recovery Mode; 2 = Partial Recovery Mode (unused at present). BIT2; IPU_NEEDED bit, if set we are in IPU_NEEDED state. BIT3; FW_INIT_STATUS done. BIT4; FWU_IN_PROGRESS
Returns	0 = Success 8193 = Heci Device not found 8204 = Heci message has incorrect message type 8213 = Heci Buffer Size is Small Error 8710 = Insufficient memory Error 8777 = Failure in Send or Receive of the Get Info Status Command. Or even when FW returns an error status after receiving command

8.2.30 FW Update Query Status Get Response (Deprecated) (*RS*)

unsigned int FWUpdate_QueryStatus_Get_Response(unsigned int* UpdateStatus, unsigned int *TotalStages, unsigned int* PercentWritten, unsigned int * LastUpdateStatus, unsigned int * LastResetType);

Purpose: This function queries FW to get response regarding the different stages of FW Update process.

Arguments	***UpdateStatus*** - indicates the current FW Update stage being executed. ***TotalStages*** – indicates the total number of FW Update stages available. ***PercentWritten*** – indicates the percentage complete of the FW Update process ***LastUpdateStatus*** **–** indicates the status of the fwupdate process just completed ***LastResetType*** **–** indicates Reset type required for the fwupdate process just completed

Returns	0= Success 1 = Invalid Manifest Data in partition 8193 = Heci Device not found 8204 = Heci message has incorrect message type 8213 = Heci Buffer Size is Small Error 8710 = Insufficient memory Error 8724 = Failure to send or receive messages to heci to get Status Info 8741 = FW returns incorrect Message Type

8.2.31 FW Update Full – Using Buffer (Deprecated)

unsigned int FwUpdateFull (char* buffer, unsigned int bufferLength, char* _pwd,int _forceResetLib, unsigned int UpdateEnvironment,_UUID OemID, UPDATE_FLAGS_LIB update_flags, void(*func)(float,float));

Purpose: This function performs the full FW Update using the Buffer provided by the calling function.

Arguments	***Buffer*** – Buffer with the update image ***Buffer Length*** – Length of buffer ***Password*** – MEBX Password ***ForceResetLib*** – Flag to perform system reset ***UpdateEnvironment*** – differentiates various firmware update process environment within the firmware (manufacturing/non-manufacturing) ***UUID OEMID*** – OEM ID ***update_flags*** – flag to indicate FW of recovery/rollback ***Func pointer*** – (bytes of Binary
Returns	0 = Success Non-zero value = Failure

8.2.32 FW Update Partial Buffer (Deprecated) (*RS*)

unsigned int FwUpdatePartialBuffer(char* buffer,unsigned int bufferLength, unsigned int PartitionID, unsigned int Flags, IPU_UPDATED_INFO *IpuUpdatedInfo, void(*func)(float, float));

Purpose: This function performs the Partial FW Update. If the requested partition is expected by the Firmware, it will search for the expected partition in the image provided, extract it and send it to the FW to perform the update. If the expected partition is not found in the image an invalid file error will be returned by the tool. If the requested partition is not expected by the firmware an error will be returned to the user.

Note: For Partial FW update the image provided must either be a Full or Partial image. A full image starts with a FPT and contains FTP and NFTP partitions. A partial image starts with either WCOD or LOCL partitions.

FWUpdate API Library supports only Partial FWUpdate for ISH only. –i is the command line switch.

Example Usage: FwUpdLclApp.efi –i <Image.bin>

Arguments	***Buffer*** - Buffer ***Buffer Length*** – Length of buffer
Returns	***Partition ID*** - denotes the partition ID, which could be WLAN (wcod) or language (locl). ***WOCD ID*** = 0x244f4357 and ***LOCL ID*** = 0x4C434F4C ***Flags***: Bit 0 of the flags is used to set allow same version update. Other bits are reserved and can be used in the future. ***IpuUpdatedInfo*** - Contain the information that is actually used to update the IPU partition. 0 = Success Non-zero value = Failure

8.2.33 PDT Data (Sensor Calibration Data) Update (Deprecated) (*RS*)

```
EFI_STATUS
HeciPdt (
    IN  char                *buffer,
    IN  UINT32        bufferLength
  );
```

Purpose: The function performs PDT Data Update i.e. Sensor Calibration Data Update.

Command Line Switch –d needs to be used in order to execute PDT Data Update.

Example for Usage:

FwUpdLclApp.efi –d <Pdt Data Binary>
FWUpdLclApp.efi –d INTC_pdt_SPT_RR3_BOM1_SENSORS

Arguments	***Buffer*** - Buffer ***Buffer Length*** – Length of buffer
Returns	***If Payload is sent to CSME successfully then Send Succeeded Message will be seen.***

8.2.34 ISH Firmware Version (Deprecated)

```
int
GetPartVersion (
    UINT32 partID,
       UINT16 *major,
       UINT16 *minor,
       UINT16 *hotfix,
       UINT16 *build);
```

Purpose: The function helps retrieve ISH Firmware Version flashed on the platform.

9 Intel® Manifest Extension Utility (Intel® MEU)

The Intel® Manifest Extension Utility (MEU) inputs a firmware binary created by a 3rd party and outputs an independent-Updatable partition (IUP) that is compressed and signed. After completing this process the signed binary can be added to the flash image using the Intel® FIT tool.

The Intel® MEU tool completes the following steps:

- Creates an Independent Updatable Partition (IUP) by adding manifest and meta-data information to the firmware.
- Calls an external LZMA tool for compression of the firmware binary. The LZMA tool is supplied with the ISH binary or may be downloaded from http://7-zip.org/sdk.html.
- Calls the OpenSSL tool as the signing infrastructure tool to sign the partition.

9.1 Usage

Refer to the *Signing & Manifesting Guide* in the latest Intel ME FW kit for details on MEU usages, signing & manifesting flows, etc.

§ §

A Intel® ME NVARs

This appendix only covers fixed offset variables that are directly available to FPT and FPTW. A complete list of NVARs can be found in the *Firmware Variable Structures for Intel® Management Engine*. All of the fixed offset variables have an ID and a name. The `-CVAR` option displays a list of the IDs and their respective names. The variable name must be entered exactly as displayed below.

This table is for reference use only and will be updated later.

Table A-1. NVARs Descriptions

Fixed Offset Name	Description	Data Length (in Bytes)	Expected Value	Reset Type	Mfg. Post EOM/ Pre EOP
Non-Application Specific Fixed Offset Item Descriptions					
MEBx Password	Overrides the MEBx default password. It must be at least eight characters and not more than 32 characters in length. All characters must meet the following: ASCII(32) <= char <= ASCII(126) Cannot contain these characters: , : " Must contain for complexity: a. At least one Digit character (0 - 9) b. At least one 7-bit ASCII non alpha-numeric character above 0x20 (e.g. ! $;) c. Both lower-case and upper case Latin. d. underscore and space are valid characters but are not used in determination of complexity. Refer section 2.7 for format and strong password requirements.	8<=N<=32	Password	ME	Yes

<table>
<tr><th>Fixed Offset Name</th><th>Description</th><th>Data Length (in Bytes)</th><th>Expected Value</th><th>Reset Type</th><th>Mfg. Post EOM/ Pre EOP</th></tr>
<tr><td>OEMSKURule</td><td>UINT32 (little endian) value. This controls what features are permanently disabled by OEM.

<b>Note:</b> The FPT command now supports changing individual bits of the OEMSKURule. It is strongly recommended to set the individual bits rather than the full 32 bit value.

<b>Note:</b> There are reserved bits that the must not be changed for proper platform operation. The user should only modify the bit(s) for the feature(s) they wish to change. This NVAR sets OEM Permanent Disable for ALL features. In addition, prior to updating or changing any of available settings it is highly recommended that the user first retrieves the current OEM Sku Rule and toggling only the desired bits, and then resave them.

This will not enable functionality that is not capable of working in the target hardware SKU. Refer respective Firmware Bring-up Guide for a list of what features are capable with what firmware bundle and Hardware SKU of Intel 9 Series Chipset.</td><td>4</td><td>Feature Capable: 1
Feature Permanently disabled: 0

<table>
<tr><th>Bit</th><th>Description</th><th>Notes</th></tr>
<tr><td>31</td><td>Reserved</td><td></td></tr>
<tr><td>30</td><td>Reserved</td><td></td></tr>
<tr><td>29:22</td><td>Reserved</td><td></td></tr>
<tr><td>21</td><td>TLS</td><td></td></tr>
<tr><td>20</td><td>DAL</td><td></td></tr>
<tr><td>19</td><td>Reserved</td><td></td></tr>
<tr><td>18</td><td>KVM</td><td>2</td></tr>
<tr><td>17</td><td>Reserved</td><td></td></tr>
<tr><td>16</td><td>ME Network Disable</td><td></td></tr>
<tr><td>15:13</td><td>Reserved</td><td></td></tr>
<tr><td>12</td><td>PAVP</td><td></td></tr>
<tr><td>11</td><td>Reserved</td><td></td></tr>
<tr><td>10</td><td>ISH</td><td></td></tr>
<tr><td>9:6</td><td>Reserved</td><td></td></tr>
<tr><td>4:5</td><td>Reserved</td><td></td></tr>
<tr><td>3</td><td>Reserved</td><td></td></tr>
<tr><td>2</td><td>Manageability and Security Application</td><td>1</td></tr>
<tr><td>1</td><td>Reserved</td><td></td></tr>
<tr><td>0</td><td>Manageability Full</td><td>1</td></tr>
</table>

