

Rocket Lake Platform System Tools - Intel® Converged Security and Management Engine Firmware 15.0

User Guide

April 2020

Revision 0.6

Intel Confidential



You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein. You agree to grant Intel a non-exclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at [Intel.com](https://www.intel.com), or from the OEM or retailer.

No computer system can be absolutely secure. Intel does not assume any liability for lost or stolen data or systems or any damages resulting from such losses.

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

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at [intel.com](https://www.intel.com), or from the OEM or retailer.

All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest Intel product specifications and roadmaps.

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

By using this document, in addition to any agreements you have with Intel, you accept the terms set forth below.

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

Intel 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 © 2020, Intel Corporation. All rights reserved.



Contents

1	Introduction	8
1.1	Terminology	8
1.2	Reference Documents	13
2	Preface	14
2.1	Overview	14
2.2	Image Editing Tools	14
2.3	Manufacturing Line Validation Tool	15
2.4	Intel® Management Engine Setting Checker Tool	15
2.5	Operating System Support	16
2.6	Generic System Requirements	16
2.7	Error Return	17
2.8	Usage of Double-Quote Character (")	17
2.9	PMX Driver Limitation	18
2.10	Control Handler Support	18
3	Intel® Flash Image Tool	19
3.1	System Requirements	19
3.2	Flash Image Details	19
3.2.1	Flash Space Allocation	20
3.3	Required Files	20
3.4	Intel® Flash Image Tool	20
3.4.1	Configuration Files	21
3.4.2	Creating New Configuration	21
3.4.3	Opening Existing Configuration	21
3.4.4	Saving Configuration	21
3.4.5	Environment Variables	21
3.4.6	Modifying the Flash Descriptor Region	24
3.4.7	Descriptor Region Length	25
3.4.8	Setting the Number and Size of the Flash Components	25
3.4.9	Region Access Control	26
3.4.10	VSCC Table	30
3.4.11	Adding New Table	30
3.4.12	Removing Existing VSCC Table	31
3.4.13	Modifying the Intel® Management Engine Region	31
3.4.14	Setting the Intel® Management Engine Region Binary File	31
3.4.15	Setting the Intel® PMC Binary File	31
3.4.16	Intel® Management Engine Section	32
3.4.17	Power	32
3.4.18	Power Delivery PD Controller Configuration	33
3.4.19	Manageability Application Section	34
3.4.20	Platform Protection	35
3.4.21	Provisioning Section	36
3.4.22	Gbe (LAN) Region Settings	38
3.4.23	Setting Gbe Region Length Option	38
3.4.24	Setting Gbe Region Binary File	38
3.4.25	Enabling/Disabling GbE Region	38
3.4.26	Modifying PDR Region	39
3.4.27	Setting PDR Region Length Option	39
3.4.28	Setting PDR Region Binary File	39



	3.4.29	Enabling/Disabling PDR Region	39
	3.4.30	Modifying BIOS Region	40
	3.4.31	Setting BIOS Region Length Parameter	40
	3.4.32	Setting the BIOS Region Binary File	40
	3.4.33	Enabling/Disabling the BIOS Region	40
	3.4.34	Building Flash Image	40
	3.4.35	Decomposing Existing Flash Image	41
	3.4.36	Command Line Interface	42
	3.4.37	Example – Decomposing Image and Extracting Parameters	43
	3.4.38	More Examples of FIT CLI	44
4		Flash Programming Tool	45
	4.1	System Requirements	45
	4.2	Flash Image Details	46
	4.3	Microsoft Windows® Required Files	46
	4.4	EFI Required Files	46
	4.5	Programming Flash Device	46
	4.5.1	Stopping Intel® ME SPI Operations	47
	4.6	Programming NVARs	47
	4.7	Usage	48
	4.8	Updating Hash Certificate through NVAR	54
	4.9	Fparts.txt File	56
	4.10	Examples	56
	4.10.1	Complete SPI Flash Device with Binary File	56
	4.10.2	Program Specific Region	57
	4.10.3	Program SPI Flash from Specific Address	58
	4.10.4	Dump Full Image	59
	4.10.5	Dump Specific Region	59
	4.10.6	Display SPI Information	60
	4.10.7	Verify Image with Errors	60
	4.10.8	Verify Image Successfully	61
	4.10.9	Get Intel® ME settings	61
	4.10.10	CVAR Configuration File Generation (-cfggen)	62
5		Intel® MEManuf and MEManufWin	65
	5.1	Windows® PE Requirements	65
	5.2	How to Use Intel® MEManuf	65
	5.3	Usage	66
	5.3.1	Host based Tests	70
	5.4	Intel® MEManuf –EOL Check	70
	5.4.1	ErrorAction Field	70
	5.4.2	MEManuf.xml File	70
	5.4.3	MEManuf –EOL Variable Check	105
	5.4.4	MEManuf –EOL Config Check	105
	5.4.5	Output/Result	106
	5.5	Examples	106
	5.5.1	Example 1	106
6		Intel® MEInfo	110
	6.1	Windows® PE Requirements	110
	6.2	Anti-rollback SVN	110
	6.3	Usage	110
	6.4	Examples	120



