

System Tools for SkyLake - Intel® Management Engine Firmware 11.0

User Guide

July 2015

Revision: 1.1

Intel Confidential



INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITCNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS OTHERWISE AGREED IN WRITING BY INTEL, THE INTEL PRODUCTS ARE NOT DESIGNED NOR INTENDED FOR ANY APPLICATION IN WHICH THE FAILURE OF THE INTEL PRODUCT COULD CREATE A SITUATION WHERE PERSONAL INJURY OR DEATH MAY OCCUR.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

All products, platforms, dates, and figures specified are preliminary based on current expectations, and are subject to change without notice. All dates specified are target dates, are provided for planning purposes only and are subject to change.

This document contains information on products in the design phase of development. Do not finalize a design with this information. Revised information will be published when the product is available. Verify with your local sales office that you have the latest datasheet before finalizing a design.

Intel® Active Management Technology (Intel® AMT) requires activation and a system with a corporate network connection, an Intel® AMT-enabled chipset, network hardware and software. For notebooks, Intel AMT may be unavailable or limited over a host OS-based VPN, when connecting wirelessly, on battery power, sleeping, hibernating or powered off. Results dependent upon hardware, setup and configuration. For more information, visit http://www.intel.com/technology/platform-technology/intel-amt.

KVM Remote Control (Keyboard, Video, Mouse) is only available with Intel Core™ i5 vPro and Core™ i7 vPro processors with Intel® Active Management technology activated and configured and with integrated graphics active. Discrete graphics are not supported.

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

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

Intel, Intel® vPro™, and the Intel logo are trademarks of Intel Corporation in the U.S. and other countries.

*Other names and brands may be claimed as the property of others.

Copyright© 2015, Intel Corporation. All rights reserved.



Contents

1 Introduction 8
1.1 Terminology 8
1.2 Reference Documents 14

2 Preface 15
2.1 Overview 15
2.2 Image Editing Tools 15
2.3 Manufacturing Line Validation Tool 16
2.4 Intel® ME Setting Checker Tool 16
2.5 Operating System Support 17
2.6 Generic System Requirements 17
2.7 Error Return 19
2.8 Usage of the Double-Quote Character (") 19
2.9 PMX Driver Limitation 20

3 Intel® Flash Image Tool 21
3.1 System Requirements 21
3.2 Flash Image Details 21
3.2.1 Flash Space Allocation 22
3.3 Required Files 22
3.4 Intel® FIT 23
3.4.1 Configuration Files 23
3.4.2 Creating a New Configuration 23
3.4.3 Opening an Existing Configuration 23
3.4.4 Saving a Configuration 23
3.4.5 Environment Variables 23
3.4.6 Modifying the Flash Descriptor Region 27
3.4.7 Descriptor Region Length 27
3.4.8 Setting the Number and Size of the Flash Components 28
3.4.9 Region Access Control 30
3.4.10 VSCC Table 33
3.4.11 Adding a New Table 33
3.4.12 Removing an Existing VSCC Table 34
3.4.13 Modifying the Intel® ME Region 35
3.4.14 Setting the Intel® ME Region Binary File 35
3.4.15 Intel® ME Section 35
3.4.16 Power 36
3.4.17 Manageability Application Section 37
3.4.18 Platform Protection 38
3.4.19 Provisioning Section 40
3.4.20 Gbe (LAN) Region Settings 42
3.4.21 Setting the Gbe Region Length Option 42
3.4.22 Setting the Gbe Region Binary File 42
3.4.23 Enabling/Disabling the GbE Region 42
3.4.24 Modifying the PDR Region 42
3.4.25 Setting the PDR Region Length Option 43



3.4.26 Setting the PDR Region Binary File 43
3.4.27 Enabling/Disabling the PDR Region 43
3.4.28 Modifying the BIOS Region 44
3.4.29 Setting the BIOS Region Length Parameter 44
3.4.30 Setting the BIOS Region Binary File 44
3.4.31 Enabling/Disabling the BIOS Region 44
3.4.32 Building a Flash Image 45
3.4.33 Decomposing an Existing Flash Image 45
3.4.34 Command Line Interface 46
3.4.35 Example – Decomposing an Image and Extracting Parameters 47
3.4.36 More Examples of FIT CLI 48

4 Flash Programming Tool 49
4.1 System Requirements 49
4.2 Flash Image Details 50
4.3 Microsoft Windows* Required Files 50
4.4 EFI Required Files 51
4.5 DOS Required Files 51
4.6 Programming the Flash Device 52
4.6.1 Stopping Intel® ME SPI Operations 52
4.7 Programming NVARS 52
4.8 Usage 53
4.9 Updating Hash Certificate Through NVAR 57
4.10 Fparts.txt File 59
4.11 Examples 59
4.11.1 Complete SPI Flash Device with Binary File 60
4.11.2 Program a Specific Region 60
4.11.3 Program SPI Flash from a Specific Address 61
4.11.4 Dump full image 61
4.11.5 Dump Specific Region 61
4.11.6 Display SPI Information 62
4.11.7 Verify Image with Errors 62
4.11.8 Verify Image Successfully 63
4.11.9 Get Intel® ME settings 63
4.11.10 CVAR Configuration File Generation (-cfggen) 64

5 Intel® MEManuf and MEManufWin 67
5.1 Windows* PE Requirements 67
5.2 How to Use Intel® MEManuf 67
5.3 Usage 68
5.3.1 Host based tests 73
5.4 Intel® MEManuf –EOL Check 73
5.4.1 MEManuf.cfg File 73
5.4.2 MEManuf –EOL Variable Check 100
5.4.3 MEManuf –EOL Config Check 101
5.4.4 Output/Result 101
5.5 Examples 101
5.5.1 Example 1 101

6 Intel® MEInfo 108
6.1 Windows* PE Requirements 108
6.2 Usage 108

6.3 Examples 118
6.3.1 Consumer Intel® ME FW SKU 118
6.3.2 Corporate Intel® ME FW SKU 121
6.3.3 Retrieve the Current Value of the Flash Version 123
6.3.4 Checks Whether the Computer has Completed the Setup and Configuration Process 123

7 Intel® ME Firmware Update 124
7.1 Requirements 124
7.2 Windows* PE Requirements 124
7.3 Enabling and Disabling Intel® FWUpdate 125
7.4 Usage 125
7.5 Examples 127
7.5.1 Updates Intel® ME with Firmware Binary File 127
7.5.2 Partial Firmware Update 128
7.5.3 Display Supported Commands 129
7.5.4 Language Codes 129

8 Intel® Manifest Extension Utility (MEU) 131
8.1 Usage 131
8.2 Examples 132
8.2.1 Generate Configuration XML template 132
8.2.2 Generate Code partition XML 132
8.2.3 Generate Compressed & Signed partition 133

9 Appendix A: Intel® ME NVARs 134

10 Appendix B: Tool Detail Error Codes 144

11 Appendix C: Tool Option Dependency on BIOS/Intel® ME Status 162



Figures

Figure 1: SPI Flash Image Regions 22
Figure 2. Environment Variables Dialog 25
Figure 3. Build Settings Dialog 27
Figure 4: PDR Region Options 43
Figure 5: BIOS Region Parameters 44
Figure 6: Raw Hash Values from Certificate File 58

Tables

Table 1: OS Support for Tools 17
Table 2: Tools Summary 19
Table 3: Flash Image Regions – Description 22
Table 4: Build Settings Dialog Options 26
Table 5: Region Access Control Table 30
Table 6: CPU/BIOS Access 31
Table 7: FIT Command Line Options 46
Table 8: Flash Image Regions – Description 50
Table 9: FPT OS Requirements 51
Table 10: Named Variables Options 52
Table 11: Command Line Options for fpt.efi, fpt.exe and fptw.exe 53
Table 12: FPT –closemnf Behavior 57
Table 13: Intel-Recommend Access Settings 57
Table 14: Options for the Tool 68
Table 15: Intel® MEManuf Test Matrix 72
Table 16: MEManuf - EOL Config Tests 101
MTable 17: Intel® MEInfo Command Line Options 108
Table 18: List of Components that Intel® MEINFO Displays 110
Table 19: Image File Update Options 126
Table 20: Options 131



Revision History

Revision	Description	Date
11.0.0.1080	Pre-Alpha Release	September 2014
11.0.0.1105	Alpha Release	January 2015
11.0.0.11xx	Engineering Release	January 2015
11.0.0.11xx	Beta Release	February 2015
11.0.0.11xx	PC Release	May 2015
11.0.0.11xx	PV Release	July 2015

§



1 Introduction

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

1.1 Terminology

Acronym / Term	Definition
3PDS	3rd Party Data Storage
AC	Alternating Current
Agent	Software that runs on a client PC with OS running
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
BBBS	BIOS Boot Block Size
BIN	Binary file
BIOS	Basic Input Output System
BIOS-FW	Basic Input Output System Firmware
BIST	Built In Self-Test
CCM	Client Control Mode (Host Based Setup and Configuration)
CLI	Command Line Interface
CRB	Customer Reference Board
DHCP	Dynamic Host Configuration Protocol
DIMM	Dual In-line Memory Module
DLL	Dynamic Link Library
DNS	Domain Naming System
EC	Embedded Controller
EEPROM	Electrically Erasable Programmable Read Only Memory
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



Acronym / Term	Definition
FCIM	Full Clock Integrated Mode
FCSS	Flex Clock Source Select
FDI	Flexible Display Interface
Intel® FIT	Intel® Flash Image Tool
FLOCKDN	Flash Configuration Lock-Down
FMBA	Flash Master Base Address
FOV	Fixed Offset Variable
FPSBA	Flash PCH Strap Base Address
Intel® FPT	Intel® Flash Programming Tool
FQDN	Fully Qualified Domain Name
FRBA	Flash Region Base Address
FW	Firmware
FWUpdate	Firmware Update
G3	A system state of Mechanical Off where all power is disconnected from the system. A G3 power state does not necessarily indicate that RTC power is removed.
GbE	Gigabit Ethernet
PCH	Peripheral Controller Hub
GPIO	General Purpose Input/output
GUI	Graphical User Interface
GUID	Globally Unique Identifier
HECI (deprecated)	Host Embedded Controller Interface
Host or Host CPU	The processor running the operating system. This is different than the management processor running the Intel® ME FW.
Host Service/ Application	An application running on the host CPU
HostIF	Host Interface
HTTP	HyperText Transfer Protocol
HW	Hardware
AMT	Intel® AMT
IBEN	Input Buffer Enable
IBV	Independent BIOS Vendor
ICC	Integrated Clock Configuration
ID	Identification

Acronym / Term	Definition
IDER	Integrated Drive Electronics Redirection
INF	An information file (.inf) used by Microsoft operating systems that support the Plug & Play feature. When installing a driver, this file provides the OS with the necessary information about driver filenames, driver components, and supported hardware.
Intel® AMT	The Intel® AMT Firmware running on the embedded processor
Intel® DAL	Intel® Dynamic Application Loader (Intel® DAL)
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
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
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
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.



Acronym / Term	Definition
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.
MAC address	Media Access Control address
MCP	Multi-Chip Package (Central Processing Unit / Platform Controller Hub)
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
OVR	Override
PAVP	Protected Video and Audio Path
PC	Personal Computer
PCI	Peripheral Component Interconnect
PCIe	Peripheral Component Interconnect Express
PDR	Platform Descriptor Region
PHY	Physical Layer
PID	Provisioning ID
PKI	Public Key Infrastructure

Acronym / Term	Definition
PM	Power Management
PRTC	Protected Real Time Clock
PSK	Pre-Shared Key
PSL	PCH Strap Length
RCS	Remote Connectivity Service
RCFG	Remote Configuration
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
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.
Sx	All S states which are different than S0
SW	Software
System States	Operating System power states such as S0, S1, S2, S3, S4, and S5.
TCP/IP	Transmission Control Protocol/Internet Protocol



Acronym / Term	Definition
TLS	Transport Layer Security
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
Windows* PE	Windows* Pre installation Environment
WIP	Work in Progress
WLAN	Wireless Local Area Network
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



1.2 Reference Documents

Document	Document No./Location
FW Bring Up Guide	Release kit
Firmware Variable Structures for Intel® Management Engine and Intel® Active Management Technology 11.0	ANACAPA document
PCH EDS	CDI
SkyLake PCH-LP SPI Programming Guide	Release kit
ISS Firmware Bring Up Guide	CDI

§



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, see the 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:

- SkyLake PCH (SPT) platform – All tools in the SkyLake PCH FW release kit are designed for 4th Generation Intel® Core™ Processor and SkyLake PCH platforms only. These tools do not work properly on any other legacy platforms (2nd or 3rd Generation Intel® Core™ Processors). Tools designed for other platforms also do not work properly on the 4th Generation Intel® Core™ Processor or SkyLake PCH platform.
- Intel® vPro™ platform – All features listed in this document are available for Intel® vPro™ platforms with Intel® ME FW 11.0. There are some features that are specifically designed for the Intel® vPro™ platform and only work on it.
- Intel® ME Firmware 11.0 SKU – A common set of tools are provided for the following Intel® ME FW 11.0 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.

2.2 Image Editing Tools

The following tools create and write flash images:

- Intel® FIT:

  Combines the Descriptor, GbE, BIOS, PDR, 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) 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 SPI image. (**Note:** The firmware update tool provided by Intel only works on the platforms that support FWUpdate feature.)

2.3 Manufacturing Line Validation Tool

The manufacturing line validation tool (Intel® MEManuf) allow the Intel® ME and Intel® AMT functionality to be tested immediately after the PCH chipset is generated. These tools are designed to be able to run quickly. They can run on simple operating systems, such as EFI, MS-DOS* 6.22, Windows* 98 DOS, FreeDOS, and DRMK DOS. The Windows* versions are written to run on Windows* 7, Windows* 8.1 and Win* PE 32 and 64. These tools are mostly run on the manufacturing line to do manufacturing testing.

2.4 Intel® ME 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 1: OS Support for Tools

Intel® ME and Manufacturing Tools	MS DOS*	Windows* 98 DOS	Free DOS	PC DOS Version 7.01	UEFI (64 bit)	Windows* PE 32 / 64 (version 3, 4 and 5.x)	Windows* 7 32/64	Windows* Server 2008 R2/2012 32/64 With the latest SP	Windows* 8.1 32/64 (MBR & uEFI)	Linux Redhat RHEL7 (RPM/Debian)	Windows 10 DT	OSX (Lion and Mountain Lion)
Intel® Flash Image Tool							X	x	x		x	x
Intel® Flash Programming Tool	x	x	x	x	x	x	X	x	x	x	x	
Intel® MEManuf Tool	x	x	x	x	x	x	X	x	x	x	x	
Intel® ME Info Tool	x	x	x	x	x	x	X	x	x	x	x	
Intel® Firmware Update Tool	x	x	x	x	x	x	X	x	x	X	x	
ICC CCT Tool	x	x	x	x	x		x	x	x			

NOTES:

1. 64 bit support does 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 Windows Server

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.

See the description of each tool for its exact requirements.



Table 2: Tools Summary

Tool Name	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	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. (See 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 the Double-Quote Character (")

The EFI version of the tools handle multi-word argument is different 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, 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.

§



3 Intel® Flash Image Tool

The Flash Image Tool (**FIT.exe**) creates and configures a complete SPI image file for SkyLake PCH-LP 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 SPI flash image:

   Binary files of the following regions:

   BIOS

   Intel integrated LAN (GbE)

   Intel® ME

   Platform Descriptor Region

   The Flash Descriptor Region created by FIT
3. The user can manipulate the completed SPI image via a GUI and change the various chipset parameters to match the target hardware. Various configurations can be saved to independent files, so 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 SPI image file; it does not program the flash device. This complete SPI 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* 7, and Windows* 8.1. The tool does not have to run on an Intel® ME-enabled system.

3.2 Flash Image Details

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

Figure 1: SPI Flash Image Regions

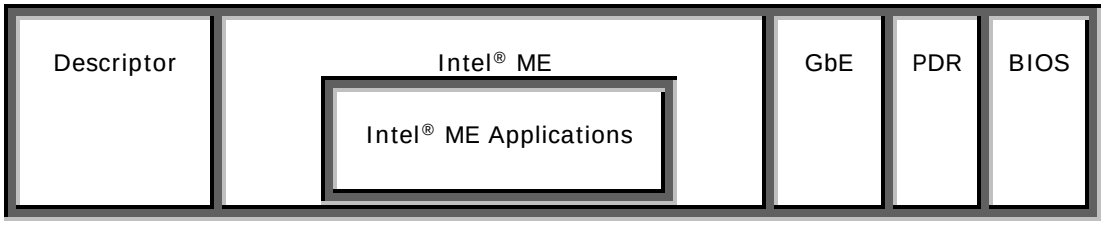


Table 3: Flash Image Regions – Description

Region	Description
Descriptor	This region contains information such as the space allocated for each region of the flash image, read-write permissions for each region, and a space which can be used for vendor-specific data. It takes up a fixed amount of space at the beginning of the flash memory. **Note:** This region MUST be locked before the serial flash device is shipped to end users. Please see 0 below for more information. Failure to lock the Descriptor Region leaves the Intel® ME device vulnerable to security attacks.
Intel® ME	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.
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® FIT

See the following for further information:

- General configuration information – See the FW Bring Up Guide from the appropriate Intel® ME FW kit.
- Detailed information on how to configure PCH Soft Straps and VSCC information – See the SkyLake PCH SPI Programming Guide.

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 a New Configuration

FIT provides a XML configuration file template that will help the user can use to 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 an Existing Configuration

To open an existing configuration file:

1. Choose File **> Open**; the **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 a 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.