1. For corporate SKUs bits 0 and 2 need to be both set to ‘1’ to allow for Intel® AMT to work.
2. KVM (bit 18) should only be set to ‘1’ when Manageability Application (bit 2) is set to ‘1’. If using a Corporate SKU, then Manageability Full (bit 0) must also be set to ‘1’</td><td>Global</td><td>No</td></tr>
</table>

<table>
<tr><th>Fixed Offset Name</th><th>Description</th><th>Data Length (in Bytes)</th><th>Expected Value</th><th>Reset Type</th><th>Mfg. Post EOM/ Pre EOP</th></tr>
<tr><td>Feature Shipment Time State</td><td>UINT32 (little endian) value. This controls what features are enabled or disabled. These features may be enabled / disabled by mechanisms such as MEBx or provisioning. This setting is only relevant for features NOT permanently disabled by the OEM Permanent Disable.
This will not enable functionality that is not capable of working in the target hardware SKU. Refer respective Firmware Bring-up Guide for a list of what features are capable with what firmware bundle and Hardware SKU of Intel 8 Series Chipset.
<b>Note:</b> The FPT command now supports changing individual bits of the Feature Ship State. It is strongly recommended to set the individual bits rather than the full 32 bit value.</td><td>4</td><td>Feature Enabled: 1
Feature Disabled: 0<table>
<tr><th>Bit</th><th>Description</th><th>Notes</th></tr>
<tr><td>31:30</td><td>Reserved</td><td></td></tr>
<tr><td>29</td><td>PTT</td><td></td></tr>
<tr><td>28:3</td><td>Reserved</td><td></td></tr>
<tr><td>2</td><td>Manageability Full</td><td></td></tr>
<tr><td>1:0</td><td>Reserved</td><td></td></tr>
</table><b>Note:</b> When disabling PTT using Feature Shipment Time state NVAR, execute a reset after executing fpt.efi –commit to ensure PTT is disabled completely.</td><td><b>Global</b></td><td><b>Yes</b></td></tr>
<tr><td>SetWLANPowerWell</td><td>Sets which power well the board uses for WLAN cards</td><td>4</td><td><b>0x80</b> = Disabled
<b>0x81</b> = Core Well || SLP_S3
<b>0x82</b> = Primary Well ||SLP_SUS
<b>0x83</b> = ME Well || SLP_A
<b>0x86</b> = WLAN Sleep via SLP_WLAN#</td><td><b>Global</b></td><td><b>No</b></td></tr>
<tr><td>OEM_TAG</td><td>A human readable 32-bit number to describe the flash image represented by value</td><td>4</td><td>Readable 32 bit hex value identifying the image. Can be empty (Null).</td><td><b>Global</b></td><td><b>No</b></td></tr>
</table>

Fixed Offset Name	Description	Data Length (in Bytes)	Expected Value	Reset Type	Mfg. Post EOM/ Pre EOP
GpioNvar	GPIO	60	GPIO groups and pad range for each grp pad# GPP_A 0-16 GPP_B 0-23 GPP_C 0-23 GPP_D 0-23 GPP_E 0-23 GPP_F 0-23 GPP_G 0-7 GPD 0-11 Example read of GPIO: Variable: "gpio" Value: 0x0000 : 00 00 00 00 04 00 00 00 06 00 00 00 01 00 00 00 0x0010 : 00 00 00 00 01 00 00 00 04 00 00 00 0C 00 00 00 0x0020 : 01 00 00 00 00 00 00 00 08 00 00 00 01 00 00 00 0x0030 : 0F 00 00 00 01 00 00 00 00 00 00 00 Note: the only locations that can be modified are underlined above. The format for updating the GPIO is as follows... GpioNvar = 0x00000000003000000011000000001000000000000000000100000000200000017000000001000000000000000000800000000300000013000000001000000000000000 RST = GPP_D_17 IRQ = GPP_C_23 DFU = GPP_D_19	ME	No
FWUpdLcl	Enabled Firmware Update Local Capability	1	0 = disabled 1 = enabled	**Global**	**Yes**
EDP_PORT_CFG	EDP Port Configuration. Up to two ports can be enabled 0x00 - 0x01 – A 0x02 - B 0x04 - C 0x08 - D 0x10 - E	1	0x00 0x01 0x02 0x03 0x04 0x05 0x06 0x08 0x09 0x0A 0x0C	**ME**	**No**
LSPCON_PORT	LSPCON Port Configuration. 0x00 - 0x02 - B 0x04 - C 0x08 - D	1	0x00 0x02 0x04 0x08	**ME**	**No**
URTC	UnConfigure On RTC	1	0 = Disabled 1 = Enabled	**ME**	**No**

Fixed Offset Name	Description	Data Length (in Bytes)	Expected Value	Reset Type	Mfg. Post EOM/ Pre EOP
DAM	DAM is a feature that allows the SUT to prepare for unlock without actually enabling debug interfaces	1	0 = Disabled 1 = Enabled	**ME**	**No**
AMT Related NVARs					
OEM Customizable Certificate 1	Cert Hash Data. Refer Certificate Hash Entry Structure definition **Note**: If the platform is un-configured the Certificate Hash will be deleted.	55 => n >= 99	Valid Certificate Hash Entry (SHA1, SHA256 or SHA384)	**ME**	**Yes**
OEM Customizable Certificate 2	Cert Hash Data. Refer Certificate Hash Entry Structure definition **Note**: If the platform is un-configured the Certificate Hash will be deleted.	55 => n >= 99	Valid Certificate Hash Entry (SHA1, SHA256 or SHA384)	**ME**	**Yes**
OEM Customizable Certificate 3	Cert Hash Data. Refer Certificate Hash Entry Structure definition **Note**: If the platform is un-configured the Certificate Hash will be deleted.	55 => n >= 99	Valid Certificate Hash Entry (SHA1, SHA256 or SHA384)	**ME**	**Yes**
Privacy/ Security Level	Redirection (KVM, SOL, IDE-r) privacy level and configuration (RCFG, CCM) settings. ***Note:*** Setting Privacy Level to its default value would cause NVARS to be reverted to their defaults disregarding changes committed to them	1	Default **0x01** Enhanced **0x02** Extreme **0x03** Default: SOL enabled = true IDER enabled = true KVM enabled = true Opt-in can be disabled= true KVM opt-in configurable remotely = true RCFG and CCM = true Enhanced: SOL enabled = true IDER enabled = true KVM enabled = true Opt-in can be disabled= false Opt-in configurable remotely = true RCFG and CCM = true Extreme: SOL enabled = false IDER enabled = false KVM enabled = false Opt-in can be disabled= false KVM opt-in configurable remotely = N/A RCFG and CCM = false	ME	No
EHBC State	Embedded Host Based Configuration State	1	0 = Disabled 1 = Enabled	ME	No

Fixed Offset Name	Description	Data Length (in Bytes)	Expected Value	Reset Type	Mfg. Post EOM/ Pre EOP
ScreenBlankingEn	Screen Blanking Enabled	1	0 = Disabled 1 = Enabled	ME	No
PKI DNS Suffix	PKI DNS Suffix. Null terminated string	32	PKI DNS Suffix in dotted string format Example: "intelFVE.com" ***Note:*** dots are acceptable only in the middle of the string	ME	Yes
CfgSrvFqdn	Configuration Server FQDN (Fully Qualified Domain Name)	256	Example: "intelFVE.com"	ME	Yes
Rcfg	R Configuration	1	0 = Disabled 1 = Enabled	ME	Yes
*Redirection	This is a bit-field Indicating the enable/disable status of Storage Redirection, SOL, and KVM features in Intel® AMT. bit[0]: 1 – Storage Redirection enabled, 0 – disabled bit[1]: 1 – SOL enabled,0 – disabled bit[2]: 1 – KVM enabled, 0 – disabled	4	Range: 0-7 Example: Value of 4 (100b) indicates that KVM is enabled. Value of 3 (011b) indicates that Storage Redirection, and SOL are enabled. Value of 7 (111b) indicates that Storage Redirection, SOL, and KVM are enabled.	ME	Yes
*OptinPolicy	Change User Opt-in (lower nibble). NONE = 0, KVM = 1, ALL = F Disable Opt-In Configurable from Remote IT (upper nibble). 0 - Opt-in is NOT Configurable from Remote IT 1 - Opt-in is Configurable from Remote IT	1	0x00 0x10 0x01 0x11 0x0F 0x1F Examples: In addition to the following, the values **may not be** configured remotely: Value of 0x00 indicates User Consent is not required. Value of 0x01 indicates User Consent is required for KVM only. Value of 0x0F indicates User Consent is required for (ALL). In addition to the following, the values **may be** configured remotely: Value of 0x10 indicates User Consent is not required. Value of 0x11 indicates User Consent is required for KVM only. Value of 0x1F indicates User Consent is required for (ALL).	ME	Yes
HostName	Set Host Name Only	64	SkyLake SunrisePoint	ME	Yes
DomainName	Set Domain Name Only	192	myserver.intel.com amr.corp.intel.com www.intel.com mymail.somecollege.edu	ME	Yes