4. Select the path and enter the file name for the configuration.
5. 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:

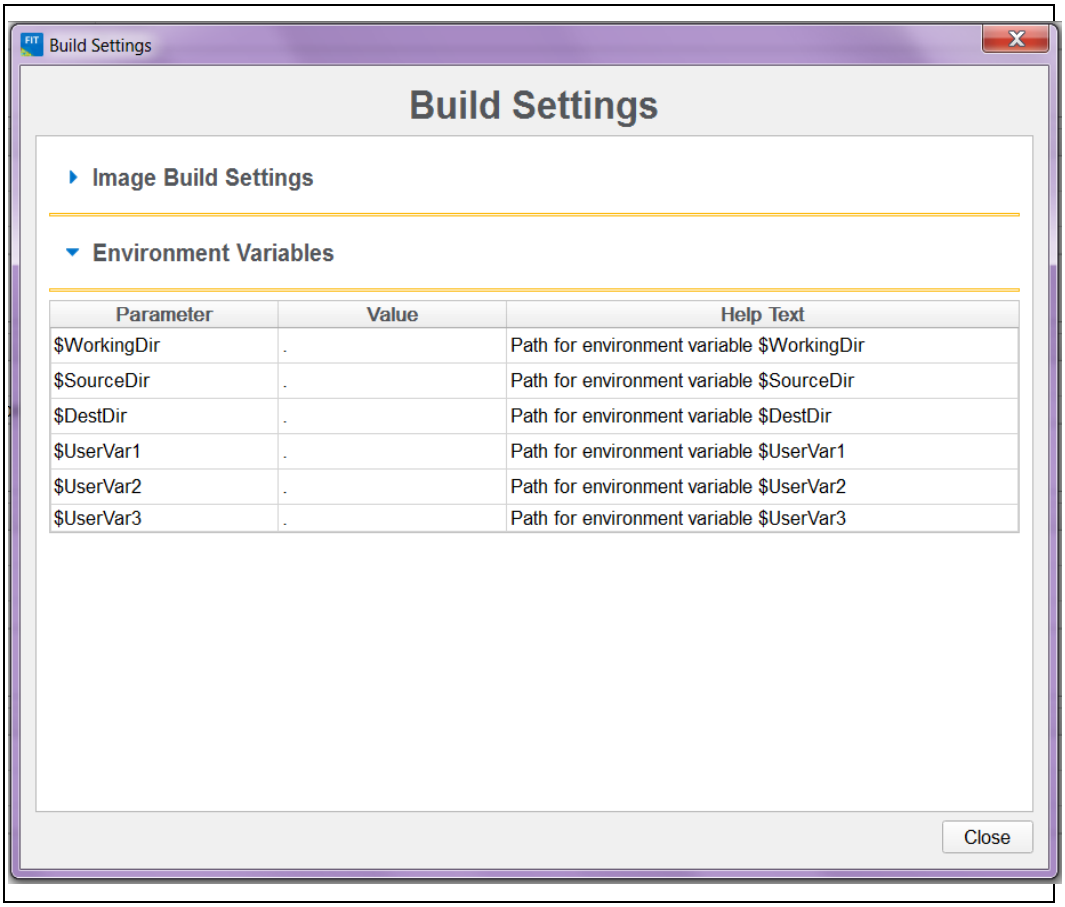
   $WorkingDir – the directory functions as a basic path variable when modified in the GUI. If $WorkingDir CLI flag is used when launching FIT GUI, then the fit.log will be created in $WorkingDir directory.

   $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 2. Environment Variables Dialog



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

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

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 Table 4.

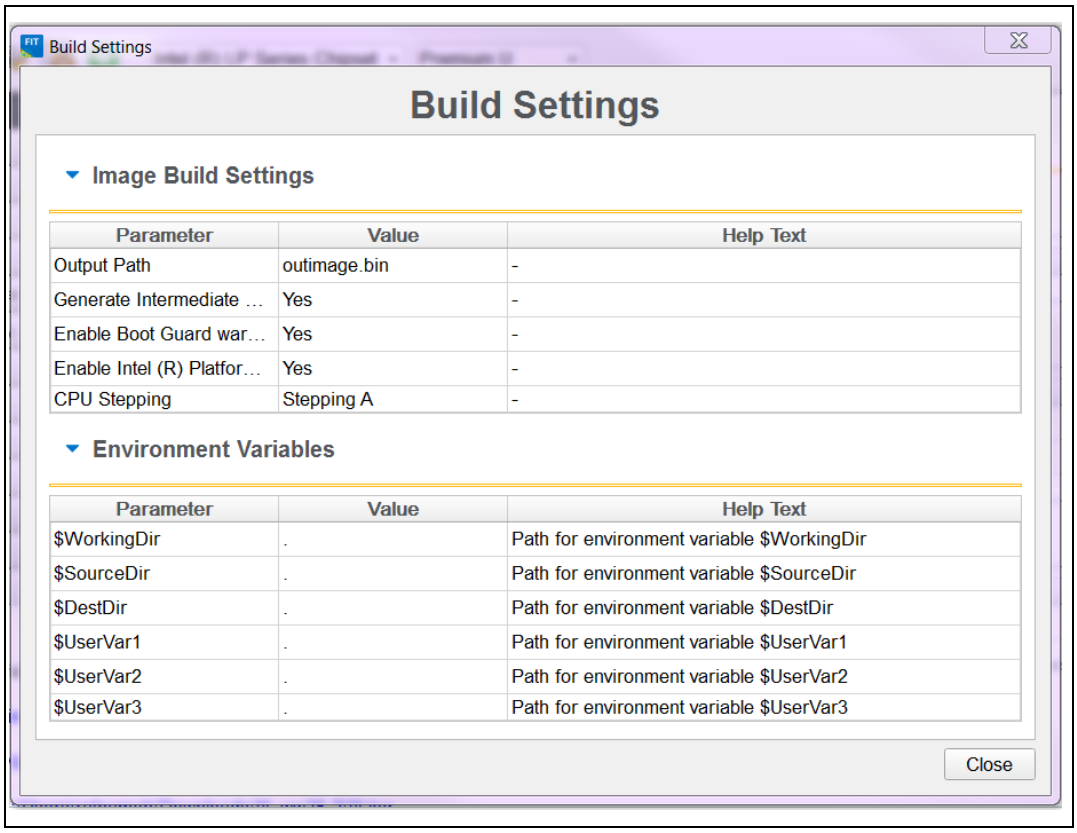
To modify the build setting:

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 4: Build Settings Dialog Options

Option	Description
Output path	The path and filename where the final image should be saved after it is built. (Note: Using the $DestDir environment variable makes the configuration more portable.)
Generate intermediate build files	Causes the application to generate separate (intermediate) binary files for each region, in addition to the final image file (see Figure 3). 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. Build Settings Dialog



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.

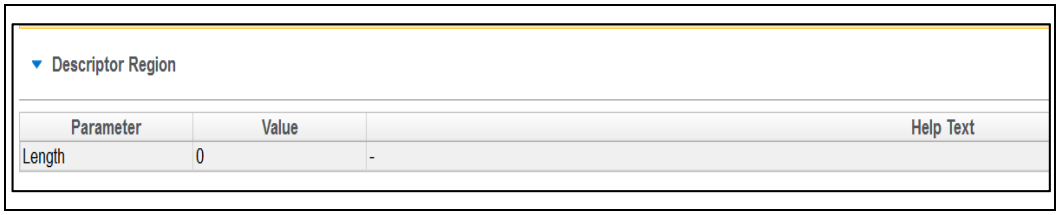
3.4.7 Descriptor Region Length

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 4. Descriptor Region Length Parameter

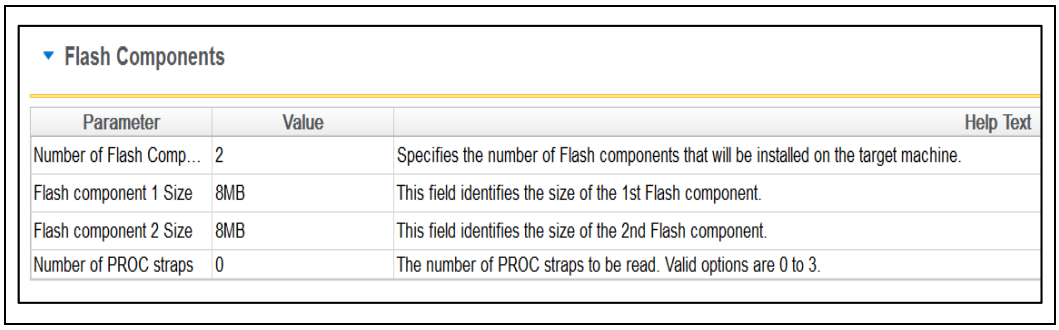


3.4.8 Setting the Number and Size of the Flash Components

To set the number of flash components:

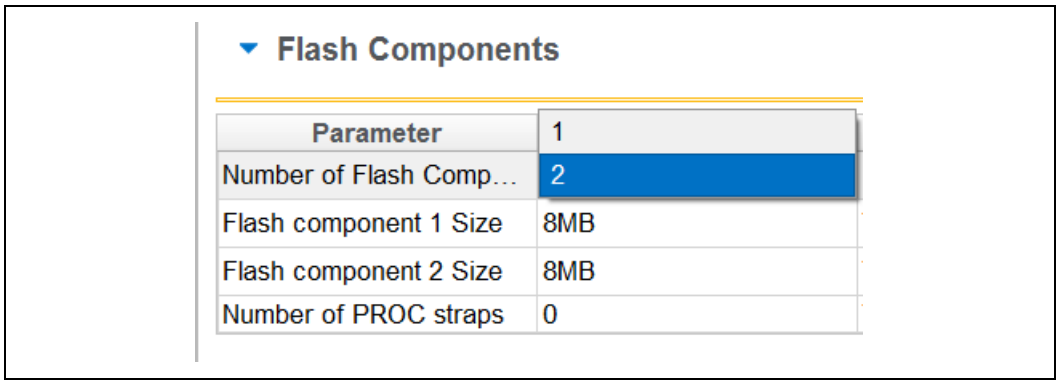
1. Select **Flash Settings** in the left pane; expand the Flash Component node in the right pane.
2. See Figure ; all the parameters in the Flash Component section are listed in the right pane.

Figure 5: Flash Settings > Flash Components



3. Double-click **Number of Flash Components** in the right pane (see Figure)
4. Enter the number of flash components (valid values are 1 or 2).

Figure 6: Flash Components Dialog



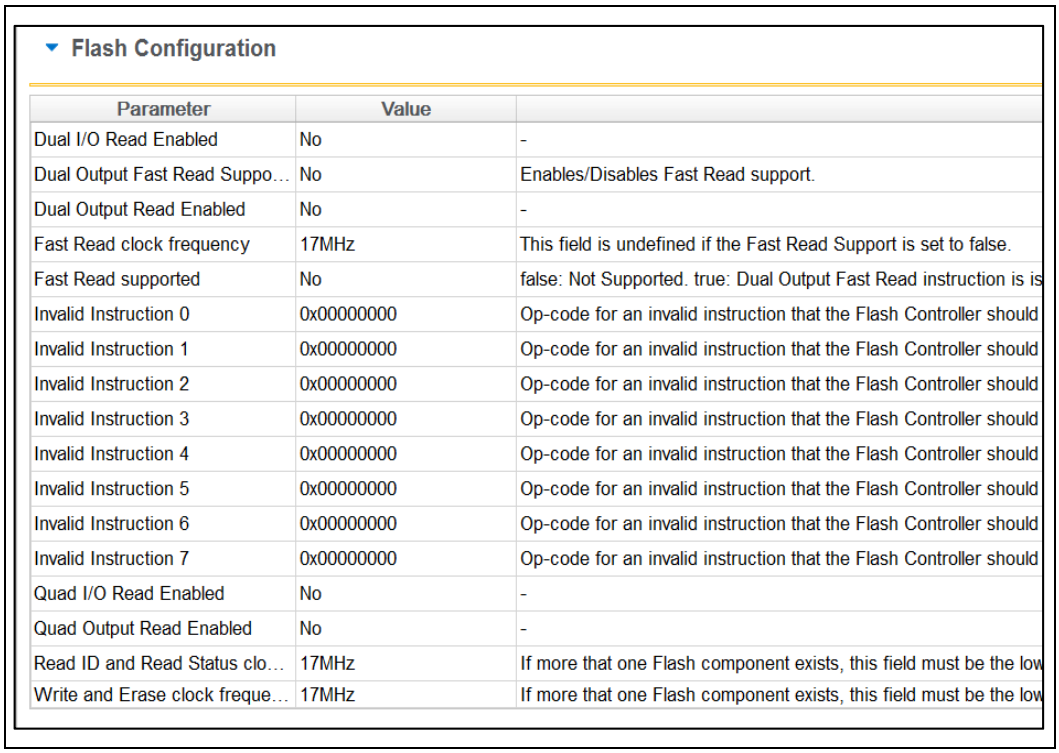


To set the size of each flash component:

1. Double-click on 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.

Figure 7: 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

3.4.9 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 5: 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 6: 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** in the right pane; the access parameters are listed in the right pane (see Figure).
2. Double-click on each parameter and set its access value in one of the following ways:

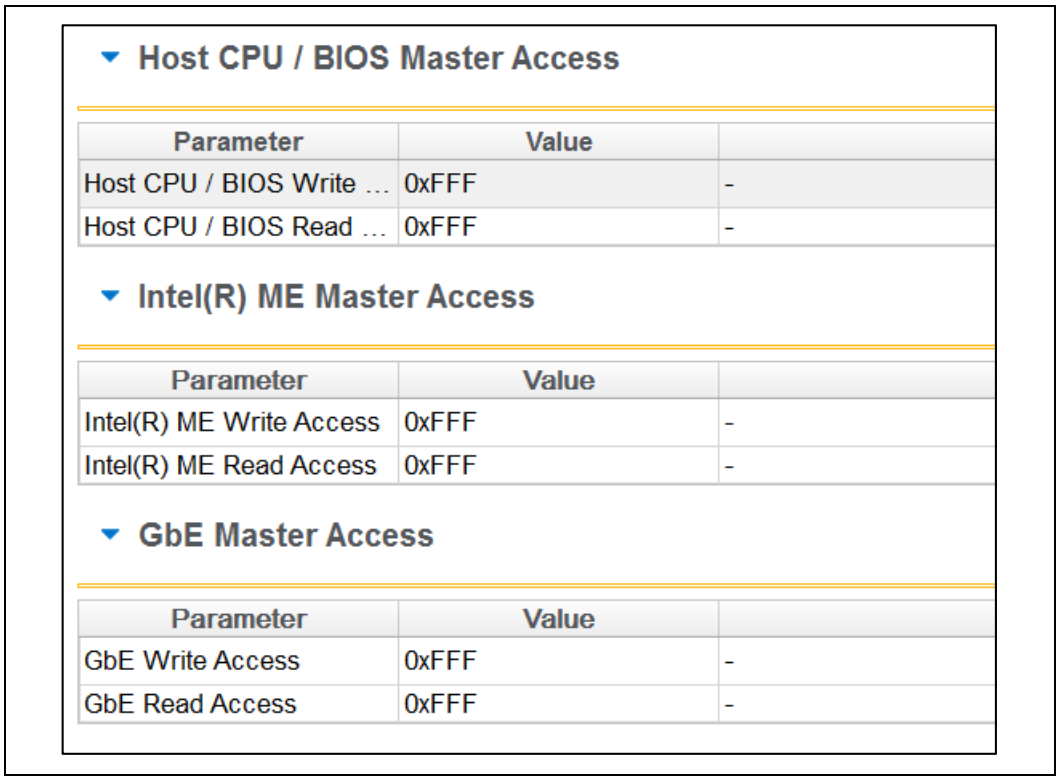
   To generate an image for debug purposes or to leave the SPI region open: select 0xFF for both read and write access in all three sections.

   To 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 setting for production (e.g., select 0x0C for Intel® ME read access and 0x0D for Intel® ME write access).

Figure 8: Descriptor Region > Master Access Section



3.4.10 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, see the SkyLake PCH-LP SPI Programming Guide, Section 6.4.)

VSCC Table can be accessed:

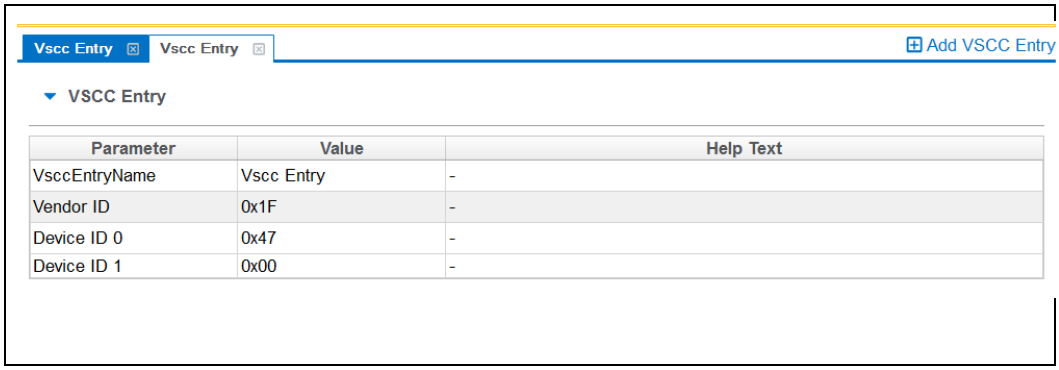
1. Select Flash Settings Tab on the left pan
2. Expand VSCC Entries on the right pan as shown in Figure 9 below:

3.4.11 Adding a New Table

To add a new table:

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

Figure 9: Add VSCC Table Entry Dialog



2. 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.)
3. User can enter into the values for the flash device. (See **Error! Reference source not found.** 11, which shows the parameters of a new VSCC table.)

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

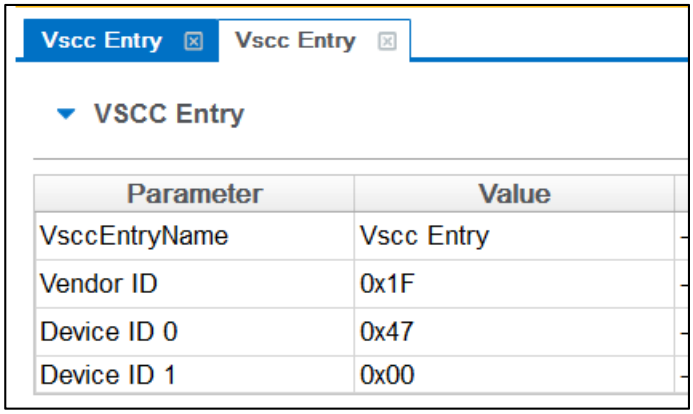
NOTES: If the descriptor region is being built manually the user will need to reference the VSCC table information for the parts being supported from the manufacturers' serial flash data sheet. The SkyLake PCH-LP SPI Programming Guide should be used to calculate the VSSC values.