Fixed Offset Name	Description	Data Length (in Bytes)	Expected Value	Reset Type	Mfg. Post EOM/ Pre EOP
CfgSrvAdr	Set Provisioning Server (IPv4/ IPv6) Address	60	Example of IPV4: 192.168.1.200 255.255.255.0	ME	Yes
CfgSrvPort	Set Provisioning Server (IPv4/ IPv6) Port	2	Within Range: 0 – 0xFFFF	ME	Yes
DisCertHash	Disable all Pre-Installed Certificate Hashes	1	0 = Disabled 1 = Enabled	ME	Yes
IdleTO	Change the Idle Timeout in minutes	2	Within Range: 1 – 0xFFFF	ME	Yes
AmtWdAutoReset	Intel® AMT Watchdog Automatic Reset enabled	1	0 = disabled 1 = Enabled	ME	No
Revenue Sharing Related NVAR Descriptions					
ODM_ID	NVAR used for setting the ODM ID Used by Intel® Services **Note:** This value can only be programmed into FW once.	4	32-bit value Value 0x00000000 < n < 0xFFFFFFFF	ME	No
SystemIntegratorID	Used for setting the System Integrator ID used by Intel® Services. **Note:** This value can only be programmed into FW once.	4	32-bit value Value 0x00000000 < n < 0xFFFFFFFF	ME	No
ReservedID	Used for setting the "Reserved" ID used by Intel® Services **Note:** This value can only be programmed into FW once.	4	32-bit value Value 0x00000000 < n < 0xFFFFFFFF	ME	No
Field Programable Fuses					
PTTEnable	**Enables / Disables the fTPM / PTT FPFs**	1	0 **= Disabled** 1 = Enabled	ME	No

- Indicates: Intel AMT KVM not supported if both HDCP Internal Display Ports (A, B, C, and D) are configured.

Note: Settings of all AMT Related parameters (All NVARs Listed under AMT Related NVARs Section) will be supported when Intel® AMT is in pre-provisioned mode only. Otherwise the settings will be ignored.

§ §

B Tool Detail Error Codes

Errors are reported in the following format for all tools:

<ErrorTypeNumber>*256+<ErrorNameNumber>

B.1 Common Error Code for FIT Tool

The below table displays the error type number and corresponding string.

Error Type Number	Corresponding String
0	No Error
1	[Action Processor]
2	[Bin Actions]
3	[Fit Converter]
4	[Csme Binary Gen]
5	[Fit Actions]
6	[Fit File I/O]
7	[Fit Utils]
8	[Framework C Lib]
9	[ME Util]
10	[Xml Processor]
11	[Fit]
12	[DNX Utils]
13	[Ifwi Actions]
14	[IMR Actions]
15	[Smip Controller]
16	[GPIO Actions]
17	[ME MFS]
18	[Nvar Actions]
19	[Manifest]
20	[Manifest Actions]
21	[Crypto Actions]
22	[CSME Actions]
23	[Elf Actions]
24	[Huffman Utils]
25	[System Resources]
26	[SysCall Actions]

The below table shows the error name number and the corresponding message.

Error Code	Error Message
0	No Error
1	Initialize Error
2	Failed to build
3	Build general error
4	Build enumeration error
5	Error building attribute
6	Error Building Resources

Error Code	Error Message
7	Decompose Error.
8	Failed to decompose SMIP data
9	Failed to decompose Image
10	Failed to decompose Region
11	Error Decomposing attribute
12	Error calling decomposition actions
13	Decomposition node not found
14	Error decomposing class
15	Decomposition not ready
16	Failed to detect and configure ROM Bypass partition
17	Failed to decompose boot partition
18	Failed to generate decomposed files
19	Failed to decompose boot partition entry
20	Error executing pre-build actions
21	Error executing post-build actions
22	Error generating intermediate build output
23	Invalid object alignment value
24	Unable to resolve parent for attribute
25	Buffer offset out of bounds
26	Unresolved native type
27	Data Conversion error
28	Invalid H file output path
29	Error rounding attribute
30	Error updating attribute
31	Error executing post-decomp actions
32	Missing XML attribute
33	Failed to sign SMIP data
34	MEU config file Error
35	Bad command line options
36	Invalid PCH SKU specified
37	Error setting the log file
38	File not found
39	File name with the PERIODS at the end is not supported
40	Could not find VSCC value for given JEDEC code
41	Failed to open new image
42	Failed to open with processed commands
43	Failed to parse XML
44	Failed to parse XML settings
45	Full image could not be written
46	Could not locate input region file
47	Invalid input file type
48	Failed to populate known good VSCC database in memory
49	User did not accept the license agreement
50	Failed to launch GUI as requested
51	Failed to build CSE region file
52	Failed to process layout file
53	Failed to process configuration file
54	Unable to open layout file
55	Invalid root name detected in configuration file
56	Unknown root node found in XML
57	Invalid XML tag
58	Invalid XML child tag association
59	XML tag exception
60	Missing XML tag
61	Error resolving data dependencies
62	Failed to generate dependency map for XML tag
63	Failed to load native type definitions
64	Invalid XML value attribute
65	Error overriding CSE version Major number
66	Error overriding CSE version Minor number
67	Error overriding CSE version Hotfix number

Error Code	Error Message
68	Error overriding CSE version Build number
69	Error overriding CSE Internal Build Version
70	Error overriding signing key
71	Error overriding MFS BinObject
72	Error overriding the active LOCL Instance Id
73	Error overriding the active WCOD Instance Id
74	Error overriding the active MDMV Instance Id
75	Error enabling partition from the command line
76	Error disabling partition from the command line
77	Error setting partition length
78	Error overriding the LOCL UPV version
79	Error overriding the WCOD UPV version
80	Error overriding the MDMV UPV version
81	Error overriding the TCB SVN number
82	Error overriding the ARB SVN number
83	Error overriding the VCN number
84	Error Overriding the Sku Attributes value
85	Error overriding the Uma size
86	Error overriding compression mode on all modules
87	Error overriding CSE region length
88	Error enabling RomBypass partition
89	Error overriding PCH version
90	Error overriding DataFormatVersion
91	No XML version was specified
92	The XML version specified is not in the proper format: x.x
93	The version of the XML file you are loading is not supported
94	The version of the XML you are loading is greater than any version this application knows about
95	Unable to open config file
96	Error overriding header files output directory
97	Error overriding the AFS SKU ID
98	Error opening file
99	Error writing to file
100	Size mismatch in file write
101	Error reading text file
102	Setting not found
103	Invalid type specified for setting
104	Unable to resolve action's target attribute
105	Unable to resolve action's source attribute
106	Failed to write data to image buffer
107	Unable to resolve attribute used in action
108	Failed to open file
109	Failed to update image buffer offset
110	Source value of length 0 must be in hex string format
111	Failed to load NVARS from path
112	Detected overflow in CalcOffset operation
113	String length too long
114	Empty input string
115	Failed attribute length limit validation
116	Failed to write buffer to file
117	Invalid JSON Parameter(s)
118	Invalid region size
119	Unable to calculate hash
120	Invalid action parameters
121	Buffer overflow detected
122	Buffer overflow detected
123	Invalid Checksum Action Parameters
124	Error Calculating round_to function
125	Missing data class
126	Invalid signing key

Error Code	Error Message
127	Failed to generate signature
128	Unable to generate intel.cfg file
129	Unable to generate intel.cfg SHA2
130	Failed to encrypt module
131	Error generating manifest independent partition
132	Error generating feature permissions extension
133	Error generating thread attributes extension
134	Error generating device attributes extension
135	Error generating mmio ranges extension
136	Error generating file producer extension
137	Failed to add group IDs to process extension
138	Error generating user info extension
139	Failed to adjust the FTUP partition length and offset
140	Found IUP partition (WCOD,MDMV,LOCL,ISH) before NFTP. This is not allowed
141	Found unexpected IUP partition. All IUP must be allocated in a contiguous block
142	Calculated FTUP partition size is smaller than FTPR size, this will break FWUpdate
143	Number of FPT entries does not fit in current FPT area supported by FTOOL
144	Unable to resolve user name
145	Unable to update partition offsets in database
146	Missing partition parameters
147	Missing partition instance
148	Unable to update partition offset
149	Error building Partial Firmware Update image
150	Failed to configure firmware runnable region
151	Unable to disable attribute
152	Invalid runnable region configuration
153	Make Module Failed
154	Get elf info failed
155	Make Module Failed
156	Parse Module metadata Failed.
157	Elf to Bin failed
158	Get Section Data failed
159	Invalid ModuleType. Module is not Process or Shared Library type
160	Failed to build shared library
161	Invalid TotalThreadStackSize value
162	Unable to get CM0HeapSize configuration parameter
163	Unable to get DefaultHeapSize configuration parameter
164	Invalid CM0Heap Value
165	Invalid DefaultHeap Value
166	Unable to find the FLREG layout entry
167	Failed to resolve region limit
168	Failed to resolve region base
169	Unable to find the Regions layout entry
170	Missing input region length configuration option
171	File Path could not be resolved
172	Invalid region size
173	Not enough flash space
174	Missing region data target
175	Unable to update region data target
176	Unable to load CSE region
177	Failed to allocate memory
178	Failed to parse CSE region
179	Not enough space to copy CSE region into image buffer
180	Unable to prepare CSE region
181	Invalid VSCC entry
182	Detected overflow in BPDT table

Error Code	Error Message
183	Invalid Descriptor offset
184	Invalid Descriptor size
185	Unable to parse ROM Bypass configuration
186	Unable to load ISH image
187	ISH image file size is too large
188	Failed to update ME Region
189	Invalid PKI Suffix:
190	Invalid Certificate Hash Format:
191	Invalid GUID format
192	Failed to parse GbE image
193	The file is not large enough to be a valid GbE
194	Invalid Region Order
195	Invalid settings combination
196	Unable to load Token
197	Unable to decompose Token
198	IDLM Binary is invalid or corrupt
199	Unable to decompose IDLM Binary
200	Failed to process VR profile selection
201	Failed to generate FW update image
202	String length is too large
203	Failed to generate CSE data partition
204	TBT Binary is invalid or corrupt
205	Chipset Init Binary is invalid or corrupt
206	Chipset Init Base Intel Recommendation table is invalid
207	Chipset Init Product version does not match the configured PCH SKU type
208	Failed to get image Metadata sub partition
209	Failed to load FITC binary to sub partition
210	Failed to generate Image Metadata partition
211	Failed to load FITC binary to sub partition
212	Failed to find child attribute
213	Failed to get Class Instance
214	Failed to map GPIOs
215	Invalid NVAR size
216	NVAR IO Error
217	Failed to set target IFWI configuration
218	Failed to load BIOS image from file
219	Failed to configure IFWI layout
220	Failed to prepare one or more IFWI components
221	Failed to load ME component
222	Detected invalid Sub-Partition
223	Detected build buffer overflow
224	Failed to update build buffer cursor offset
225	Failed to prepare ME BUP Sub-Partition
226	Failed to calculate boot partition sizes
227	Unable to update region data target
228	Failed to get ME Sub-Partitions
229	Failed to load OEM Key Manifest input file
230	Failed to add OEM Key Manifest to IFWI image
231	Failed to build SMIP data
232	Failed to load SMIP intermediate file
233	Failed to add SMIP Sub-Partition to IFWI image
234	Failed to add ROMB partition to IFWI image
235	Failed to load input file
236	Unable to determine image type
237	Detected invalid DNX image format.
238	Unable to detect number of flash components setting
239	Unable to resolve flash image size
240	Failed to validate Key Manifests
241	Failed to validate Public key hash
242	Failed to calculate BPDT Checksum
243	Failed to calculate and set required image padding

Error Code	Error Message
244	Invalid Manifest Extension Utility path
245	Utility to sign the SMIP data
246	Invalid signing key path
247	Invalid signing tool path
248	Failed to load data sub-partition
249	General error
250	Missing configuration attribute
251	Unable to set configuration value
252	IMR range value out of range
253	Unable to round up IMR value
254	Total IMR size exceeding maximum size
255	Invalid action parameters
256	Invalid attribute parameters
257	Missing JSON parameter in NVAR Action
258	Unable to convert NVAR index to U32
259	Unable to convert NVAR offset to U32
260	Unable to convert NVAR bitHi to U32
261	Unable to convert NVAR bitLo to U32
262	Unable to convert NVAR field size to U32
263	Unable to convert NVAR file size to U32
264	Failed to write NVAR
265	Failed to read NVAR
266	Invalid action parameter
267	Invalid target name
268	Invalid bitfield length specified
269	Error updating configuration variable
270	Could not load binary file
271	Could not resize NVAR for binary file
272	Specified variable size will not fit into fixed-size NVAR file
273	Specified variable size will not fit into cell
274	Specified offset is larger than NVAR size
275	Could not adjust NVAR params
276	Could not write NVAR value
277	Could not read NVAR value
278	Failed to write binary file for NVAR
279	Certificate NVAR size mismatch
280	Certificate NVAR name field size mismatch
281	Failed to save intermediate file
282	Unable to access data
283	Detected duplicate syscall id
284	Detected duplicate syscall name
285	Detected duplicate syscall group name
286	Detected loop in syscall group dependencies
287	Detected invalid syscall group name
288	Detected invalid syscall group raw value
289	Detected invalid syscall name in group definition
290	Detected invalid syscall id
291	Detected invalid syscall group id used in process module
292	Failed to generate header file definitions
293	Invalid Group Value
294	SystemResources Class has not been initialized
295	Internal error
296	Failed to get active module names
297	Unable to resolve type
298	Failed to generated system resources report
299	Failed to generate source code for bus driver
300	Detected duplicate process name
301	Detected duplicate process id
302	File write error
303	Detected duplicate user name
304	Detected duplicate special file label

Error Code	Error Message
305	Detected duplicate service name
306	Detected duplicate group id
307	Detected duplicate user id
308	Error opening file for read
309	Error reading file data
310	Size requested was too large
311	Error opening file for write
312	Error appending to file
313	Creating directory structure
314	Error running LZMA compression
315	Error running LZMA extraction
316	Wrong format found
317	Unknown Project
318	Invalid data pointer
319	Out of memory
320	Unable to remove file entry from FCS table
321	MFS was not initialized
322	FCS was not initialized
323	Failed to process FCS entries
324	Detected duplicate special file label
325	FileEntry already being used by another FCS table
326	Failed to create FCS handle
327	Failed to get file from FCS
328	Failed to get file attributes from FCS
329	Failed to add new file to FCS
330	Failed to delete file from FCS
331	Failed to flush FCS buffer into memory
332	Invalid FCS file
333	Failed to terminate MFS library
334	Failed to get file size from MFS
335	Failed to delete file from MFS
336	Failed to decompose CSE image
337	Failed to initialize MFS
338	Failed to load Intel.cfg table
339	Failed to load Fit.cfg table
340	Failed to create the current values table
341	Failed to generate the current values table
342	Failed to create new Fit.cfg table
343	NVAR Access error
344	FW Code Generation Error
345	Invalid ME Version
346	Module Not Found
347	Action not found
348	Action failed to execute
349	Failed to process input XML
350	Invalid command line options
351	Failed to save XML
352	Invalid XML template option specified
353	Invalid Manifest Version specified on CLI
354	Unable to load tool config xml
355	Unsupported signing tool specified
356	Invalid signing tool configuration
357	Invalid decomp binary type specified
358	File is not a valid XML file
359	Invalid manifest index value
360	Error finding manifests in file
361	Path provided is not a valid directory
362	Unable to find files
363	Unable to read file
364	Failed to import manifest(s)
365	Failed to resign manifest(s)

Error Code	Error Message
366	Failed to generate public key hash
367	Failed to export manifest(s)
368	Buffer overflow detected
369	Buffer overflow detected
370	Failed to load file
371	Invalid value specified
372	Failed to parse Part IDs
373	Failed to save Part ID to file
374	Unable to remove directory
375	Signature verification failed
376	Failed to generated Boot Partition Manifest
377	Invalid DnxRecoveryImage configuration
378	Failed to generate DNX image
379	Invalid ME
380	Error Parsing Manifest
381	Error Parsing Missing Partition
382	Error Modifying Invalid ME
383	Error Modifying WCOD
384	Error Modifying LOCL
385	Utility to build the DNX image
386	Utility DNX configuration file
387	Invalid OEM Key Manifest path
388	Compressor unexpected exit code
389	Unable to get process uncompressed size
390	Unable to load file
391	Invalid LUT size

B.2 Command line tools errors

Error Code	Error Message
0	Success
1	Tool common error
2	Passed with warning
3	Internal Error. Unexpected error occurred
4	Unsupported OS
5	Memory allocation error occurred
6	Error accessing the function “GetSystemFirmwareTable” from kernel32.dll
7	The function “GetSystemFirmwareTable” failed with Windows Error Code: %d
8	Error accessing the kernel32.dll
9	Commit Anti Rollback SVN failed
10	Error occurred while reading the file
11	Error getting current working directory path
12	Error getting current working directory permission
13	An unknown error occurred while opening the file
14	An unknown error occurred while working with the file
15	Error occurred while writing to the file
16	Error while trying to read the signature of the file %s

Error Code	Error Message
17	The file %s, is not signed by Intel(R) Embedded Subsystems and IP Blocks Group
18	Invalid certificate information residing in file %s
19	Failed to write 0x%02X to IO Port 0x%04X
20	Cannot locate ME device
21	Write register failure
22	Circular buffer overflow
23	Communication error between application and Intel(R) ME module
24	Unsupported HECI bus message protocol version
25	HECI Timeout
26	Unexpected result in command response
27	Cannot find host client
28	Cannot find ME client
29	Failure occurred during ME disconnect
30	Client already connected
31	No free connection available
32	Flow control error
33	No message
34	Buffer size is too large
35	Buffer is too small
36	%s is too long
37	Invalid command line option(s)
38	The following Parameter is not a valid option: %s
39	PCH is not supported
40	Internal Error (Safe function wrapper error: Invalid size)
41	Internal Error (Safe function wrapper error: compose string from list)
42	Internal Error (Safe function wrapper error: compose string)
43	Internal Error (Safe function wrapper error: memncpy)
44	Internal Error (Safe function wrapper error: strncpy)
45	Internal Error (Safe function wrapper error: strncat)
46	Internal Error (Safe function wrapper error: strtok)
47	Printf function failed
48	Failed getting variable %s value
49	The variable %s is supported on Corporate SKU only
50	Unable to find matching LOCL
51	Could not access PCI device
52	Unable to load library
53	Unable to change permission
54	Unable to perform request due to permission failure
55	Cannot find requested device
56	Unable to perform CreateFile
57	The FPF compare failed
58	The CSE File Component requested, %s, is not valid for this operation