3.4.12 Removing an 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 as shown in Figure 12.

Figure 10: Deleting VSCC Table Entry Dialog





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

3.4.13 Modifying the Intel® ME 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.

3.4.14 Setting the Intel® ME Region Binary File

To select the Intel® ME region binary file:

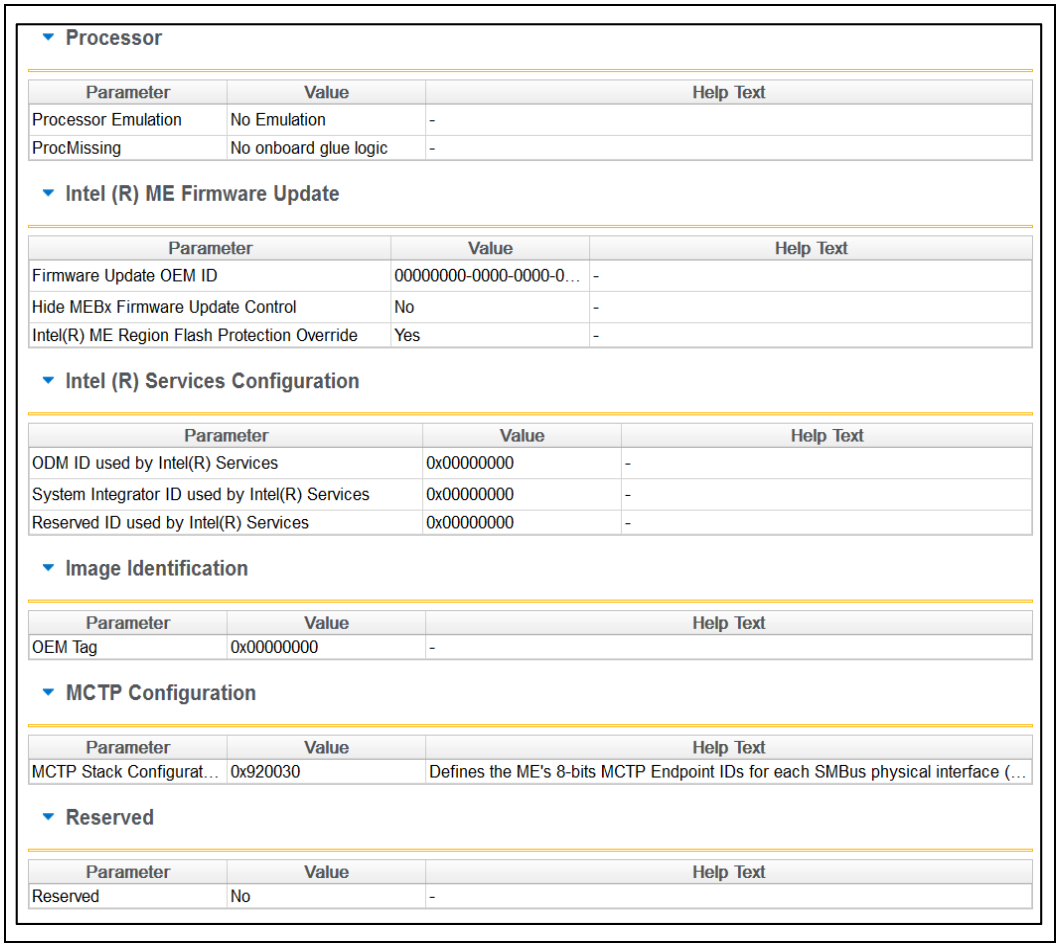
1. Select the Intel® ME Region available under Flash Layout tab on the left pane.
2. Double-click on the **Binary file parameter** in the list; select the Intel® ME file to be used.
3. Click **0K** to update the parameter; when the flash image is built, the contents of this file is copied into the Intel® ME Region.

3.4.15 Intel® ME Section

This section describes Intel® ME FW Kernel parameters. (See the FW Bringup guide for general information and see Appendix for more details.)

Click on Intel® ME Kernel Tab on the left pane to configure Intel® ME parameters. The parameter values can be found in the Help Text next to the parameter value as shown in Figure .

Figure 11: Intel® ME Kernel

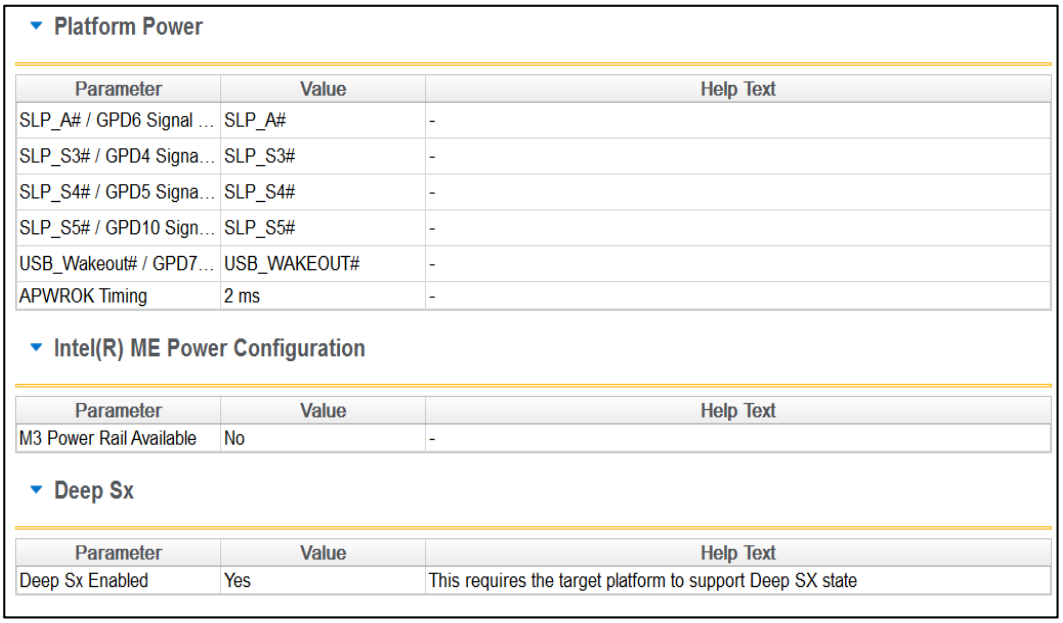


3.4.16 Power

This section describes the platform power configuration settings.

Click on the Power tab on the left pane to configure power parameters. (See Figure 14)

Figure 12: 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

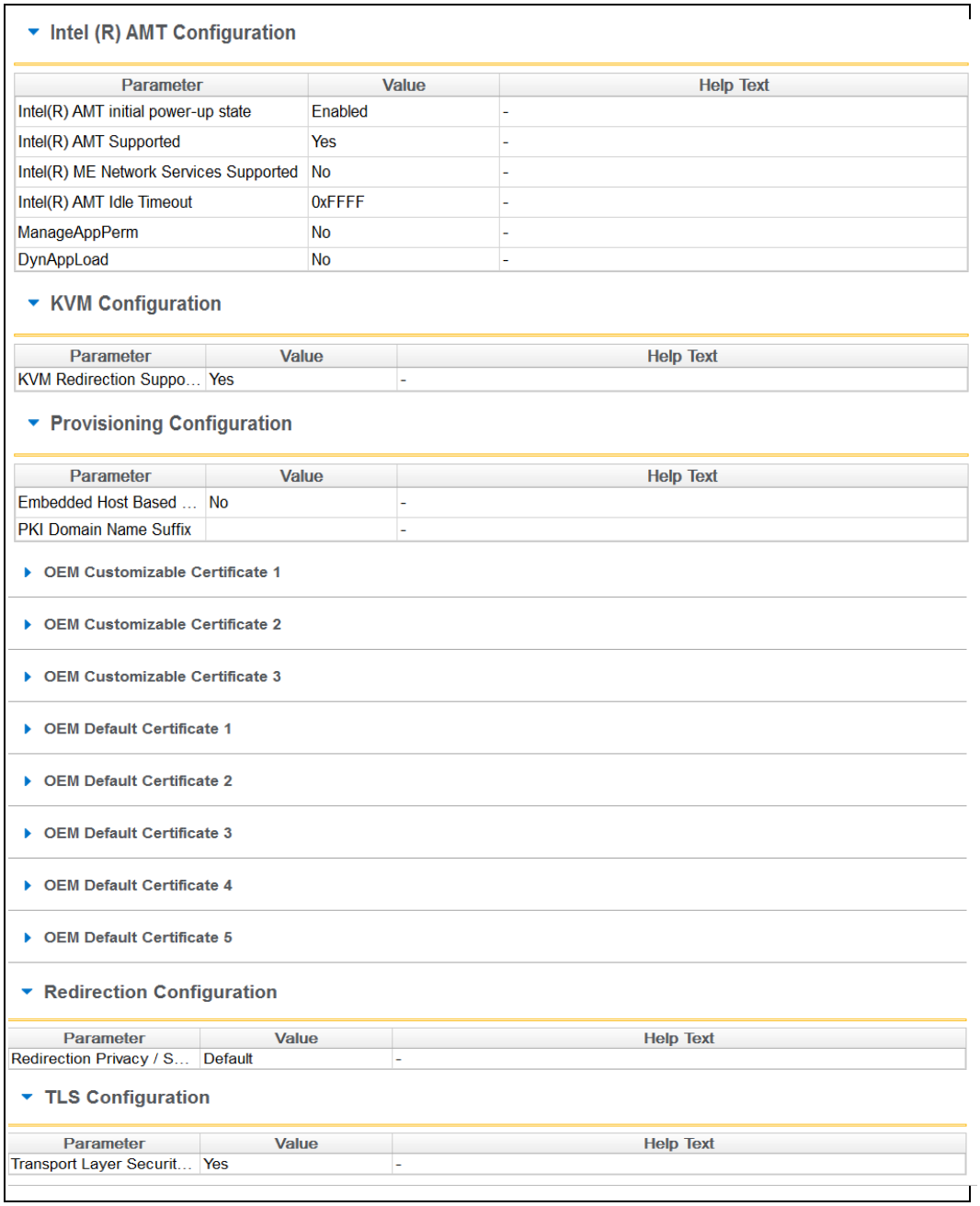
3.4.17 Manageability Application Section

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

This section describes the Manageability Application parameters. (See the 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 on Intel® AMT Tab on the left tab to configure Intel® AMT parameters.

Figure 13: 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	-

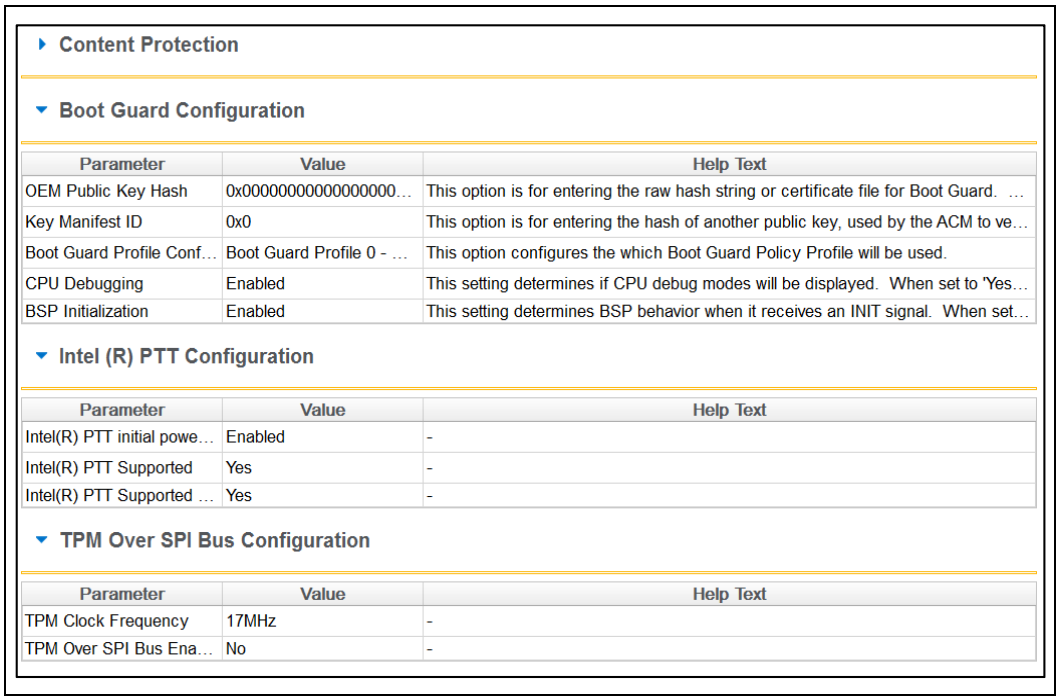
TLS Configuration

Parameter	Value	Help Text
Transport Layer Securit…	Yes	-

3.4.18 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. (See the FW Bringup guide for general information and see Appendix E for more details.)

Figure 14: Platform Protection Section



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 choose and the FW SKU they load on that platform.

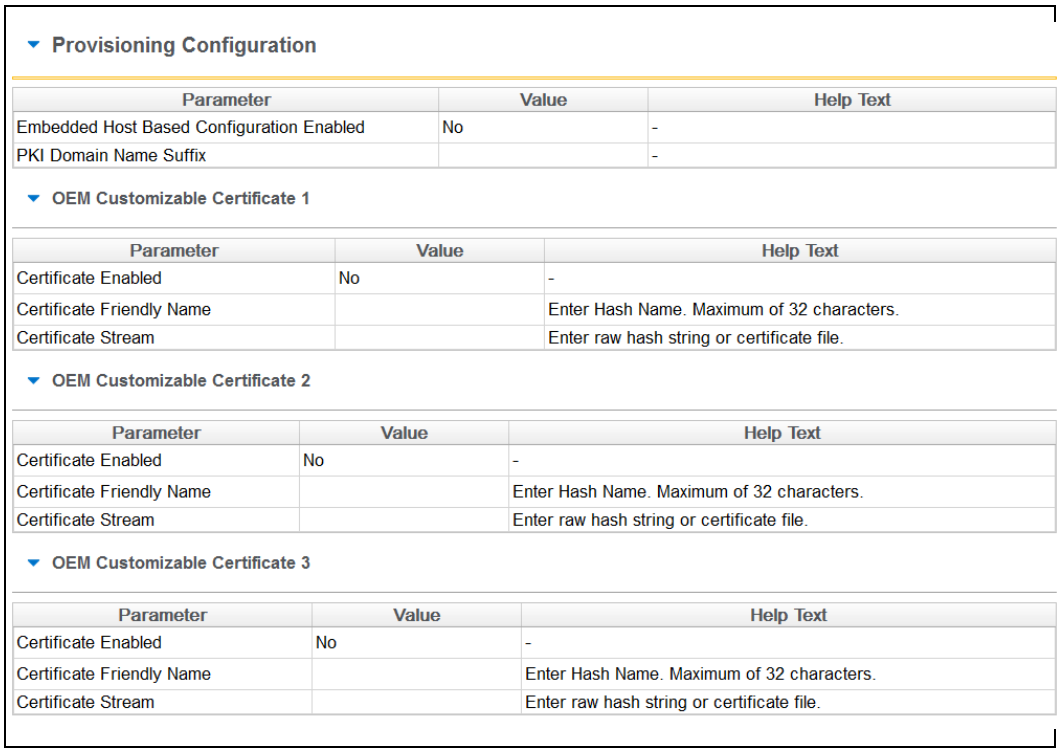
- Intel® Platform Trusted Technology
- Intel® Content Protection

3.4.19 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 and see Appendix E for more details.

Click on Intel® AMT tab on the left pane to specify the OEM settings.

Figure 15: Provisioning Configuration Section



▼ Provisioning Configuration

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

▼ OEM Customizable Certificate 1

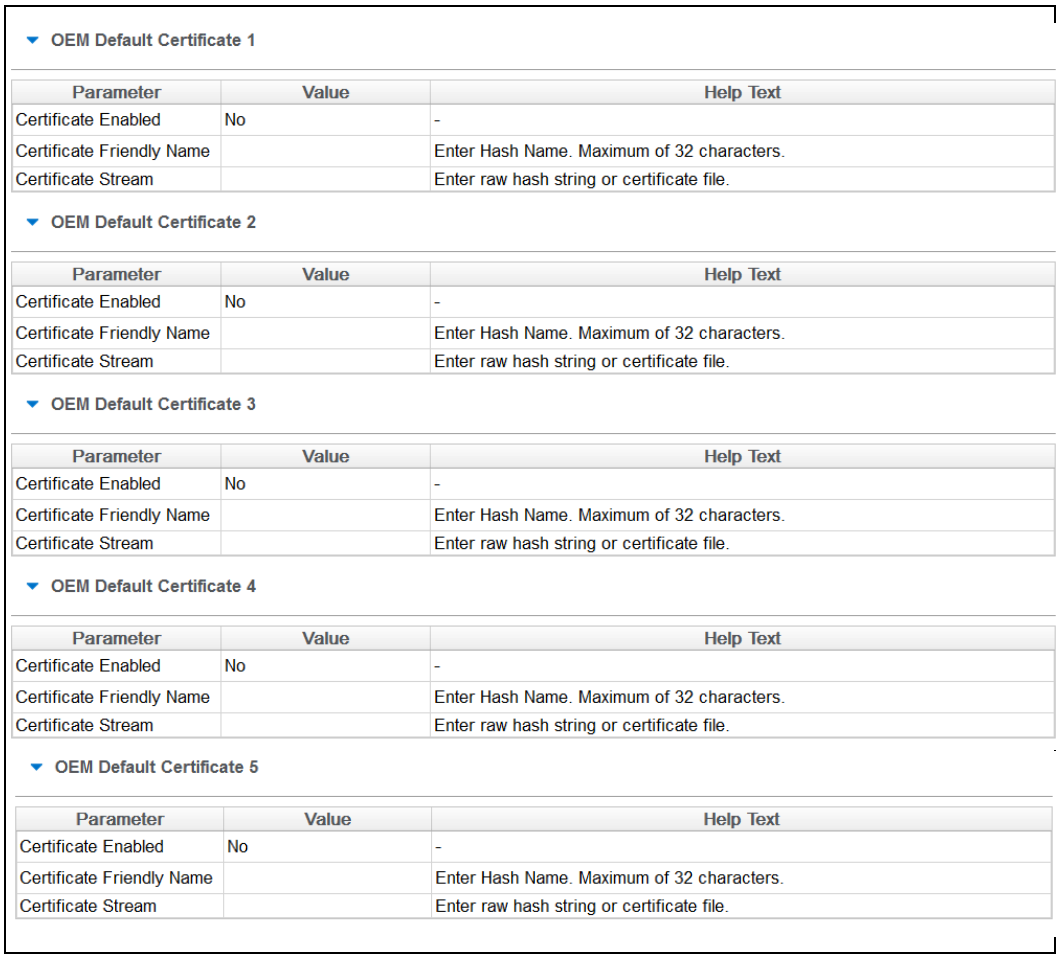
Parameter	Value	Help Text
Certificate Enabled	No	-
Certificate Friendly Name		Enter Hash Name. Maximum of 32 characters.
Certificate Stream		Enter raw hash string or certificate file.