Error Code	Error Message
59	The CSE File Component requested, ID is not valid for this operation
60	Failed to read FPT NVARs config file. %s
61	Fail to read FW Status Register value
62	Fail to create verbose log file
63	Unknown or unsupported hardware platform. %s)
64	Failed to initialize SPI interface
65	Could not update [%s]
66	Cannot update %s. Invalid data length
67	Feature not found
68	Feature not available
69	Anti-Rollback SVN feature is disabled
70	%s actual value is - %s
71	FW status test failed
72	Boot Guard status test failed
73	Parameter %s - %s
74	The value of %s is missing
75	Failed to communicate with CSME. This tool must be run from a privileged account
76	Master Access config file value for %s format is invalid
77	Failed to retrieve feature
78	Master Access config file value for %s exceed maximum allowed value
79	Failed to retrieve Intel (R) FIT version
80	Failed to retrieve Intel (R) Internal Build Version
81	Ambiguous Master Access value. Master Access config file region %s defined more than once
82	MEManuf Operation Failed
83	Invalid Access node name in Master Access configuration file
84	Invalid RequiredValue node name in Master Access configuration file
85	Intel(R) test failed to start, error 0x%X returned
86	NA
87	Intel(R) test timeout (exceeded 30 seconds)
88	Intel(R) ME test is currently running, try again later
89	MEManuf EOL & BIST config file generation failed
90	M3 results are not available from SPI. Please run -test option to perform the BIST test
91	Could not read M3 results from SPI
92	SMBus hardware is not ready
93	Internal error - SMBus Read Byte PEC failure
94	SMBus encountered time-out
95	Signature: invalid! No more information can be displayed
96	Internal error - Failed to match
97	Internal error - Out of memory
98	Internal error - Unable to get current PP
99	Failed to retrieve test result from SPI

Error Code	Error Message
100	Failed to retrieve power package setting
101	Failed to retrieve power rule from SPI
102	WLAN power well setting is set incorrectly
103	Failed to retrieve test result from SPI
104	Internal error - Failed to retrieve Platform Attribute
105	Failed to retrieve PROC_MISSING NVAR setting
106	PROC_MISSING NVAR setting is set incorrectly
107	Failed to retrieve password from SPI
108	Internal error - Password length is incorrect
109	Internal error - Modified local password
110	Internal error - Invalid password
111	Boot Guard Self-Test Failed
112	Intel integrated LAN setting is set incorrectly
113	Intel LAN Connected Device (PHY) physical connectivity error with ME
114	Internal error - Illegal data length
115	Internal error - Illegal data value
116	EHBC State Test Failed - Error while reading data from flash
117	EHBC State Test Failed - Contradiction with current Privacy Level
118	Current WLAN does not match micro-code, please update WLAN micro-code in FW
119	Communication with WLAN device failed
120	Length of OEM Customizable Certificate Friendly Name setting is set incorrectly
121	OEM Customizable Certificate Stream setting is set incorrectly
122	OEM Customizable Certificate Hash Algorithm setting is set incorrectly
123	Length of OEM Customizable Certificate Stream is set incorrectly
124	Internal error - Unable to compress
125	The compressed data is incorrect
126	USBr EHCI 1 Enabled and/or USBr EHCI 2 Enabled setting is set incorrectly
127	KVM device is already in use by other components
128	Failed to retrieve power source
129	Power source is not AC
130	LAN power well setting is set incorrectly
131	WLAN power well setting is set incorrectly
132	System UUID actual value is all 0x00
133	System UUID actual value is all 0xFF
134	Security Descriptor Override Strap (SDO) is enabled
135	End-Of-Post message is not sent
136	Unable to determine Intel(R) ME Manufacturing Mode status
137	Intel(R) ME is still in Manufacturing Mode
138	BIOS has granted Intel(R) Gbe and/or ME access to its region
139	%s mismatch, actual value is - %s
140	NA

Error Code	Error Message
141	Cannot run the command since Intel(R) AMT is not available
142	MFS is corrupted
143	Using wrong PCH SKU Emulation via Intel (R) FIT vs what is the actual HW Type
144	Cannot perform hibernation. Please manually reboot the system
145	MEManuf Test Failed
146	Test is enabled by the user but is unknown by the platform - %s
147	Attempting to add sibling to XML root node
148	File size is zero
149	XML parsing failed
150	XML parsing encountered data overflow
151	Invalid XML error code conversion
152	XML parser - out of memory error
153	Missing RequiredValue xml node in Master Access configuration file
154	Incorrect region name in Master Access configuration file
155	Failed to retrieve list of BIST tests to run from FW
156	Unexpected failure when retrieving BIST results
157	Retrieving the EOL Config list of tests failed
158	Retrieving the EOL Var list of tests failed
159	No name attribute specified for test: %s
160	Failed to parse configuration file provided
161	No output file path specified to write configuration file
162	No data to write to configuration file
163	Invalid ErrAction specified
164	The 2 SPI flash devices do not have compatible command sets
165	No SPI flash device could be identified. Please verify if Fparts.txt has support
166	Failed to allocate memory for the flash part definition file %s
167	Parsing file failed
168	Protected Range Registers are currently set by BIOS, preventing flash access. Please contact the target system BIOS vendor for an option to disable Protected Range Registers
169	Hardware sequencing failed. Make sure that you have access to target flash area
170	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
171	An attempt was made to read beyond the end of flash memory
172	Software sequencing failed. Make sure that you have access to target flash area
173	Invalid Block Erase Size
174	Invalid Write Granularity value
175	Invalid Enable Write Status Register Command value
176	The supplied zero-based index of the SPI Device is out of range
177	Invalid descriptor region

Error Code	Error Message
178	Region does not exist
179	An attempt was made to write beyond the end of flash memory
180	An attempt was made to erase beyond the end of flash memory
181	The address 0x%08X of the block to erase is not aligned correctly
182	Hardware timeout occurred in SPI device
183	There are no supported SPI flash devices installed. Please check connectivity
184	Unrecognized value in the HSFSTS register
185	AEL is not equal to zero
186	FCERR is not equal to zero
187	Checking variable %s failed
188	Invalid value for %s CVAR
189	Invalid Manufacturing Line Configurable variable name %s
190	File does not exist
191	End of Manufacturing Operation failure - Verification failure on Descriptor Lock settings
192	Unable to get master base address from the descriptor
193	Password does not match the criteria
194	Invalid length of Manufacturing Line Configurable value. Check configuration file for correct length
195	Invalid hash certificate file
196	End of Manufacturing Operation failure - Verification failure on ME Manufacturing Mode Done settings
197	cfg_rules: the requested rule change is not supported after end of manufacturing
198	Invalid parameter value specified by user. Use -? option to see help
199	ME disabled
200	Failed to get information about the installed flash devices
201	An error occurred reading the flash descriptor signature
202	Flash descriptor does not have correct signature
203	The attempt to commit the Manufacturing Line Configurables has failed
204	Access was denied opening file
205	Failed to read the entire file into memory. File: %s
206	The address is outside the boundaries of the flash area
207	Unable to write data to flash. Address 0x%x
208	Data verify mismatch found
209	Failed to write the entire flash contents to file
210	An error occurred reading the flash mapping data
211	System booted in Non-Descriptor mode, but the flash appears to contain a valid signature
212	An error occurred reading the flash components data
213	An error occurred reading the flash region base/limit data
214	An error occurred reading the flash master access data
215	Flash is not blank
216	PAVP OEM config data: invalid EDP port value

Error Code	Error Message
217	Setting Global Reset Failed
218	ME disable not needed
219	ME already disabled
220	The request to disable the ME failed
221	There is a problem with the GbE binary which prevents saving the data
222	A required parameter is missing
223	Committing the FPF is not allowed at this time
224	The FPF has already been committed
225	PAVP OEM config data: invalid LSPCON port value
226	Committing a specific FPF is not supported. Consider committing all the FPFs
227	Keybox file size invalid
228	Invalid all hashes state file
229	Invalid idle timeout file
230	Invalid provisioning state file
231	CEK is invalid
232	CEK is not available
233	Cannot provision after EOM
234	Invalid redirection state file
235	Bad CRC
236	Bad Magic
237	Invalid EHBC state file
238	Keybox is not provisioned
239	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
240	User selected to cancel the operation
241	Internal error - Invalid HECI response length
242	Error determining possible system states
243	Cannot locate MEI driver
244	Unexpected internal FW error occurred. Object was not found
245	Invalid State found for test - %s
246	ISH Internal Error
247	IUP Not Found
248	Cannot locate HID device
249	Incorrect Report ID received
250	MCTP SMBUS test failed
251	Invalid config file. State was not found for test - %s
252	Invalid config file. RequiredValue was not found for test - %s
253	Invalid config file. \ErrAction\ was not found for test - %s
254	Unable to validate address range
255	Memory window is not set, or device is not armed for operation
256	Sensor could not be found. Either no sensor is connected, the sensor has not yet initialized, or the system is improperly configured

Error Code	Error Message
257	Not enough memory/storage for requested operation
258	Used in TOUCH_SENSOR_HID_READY_FOR_DATA_RSP to indicate sensor has been disabled or reset and must be reinitialized
259	Used to indicate compatibility revision check between sensor and ME failed, or protocol ver between ME/HID/Kernels failed
260	Indicates sensor went through an unexpected reset
261	Requested sensor reset failed to complete
262	Operation timed out
263	Test mode pattern did not match expected values
264	Indicates sensor reported fatal error during reset sequence. Further progress is not possible
265	Indicates sensor reported non-fatal error during reset sequence. HID/ BIOS logs error and attempts to continue
266	Indicates sensor reported invalid capabilities, such as not supporting required minimum frequency or I/O mode
267	Indicates that command cannot be complete until ongoing Quiesce I/O flow has completed
268	Cannot find the NVAR file; the system maybe in EOM
269	Invalid cfg rule data
270	Cannot access the NVAR file attributes
271	Failed to hash CSE file data
272	Operation is not allowed after EOM
273	Used an invalid input parameter to access the NVAR file
274	FPF is not written
275	Invalid privacy level file
276	File is invalid
277	Cannot provision after EOM
278	Certificate verification failed
279	HDCP Rx is not provisioned
280	Invalid string value entered for the Manufacturing Line Configurable
281	Detected ME in recovery mode
282	FW returned status: Erase token failure
283	Detected invalid data size
284	Detected invalid hex value
285	Failed to retrieve 5K port setting
286	Failed to retrieve LSPCON Port setting
287	Display port settings are not correct
288	EC Region write access permissions don't match Intel recommended values
289	Unexpected size found in the file %s. Expected: 0x%X. Received: 0x%X
290	Unable to execute command in this Firmware State. Please reboot
291	GBE Region write access permissions don't match Intel recommended values
292	GPIO file contains GPIO pin assignments that are not multiples of the GPIO pin data structure
293	ME Region write access permissions don't match Intel recommended values

Error Code	Error Message
294	Mismatch on FPF file %s - UEP: %s, FPF HW: %s
295	FPFs are not committed to HW
296	BIOS Region write access permissions don't match Intel recommended values
297	Failed to read FPF HW
298	SOC Config Lock is not set
299	Lock bit FPF is not set on file
300	Failed to read FPF in UEP
301	FW Update OEM ID incorrectly set to 00 or FF
302	Unable to determine FW Update OEM ID status
303	BIOS Region read access permissions don't match Intel recommended values
304	ME Region read access permissions don't match Intel recommended values
305	GBE Region read access permissions don't match Intel recommended values
306	EC Region read access permissions don't match Intel recommended values
307	RPMC SPI device did not initialize RPMC support correctly, RPMC SPI device needs replacement/ refurbishment
308	RPMC SPI device has not been bound to the platform yet, RPMC manufacturing process is not complete
309	HW Binding state is not enabled
310	The %s var is not updatable
311	The variable %s is not supported on this platform
312	PCH is unlocked. Disable Delayed Authentication Mode and retry
313	Test required value format is not valid
314	Invalid BootGuard configuration
315	Minimum ARB SVN value set on current platform does not match corresponding ARB SVN in FW image
316	Unexpected internal FW error occurred. Invalid parameter
317	Platform name for this PCH type not found or not exists
318	Clear option is not supported for FPFs
319	This command cannot be processed on platforms using %s as the storage type
320	This command cannot be processed. Region is not supported on this platform
321	The maximum number of updated NVARs has been reached
322	Invalid value for this CVAR
323	The VAR compare failed
324	Fatal flash logs exist in NVM
325	Request and Reply messages' size mismatch
326	Intel (R) ME Interface: Unsupported message type
327	Specified partition was not found in the Update Image
328	FPT is not found in the image
329	Full FW Update using same version is not allowed. Include -allowsv in command line to allow it

Error Code	Error Message
330	Restore Point Image Failure. Reboot may be required
331	Invalid Partition ID. Use a Partition ID which is possible to do Partial FW Update on
332	The partition provided is not supported by the platform
333	The requested size of partition to read/write/erase exceeds the actual partition size
334	Firmware Update operation not initiated because a firmware update is already in progress
335	Sku capabilities bits are different between the Update Image and the Flash Image
336	Major version number of Update Image is not the same as major version number of Flash Image
337	Firmware update failed due to an internal error. The total size of the backup partitions is bigger than NFTP size. Can happen in Consumer, when not setting fixed partitions sizes in build
338	Firmware update failed due to an internal error caused by a failure in event publishing
339	FW Flash read/write/erase operation failed
340	Update operation timed-out; cannot determine if the operation succeeded
341	FW Update is disabled. MEBX has options to disable / enable FW Update
342	Firmware update cannot be initiated because the OEM ID given for FW Update did not match the OEM ID in the FW
343	Display FW Version failed
344	Update was blocked by one of the FW modules
345	Firmware update failed due to an internal error. Write file failed: error occurred in write() or number of bytes written is not the same as file length
346	Sanity check in erase/write of partitions. Error might have happened when size of partition is not 4K aligned
347	Firmware update failed due to an internal error. Firmware returns invalid flash code partition
348	Firmware update failed due to an internal error NFTP is corrupted, CSE is in Recovery Mode
349	Host reset is required after the last FW Update operation
350	Update to Image with lower TCB SVN is not allowed
351	Partial update is allowed only to the expected instance ID of an IUP The Update Image contains IUP with instance ID that is not the currently expected one by the FW. To update LOCL, please use The Intel Management and Security Status (IMSS) tool
352	Partial Update is not allowed, because CSE is in Recovery Mode
353	Partial Update of an IUP was requested, but this IUP doesn't exist in the Flash Image
354	Get Restore Point Image is not allowed, because FW Update is in progress. (The regular FW Update will continue
355	Update to Image with lower VCN is not allowed
356	SVN invalid: SVN larger than 254 is not allowed
357	SVN partition is full, so cannot update to higher SVN
358	Restore Point Image was requested, but it is not allowed because CSE is in Recovery Mode

Error Code	Error Message
359	Display Partition Version failed
360	Restore Point Image was requested, but there was Full/Partial FW Update before without Restart after it
361	Update to incompatible PMC: The PMC instance ID is different, which may be due to H/LP SKU incompatibility
362	Update to incompatible H/LP SKU image
363	Update Image length is bigger than the expected size of the image according to its size in the flash. For example: Error on updating from Consumer to Corporate
364	Manifest size in Update Image is bigger than 8KB, or exceeds the Update Image size
365	Failed to open loader (DEV_FD_LDR_VERIFY_MAN) to verify manifest
366	Failed to open loader (DEV_FD_LDR_VERIFY_MAN) to install / uninstall keys
367	Failed to verify signature of OEM or RoT key manifests. For example: Error on update from Production to Pre-Production
368	ldr_uninstall_keys() failed - uninstall keys for OEM partitions (ISHC/ IUNP)
369	Call to sku_mgr functions failed
370	Call to cfgmgr functions failed. cfgmgr_get_rule(), cfgmgr_set_rule()
371	Manifest not found in partition (in Update or Flash Image)
372	Crypto operation (calculating hash of partition) failed
373	Loader failed to verify manifest signature of FTPR. Production vs. Pre-Production
374	Loader failed to verify manifest signature of NFTP
375	Loader failed to verify manifest signature of IDLM
376	Loader failed to verify manifest signature of RBE
377	Loader failed to verify manifest signature of PMC
378	Loader failed to verify manifest signature of OEM KM
379	Loader failed to verify manifest signature of WCOD
380	Loader failed to verify manifest signature of LOCL
381	Loader failed to verify manifest signature of PCHC
382	Loader failed to verify manifest signature of IOMP
383	Loader failed to verify manifest signature of MGPP
384	Loader failed to verify manifest signature of TBTP
385	Loader failed to verify manifest signature of ISHC
386	Loader failed to verify manifest signature of IUNIT
387	Some manifest extension is missing in FTPR
388	Some manifest extension is missing in NFTP
389	Some manifest extension is missing in IDLM
390	Some manifest extension is missing in RBE
391	Some manifest extension is missing in PMC. Wrong MEU Tool was used to create the partition
392	Some manifest extension is missing in OEM KM. Wrong MEU Tool was used to create the partition
393	Some manifest extension is missing in WCOD
394	Some manifest extension is missing in LOCL