▼ 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.

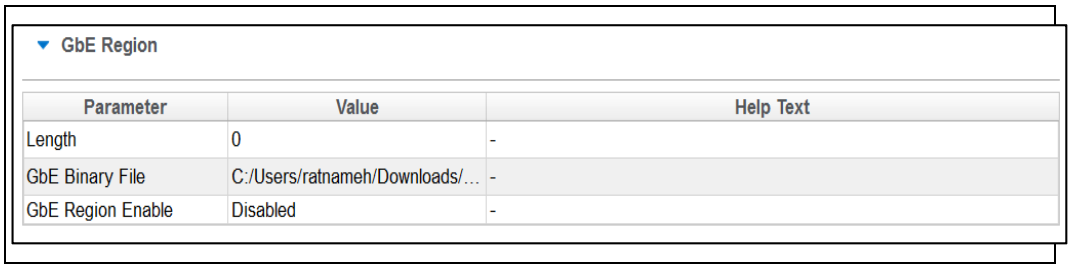
▼ OEM Default Certificate 5

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.

3.4.20 Gbe (LAN) Region Settings

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

Figure 16: GbE Region Options



3.4.21 Setting the Gbe Region Length Option

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

3.4.22 Setting the 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.

3.4.23 Enabling / Disabling the 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.
2. Choose **Disable Region** from the drop down. When the flash image is built it will not contain a GbE Region.

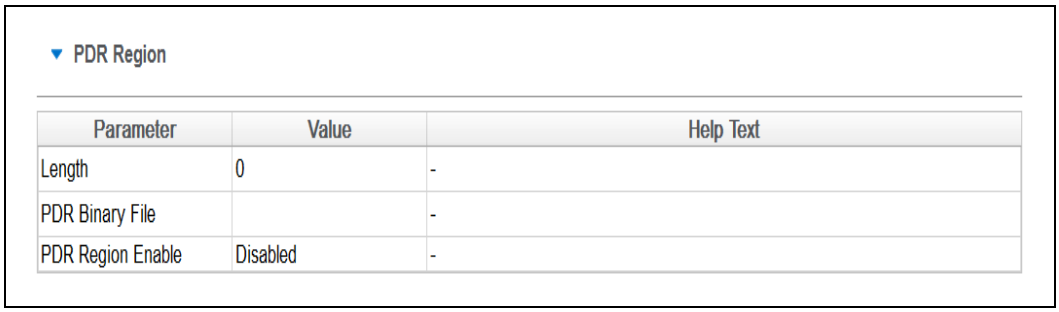
To enable the GbE Region:

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.

3.4.24 Modifying the PDR Region

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

Figure 4: PDR Region Options



3.4.25 Setting the PDR Region Length Option

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

3.4.26 Setting the 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.

3.4.27 Enabling / Disabling the PDR Region

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

To disable the PDR Region:

1. Click on 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.

NOTES: This region is disabled by default.

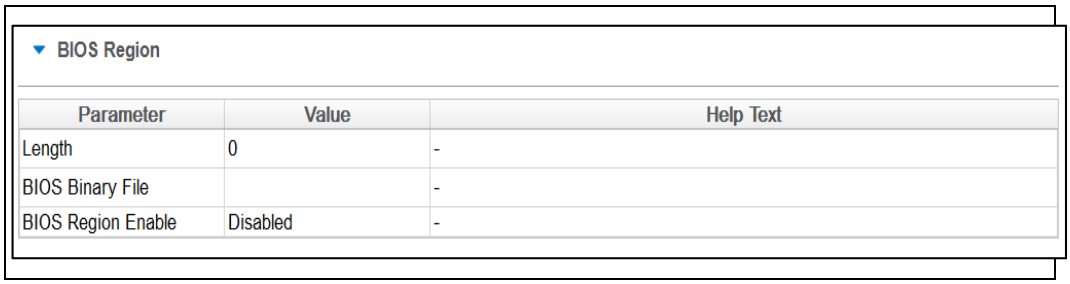
To enable the PDR Region:

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.

3.4.28 Modifying the 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 5: BIOS Region Parameters



3.4.29 Setting the BIOS Region Length Parameter

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

3.4.30 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.

3.4.31 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 PDR 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.

3.4.32 Building a 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 (see Section Note:).
- 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.

3.4.33 Decomposing an 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 (see 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.

3.4.34 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 7: 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.

3.4.35 Example – Decomposing an 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 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.

3.4.36 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\ME9.0_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

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.

NOTES: 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 MS DOS 6.22, DRMKDOS, and 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* 7 64/32 bit and Windows* 8.1 64/32 bit.

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 an operating system to run on. It is designed to deliver a custom image to a computer that is already able to boot and is not a means to get a blank system up and running. 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

A flash image is composed of up to five 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 19: Flash Image Regions

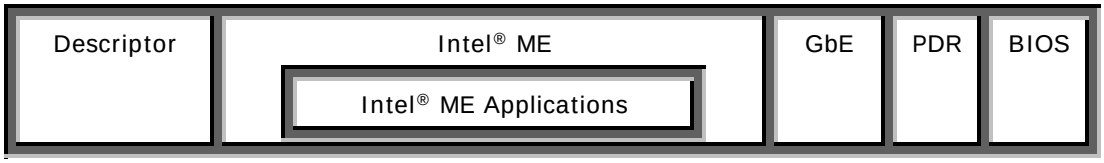


Table 8: Flash Image Regions – Description

Component	Description
Descriptor	Region that takes up a fixed amount of space at the beginning of the flash memory. Contains information such as: Space allocated for each region of the flash image. Read/write permissions for each region. A space that can be used for vendor-specific data.
Intel® ME	Contains code and configuration data for Intel® ME applications, such as Intel® AMT technology.
GbE	Contains code and configuration data for GbE.
BIOS	Contains code and configuration data for the entire platform.
PDR	Region that allows system manufacturers to define custom features for the platform.

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 9: 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

NOTES: 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 the Flash Device

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

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 see the PCH Intel® ME BIOS writer's guide).

NOTES: 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.

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

Table 10: 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 the NVAR name
fpt.exe –IN <Text file>	Accepts cfggen file with values set and will use setfile to update

See Appendix A for a description of all the NVAR parameters.



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 -? option.

The commands in 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]
[-SAVEMAC] [SAVESXID] [-B] [-E][REWRITE] [-ADDRESS|A] [-LENGTH|L]
[-CVARS] [-CFGGEN] [-U] [-O] [-IN] [-N][-V] [-LOCK]
[-PSKFILE] [-CLOSEMNF] [-GRESET] [-PAGE] [-SPIBAR] [-R] [-VARS]
[-COMMIT][-HASHED] [-DISABLEME] [-READFPF] [-READFPFATTRIB]
[-COMPAREFPF] [-FPFS] [-COMMITFPFS] [-RPMCBIND]
```

Table 11: 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.
-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.
-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.

Option	Description
-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.
-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.
-GBE: the de	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.
-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.
-U:	Update. Updates the NVARs in the flash. 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.
-O <file>	Output File. The file used by FPT to output NVAR information.



Option	Description
-IN <file>	Input File. The file used by FPT for NVAR input. This option flag must be followed by a text file (i.e., fpt –u –in FPT.cfg). The tool updates the NVARs contained in the text file with the values provided in the input file. User can also use FPT –cfggen to generate this file.
-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.
-PSKFILE <file>	PID/PPS/Password pair files. Specifies the input file that contains the one or more PID/PPS/Password key value pairs. This option is used to update the PID, PPS, and Password NVARs whose values are read from the input file. This option only support version 1 FiletypeHeader UUID
-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 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 & 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.

Option	Description
-GRESET <NO> :	Global Reset. FPT performs a global reset. On mobile platforms this includes driving GPIO30 low. Mobile platforms require a SUS Well power-down acknowledge-driven low before the global reset occurs or the platform may not boot up from the reset. The "NO" afterwards disables the driving of GPIO30 for mobile SKUs.
-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.
-CFGGEN	NVAR Input file generation option. This creates a file which can be used to update the line configurable NVARS.
-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.
-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.
-COMMITFPF	Commits NVAR values to FPF via firmware and prevents further modification of FPFs
-PAGE	Pauses the screen when a page of text has been reached. Hit any key to continue.
-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	This option will allow the tool operator to temporarily disable the Intel® Management Engine until the next Global Reset or G3.
-READFPF<name>	Displays programmed FPF values.
-READFPFATTRIB<name>	Displays the Attributes of the FPF values.
-COMPAREFPF<name>	Compare the FPF with a value passed in by the user.
-FPFS	Displays a list of the FPFs
-COMMITFPFS<name>	Commit the FPFs permanently into the MCP.
-RPMCBIND	Binds the Real-time Monotonic Counters in the SPI part to the MCP.
-REWRITE	Allows to rewrite the SPI with file data even if flash is identical.



Table 12: 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 13: 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

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

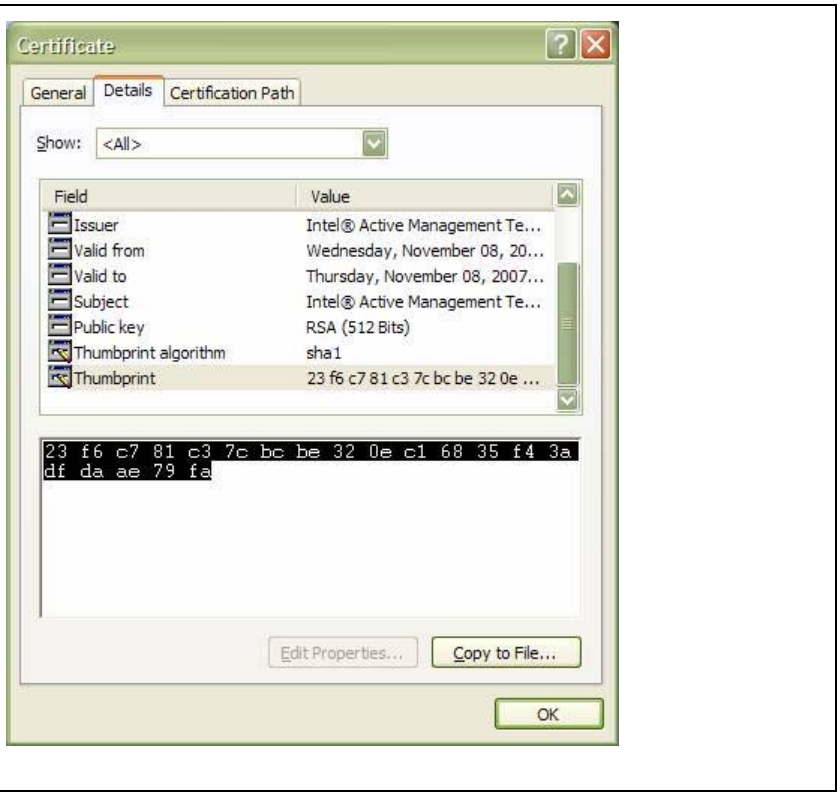
There are 3 OEM Customizable certificate hash values that can be stored in the Intel® ME region:

- 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:

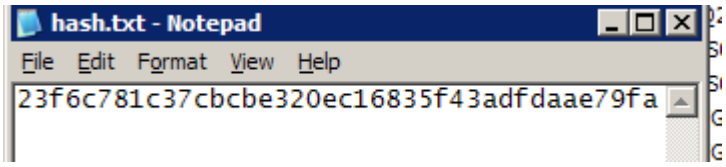
1. Copy the raw hash values from a valid certificate file.

Figure 6: Raw Hash Values from Certificate File



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

Figure 21: Sample Hash.txt File



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   = 23f6c781c37cbce320ec16835f43adfdaae79fa
```

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

NOTES: 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 SkyLake PCH-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 Image, following BIOS settings need to be done manually otherwise Error might 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 -> CPU CONFIGURATION -> FLASH WEAR OUT PROTECTION : Disabled

```
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 a 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 a 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:  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)
    --- Flash Image Information --
    Signature: VALID
    Signature: VALID
        Number of Flash Components: 1
            Component 1 - 8192KB (65536Kb)
        Regions:
            Descriptor - Base: 0x000000, Limit: 0x000FFF
            BIOS       - Base: 0x580000, Limit: 0x7FFFFF
            ME         - Base: 0x003000, Limit: 0x57FFFF
            GbE        - Base: 0x001000, Limit: 0x002FFF
            PDR        - Not present
        Master Region Access:
            CPU/BIOS - ID: 0x0000, Read: 0xFF, Write: 0xFF
            ME       - ID: 0x0000, Read: 0xFF, Write: 0xFF
            GbE      - ID: 0x0118, Read: 0xFF, Write: 0xFF
Total Accessible SPI Memory: 8192KB, Total Installed SPI Memory: 16384KB
FPT Operation Passed
```

This command displays information about the flash devices present in the computer. The base address refers to the start location of that region and the limit address refers to the end of the region. If the flash device is not specified in **fparts.txt**, 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
```

Please note that 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 CVAR 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, CVARs, that are displayed with values
;   indicates that the CVAR has already been given a value,
;   but has not yet been committed.  Entries without values
;   indicates that the CVAR has not been written, at least
;   since the system reset or use of the '-commit' command.
;
MEBxPassword =

;   OEMSkuRule:  Entering a value for the complete 32-bit CVAR entry
;   below and bit-wise entries are mutually exclusive.  Entering a value
for
;   the complete CVAR will cause the program to ignore any bit-wise
entries.
;
;   Valid entries for the bit-wise values are "enable", "disable",
;   "NoChange", or no value at all (i.e. blank). The values are not case
;   sensitive. Invalid bit-wise values will cause FPT to display a
warning
;   and ignore the bit-wise entry being updated.
;
OEMSkuRule =
    Enable Intel (R) Standard Manageability; Disable Intel (R) AMT =
    Manageability Application =
    Intel (R) Anti-Theft Technology =
```



```
    PAVP                  =
    Intel (R) ME Network Service =
    KVM                   =
    TLS                   =
    Service Advertisement & Discovery =
    Near Field Communication Enabled =


;   FeatureShipState:  Entering a value for the complete 32-bit CVAR
entry
;   below and bit-wise entries are mutually exclusive.  Entering a value
for
;   the complete CVAR will cause the program to ignore any bit-wise
entries.
;
;   Valid entries for the bit-wise values are "enable", "disable",
;   "NoChange", or no value at all (i.e. blank). The values are not case
;   sensitive. Invalid bit-wise values will cause FPT to display a
warning
;   and ignore the bit-wise entry being updated.
;
FeatureShipState =
    Manageability Application =


SetWLANPowerWell =

OEM_TAG =

FtpmEnable =

PID =

PPS =

;  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      =
OEMCustomCert1 FriendlyName  =
OEMCustomCert1 RawHashFile   =

;  OEMCustomCert2 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   =

;  OEMCustomCert3 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   =

USBrSettings =

Privacy/SecurityLevel =

EhbcState =

ODM_ID =

SystemIntegratorId =

ReservedId =

ATFPOPHard =

ATFPOPSoft =
```

§



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* 7 64/32 bit and Windows* 8.1 64/32 bit.

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.

VSCCCOMMN.bin is required to verify the VSCC entry on the platform. This file must be in same folder as the MEManuf executable or MEManuf reports an error.

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]
        [-NONFC] [-NFC] [-NORPMC] [-RPMC] [-NOISH] [-ISH][-ISH? or h]
         [-ISH test<Level>]
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 14: Options for the Tool

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 doesn't 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 isn't 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. **Note:** Full BIST will not run if the Mobile platform is on DC power only.
-EXP	Shows examples of how to use the tools.
-H or -?	Displays the help screen.
-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.



Option	Description
-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
-NORPMC	Disables RPMC related test cases
-RPMC	Enables RPMC related test cases

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 ME11 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). Intel® MEManuf –EOL Check.) When –f flag is used along with a file name, the tool will load the file as the configuration file, instead of using MEManuf.cfg.
-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.cfg before using it. It is highly recommended to use this option to generate a new MEManuf.cfg with an up-to-date variable names list before using the Intel® MEManuf End-Of-Line check feature.
-VERBOSE <file>	Displays the debug information of the tool or stores it in a log file.
–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



Option	Description
-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
-NONFC	This option will skip NFC test
-NFC	This option will force NFC test. NFC BIST consists of two tests: 1. HW connectivity between ME and the NFC module 2. RF test of the module
-NOISH	This option will skip ISH tests
-ISH	This option will force ISH tests
-ISH -? Or -h	Shows ISH Library help usage
-ISH -exp	Shows command line example
-ISH –test<Level>	Execute self-test. Level 0: configuration test – checks that all algorithms have required reporters. Level 1: - Basic connectivity test to each sensor Level 2: Check that calibration data exists for selected sensors (the driver will decide if calibration is mandatory or not) Level 3: Sensor BIST for each sensor that supports BIST.

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

Table 15: 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

NOTES: ICC data check is performed for all options.

NOTES: The Full BIST test for ME11 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. Please refer that section.

5.4.1 MEManuf.cfg File

The MEManuf**.cfg** 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**.cfg** 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.