Error Code	**Error Message**
395	Some manifest extension is missing in PCHC. Wrong MEU Tool was used to create the partition
396	Some manifest extension is missing in IOMP. Wrong MEU Tool was used to create the partition
397	Some manifest extension is missing in MGPP. Wrong MEU Tool was used to create the partition
398	Some manifest extension is missing in TBTP. Wrong MEU Tool was used to create the partition
399	Some manifest extension is missing in ISHC. Wrong MEU Tool was used to create the partition
400	Some manifest extension is missing in IUNIT. Wrong MEU Tool was used to create the partition
401	FTPR partition hash and calculated hash are not the same. If partition hash is zero - wrong MEU Tool was used to create the partition
402	NFTP partition hash and calculated hash are not the same. If partition hash is zero - wrong MEU Tool was used to create the partition
403	DLMP partition hash and calculated hash are not the same. If partition hash is zero - wrong MEU Tool was used to create the partition
404	RBEP partition hash and calculated hash are not the same. If partition hash is zero - wrong MEU Tool was used to create the partition
405	PMCP partition hash and calculated hash are not the same. If partition hash is zero - wrong MEU Tool was used to create the partition
406	OEMP partition hash and calculated hash are not the same. If partition hash is zero - wrong MEU Tool was used to create the partition
407	WCOD partition hash and calculated hash are not the same. If partition hash is zero - wrong MEU Tool was used to create the partition
408	LOCL partition hash and calculated hash are not the same. If partition hash is zero - wrong MEU Tool was used to create the partition
409	PCHC partition hash and calculated hash are not the same. If partition hash is zero - wrong MEU Tool was used to create the partition
410	IOMP partition hash and calculated hash are not the same. If partition hash is zero - wrong MEU Tool was used to create the partition
411	MGPP partition hash and calculated hash are not the same. If partition hash is zero - wrong MEU Tool was used to create the partition
412	TBTP partition hash and calculated hash are not the same. If partition hash is zero - wrong MEU Tool was used to create the partition
413	ISHC partition hash and calculated hash are not the same. If partition hash is zero - wrong MEU Tool was used to create the partition
414	IUNP partition hash and calculated hash are not the same. If partition hash is zero - wrong MEU Tool was used to create the partition
415	Place holder. This error code will not be returned by the FW
416	Place holder. This error code will not be returned by the FW
417	Place holder. This error code will not be returned by the FW
418	Place holder. This error code will not be returned by the FW
419	PMCP must have the same major API version as the version inside the list in FTPR, in the Update Image for Full Update
420	OEMP must have the same major API version as the version inside the list in FTPR, in the Update Image for Full Update
421	WCOD must have the same major API version as the version inside the list in FTPR, in the Update Image for Full Update, in the Flash Image for Partial Update

Error Code	Error Message
422	LOCL must have the same major API version as the version inside the list in FTPR, in the Update Image for Full Update, in the Flash Image for Partial Update
423	PCHC must have the same major API version as the version inside the list in FTPR, in the Update Image for Full Update, in the Flash Image for Partial Update
424	IOMP must have the same major API version as the version inside the list in FTPR, in the Update Image for Full Update, in the Flash Image for Partial Update
425	MGPP must have the same major API version as the version inside the list in FTPR, in the Update Image for Full Update, in the Flash Image for Partial Update
426	TBTP must have the same major API version as the version inside the list in FTPR, in the Update Image for Full Update, in the Flash Image for Partial Update
427	ISHC must have the same major API version as the version inside the list in FTPR, in the Update Image for Full Update, in the Flash Image for Partial Update
428	IUNP must have the same major API version as the version inside the list in FTPR, in the Update Image for Full Update, in the Flash Image for Partial Update
429	The size of an Update partition size is bigger than the size of the Flash partition
430	Location of partition to backup is not inside NFTP
431	The number of IUPs in the Update/Flash Image is bigger than MAX_IUPS
432	Partition name inside IUPs list (in FTPR manifest extension) is not IUP
433	Non-optional IUP (like LOCL, WCOD) inside IUPs list (in FTPR manifest extension) is not in the Update Image
434	PMC partition is not in the Update Image
435	It is not allowed to do Partial Update on this partition
436	It is not allowed to do Partial Update on Type-C partitions, according to NVAR
437	RBEP and NFTP must have the same version as FTPR, in the Update Image
438	RBEP and NFTP must have the same SVN as FTPR, in the Update Image
439	RBEP and NFTP must have the same VCN as FTPR, in the Update Image
440	Non-optional IUPs (like LOCL, WCOD) must have the same major build version as FTPR, in the Update Image for Full Update, in the Flash Image for Partial Update
441	Update IUP must not have SVN smaller than SVN of Flash IUP
442	Update Image length is not the same as Flash Image length
443	Update IUP must not have VCN smaller than VCN of Flash IUP
444	Update from PV bit ON to PV bit OFF is not allowed
445	Update to PV bit OFF on Revenue platform is not allowed
446	Update to higher SVN must be an upgrade - to higher build version
447	Update to higher SVN must be to a higher Hot Fix number (the third number in the build version)
448	Non-optional IUP (like LOCL, WCOD) inside IUPs list (in FTPR manifest extension) is not in the Flash Image

Error Code	Error Message
449	A partition that was searched in the Update Image is not in it
450	Update between engineering build vs regular build is not allowed. Both builds have to be the same type: regular or engineering build. Engineering build is 7000 and above. Regular build is below 7000
451	OEM KM partition is not in the Update Image, but ISHC/IUNP is in the Update Image, which is not allowed
452	ISHC/IUNP do not exist in the same way in the Update Image and in the Flash Image
453	OEM KM partition is not in the Flash Image, but it is in the Update Image, which is not allowed.")
454	Partial FW Update: The Update Image contains IUP that is different than the one that was requested to be updated in the Partial Update command
455	The Partial Update Image size is different than the size of the IUP in it (as it is in the manifest). This means that the Update Image contains more (or less) than the IUP partition.
456	Bug: Open of IUP path failed. Need to add the path in Storage or add permissions to FW Update process.
457	Bug: spi_flash_partition_updated() failed. This updates the files (in the file system) of the newly updated IUP, after Partial Update (without reset).
458	Update Rule file contains invalid value. (This file holds the MEBX option for FW Update: values: disable / enable / password protected).
459	Call to pwr function failed. pwr_state_get_last_reset_reason()
460	Call to spi function failed. spi_flash_get_override_strap()
461	Get Restore Point Image is not allowed, because a previous Get Restore Point operation already started. Both operations will be aborted. (Get Restore Point can be started again after this).
462	Bug: Get Restore Point Image Data: The offset of Restore Point Image is bigger than the Image length.
463	Heci message length is not as expected.
464	FWU_START_MSG Heci message contains invalid value in UpdateEnvironment. Value should be FWU_ENV_MANUFACTURING. (Other possible value: FWU_ENV_IFU is obsolete)
465	FWU_DATA Heci command was sent, but the FW Update wasn't started with FWU_START Heci command before it
466	Call to storage_nvm function failed
467	FW Update is not possible on UFS Flash after End of Post (after the OS is running). It is possible only before the OS is running using Bios Capsule Update
468	DPHY must have the same major API version as the version inside the list in FTPR, in the Update Image for Full Update, in the Flash Image for Partial Update
469	DPHY partition hash and calculated hash are not the same. If partition hash is zero - wrong MEU Tool was used to create the partition.
470	Some manifest extension is missing in DPHY. Wrong MEU Tool was used to creat
471	Loader failed to verify manifest signature of DPHY
472	Update to higher TCB SVN must be also to higher ARB SVN
473	Invalid Partition ID. Use a Partition ID which is on the Flash Image
474	Display Partition Vendor ID failed

Error Code	Error Message
475	Wrong structure of Update Image (manifests, $CPD), complete_partition_length is 0, no module or metadata of preupdate inside UPDT partition.
476	Flash Image content is invalid (partitions/manifests sizes, locations, structures).
477	FW Update process called to PG entry override (sys_pg_override()) at the start of the update, and it returned error.
478	clear_ipk_valid_bit() returned error. This function prevents CSE from entering M3 after FW Update, and instead CSE will go into MOff.
479	Error when flushing NVM to UMA space (before rewriting flash)
480	FWU_END Heci command was sent, but there was no FWU_DATA command before it
481	FWU_DATA Heci command has invalid data length (too big)
482	FW Update process received Heci command message with unknown command type
483	Cannot obtain ME Mode
484	Local FW Update only supported when ME Mode=Normal
485	BIOS does not support boot measurements
486	BIOS does not support Trusted Device Setup boot
487	NA
488	ODM ID \\ System Integrator ID \\ Reserved ID: value already set
489	File already exists
490	ME FW version mismatch, actual value is - %s
491	Intel(R) Gbe version mismatch, actual value is - %s
492	BIOS version mismatch, actual value is - %s
493	System UUID mismatch, actual value is - %s
494	Intel(R) Wired LAN MAC address mismatch, actual value is - %s
495	Intel(R) Wireless LAN MAC address mismatch, actual value is - %s
496	Wireless LAN micro-code mismatch, actual value is - %s
497	Firmware Update OEM ID mismatch, actual value is - %s
498	Touch - Vendor ID mismatch, actual value is - %s
499	Invalid PKI suffix file
500	Update to Image with lower ARB SVN is not allowed
501	RBEP and NFTP must have the same unique build as FTPR, in the Update Image
502	Disable FIPS mode failed
503	RPMB fuse is set. Cannot commit FPFs
504	PCHC partition is not in the Update Image
505	Mismatch between FPF UEP and HW values
506	Invalid Update Image length, size is smaller than required
507	The internal structure of the Update Image is corrupted
508	Update Image has wrong structure for Full Update operation
509	Update Image has wrong structure for Partial Update operation
510	Mandatory partitions (FTPR / NFTP / RBEP) were not found in the Update Image
511	Number of IUPs in FW exceeds allowed maximum