Here is an example of the new xml configuration file:

```
<?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, ishbist, ishchub). -->
     <!-- 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="VDM - General : VDM engine">
```

```
    <!-- The commented fields bellow 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 bellow CANNOT be edited. Any edits will
be ignored by the tool -->
    <!-- Description>Test KVM sampling 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="SMBus - SMBus : Read byte">
    <!-- The commented fields bellow 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="Policy Kernel - ME Password : Validate MEBx
password">
    <!-- The commented fields bellow 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 bellow 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 : M3 Power Rails
Available">
    <!-- The commented fields bellow CANNOT be edited. Any edits will
be ignored by the tool -->
    <!-- Description>Only on mobile or desktop.  Test fails if M3
power well rule is not set to
MEFWCAPS_M3_PWR_RAILS_AVAILABL.</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 - ME Configuration : PROC_MISSING">
    <!-- The commented fields bellow 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></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 Configuration : Wlan Power Well">
    <!-- The commented fields bellow CANNOT be edited. Any edits will
be ignored by the tool -->
    <!-- Description>WLAN power well setting.</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 - Power Package : Live Heap Test">
    <!-- The commented fields bellow 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="Policy Kernel - Embedded Controller : Power source
type">
    <!-- The commented fields bellow 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></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 bellow 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 bellow 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 bellow CANNOT be edited. Any edits will
be ignored by the tool -->
    <!-- Description>LAN test runs only if AMT is not permenantely
disabled and we are not in small business mode or 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 bellow CANNOT be edited. Any edits will
be ignored by the tool -->
    <!-- Description>LAN test runs only if AMT is not permenantely
disabled and we are not in small business mode or 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 - LAN : Connectivity to NIC in M3">
    <!-- The commented fields bellow CANNOT be edited. Any edits will
be ignored by the tool -->
    <!-- Description>LAN test runs only if AMT is not permenantely
disabled and we are not in small business mode or 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 bellow CANNOT be edited. Any edits will
be ignored by the tool -->
    <!-- Description>LAN test runs only if AMT is not permenantely
disabled and we are not in small business mode or 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 bellow CANNOT be edited. Any edits will
be ignored by the tool -->
    <!-- Description>Check while both EHBC and privacy level are
available, (PrivLevel != PRIVACY_LEVEL_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 bellow CANNOT be edited. Any edits will
be ignored by the tool -->
    <!-- Description>Check if EHBC state is avaliable.</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 bellow 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="Common Services - General : Valid FOV number %d">
    <!-- The commented fields bellow CANNOT be edited. Any edits will
be ignored by the tool -->
    <!-- Description>Checks if there were any issues when FOV's were
copied into system.</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 - Power : M3 power rail supported">
    <!-- The commented fields bellow CANNOT be edited. Any edits will
be ignored by the tool -->
    <!-- Description>Run the tests verifying the internal
variables.</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 - Power : Valid LAN power well">
    <!-- The commented fields bellow 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="ISH  service - ISH Service Tests : IUP Presence">
    <!-- The commented fields bellow CANNOT be edited. Any edits will
be ignored by the tool -->
    <!-- Description>Test ISH services</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="GuC Encryption Key FPF">
    <!-- The commented fields bellow CANNOT be edited. Any edits will
be ignored by the tool -->
    <!-- Description>Check GuC Encryption Key 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"
example="04ABF345031DEFA2B7E898791045ABDEF23549A00135782937ABDEEFFA10
EF33"> </RequiredValue>
</eolconfig>
<eolconfig name="BSMM SVN FPF">
    <!-- The commented fields bellow CANNOT be edited. Any edits will
be ignored by the tool -->
    <!-- Description>Check fpf bsmm svn 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="Not set/2 digit hex number with 0x prefix"
example="0xB4"> </RequiredValue>
</eolconfig>
<eolconfig name="KM SVN FPF">
    <!-- The commented fields bellow CANNOT be edited. Any edits will
be ignored by the tool -->
    <!-- Description>Check fpf km svn 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="Not set/2 digit hex number with 0x prefix"
example="Not set"> </RequiredValue>
</eolconfig>
<eolconfig name="ACM SVN FPF">
    <!-- The commented fields bellow CANNOT be edited. Any edits will
be ignored by the tool -->
    <!-- Description>Check fpf acm 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="Not set/2 digit hex number with 0x prefix"
example="0xB4"> </RequiredValue>
```

```
</eolconfig>
<eolconfig name="Enforcement Policy FPF">
    <!-- The commented fields bellow 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>Disabled</State>
    <ErrAction>ErrorContinue</ErrAction>
    <RequiredValue format="2 digit hex number with 0x prefix"
example="0x4A"> </RequiredValue>
</eolconfig>
<eolconfig name="BSP Initialization FPF">
    <!-- The commented fields bellow CANNOT be edited. Any edits will
be ignored by the tool -->
    <!-- Description>Check fpf bsp 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/Disabled" example="Disabled">
</RequiredValue>
</eolconfig>
<eolconfig name="CPU Debugging FPF">
    <!-- The commented fields bellow CANNOT be edited. Any edits will
be ignored by the tool -->
    <!-- Description>Check fpf cpu debug 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/Disabled" example="Enabled">
</RequiredValue>
</eolconfig>
<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>Disabled</State>
    <ErrAction>ErrorContinue</ErrAction>
    <RequiredValue format="Not set/Enabled/Disabled" example="Not
set"> </RequiredValue>
</eolconfig>
<eolconfig name="Key Manifest ID FPF">
    <!-- The commented fields bellow CANNOT be edited. Any edits will
be ignored by the tool -->
    <!-- Description>Check kmid 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="2 digit hex number with 0x prefix"
example="0xA4"> </RequiredValue>
</eolconfig>
<eolconfig name="Verified Boot FPF">
    <!-- The commented fields bellow CANNOT be edited. Any edits will
be ignored by the tool -->
    <!-- Description>Check fpf verified boot 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/Disabled" example="Disabled">
</RequiredValue>
</eolconfig>
<eolconfig name="Measured Boot FPF">
    <!-- The commented fields bellow CANNOT be edited. Any edits will
be ignored by the tool -->
    <!-- Description>Check fpf measure boot 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/Disabled" example="Enabled">
</RequiredValue>
</eolconfig>
<eolconfig name="Protect BIOS Environment FPF">
    <!-- The commented fields bellow CANNOT be edited. Any edits will
be ignored by the tool -->
    <!-- Description>Check fpf protect bios env 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/Disabled" example="Disabled">
</RequiredValue>
</eolconfig>
<eolconfig name="Force Boot Guard ACM FPF">
    <!-- The commented fields bellow CANNOT be edited. Any edits will
be ignored by the tool -->
    <!-- Description>Check fpf force boot 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/Disabled" example="Enabled">
</RequiredValue>
</eolconfig>
<eolconfig name="OEM Public Key Hash FPF">
    <!-- The commented fields bellow CANNOT be edited. Any edits will
be ignored by the tool -->
    <!-- Description>Check fpf oem key hash 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"
example="04ABF345031DEFA2B7E898791045ABDEF23549A00135782937ABDEEFFA10
EF33"> </RequiredValue>
</eolconfig>
<eolconfig name="Wireless LAN micro-code mismatch">
    <!-- The commented fields bellow CANNOT be edited. Any edits will
be ignored by the tool -->
    <!-- Description>Check ucode WLAN against programmed
ucode</Description -->
    <!-- IntelRequired>False</IntelRequired -->
    <!-- Dependencies>VPRO|WLAN|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="Yes/No -OR- 1/0" example="1">
</RequiredValue>
</eolconfig>
<eolconfig name="GBE version">
    <!-- The commented fields bellow CANNOT be edited. Any edits will
be ignored by the tool -->
    <!-- Description>Check Gbe Version against expected
value</Description -->
    <!-- IntelRequired>False</IntelRequired -->
    <!-- Dependencies>LAN</Dependencies -->
    <!-- End of uneditable fields -->
    <!-- Please edit the fields below ONLY with the State or ErrAction
-->
    <State>Disabled</State>
    <ErrAction>ErrorContinue</ErrAction>
    <RequiredValue format="major_ver.minor_ver" example="0.6">
</RequiredValue>
</eolconfig>
<eolconfig name="BIOS version">
    <!-- The commented fields bellow 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>Disabled</State>
    <ErrAction>ErrorContinue</ErrAction>
    <RequiredValue format="Customer specific"
example="HSWLPTU1.86C.0117.R00.1303102001"> </RequiredValue>
</eolconfig>
```



```
<eolconfig name="ME FW version">
    <!-- The commented fields bellow 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>Disabled</State>
    <ErrAction>ErrorContinue</ErrAction>
    <RequiredValue format="major_ver.minor_ver.hotfix_ver.build_num H
| LP | ULT" example="11.0.10.1040 LP"> </RequiredValue>
</eolconfig>
<eolconfig name="RPMC Check">
    <!-- The commented fields bellow CANNOT be edited. Any edits will
be ignored by the tool -->
    <!-- Description>Check RPMC</Description -->
    <!-- IntelRequired>False</IntelRequired -->
    <!-- Dependencies>RPMC</Dependencies -->
    <!-- End of uneditable fields -->
    <!-- Please edit the fields below ONLY with the State or ErrAction
-->
    <State>Disabled</State>
    <ErrAction>ErrorContinue</ErrAction>
</eolconfig>
<eolconfig name="System UUID">
    <!-- The commented fields bellow CANNOT be edited. Any edits will
be ignored by the tool -->
    <!-- Description>Check System UUID against programmed
value</Description -->
    <!-- IntelRequired>True</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 bellow CANNOT be edited. Any edits will
be ignored by the tool -->
    <!-- Description>Check Wireless MAC address</Description -->
    <!-- IntelRequired>True</IntelRequired -->
    <!-- Dependencies>VPRO|WLAN</Dependencies -->
    <!-- End of uneditable fields -->
    <!-- Please edit the fields below ONLY with the State or ErrAction
-->
    <State>Disabled</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 bellow CANNOT be edited. Any edits will
be ignored by the tool -->
    <!-- Description>Check MAC address</Description -->
    <!-- IntelRequired>True</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="6 hex pairs separated by ':'"
example="00:01:12:A2:3B:45"> </RequiredValue>
</eolconfig>
<eolconfig name="CF9GR lock check">
    <!-- The commented fields bellow CANNOT be edited. Any edits will
be ignored by the tool -->
    <!-- Description>Check CF9CR lock register</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="Security Descriptor Override (SDO) check">
    <!-- The commented fields bellow CANNOT be edited. Any edits will
be ignored by the tool -->
    <!-- Description>Check SDO pin</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="Flash Region Access Permissions">
    <!-- The commented fields bellow CANNOT be edited. Any edits will
be ignored by the tool -->
    <!-- Description>Check flash access</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="ME Manufacturing Mode status">
    <!-- The commented fields bellow 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="BIOS VSCC check">
    <!-- The commented fields bellow CANNOT be edited. Any edits will
be ignored by the tool -->
    <!-- Description>Check programmed BIOS VSCC 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="ME VSCC check">
    <!-- The commented fields bellow CANNOT be edited. Any edits will
be ignored by the tool -->
    <!-- Description>Check programmed ME VSCC 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 bellow 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="GuC Encryption Key">
    <!-- 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="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="BSP Initialization">
    <!-- 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="False/True/00/01" example="False">
</RequiredValue>
</eolvar>
<eolvar name="CPU Debugging">
    <!-- 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="False/True/00/01" example="False">
</RequiredValue>
</eolvar>
<eolvar name="Boot Guard Profile Configuration">
    <!-- 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="No_FVME/VE/VME/VM/FVE/FVME"
example="No_FVME"> </RequiredValue>
</eolvar>
<eolvar name="Key Manifest ID">
    <!-- 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="OEM Public Key Hash">
    <!-- 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="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="Embedded Host Based Configuration Enabled">
    <!-- 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>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="Enabled/Disabled/00/01" example="Enabled">
</RequiredValue>
</eolvar>
<eolvar name="PKI Domain Name Suffix">
    <!-- 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>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" example="0x0"> </RequiredValue>
</eolvar>
<eolvar name="MCTP PCIe Enabled">
    <!-- 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="False/True/00/01" example="False">
</RequiredValue>
</eolvar>
<eolvar name="MCTP eSPI Enabled">
    <!-- 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="False/True/00/01" example="False">
</RequiredValue>
</eolvar>
<eolvar name="MCTP Device Ports">
    <!-- 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="Reserved 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="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="ODM 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="NFC SMBus Address">
    <!-- 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="0x28 - NXP/0x29 - NXP/0x2A - NXP/0x2B -
NXP/00/Not Available" example="0x28 - NXP"> </RequiredValue>
</eolvar>
<eolvar name="Enable Near Field Communication">
    <!-- 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="Enabled/Disabled/00/01" example="Enabled">
</RequiredValue>
</eolvar>
<eolvar name="Intel(R) PTT initial power-up state">
    <!-- 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="Enabled/Disabled/00/01" example="Enabled">
</RequiredValue>
</eolvar>
<eolvar name="Intel(R) AMT initial power-up state">
    <!-- 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="Enabled/Disabled/00/01" example="Enabled">
</RequiredValue>
</eolvar>
<eolvar name="Intel(R) ME Network Services Supported">
    <!-- 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="No/Yes/00/01" example="No">
</RequiredValue>
</eolvar>
<eolvar name="Transport Layer Security Supported">
    <!-- 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="No/Yes/00/01" example="No">
</RequiredValue>
</eolvar>
<eolvar name="KVM Redirection Supported">
    <!-- 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="No/Yes/00/01" example="No">
</RequiredValue>
</eolvar>
<eolvar name="PAVP Supported">
    <!-- 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="No/Yes/00/01" example="No">
</RequiredValue>
</eolvar>
<eolvar name="Intel(R) AMT Supported">
    <!-- 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="No/Yes/00/01" example="No">
</RequiredValue>
</eolvar>
<eolvar name="Intel(R) PTT Supported">
    <!-- 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="Enabled/Disabled/00/01" example="Enabled">
</RequiredValue>
</eolvar>
<eolvar name="Auto BIST Config Status">
    <!-- 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="Enabled/Disabled/00/01" example="Enabled">
</RequiredValue>
</eolvar>
<eolvar name="OEM Tag">
    <!-- 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="Processor Emulation">
    <!-- 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="No
Emulation/vPro/Core/Celeron/Pentium/Xeon/Xeon Manageability Capable"
example="No Emulation"> </RequiredValue>
</eolvar>
<eolvar name="PROC_MISSING">
    <!-- 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="No onboard glue logic" example="No onboard
glue logic"> </RequiredValue>
</eolvar>
<eolvar name="Firmware Update OEM ID">
    <!-- 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="Intel(R) ME Region Flash Protection Override">
    <!-- 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="False/True/00/01" example="False">
</RequiredValue>
</eolvar>
<eolvar name="M3 Power Rail Available">
    <!-- 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="False/True/Not Available/Available/00/01"
example="False"> </RequiredValue>
</eolvar>
<eolvar name="Debug Override Production Silicon">
    <!-- 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="Debug Override Pre-Production Silicon">
    <!-- 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="WLAN Power Well">
```



```
    <!-- 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="Disabled/Sus Well/ME Well/SLP_M#||SPDA/WLAN
Sleep via SLP_WLAN#/80/82/83/84/85/86" example="Disabled">
</RequiredValue>
</eolvar>
<eolvar name="LAN Power Well">
    <!-- 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="Core Well/Sus Well/ME
Well/SLP_LAN#(MGPIO3)/00/01/02/03" example="Core Well">
</RequiredValue>
</eolvar>
<eolvar name="Firmware KVM Screen Blanking">
    <!-- 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>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/Yes/00/01" example="No">
</RequiredValue>
</eolvar>
<eolvar name="Intel(R) AMT Idle Timeout">
    <!-- 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>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" example="0x0"> </RequiredValue>
</eolvar>
<eolvar name="Redirection Privacy / Security Level">
    <!-- 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>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/Enhanced/Extreme/01/02/03"
example="Default"> </RequiredValue>
</eolvar>
<eolvar name="OEM Default Certificate 5 Stream">
    <!-- 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>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" example="0x0"> </RequiredValue>
</eolvar>
<eolvar name="OEM Default Certificate 5 Friendly Name">
    <!-- 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>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 bellow 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/True/Not Active/Active/00/01"
example="False"> </RequiredValue>
</eolvar>
<eolvar name="OEM Default Certificate 4 Stream">
    <!-- 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>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" example="0x0"> </RequiredValue>
</eolvar>
<eolvar name="OEM Default Certificate 4 Friendly Name">
    <!-- 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>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 bellow 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/True/Not Active/Active/00/01"
example="False"> </RequiredValue>
</eolvar>
<eolvar name="OEM Default Certificate 3 Stream">
    <!-- 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>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" example="0x0"> </RequiredValue>
</eolvar>
<eolvar name="OEM Default Certificate 3 Friendly Name">
    <!-- 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>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 bellow 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/True/Not Active/Active/00/01"
example="False"> </RequiredValue>
</eolvar>
<eolvar name="OEM Default Certificate 2 Stream">
    <!-- 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>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" example="0x0"> </RequiredValue>
</eolvar>
<eolvar name="OEM Default Certificate 2 Friendly Name">
    <!-- 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>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 bellow 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/True/Not Active/Active/00/01"
example="False"> </RequiredValue>
</eolvar>
<eolvar name="OEM Customizable Certificate 3 Stream">
    <!-- 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>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" example="0x0"> </RequiredValue>
</eolvar>
<eolvar name="OEM Customizable Certificate 3 Friendly Name">
    <!-- 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>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 bellow 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/True/Not Active/Active/00/01"
example="False"> </RequiredValue>
</eolvar>
<eolvar name="OEM Customizable Certificate 2 Stream">
    <!-- 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>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" example="0x0"> </RequiredValue>
</eolvar>
<eolvar name="OEM Customizable Certificate 2 Friendly Name">
    <!-- 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>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 bellow 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/True/Not Active/Active/00/01"
example="False"> </RequiredValue>
</eolvar>
<eolvar name="OEM Customizable Certificate 1 Stream">
    <!-- 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>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" example="0x0"> </RequiredValue>
</eolvar>
<eolvar name="OEM Customizable Certificate 1 Friendly Name">
    <!-- 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>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 bellow 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/True/Not Active/Active/00/01"
example="False"> </RequiredValue>
</eolvar>
<eolvar name="OEM Default Certificate Stream">
    <!-- 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>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" example="0x0"> </RequiredValue>
</eolvar>
<eolvar name="OEM Default Certificate Friendly Name">
    <!-- 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>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 bellow 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/True/Not Active/Active/00/01"
example="False"> </RequiredValue>
</eolvar>
<eolvar name="Config Server FQDN">
    <!-- 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="String" example="Any"> </RequiredValue>
</eolvar>
<eolvar name="FeatureShipState">
    <!-- 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="OEMSkuRule">
    <!-- 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="MEBxPassword">
    <!-- 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>
  <!-- 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: Create a line that begins with SubTestName="<specific sub test name>".

Here are some other examples that explain how to use this feature:

- To run a GbE version check defined under "Platform Configuration Checkings", a valid GbE version should be equal to string 1.2.3:

  SubTestName="GBE version", Reqval="1.2.3"

- To run the Variable check defined for "Remote Connectivity Service Enabler ID", a valid ID should be equal to string 550e8400-e29b-41d4-a716-446655440000:

  ```
  SubTestName="Remote Connectivity Service Enabler ID", Reqval="
  550e8400-e29b-41d4-a716-446655440000"
  ```

5.4.2 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.

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



5.4.3 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 16: 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.

5.4.4 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 1

5.5.1.1 Example for Consumer Intel® ME FW SKU

```
MEManuf -verbose

    Intel(R) MEManuf Version: 11.0.0.xxxx
    Copyright(C) 2005 - 2014, Intel Corporation. All rights reserved.