Error Code	Error Message
512	NA
513	Missing a required partition manifest in the Update Image
514	Missing a required partition manifest extension in the Update Image
515	The VAR invalid data size
516	Update Image size exceeds allocated buffer
517	FW failed to read FWSTS register
518	Firmware update failed due to an internal error. Read file failed: error occurred in read() or number of bytes read is not the same as file length.")
519	PG in progress, no override is allowed during such state
520	Full FW Update using same version is not allowed. Include /s in command line to allow it
521	WLAN uCode is already updated to the expected instance. Include -allowsv in command line to force update.
522	FW failed to set ISH configuration file
523	PCIe connectivity failure. Unable to connect to vPro NIC through designated bus.
524	SMBUS connectivity failure. Unable to connect to vPro NIC through designated bus.
525	Conflict in OEM Data: Overlapping values of LSPCON Port Config and eDP Port Config found.
526	Invalid configuration server FQDN value.
527	Invalid host FQDN file.
528	One or more GPIO pads provided in file have invalid ownership mode set.
529	One or more GPIO pads provided in file have invalid pad mode set.
530	Two or more GPIO pads provided in file have same feature field value set.
531	One or more GPIO pads provided in file have invalid feature field value set.
532	Invalid cert hash file.
533	Invalid host FQDN domain name.
534	Invalid host FQDN hostname.
535	ODM ID \\ System Integrator ID \\ Reserved ID: invalid size.
536	ODM ID \\ System Integrator ID \\ Reserved ID: invalid value.
537	One or more GPIO pads provided in file have invalid pad address set (group / pad number).
538	Two or more GPIO pads provided in file have same pad address set.
539	Update this var is not supported if AMT is provisioned.
540	Unsupported combination of EHBC state and privacy level files.
541	CSE is in Recovery Mode but FWSTS registers report Normal Mode.
542	The Flash Image that was burned on the platform was corrupted. CSE is in Recovery Mode at first boot.
543	Clear option is not supported for Hashed vars
544	FW returned status: ICC_STATUS_FAILURE
545	FW returned status: ICC_STATUS_INCORRECT_API_VERSION
546	FW returned status: ICC_STATUS_INVALID_FUNCTION
547	FW returned status: ICC_STATUS_INVALID_BUFFER_LENGTH

Error Code	Error Message
548	FW returned status: ICC_STATUS_INVALID_PARAMS
549	FW returned status: ICC_STATUS_FLASH_WEAR_OUT_VIOLATION
550	FW returned status: ICC_STATUS_FLASH_CORRUPTION
551	FW returned status: ICC_STATUS_PROFILE_NOT_SELECTABLE_BY_BIOS
552	FW returned status: ICC_STATUS_TOO_LARGE_PROFILE_INDEX
553	FW returned status: ICC_STATUS_NO_SUCH_PROFILE_IN_FLASH
554	FW returned status: ICC_STATUS_CMD_NOT_SUPPORTED_AFTER_END_OF_POST
555	FW returned status: ICC_STATUS_NO_SUCH_RECORD
556	FW returned status: ICC_STATUS_FILE_NOT_FOUND
557	FW returned status: ICC_STATUS_INVALID_RECORD_FORMAT
558	FW returned status: ICC_STATUS_TOO_LARGE_UOB_RECORD
559	FW returned status: ICC_STATUS_CLOCK_NOT_CONFIGURABLE
560	FW returned status: ICC_STATUS_REGISTER_IS_LOCKED
561	FW returned status: ICC_STATUS_NO_VALID_PRE_UOB
562	FW returned status: ICC_STATUS_NO_VALID_PERM_UOB
563	FW returned status: ICC_STATUS_NO_DATA_FOR_THIS_CLOCK
564	FW returned status: ICC_STATUS_PROFILE_INDEX_IS_CURRENT
565	FW returned status: ICC_STATUS_NO_BCLK_ADJUSTMENT_FOUND
566	FW returned status: ICC_STATUS_WARM_RESET_RAMP_NOT_SUPPORTED
567	FW returned status: ICC_STATUS_UOB_RECORD_IS_ALREADY_INVALID
568	FW returned status: ICC_STATUS_NO_PROFILES_EXIST
569	FW returned status: ICC_STATUS_AUTH_FAILURE
570	FW returned status: ICC_STATUS_ERROR_READING_FILE
571	FW returned status: ICC_STATUS_RANGE_VIOLATION_FREQ_TOO_HIGH
572	FW returned status: ICC_STATUS_HW_VIOLATION_FREQ_TOO_HIGH
573	FW returned status: ICC_STATUS_PENDING_REVERT_TO_DEFAULT
574	FW returned status: ICC_STATUS_PENDING_SET_PROFILE
575	FW returned status: ICC_STATUS_UNVALID_PROFILE
576	FW returned status: ICC_STATUS_UNVALID_OEM_DATA
577	FW returned status: ICC_STATUS_ERROR_READING_DYNAMIC_RECORD
578	FW returned status: ICC_STATUS_RANGE_VIOLATION_FREQ_TOO_LOW
579	FW returned status: ICC_STATUS_HW_VIOLATION_FREQ_TOO_LOW
580	FW returned status: ICC_STATUS_GET_REGISTER_NO_SUCH_REG
581	FW returned status: ICC_STATUS_SSC_MODE_CHANGE_NOT_SUPPORTED
582	FW returned status: ICC_STATUS_RANGE_VIOLATION_SSC_TOO_HIGH
583	FW returned status: ICC_STATUS_SURVIVABILITY_SYNC_DISABLED
584	FW returned status: ICC_STATUS_WARM_RESET_FREQ_TOO_LOW
585	FW returned status: ICC_STATUS_NO_SUCH_TARGET_ID
586	FW returned status: ICC_STATUS_NO_SUCH_REGISTER
587	FW returned status: ICC_STATUS_INVALIDATE_SUCCESSFUL
588	FW returned status: ICC_STATUS_BUFFER_TOO_SMALL

Error Code	Error Message
589	FW returned status: ICC_STATUS_VALID_UOB_ALREADY_PRESENT
590	FW returned status: ICC_STATUS_WAITING_FOR_POWER_CYCLE
591	FW returned status: ICC_STATUS_SURVIVABILITY_TABLE_ACCESS_VIOLATION
592	FW returned status: ICC_STATUS_SURVIVABILITY_TABLE_TOO_LARGE
593	FW returned status: ICC_STATUS_NO_SUCH_EID
594	FW returned status: ICC_STATUS_SUCCESS_TRANSLATE_ONLY
595	FW returned status: ICC_STATUS_PCIE_FAIL_READING_DATA
596	FW returned status: ICC_STATUS_PCIE_FAIL_WRITING_DATA
597	FW returned status: ICC_STATUS_PCIE_CONFIG_INVALID
598	FW returned status: ICC_STATUS_CMD_NOT_SUPPORTED_BEFORE_DID
599	FW returned status: ICC_STATUS_FIA_MUX_CONFIG_SKU_MISMATCH
600	FW returned status: ICC_STATUS_FIA_MUX_NO_CONFIG_FOUND
601	FW returned status: ICC_STATUS_FIA_MUX_ERROR_GETTING_LANES_LIMIT
602	FW returned status: ICC_STATUS_FIA_MUX_ERROR_READING_CONF_FROM_FILE
603	FW returned status: ICC_STATUS_FIA_MUX_ERROR_PROMPTING_TO_GLOBAL_RESET
604	FW returned status: ICC_STATUS_FIA_MUX_INVALID_FIA_MUX_CONFIG
605	FW returned status: ICC_STATUS_FIA_MUX_ERROR_WRITING_CONF_TO_FILE
606	FW returned status: ICC_STATUS_FIA_MUX_ERROR_READING_CONF_FROM_STRAPS
607	FW returned status: ICC_STATUS_MAX_BUNDLES_PER_RECORD_REACHED
608	FW returned status: ICC_STATUS_PLL_UNSUPPORTED
609	FW returned status: ICC_STATUS_DATA_ITEM_UNSUPPORTED
610	FW returned status: ICC_STATUS_OEM_PROFILE_CRDR_VIOLATION
611	FW returned status: ICC_STATUS_OEM_PROFILE_CRDR_VIOLATION
612	FW returned unknown status.
613	Invalid argument.
614	AMT Ipv4 Interface is disabled.
615	Interface does not exists.
616	Invalid user consent policy file.
617	Generating file in System Folder is not allowed.
618	Input Configuration file contains CSE file name duplicates.
619	Update of partition between engineering build vs regular build is not allowed.
620	Unknown hardware platform.
621	Unsupported hardware platform %s The %s will actually be printed as: "HW: %s. Supported HW: %s." For example: "Unsupported hardware platform. HW: Cannonlake Platform. Supported HW: Icelake Platform."

C Tool Option Dependency on BIOS/Intel® ME Status

Tools' Options	Intel® ME End-of-Manufacturing NVAR		End of Post		CF9GR Locking	
	Set	**Not Set**	**Yes**	**No**	**Yes**	**No**
FPT -Greset	Not related	Not related	Not related	N/A Not related	Fail – DOS	Work
FPT –R	Depends on End of post status	Work	Depends on Intel® ME manufacturing mode donebit status	Work	Not related	Not related
Intel® MEINFO – EOL config	Depends on End of post status	Work	Depends on Intel® ME manufacturing mode donebit status	Work	Not related	Not related
All options for UpdPARAM	Not related	Not related	Fail	Work	Not related	Not related

§ §