    FW Status Register1: 0x1E000255
    FW Status Register2: 0x62000006

      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
```

5.5.1.2 Example for Corporate Intel® ME FW SKU

```
MEManuf -verbose


Intel(R) MEManuf Version: 11.0.0.xxxx
Copyright(C) 2005 - 2014, Intel Corporation. All rights reserved.


FW Status Register1: 0x1E000255
FW Status Register2: 0x68000006

  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
```



5.5.1.3 Examples for ISH

Test 0

```
Intel(R) MEManuf Version: 11.0.0.1100
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
  SPI Flash Log:                        Not Present
  Phase:                                HOSTCOMM Module
  ICC:                                  Valid OEM data, ICC programmed
  ME File System Corrupted:             No

FW Capabilities value is 0x31103E40
Feature enablement is 0x31103E40
Platform type is 0x42110341
No Intel vPro Wireless device was found
Feature enablement is 0x31103E40

Self Test Result:
Test Level: configuration
Result: test pass for all sensors

MEManuf Operation Passed
```

Test 1

```
Intel(R) MEManuf Version: 11.0.0.1100
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
  SPI Flash Log:                      Not Present
  Phase:                              HOSTCOMM Module
  ICC:                                Valid OEM data, ICC programmed
  ME File System Corrupted:           No

FW Capabilities value is 0x31103E40
Feature enablement is 0x31103E40
Platform type is 0x42110341
No Intel vPro Wireless device was found
Feature enablement is 0x31103E40

Self Test Result:
Test Level: connectivity
Result: test pass for all sensors

MEManuf Operation Passed.
```

Test 2

```
Intel(R) MEManuf Version: 11.0.0.1100
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
  SPI Flash Log:                      Not Present
  Phase:                              HOSTCOMM Module
  ICC:                                Valid OEM data, ICC programmed
  ME File System Corrupted:           No

FW Capabilities value is 0x31103E40
Feature enablement is 0x31103E40
Platform type is 0x42110341
No Intel vPro Wireless device was found
Feature enablement is 0x31103E40
```



```
Self Test Result:
Test Level: calibration
Result: test pass for all sensors

MEManuf Operation Passed
```

Test 3

```
Intel(R) MEManuf Version: 11.0.0.1100
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
  SPI Flash Log:                        Not Present
  Phase:                                HOSTCOMM Module
  ICC:                                  Valid OEM data, ICC programmed
  ME File System Corrupted:             No

FW Capabilities value is 0x31103E40
Feature enablement is 0x31103E40
Platform type is 0x42110341
No Intel vPro Wireless device was found
Feature enablement is 0x31103E40

Self Test Result:
Test Level: BIST
Result: test pass for all sensors

MEManuf Operation Passed
```

§



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* 7 64/32 bit and Windows* 8.1 64/32 bit.

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] [-ISH FWStat] [-ISH ? or h]
           [ISH sensorInfo] [-FEAT<ISHParmName> -VALUE <value>]
           [-ISH exp][ISH]

MEInfo.efi [-EXP] [-H|?] [-VER] [-FITVER] [-FEAT] [-VALUE] [-FWSTS]
           [-VERBOSE] [-PAGE][-ISH] [-ISH FWStat] [-ISH ? or h]
           [ISH sensorInfo] [-FEAT<ISHParmName> -VALUE <value>]
           [-ISH exp][ISH]
```

M



Table 17: Intel® MEInfo Command Line Options

Option	Description
-FEAT < name> -VALUE <value>	Compares the value of the given feature name 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. 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
-FEAT <name>	Retrieves the current value for the specified feature. If the feature name is more than one word, the entire feature name must be enclosed in quotation marks. 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: 0x1E000255 FW Status Register2: 0x69000006 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
-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.
-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.

Option	Description
-ISH	Display information for ISH.
-ISH FWStat	This displays the ISH Firmware status.
-ISH ? or h	This displays the Intel® MEInfo ISH help screen.
-ISH SensorInfo	This displays information about all of the sensors connected to the ISH.
-FEAT <ISHParmName> -VALUE <value>	This checks if an ISH Parameter matches the given value being parsed in.
No option:	If the tool is invoked without parameters, it reports information for all components listed in Table 18 below for full SKU FW.

Table 18: 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: 11.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 11.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
VendorID	Intel® ME Kernel	X	X	N/A	A number (in Hex)
PCH Version	Intel® ME Kernel	X	X	N/A	A version string
FW Version	Intel® ME Kernel	X	X	N/A	Version string 11.x.y.ZZZZ; where x=minor, y = HF/MR, ZZZZ = Build Number.



Feature Name	Feature Data Source (Intel® ME Kernel/ Intel® AMT/SW/ Other)	Consumer SKU	Corporate SKU	Specific Feature Dependency	Field Value
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
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.
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
BIOS and GbE Config Lock	Other (Directly reading from SPI)	X	X	N/A	Enabled/Disabled/ Unknown If shown as enabled, both FLOCKDN for BIOS and Gbe are set. If shown as disabled, either/all FLOCKDN for

Feature Name	Feature Data Source (Intel® ME Kernel/ Intel® AMT/SW/ Other)	Consumer SKU	Corporate SKU	Specific Feature Dependency	Field Value
					BIOS and Gbe are not set.
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
SPI Flash ID	Other (Directly reading from SPI)	X	X	Only when there are flash parts HW installed	A JEDEC ID number (in Hex)
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
Independent Firmware Recovery	FWU	All	All	Only when Windows* IFR Agent is installed and the FW image has IFR set to 'true'	Enabled/Disabled
RPMC Supported	Intel® ME Kernel	All	All		Yes / No
RPMC Counters	Intel® ME Kernel	All	All		A number (in Hex)
RPMC Status	Intel® ME Kernel	All	All		Enabled
NFC FW Version	NFC	Both	All	N/A	A version string. If NFC HW device is not found/accessible, display "Not Available"
NFC Loader Version	NFC	Both	All	N/A	A version string. If NFC HW device is not found/accessible, display "Not Available"
Link Status	Intel® AMT	X	X	Intel® AMT CEM (a.k.a Common Service) is used. Not available on Corporate Sku	Link up/down
Intel® AMT State	Intel® ME Kernel	N/A	X	Both Full Manageability and Manageability Application has to be PRESENT (Capable)	Enabled/Disabled
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

Feature Name	Feature Data Source (Intel® ME Kernel/ Intel® AMT/SW/ Other)	Consumer SKU	Corporate SKU	Specific Feature Dependency	Field Value
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	X	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 “=”)
Wireless MAC Address	Intel® AMT	X	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	X	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	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
				Sku	
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	X	X	Not available on Corporate SKU. Only shown when AMT is enabled	Default/Enhanced/Extreme/ Unknown
FWSTS	Intel® ME Kernel	X	X	N/A	Two 32bit Hexadecimal numbers and their bit definition breakdown
Wireless Micro-code Mismatch	FWU	Corporate	All	N/A	Yes: FW has detected a ucode mismatch, and partial FWUpdate needs to be performed
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
OEM Public Key Hash FPF	Intel® ME Kernel	All	All	BIOS	Yes / No

Feature Name	Feature Data Source (Intel® ME Kernel / Intel® AMT / SW / Other)	Consumer SKU	Corporate SKU	Specific Feature Dependency	Field Value
OEM Public Key Hash ME	Intel® ME Kernel	All	All	BIOS	SHA-256bit Hash entry
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	
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
Protect BIOS Environment	Intel® ME Kernel	All	All	BIOS	Yes / No
CPU Debugging	Intel® ME Kernel	All	All	BIOS	Enabled / Disabled
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
Key Manifest ID	Intel® ME Kernel	All	All	BIOS	Hash of Public Key to verify Boot Policy Manifest
Enforcement Policy	Intel® ME Kernel	All	All	BIOS	Unrestricted / Remediation / Restricted
PTT	Intel® ME Kernel	All	All	BIOS	Enabled / Disabled
EK Revoke	Intel® ME Kernel	All	All	BIOS	Revoked / Not Revoked
Integrated Sensor Solution FW State	ISH	All	All	ISH Firmware	Responding / Not Responding
FW Status	ISH	All	All	ISH Firmware	Sensors Apps Responding / Sensor Apps Not Responding
Integrated Sensor Solution FW Version	ISH	All	All	ISH Firmware	Version string
Module Status	ISH	All	All	ISH Firmware	Module x Status: Loaded /



Feature Name	Feature Data Source (Intel® ME Kernel/ Intel® AMT/SW/ Other)	Consumer SKU	Corporate SKU	Specific Feature Dependency	Field Value
					Not Loaded
Extended Modules FW Status	ISH	All	All	ISH Firmware	Version string
Extended Modules FW Versions	ISH	All	All	ISH Firmware	Version string
HECI Driver Version	ISH	All	All	ISH Firmware	Version string
PCI Bus Driver Version	ISH	All	All	ISH Firmware	Version string
Integrated Sensor Solution Driver Version	ISH	All	All	ISH Firmware	Version string
Sensors Information	ISH	All	All	ISH Firmware	Information on the various Sensors configured on the platform.

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.

6.3.1 Consumer Intel® ME FW SKU

```
MEINFOWIN.exe

Intel(R) MEInfo Version: 11.0.0.xxxx
Copyright(C) 2005 - 2014, Intel Corporation. All rights reserved.

Intel(R) ME code versions:

BIOS Version                                    SKLSE2P1.86C.B053.R02.1409290036
MEBx Version                                    11.0.0.xxxx
GbE Version                                     0.7
Vendor ID                                       8086
PCH Version                                     1
FW Version                                      11.0.0.xxxx LP
LMS Version                                     Not Available
MEI Driver Version                              11.0.0.xxxx
Wireless Hardware Version                       Not Available
Wireless Driver Version                         Not Available

FW Capabilities                                 0x31103E40

    Intel(R) Capability Licensing Service - PRESENT/ENABLED
    Protect Audio Video Path - PRESENT/ENABLED
    Intel(R) Dynamic Application Loader - PRESENT/ENABLED
    Intel(R) Platform Trust Technology - PRESENT/ENABLED
    Intel(R) Network Frame Forwarder - PRESENT/ENABLED

TLS                                             Disabled
Last ME reset reason                            Global system reset
Local FWUpdate                                  Enabled
BIOS Config Lock                                Disabled
GbE Config Lock                                 Disabled
Host Read Access to ME                          Enabled
Host Write Access to ME                         Enabled
SPI Flash ID #1                                 EF4018
SPI Flash ID VSCC #1                            0
SPI Flash ID #2                                 Unknown
SPI Flash ID VSCC #2                            Unknown
SPI Flash BIOS VSCC                             20042004
BIOS boot State                                 Post Boot
OEM ID                                          00000000-0000-0000-0000-000000000000
Capability Licensing Service                    Enabled
OEM Tag                                         0x00000000
Slot 1 Board Manufacturer                       0x00000001
Slot 2 System Assembler                         0x00000002
Slot 3 Reserved                                 0x00000003
M3 Autotest                                     Disabled
C-link Status                                   Enabled
Independent Firmware Recovery                   Enabled
RPMC Supported                                  Not Available
RPMC Counters                                   Not Available
RPMC Status                                     Not Available
OEM Public Key Hash FPF                         Not set
OEM Public Key Hash ME
    0000000000000000000000000000000000000000000000000000000000000000
ACM SVN FPF                                     0x0
KM SVN FPF                                      0x0
BSMM SVN FPF                                    0x0
GuC Encryption Key FPF                          Not set

Error 10: Failed getting variable "GuC Encryption Key" value
```



```
GuC Encryption Key ME                       Not set

                                            FPF                     ME
                                            ---                     --
Force Boot Guard ACM                        Not set                 Disabled
Protect BIOS Environment                    Not set                 Disabled
CPU Debugging                               Not set                 Disabled
BSP Initialization                          Not set                 Disabled
Measured Boot                               Not set                 Disabled
Verified Boot                               Not set                 Disabled
Key Manifset ID                             Not set                 0x0
Enforcement Policy                          Not set                 0x0
PTT                                         Not set                 Enabled
EK Revoke State                             Not Revoked

Integrated Sensor Solution FW State         responding
FW Status                                   sensors apps running
Integrated Sensor Solution FW Version       3.0.0.1016
Module Status                               Module 1397c90 Status: loaded
Extended Modules FW Versions                1.0.0.0
HECI Driver Version                         3.0.0.1030
PCI Bus Driver Version                      3.0.0.1030
Integrated Sensor Solution Driver Version   3.0.0.1016
Sensors Information


                                            Sensor LUID: 10073
                                            Sensor Name: motion accelerometer 3D
                                            Vendor: ST Micro
                                            Sensor Sub Type: LSM303D
                                            Bus Type: I2C
                                            Bus Address: 1d
                                            Calibration Status: set

                                            Sensor LUID: 20041
                                            Sensor Name: light ambientlight
                                            Vendor: Capella
                                            Sensor Sub Type: CM32181
                                            Bus Type: I2C
                                            Bus Address: 48
                                            Calibration Status: set

                                            Sensor LUID: 10076
                                            Sensor Name: motion gyrometer 3D
                                            Vendor: ST Micro
                                            Sensor Sub Type: L3GD20H
                                            Bus Type: I2C
                                            Bus Address: 6b
                                            Calibration Status: set

                                            Sensor Name: activity PDR
                                            Vendor: Intel
                                            Sensor Sub Type: no sub type
                                            Bus Type: no BUS entry
                                            Calibration Status: not set

                                            Sensor LUID: 20C
                                            Sensor Name: Intel oreintation
                                            Vendor: Intel
                                            Sensor Sub Type: no sub type
                                            Bus Type: no BUS entry
                                            Calibration Status: not set

                                            Calibration Status: not set

                                            Sensor LUID: 211
                                            Sensor Name: Intel gesture
                                            Vendor: Intel
                                            Sensor Sub Type: no sub type
                                            Bus Type: no BUS entry
                                            Calibration Status: not set
```

```
Sensor LUID: 212
Sensor Name: Intel gesture
Vendor: Intel
Sensor Sub Type: no sub type
Bus Type: no BUS entry
Calibration Status: not set

Sensor LUID: 83
Sensor Name: orientation compass 3D
Vendor: Intel
Sensor Sub Type: no sub type
Bus Type: no BUS entry
Calibration Status: not set
```



6.3.2 Corporate Intel® ME FW SKU

```
MEINFOWIN.exe
 Intel(R) MEInfo Version: 11.0.0.xxxx
 Copyright(C) 2005 - 2014, Intel Corporation. All rights reserved.

 Intel(R) ME code versions:

 BIOS Version                                SKLSE2P1.86C.B053.R02.1409290036
 MEBx Version                                11.0.0.xxxx
 GbE Version                                 0.7
 Vendor ID                                   8086
 PCH Version                                 1
 FW Version                                  11.0.0.xxxx LP
 LMS Version                                 Not Available
 MEI Driver Version                          11.0.0.xxxx
 Wireless Hardware Version                   Not Available
 Wireless Driver Version                     Not Available

 FW Capabilities                             0x31103E40

     Intel(R) Active Management Technology - PRESENT/ENABLED
     Intel(R) Capability Licensing Service - PRESENT/ENABLED
     Protect Audio Video Path - PRESENT/ENABLED
     Intel(R) Dynamic Application Loader - PRESENT/ENABLED
     Intel(R) Platform Trust Technology - PRESENT/ENABLED
     Intel(R) Network Frame Forwarder - PRESENT/ENABLED

 Intel(R) AMT State:                         Enabled
 TLS:                                        Disabled
 Last ME reset reason:                       Power up
 Local FWUpdate:                             Enabled
 BIOS Config Lock:                           Enabled
 GbE Config Lock:                            Enabled
 Host Read Access to ME:                     Enabled
 Host Write Access to ME:                    Enabled
 SPI Flash ID #1:                            EF4017
 SPI Flash ID VSCC #1:                       20252025
 SPI Flash ID #2:                            EF4017
 SPI Flash ID VSCC #2:                       20252025
 SPI Flash BIOS VSCC:                        20252025
 BIOS boot State:                            Post Boot
 OEM Id:                                     00000000-0000-0000-0000-000000000000
 Link Status:                                Link down
 System UUID:                                88888888-8887-8888-8888-878888888888
 MAC Address:                                88-88-88-88-87-88
 IPv4 Address:                               0.0.0.0
 IPv6 Enablement:                            Disabled
 Privacy/Security Level:                     Default
 Configuration state:                        Not started
 Provisioning Mode:                          PKI
 Capability Licensing Service:               Enabled
 OEM Tag:                                    0x00000000
 Slot 1 Board Manufacturer:                  Unused
 Slot 2 System Assembler:                    Unused
 Slot 3 Reserved:                            Unused
 CM3 Autotest:                                Disabled
 C-link Status:                              Enabled
 Wireless Micro-code Mismatch:               No
 Wireless Micro-code ID in Firmware:         0x08B1
 Wireless LAN in Firmware:                   Intel(R) Centrino(R) Advanced-AC 7260
 Wireless Hardware ID:                       No Intel WLAN card installed
 Wireless LAN Hardware:                      No Intel WLAN card installed
 Localized Language:                         English
 Independent Firmware Recovery:              Enabled
  Message Data    [File]: 00 00 00 00 0A 00 00 00
  NOTE: (Regardless of size, display first 32-bytes)
 Verifying Command Status...
 OEM Public Key Hash (FPF):                  not set
 OEM Public Key Hash (ME):                   00000000
 ACM SVN FPF:                                0x0
 KM SVN FPF:                                 0x0
```

```
BSMM SVN FPF:                           0x0
pGetOemSecureBootPolicyAck->Status: 250
 Message Data   [File]: 00 00 00 00 00 00 00 00
 NOTE: (Regardless of size, display first 32-bytes)
Verifying Command Status...

                                        FPF                 ME
                                        ---                 --
Force Boot Policy:                      not set             Disabled
Protect BIOS Environment:               not set             Disabled
CPU Debug Disabled:                     not set             Disabled
BSP Initialization Disabled:            not set             Disabled
 Message Data   [File]: 00 00 00 00 00 00 00 00
 NOTE: (Regardless of size, display first 32-bytes)
Verifying Command Status...
Perform Measured Boot:                  not set             Disabled
Perform Secure Boot:                    not set             Disabled
 Message Data   [File]: 00 00 00 00 00 00 00 00
 NOTE: (Regardless of size, display first 32-bytes)
Verifying Command Status...
Key Manifest ID:                        not set             0x0
 Message Data   [File]: 00 00 00 00 00 00 00 00
 NOTE: (Regardless of size, display first 32-bytes)
Verifying Command Status...
Enforcement Policy:                     not set             0x0
 Message Data   [File]: 00 00 00 00 00 00 00 00
 NOTE: (Regardless of size, display first 32-bytes)
Verifying Command Status...
"Enable Intel (R) Platform Trusted Technology"
FTPM:                                   not set             not set
```



6.3.3 Retrieve the Current Value of the Flash Version

```
C:\ MEINFO.exe -feat "BIOS boot state"
Intel(R) MEINFO Version: 11.0.0.xxxx
Copyright(C) 2005 - 2014, Intel Corporation. All rights reserved.

BIOS boot State: Post Boot

> MEINFO.efi -feat ``^"BIOS boot state"^''
Intel(R) MEINFO Version: 11.0.0.xxxx
Copyright(C) 2005 - 2014, Intel Corporation. All rights reserved.


BIOS boot State: Post Boot
```

6.3.4 Checks Whether the Computer has Completed the Setup and Configuration Process

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

Intel(R) MEINFO Version: 11.0.0.xxxx
Copyright(C) 2005 - 2014, 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: 11.0.0.xxxx
Copyright(C) 2005 - 2014, 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 and LOCL partitions that Intel provides on the OEM website. Intel® FWUpdate updates the entire Intel® ME code area. In addition FWUpdate local can perform a partial update to change / update the WCOD or LOCL portions.

The image file that the tool uses for the update is the same image file that is used

by the FIT tool to create a firmware image for use in the SPI. A sample FW image

file for updating would be '**ME11.0_5M_Production.bin**'. These files are located in the

'Image Components\ME' sub-folder of the firmware kit.

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.

NOTES: 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

FWUT.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.

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 please refer to Intel® MEBx user guide.

7.4 Usage

NOTES: 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] [-GENERIC]
              [-SAVE] [-FWVER] [-ALLOWSV] [-FORCERESET]
              [-OEMID]
```

```
FWUpdLcl.efi  [-H|?] [-VER] [-EXP] [-VERBOSE] [-F] [-Y] [-SAVE]
              [-FWVER] [-ALLOWSV] [-FORCERESET] [-OEMID]
```

NOTES: Image File is the image file of the FW to be updated. Is the same image file used by FIT.

Table 19: 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.
-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 <wcod or locl>	This option is always used along with the –F option. The partition ID is requested using the “partid” option, which takes in wcod or locl string as input. 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 and invalid file error will be returned by the tool. Also, if the requested partition is not expected by the firmware and 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.
-GENERIC	Intel® MEI. Specifies that the tool performs the update over the Intel® MEI interface. Intel® MEI is used even if the FW supports a network-based update. **Note**: This option is only supported in the Windows* version of the tool.
-FWVER	Display FW version
-H or -?	Displays the list of command line options supported by the Intel® MEINFO tool.
-EXP	Shows examples about how to use the tools.



Option	Description
-VER	Shows the version of the tools.

7.5 Examples

7.5.1 Updates Intel® ME with Firmware Binary File

This command updates Intel® ME with FW.BIN file. If the firmware on current platform is newer than then version in FW.BIN file, tools will promote a warning to let user know there will be a firmware downgrade (rollback) event 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.

```
FWUT.exe -f FW.BIN

EFI:
FWUT.efi -f FW.BIN
```



7.5.2 Partial Firmware Update

This command will perform a partial update of the FW via Intel® MEI for either the wcod or locl partitions.

```
FWUT.exe -f FW.bin -partid <wcod or locl or ishc>

EFI:
FWUT.efi -f upd.bin -partid <wcod or locl or ishc>
```

Non-Verbose Mode

```
C:\>FWUT.exe –f FW.BIN.bin –partid WCOD

Intel (R) Firmware Update Utility version 11.0.0.xxxx
Copyright (C) 2007-2014, Intel Corporation.  All rights reserved.

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

FW Update:  [ 100% (Stage: 31 of 19)(|)]
FW Update is completed successfully.
```

Verbose Mode

```
C:\>FWUT.exe –f FW.BIN.bin –partid WCOD -verbose

Intel (R) Firmware Update Utility version 11.0.0.xxxx
Copyright (C) 2007-2014, 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.
```



7.5.3 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 '-' removed. 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:\> FWUT.exe -exp partid

Intel (R) Firmware Update Utility version 11.0.0.xxxx
Copyright (C) 2007-2014, 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>]]
     [Y] [PASS<pass>]
  2. F<file> PARTID[<Partition ID>] INSTID[<Instance ID>]
     [FORCERESET] [VERBOSE[<file>]] [Y] [PASS<pass>]

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

EFI:

```
> FWUT.efi -exp partid

Intel (R) Firmware Update Utility version 11.0.0.xxxx
Copyright (C) 2007-2014, 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>]]
     [Y] [PASS<pass>]
  2. F<file> PARTID[<Partition ID>] INSTID[<Instance ID>]
     [FORCERESET] [VERBOSE[<file>]] [Y] [PASS<pass>]

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

7.5.4 Language Codes

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 Intel® Manifest Extension Utility (MEU)

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

The Intel® Manifest Extension Utility (MEU) requires administrator privileges to run under Windows* OS. The user needs to use the Run as Administrator option to open the CLI in Windows* 7 64/32 bit and Windows* 8.1 64/32 bit.

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
- Calls the signing infrastructure tool to sign the partition.

8.1 Usage

The executable can be invoked by:

```
MEU.exe [-exp] [-h|?] [-version|ver] [-binlist] [-o] [-f] [-gen]
        [-cfg] [-w] [-s] [-d] [-u1] [-u2] [-u3] [-mnver] [-key]
```

Table 20: Options

Option	Description
-H or -?:	Displays the list of command line options supported by the Intel® MEU tool.
-EXP	Shows examples about how to use the tools.
-VER	Shows the version of the tools.
-binlist	Displays a list of supported binary types.
-o <filename>	Overrides the output file path.
-f <filename>	Specifies input XML file.
-gen <type>	Specifies the binary type for which to generate a template XML file.
-cfg <filename>	Overrides the path to the tool config XML file.
-w <path>	Overrides the $WorkingDir environment variable.
-s <path>	Overrides the $SourceDir environment variable.
-d <path>	Overrides the $DestDir environment variable.

Option	Description
-u1 <path>	Overrides the $UserVar1 environment variable.
-u2 <path>	Overrides the $UserVar2 environment variable.
-u3 <path>	Overrides the $UserVar3 environment variable.
-mnver <value>	Overrides the version of the output binary. (Format: Major.Minor.Hotfix.Build)
-key <path>	Overrides the signing key in the tool config XML file.

8.2 Examples

8.2.1 Generate Configuration XML template

This command will generate the configuration XML template file using MEU.

Windows / WinPE:

```
MEU.exe -gen meu_config

================================================================================
SPT Manifest Extension Utility. Version: 11.0.0.1105
Copyright (c) 2013 - 2014, Intel Corporation. All rights reserved.
1/21/2015 - 2:14:28 pm
================================================================================
Command Line: meu.exe -gen meu_config
Saving XML ...

XML file written to meu_config.xml
```

8.2.2 Generate Code partition XML

This command will generate the Code partition XML file using MEU.

Windows / WinPE:

```
MEU.exe -gen CodePartition

================================================================================
SPT Manifest Extension Utility. Version: 11.0.0.1105
Copyright (c) 2013 - 2014, Intel Corporation. All rights reserved.
1/21/2015 - 2:14:16 pm
================================================================================

Command Line: meu.exe -gen CodePartition

Saving XML ...
```

```
XML file written to CodePartition.xml
```

8.2.3 Generate Compressed & Signed partition

This command will create the compressed & signed partition using MEU.

Windows / WinPE:

```
MEU.exe -f CodePartition.xml -o ISHC_MEU.bin

================================================================================
SPT Manifest Extension Utility. Version: 11.0.0.1105
Copyright (c) 2013 - 2014, Intel Corporation. All rights reserved.
1/21/2015 - 2:23:25 pm
================================================================================

Command Line: meu.exe -f CodePartition.xml -o ISHC_MEU.bin

Executing pre-build actions

Building objects

Processing attribute: CodePartition

Executing post-build actions


Full Flash image written to C:\...\ISHC_MEU.bin
```

9 Appendix 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 20: 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 **See 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.
Notes:
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. There is NO ability to change features one at a time. This NVAR sets OEM Permanent Disable for ALL features. In addition prior 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. Please see the 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>Near Field Communication</td><td>3</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>5</td><td>Reserved</td><td></td></tr><tr><td>4:3</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’.
3. When configuring OEM Sku Rule for NFC the NfcSmbusAddr and NfcGpioIrq NVARs must also be programmed at the same time.</td><td>Global</td><td>Yes</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. Please see the 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.

Notes:

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. There is NO ability to change features one at a time. This NVAR sets OEM Permanent Disable for ALL features. In addition prior updating or changing any of available settings it is highly recommended that the user first retrieves the current Feature Shipment Time State and toggling only the desired bits, and then resave them.</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>

Note: When disabling PTT using Feature Shipment Time state NVAR, please execute a reset after executing fpt.efi –commit to ensure PTT is disabled completely.</td>
<td>Global</td>
<td>Yes</td>
</tr>
</table>



Fixed Offset Name	Description	Data Length (in Bytes)	Expected Value	Reset Type	Mfg. Post EOM / Pre EOP
SetWLANPowerWell	Sets which power well the board uses for WLAN cards	4	**0x80** = Disabled **0x81** = Core Well || SLP_S3 **0x82** = Primary Well ||SLP_SUS **0x83** = ME Well || SLP_A **0x86** = WLAN Sleep via SLP_WLAN#	**Global**	**Yes**
OEM_TAG	A human readable 32-bit number to describe the flash image represented by value	4	Readable 32 bit hex value identifying the image. Can be empty (Null).	**Global**	**Yes**

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 NFC_RST, GPP_E, 6 NFC_IRQ, GPP_E, 12 NFC_DFU, GPP_B, 15 Note: the only locations that can be modified are underlined above. The format for updating the GPIO is as follows... GpioNvar = 0x00000000030000001100000001000000000000001000000020000001700000001000000000000000800000003000000130000000100000000000000 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**



Fixed Offset Name	Description	Data Length (in Bytes)	Expected Value	Reset Type	Mfg. Post EOM / Pre EOP
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	**Global**	**No**
LPSCON_PORT	LSPCON Port Configuration. 0x00 - 0x02 - B 0x04 - C 0x08 - D	1	0x00 0x02 0x04 0x08	**Global**	**No**
AMT Related NVARs					
OEM Customizable Certificate 1	Cert Hash Data. See Certificate Hash Entry Structure definition **Note: If the platform is un-configured the Certificate Hash will be deleted.**	55 => n >= 83	Valid Certificate Hash Entry (SHA1, SHA256 or SHA384)	**ME**	**Yes**
OEM Customizable Certificate 2	Cert Hash Data. See Certificate Hash Entry Structure definition **Note: If the platform is un-configured the Certificate Hash will be deleted.**	55 => n >= 83	Valid Certificate Hash Entry (SHA1, SHA256 or SHA384)	**ME**	**Yes**
OEM Customizable Certificate 3	Cert Hash Data. See Certificate Hash Entry Structure definition **Note: If the platform is un-configured the Certificate Hash will be deleted.**	55 => n >= 83	Valid Certificate Hash Entry (SHA1, SHA256 or SHA384)	**ME**	**Yes**

Fixed Offset Name	Description	Data Length (in Bytes)	Expected Value	Reset Type	Mfg. Post EOM / Pre EOP
Privacy/Security Level	Redirection (KVM, SOL, IDE-r) privacy level and configuration (RCFG, CCM) settings.	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**
NfcSmbusAddr	NFC Radio SMBus Address	1	0x28 - NXP 0x29 - NXP 0x2A - NXP 0x2B - NXP **Note:** When configuring NFC using all related NVAR options must be programmed at the same time.	**Global**	**No**
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	**ME**	**Yes**



Fixed Offset Name	Description	Data Length (in Bytes)	Expected Value	Reset Type	Mfg. Post EOM / Pre EOP
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	Redirection (Storage, SOL, KVM)
	struct{
	
	UINT8 StorageState : 1,//Storage redir is enabled
	Sol : 1,//SOL is enabled
	Kvm : 1,//KVM is enabled
	Reserved : 5;
	}
		1	Range: 0-7	**ME**	**Yes**

*OptinPolicy	Change User Opt-in (lower nibble). NONE = 0, KVM = 1, IDER = 2, BOOT_OPTIONS = 4, 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 0x02 0x12 0x04 0x14 0x0F 0x1F	**ME**	**Yes**



Fixed Offset Name	Description	Data Length (in Bytes)	Expected Value	Reset Type	Mfg. Post EOM / Pre EOP

DomainHost	Set Domain Name and Set Host Name from FQDN input. A device with a local hostname myhost and a parent domain name "example.com" has the fully qualified domain name myhost.example.com. The FQDN uniquely distinguishes the device from any other hosts called "myhost" in other domains.
	#define NET_HOSTNAME_MAX_LENGTH 64
	#define NET_DOMAIN_NAME_MAX_LENGTH (256-64)
	typedef struct {
	CHAR8 Hostname[NET_HOSTNAME_MAX_LENGTH];
	CHAR8 DomainName[NET_DOMAIN_NAME_MAX_LENGTH];
	}
	HOST_FQDN_NAMED_VAR;
		256	Examples of FQDN:
			myserver.intel.com
			amr.corp.intel.com
			www.intel.com
			mymail.somecollege.edu
				ME	**Yes**

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



Fixed Offset Name	Description	Data Length (in Bytes)	Expected Value	Reset Type	Mfg. Post EOM / Pre EOP
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 Programming 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, D) are configured.

§



10 Appendix B: Tool Detail Error Codes

A.1 Common Error Code for All Tools

Error Code	Error Message	Response
0	Success	
1	Memory allocation error occurred	Make sure there is enough memory in the system
2	Invalid descriptor region	Check descriptor region
3	Region does not exist	Check region to be programmed
4	Failure. Unexpected error occurred	Contact Intel
5	Invalid data for Read ID command	Contact Intel
6	Error occurred while communicating with SPI device	Check SPI device
7	Hardware sequencing failed. Make sure that access permissions are correct for the target flash area	Check descriptor region access settings
8	Software sequencing failed. Make sure that access permissions are correct for the target flash area	Check descriptor region access settings
9	Unrecognized value in the HSFSTS register	Unrecognized value in the HSFSTS register
10	Hardware Timeout occurred in SPI device	Hardware Timeout occurred in SPI device
11	AEL is not equal to zero	AEL is not equal to zero
12	FCERR is not equal to zero	FCERR is not equal to zero
25	The host CPU does not have writes access to the target flash area. To enable write access for this operation the user needs to modify the descriptor settings to give host access to this region.	Check descriptor region access settings
26	The host CPU does not have read access to the target flash area. To enable read access for this operation the user needs to modify the descriptor settings to give host access to this region.	Check descriptor region access settings
27	The host CPU does not have erase access to the target flash area. To enable erase access for this operation the user needs to modify the descriptor settings to give host access to this region.	Check descriptor region access settings



Error Code	Error Message	Response
28	Protected Range Registers are currently set by BIOS, preventing flash access. Contact the target system BIOS vendor for an option to disable Protected Range Registers.	Assert Flash Descriptor Override Strap (GPIO33) to Low, Power Cycle, and Retry. If Protected Range Registers (memory location: SPIBAR + 74h -> 8Fh) are still set, contact the target BIOS vendor.
50	General Erase failure	Attempt the command again. If it fails again, contact Intel.
51	An attempt was made to read beyond the end of flash memory	Check address
52	An attempt was made to write beyond the end of flash memory	Check address
53	An attempt was made to erase beyond the end of flash memory	Check address
54	The address <address> of the block to erase is not aligned correctly	Check address
55	Internal Error	Contact Intel
56	The supplied zero-based index of the SPI Device is out of range.	The supplied zero-based index of the SPI Device is out of range.
57	AEL or FCERR is not equal to zero for Software Sequencing	AEL or FCERR is not equal to zero for Software Sequencing
75	File not found	Check file location
76	Access was denied opening the file	Check file location
77	An unknown error occurred while opening the file	Verify the file is not corrupt
78	Failed to allocate memory for the flash part definition file	Check system memory Verify the file is not corrupt
79	Failed to read the entire file into memory	Check system memory Verify the file is not corrupt
80	Parsing of file failed	Check system memory Verify the file is not corrupt
100	This error can occur if both Software and Hardware sequencing are not available and the SPI Flash configuration registers are write protected by the Flash Configuration Lock-Down bit (FLOCKDN). Contact the BIOS vendor to unlock this bit or enable hardware sequencing in descriptor mode.	Check with BIOS vendor or SPI programming Guide

Error Code	Error Message	Response
101	No SPI flash device could be identified. Please verify if Fparts.txt has support for this part	Verify Fparts.txt contains device supported.
102	Failed to read the device ID from the SPI flash part	Verify Fparts.txt has correct values
103	There are no supported SPI flash devices installed. Check connectivity and orientation of SPI flash device	Verify Fparts.txt has correct values. Check SPI Device
104	The two SPI flash devices do not have compatible command sets	Verify both SPI devices on the system are compatible
105	An error occurred while writing to the write status register of the SPI flash device. This program will not be able to modify the SPI flash	Check SPI Device
202	Confirmation is not received from the user to perform operation.	
203	Flash is not blank	
204	Data verify mismatch found	
205	Unexpected failure occurred	
207	Invalid parameter value specified by user. The option specified cannot be run on a platform with Intel® ME Ignition FW	
208	Intel® ME is disabled	
209	Intel® ME failed to reset	
210	Requesting Intel® ME FW Reset failure.	
211	Communications error between FPT and the Intel® ME.	
212	The request to disable the Intel® ME failed.	
213	Intel® ME disable is not required	
214	Intel® ME is already disabled	
215	The attempt to commit the NVARs has failed.	
216	The Close Manufacturing process failed.	
217	Setting Global Reset Failed	
240	Access was denied opening the file	
241	Access was denied creating the file	
242	An unknown error occurred while opening the file	
243	An unknown error occurred while creating	
244	Not a valid file	
245	File not found error	
246	Failed to read the entire file into memory	
247	Failed to write the entire flash contents to file	



Error Code	Error Message	Response
248	File already exists	
249	The file is longer than the flash area to write.	
250	The file is smaller than the flash area to write.	
251	Length of image file extends past the flash area.	
252	Image file not found.	
253	File does not exist	
254	Not able to open the file	
255	Error occurred while reading the file	
256	Error occurred while writing to the file	
280	Failed to disable write protection for the BIOS space	
281	The Enable bit in the LPC RCBA register is not set. The value of this register cannot be used as the SPI BIOS base address.	
282	Failed to get information about the installed flash devices	
283	Unable to write data to flash.	
284	Fail to load driver (PCI access for Windows*). The tool needs to run with an administrator privilege account.	
320	FPT General failure error	
321	The address is outside the boundaries of the flash area.	
360	Invalid Block Erase Size value in	
361	Invalid Write Granularity value in	
362	Invalid Enable Write Status Register Command value	
363	Invalid Chip Erase Timeout value	
360	Invalid Block Erase Size value in	
361	Invalid Write Granularity value in	
362	Invalid Enable Write Status Register Command value	
363	Invalid Chip Erase Timeout value	
360	Invalid Block Erase Size value in	
361	Invalid Write Granularity value in	
362	Invalid Enable Write Status Register Command value	

Error Code	Error Message	Response
363	Invalid Chip Erase Timeout value	
440	Invalid Fixed Offset variable name	
441	NVAR invalid variable ID	
442	Param file is already opened	
443	NVAR exists already	
444	Invalid name or Id of NVAR	
445	Invalid length of NVAR value. Check NVAR configuration file for correct length	
446	Password does not match the criteria.	
447	Error occurred while reading NVAR configuration file	
448	Invalid hash certificate file	
449	Valid PID/PPS/Password records are not found in	
450	Invalid Intel® ME Manufacturing Mode Done value entered	
451	Unable to get master base address from the descriptor.	
452	Verification of End Of Manufacturing settings failed	
453	End Of Manufacturing Operation failure - Verification failure on Intel® ME Manufacturing Mode Done settings	
454	End Of Manufacturing Operation failure - Verification failure on Intel® ME Manuf counter.	
455	End Of Manufacturing Operation failure - Verification failure on Descriptor Lock settings.	
456	Invalid hexadecimal value entered for the NVAR	
457	Parsing of file failed	
480	The setup file header has an illegal UUID	
481	The setup file version is unsupported	
482	A record has been encountered that does not contain an entry with the Current Intel® MEBx Password	
483	The given buffer length is invalid	
484	the record chunk count cannot contain all of the setup file record data	
485	the setup file header indicates that there are no valid records (RecordsConsumed >= RecordCount)	
486	the given buffer is invalid	



Error Code	Error Message	Response
487	A record entry with an invalid Module ID was encountered.	
488	A record was encountered with an invalid record number.	
489	The setup file header contains an invalid module ID list.	
490	The setup file header contains an invalid byte count.	
491	The setup file record id is not found	
492	The list of data record entries is invalid.	
493	The CurrentMEBx password is invalid.	
494	The NewMEBx password is invalid.	
495	The PID is invalid.	
496	The PPS is invalid.	
497	The PID checksum failed.	
498	The PPS checksum failed.	
499	The data record is missing a CurrentMEBx password entry.	
500	The data record is missing a NewMEBx password entry.	
501	The data record is missing a PID entry.	
502	The data record is missing a PPS entry.	
503	The header chunk count cannot contain all of the setup file header data.	
504	The requested index is invalid.	
505	Failed to write to the given file.	
506	Failed to read from the given file.	
507	Failed to create random numbers.	
508	The data record is missing a PKI DNS Suffix entry.	
509	The data record is missing a Config Server FQDN entry.	
510	The data record is missing a ZTC entry.	
511	The data record is missing a Pre-Installed Certificate enabled entry.	
512	The data record is missing a User defined certificate config entry.	



Error Code	Error Message	Response
513	The data record is missing a User defined certificate Add entry.	
514	The data record is missing a SOL/IDER enable entry.	
515	OEM Firmware Update Qualifier data missing in USB file.	
1000	Invalid command line option(s)	
1001	Unsupported OS	
8192	General error	
8193	Cannot locate Intel® ME device	
8194	Memory access failure	
8195	Write register failure	
8196	OS failed to allocate memory	
8197	Circular buffer overflow	
8198	Not enough memory in circular buffer	
8199	Communication error between application and Intel® ME <HECI command name>	Contact Intel
8200	Unsupported HECI bus message protocol version	
8201	Unexpected interrupt reason	
8202	Intel® AMT device unavailable	
8203	Unexpected result in command response <HECI command name>	Contact Intel
8204	Unsupported message type	
8205	Cannot find host client	
8206	Cannot find Intel® ME client	
8207	Client already connected	
8208	No free connection available	
8209	Illegal parameter	
8210	Flow control error	
8211	No message	
8212	Requesting HECI receive buffer size is too large	
8213	Application or driver internal error	
8214	Circular buffer not empty	



A.2 Firmware Update Errors

Error Code	Error Message
0	Success
1	An internal error to the AMT device has occurred haltrcfg related
2	Intel® AMT Status is not ready
3	Invalid Intel® AMT Mode
4	An internal error to the Intel® AMT device has occurred
8193	Intel® ME Interface : Cannot locate Intel® ME device driver
8704	Firmware update operation not initiated due to a SKU mismatch
8705	Firmware update not initiated due to version mismatch
8706	Firmware update not initiated due to integrity failure or invalid FW image
8707	Firmware update failed due to an internal error
8708	Firmware Update operation not initiated because a firmware update is already in progress
8710	Firmware update tool failed due to insufficient memory
8713	Firmware update not initiated due to an invalid FW image header
8714	Firmware update not initiated due to file open or read failure
8716	Invalid usage
8718	Update operation timed-out; cannot determine if the operation succeeded
8719	Firmware update cannot be initiated because Local Firmware update is disabled
8722	Intel® ME Interface : Unsupported message type
8723	No Firmware update is happening
8724	Platform did not respond to update request.
8725	Failed to receive last update status from the firmware
8727	Firmware update tool failed to get the firmware parameters
8728	This version of the Intel I® FW Update Tool is not compatible with the current platform.
8741	FW Update Failed.
8743	Unknown or unsupported Platform.
8744	OEM ID verification failed.
8745	Firmware update cannot be initiated because the OEM ID provided is incorrect
8746	Firmware update not initiated due to invalid image length
8747	Firmware update not initiated due to an unavailable global buffer

Error Code	Error Message
8748	Firmware update not initiated due to invalid firmware parameters
8754	Encountered error writing to file.
8757	Display FW Version failed.
8758	The image provided is not supported by the platform.
8759	Internal Error.
8760	Update downgrade vetoed.
8761	Firmware write file failure.
8762	Firmware read file failure.
8763	Firmware delete file failure.
8764	Partition layout NOT compatible.
8765	Downgrade NOT allowed, data mismatched.
8766	Password did not match.
8768	Password Not provided when required.
8769	Polling for FW Update Failed.
8772	Invalid usage, -allowsv switch required to update the same version firmware
8778	Unable to read FW version from file. Please verify the update image used.
8787	Password exceeded maximum number of retries.

A.3 Intel® MEManuf Errors

Error Codes	Error Messages
9248	Intel® ME internal communication error (BIST)
9249	Intel® ME internal communication error (FW)
9250	Used by IBX, not used by CPT, ME8
9251	Fail to create verbose log file %s. Where %s is the log file name user specified.
9252	Used by IBX, not used by CPT, ME8
9254	Used by IBX, not used by CPT, ME8
9255	Internal error
9256	Communication error between host application and Intel® ME FW
9257	Cannot run the command since Intel® AMT is not available



Error Codes	Error Messages
9261	Hibernation isn't supported by the OS, Intel® ME test cannot run
9262	Used by IBX, not used by CPT, ME8
9263	Used by IBX, not used by CPT, ME8
9264	Used by IBX, not used by CPT, ME8
9265	Used by IBX, not used by CPT, ME8
9266	Used by IBX, not used by CPT, ME8
9267	Fail to establish a communication with SPI flash interface
9268	Fail to load vsccommn.bin
9269	Zero flash device found for VSCC check
9270	Fail to load driver (PCI access for Windows*) Tool needs to run with an administrator priviledge account.
9271	Flash ID 0x%06X Intel® ME VSCC mismatch Programmed value of 0x%X doesn't match the recommended value of 0x%X See PCH SPI programming Guide for more details
9272	No recommended ME VSCC value found for flash ID 0x%06X
9273	Intel® VE is disabled by PCH SoftStrap, not used by ME8
9275	Used by IBX, not used by CPT
9276	Fail to read FW Status Register value 0x%X
9277	Intel® VE internal error, not used by ME8
9278	Cannot locate hardware platform identification This program cannot be run on the current platform. Unknown or unsupported hardware platform or A %s hardware platform is detected This program cannot be run on the current platform.

Error Codes	Error Messages
	Unknown or unsupported hardware platform Where %s is the official name of the hardware platform
9279	SPI flash Intel® ME region is not locked
9280	Intel® Gbe/ME has read or write access to BIOS region
9281	SPI flash descriptor region is not locked
9282	BIOS has granted Intel® Gbe and/or Intel® ME access to its region
9283	Region access permissions don't match Intel recommended values
9284	Read firmware flash master region permission failure
9285	Used by IBX, not used by CPT, ME8
9286	Used by IBX, not used by CPT, ME8
9287	Used by IBX, not used by CPT, ME8
9288	Used by IBX, not used by CPT, ME8
9289	Used by IBX, not used by CPT, ME8
9290	Used by IBX, not used by CPT, ME8
9291	Used by IBX, not used by CPT, ME8
9292	The SKU does not have any test assigned to be run -S4 Intel® AMT test only runs under Windows*, not used by ME8
9295	Used by IBX, not used by CPT, ME8
9296	MEINFO Test Failed Or MEINFO End-Of-Line Test Failed Or MEINFO Operation Failed
9297	Intel® NAND needs to be enabled to perform the test, not used by ME8
9298	Used by IBX, not used by CPT
9299	Single flash part found, Flash Partition Boundary Address must be zero



Error Codes	Error Messages
9300	Flash Partition Boundary Address should be in between flash parts
9301	The two flash parts on this platform require different BIOS VSCC values
9302	Intel® NAND module test failed (feature not enabled), not used by ME8
9303	Memory allocation failed for checking variable "<Variable Name>"
9304	Variable "<Variable Name>" mismatch, actual value is - <Variable Value>
9305	Intel® ME firmware version mismatch, actual value is - <Version String> Intel® Gbe version mismatch, actual value is - <Version String> BIOS version mismatch, actual value is - <Version String>
9306	System UUID mismatch, actual value is - <UUID> System UUID mismatch, feature is not supported
9307	Intel® Wired/Wireless LAN MAC address mismatch, feature is not supported Intel® Wired/Wireless LAN MAC address mismatch, actual value is - <MAC Address>
9308	Security Descriptor Override Strap (SDO) is enabled
9309	End-Of-Post message is not sent
9310	Unable to determine Intel® ME Manufacturing Mode status Intel® ME is still in Manufacturing Mode
9311	Intel® ME test failed to start, error 0x%X returned
9312	Intel® ME test timeout (exceeded 30 seconds)
9313	No Intel® ME test result to retrieve, not used by ME8
9314	Intel® ME test result reports error(s), not used by ME8
9315	Intel® ME test is currently running, try again
9316	Intel® ME cannot run Full BIST. Possible Causes: (1) Power package 2 not supported, (2) This is a mobile system with DC power
9317	No valid OEM ICC data programmed
9318	MEINFO End-Of-Line Test config file generation failed
9319	CIRA service button is broken, not used by ME8

Error Codes	Error Messages
9320	Internal error
9321	MEINFO End-Of-Line Test Failed
9322	MEINFO Operation Failed
9324	CM3 results are not available from SPI. Please run –test option to perform the BIST test
9325	Failed to delete CM3 results from SPI
9326	CM3 test failed
9327	CM3 test failed
9328	Internal error
9329	Internal error
9330	Internal error
9331	SMBus hardware is not ready
9332	Internal error
9333	SMBus encountered time-out
9334	Failed to retrieve password from SPI
9335	Internal error
9336	Internal error
9337	Internal error
9338	Failed to retrieve test result from SPI
9339	Failed to retrieve power rule from SPI
9340	Failed to retrieve power source
9341	Failed to retrieve PROC_MISSING_NVAR setting
9342	PROC_MISSING_NVAR setting is set incorrectly
9343	Internal error
9344	Failed to retrieve power package setting
9345	Failed to retrieve CM3 Power Rails Availability setting
9346	CM3 Power Rails Availability setting is set incorrectly



Error Codes	Error Messages
9347	Power source is not AC
9348	Internal error
9349	Internal error
9350	Internal error
9351	Length of OEM Customizable Certificate Friendly Name setting is set incorrectly
9352	OEM Customizable Certificate Stream setting is set incorrectly
9353	OEM Customizable Certificate Hash Algorithm setting is set incorrectly
9354	Length of OEM Customizable Certificate Stream is set incorrectly
9355	Current WLAN does not match micro-code, please update WLAN micro-code in FW
9356	Communication with WLAN device failed
9357	WLAN power well setting is set incorrectly
9358	LAN power well setting is set incorrectly
9359	Power Pkg 2 Supported is set incorrectly
9360	USBr EHCI 1 Enabled and/or USBr EHCI 2 Enabled setting is set incorrectly
9361	KVM device is already in use by other components
9362	Internal error
9363	Internal error
9364	The compressed data is incorrect
9365	Intel integrated LAN setting is set incorrectly
9366	Intel LAN connected Device (PHY) physical connectivity error with ME
9367	Firmware is in recovery mode
9368	SMBus address is not configured correctly
9369	Could not register for SMBus alert
9370	Communication interference
9371	SMBUS connection failed. Check connection or SMBUS address

Error Codes	Error Messages
9372	GPIO connection failed. Check connection or GPIO configuration
9373	NFC Radio – Unknown error
9374	NFC RF Test – Error returned from radio
9375	NFC RF Test – Communication interference or bad response returned from radio
9376	NFC RF Test – Timeout

A.4 Intel® MEINFO Errors

Error Code	Error Messages
9450	Communication error between application and Intel® AMT module (Manageability client)
9451	Communication error between application and Intel® AMT module (PTHI client)
9452	Communication error between application and Intel® ME module (iCLS client)
9455	Failed to read FW Status Register value 0x%X
9457	Failed to create verbose log file %s: Where %s is the log file name user specified
9458	Communication error between application and Intel® ME module (FW Update client)
9459	Internal error (Could not determine FW features information)
9460	Cannot locate hardware platform identification This program cannot be run on the current platform. Unknown or unsupported hardware platform Or A %s hardware platform is detected This program cannot be run on the current platform. Unknown or unsupported hardware platform Where %s is the official name of the hardware platform
9461	Communication error between application and Intel® ME module (HCI client)
9462	Communication error between application and Intel® ME module (Kernel Client)
9467	Cannot use zero as SPI Flash ID index number
9468	Couldn't find a matching SPI Flash ID
9469	Access to SPI Flash device(s) failed
9470	Failed to load driver (PCI access for Windows*) Tool needs to run with an administrator privilege account.
9471	Invalid feature name XXXXX:



Error Code	Error Messages
	Where XXXXX is the feature name
9472	XXXXX feature was not available: Where XXXXX is the feature name
9473	XXXXX actual value is – YYYYY: Where XXXXX is the feature name Where YYYY is the feature value
9474	Error reporting revenue share information – Invalid index used
9475	Error reporting revenue share information – Index already in use
9476	Error reporting revenue share information – Slot is empty

A.5 FPT Errors

Error Code	Error
Invalid Parameters	
200	Invalid parameter value specified by the user. Use -? Option to see help.
Invalid Verbose File	
254	Not able to open the file <FILENAME>.
Unsupported Platform	
201	<EXENAME> cannot be run on the current platform. Please contact your vendor.
Unsupported OS	
9254	Unsupported OS
Commit NVARs Operation	
517	Get NVAR - Read Failed
518	Get NVAR - Invalid NVAR specified
519	Get NVAR - Out of Memory
520	Get NVAR - Blob Integrity Failed
8193	Intel® ME Interface : Cannot locate Intel® ME device driver
8199	Intel® ME Interface : Intel® ME Device not ready for data transmission
8204	Intel® ME Interface : Unsupported message type
8213	Intel® ME Interface : Buffer too small
Compare NVAR(s) Operation	
518	Get NVAR - Invalid NVAR specified

Error Code	Error
519	Get NVAR - Out of Memory
520	Get NVAR - Blob Integrity Failed
8193	Intel® ME Interface : Cannot locate Intel® ME device driver
8199	Intel® ME Interface : Intel® ME Device not ready for data transmission
8204	Intel® ME Interface : Unsupported message type
8213	Intel® ME Interface : Buffer too small
Retrieve NVAR Operation	
518	Get NVAR - Invalid NVAR specified
519	Get NVAR - Out of Memory
520	Get NVAR - Blob Integrity Failed
8193	Intel® ME Interface : Cannot locate Intel® ME device driver
8199	Intel® ME Interface : Intel® ME Device not ready for data transmission
8204	Intel® ME Interface : Unsupported message type
8213	Intel® ME Interface : Buffer too small
Updating Parameters Operations	
493	The Current MEBx Password is invalid.
506	Failed to read from the given file.
3003	Error occurred while opening image file
3004	Parsing of image file failed
3005	Heci communication failed
3006	File does not exist
3007	Operating system is not supported
3008	Intel® AMT Internal error occurred
3009	User defined certificate hash table is full
3010	Unable to start HECI
3011	Invalid input file name
3012	Chipset not supported by the tool
3013	PID value is NULL
3014	PPS value is NULL
3015	Configuration Server FQDN value is NULL
3016	PKI DNS Suffix value is NULL
3017	Host Name value is NULL
3018	Domain Name value is NULL



Error Code	Error
3054	Unable to create Logfile
3055	System failed to retrieve current firmware feature state.
3056	Unable to Save updated parameter as factory defaults on FW image.
3057	Unable to complete NVAR commit option.

§

11 Appendix 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