6.4.1	Corporate Intel® ME FW SKU.....	120
6.4.2	Retrieve Current Value of Flash Version	124
6.4.3	Checks Whether Computer Has Completed Set-up and Configuration Process.....	125
7	Intel® ME Firmware Update	126
7.1	Requirements	126
7.2	Windows® PE Requirements.....	126
7.3	Enabling and Disabling Intel® FWUpdate	127
7.4	FWUpdate Flows.....	127
7.4.1	Full FWUpdate	127
7.4.2	Partial FWUpdate	127
7.5	Usage.....	128
7.6	Examples	130
7.6.1	Updates Intel® ME with Firmware Binary File	130
7.6.2	Partial Firmware Update	130
7.6.3	Display Supported Commands.....	131
7.6.4	Language Codes.....	131
8	UEFI Sample Application Leveraging FWUpdate API Library	133
8.1	Getting Started - FWUpdate Library	133
8.1.1	Introduction	133
8.1.2	Environment.....	133
8.1.3	Setup	133
8.1.4	Sample App.....	133
8.2	Function Description	144
8.2.1	Get Interfaces.....	144
8.2.2	Get Last Status	144
8.2.3	Check Policy	145
8.2.4	Check Policy Buffer.....	145
8.2.5	Verify OEM Id	146
8.2.6	Get Ipu Partition Attributes	146
8.2.7	Get FW Update Info Status	146
8.2.8	FW Update Query Status Get Response	147
8.2.9	FW Update Full – Using Buffer	147
8.3	FW Update Partial Buffer	148
8.3.1	PDT Data (Sensor Calibration Data) Update.....	149
8.3.2	ISH Firmware Version	149
9	Intel® Manifest Extension Utility (Intel® MEU).....	150
9.1	Usage.....	150
Appendix A	: Intel® ME NVARs	151
Appendix B	: Tool Detail Error Codes	163

Figures

Figure 3-1. SPI Flash Image Regions	19
Figure 3-2. Environment Variables Dialog	22
Figure 3-3. Build Settings Dialog	24
Figure 3-4. FWUpdate image build icon	24



3Figure 3-5. Descriptor Region Length Parameter	25
Figure 3-6. Flash Settings > Flash Components	25
Figure 3-7. Flash Settings → Flash Configuration	26
Figure 3-8. Descriptor Region → Master Access Section	29
Figure3-9. Add VSCC Table Entry Dialog	30
Figure 3-10. Deleting VSCC Table Entry Dialog	31
Figure 3-11. Intel® ME Kernel	32
Figure 3-12. Power.....	33
Figure 3-13. Power Delivery PD Controller Configuration	33
Figure 3-14. Manageability Application Section	35
Figure 3-15. Provisioning Configuration Section	37
Figure 3-16. Provisioning Configuration Section (Cont..)	38
Figure 3-17. GbE Region Options.....	38
Figure 3-18. PDR Region Options	39
Figure 3-19. BIOS Region Parameters	40
Figure 4-1. Raw Hash Values from Certificate File	55
Figure 4-2. Sample Hash.txt File	55

Tables

Table 2-1. OS Support for Tools	16
Table 2-2. Tools Summary	17
Table 3-1. Flash Image Regions – Description	20
Table 3-2. Build Settings Dialog Options.....	23
Table 3-3. Region Access Control Table	26
Table 3-4. CPU/BIOS Access.....	28
Table 3-5. FIT Command Line Options	42
Table 4-1. Named Variables Options	47
Table 4-2. Command Line Options for fpt.efi, fpt.exe and fptw.exe.....	49
Table 4-3. FPT–closemef Behavior	54
Table 4-4. Intel-Recommend Access Settings	54
Table 5-1. Options for Tool	66
Table 5-2. Intel® MEManuf Test Matrix	69
Table 5-3. MEManuf - EOL Config Tests	106
Table 6-1. Intel® MEInfo Command Line Options	111
Table 6-2. List of Components that Intel® MEINFO Displays.....	112
Table 7-1. Image File Update Options	129



Revision History

Revision Number	Description	Date
0.6	Initial Release.	April 2020

§ §



1 Introduction

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

1.1 Terminology

Acronym/Term	Definition
3PDS	3rd Party Data Storage
AC	Alternating Current
Agent	Software that runs on a client PC with OS running
AMT	Intel® AMT
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
BBBS	BIOS Boot Block Size
BIN	Binary file
BIOS	Basic Input Output System
BIOS-FW	Basic Input Output System Firmware
BIST	Built In Self-Test
CCM	Client Control Mode (Host Based Setup and Configuration)
CLI	Command Line Interface
CM0	Intel® ME power state where all HW power planes are activated. Host power state is S0.
CM1	Intel® ME power state where all HW power planes are activated but the host power state is different than S0. (Some host power planes are not activated.) The Host PCI-E* interface is unavailable to the host SW. This power state is not available in Cougar Point.
CM3	Intel® ME power state where all HW power planes are activated but the host power state is different than S0. (Some host power planes are not activated.) The Host PCI-E* interface is unavailable to the host SW. The main memory is not available for Intel® ME use.
CM-Off	No power is applied to the management processor subsystem. Intel® ME is shut down.
CRB	Customer Reference Board
DHCP	Dynamic Host Configuration Protocol
DIMM	Dual In-line Memory Module
DLL	Dynamic Link Library
DNS	Domain Naming System

Acronym/Term	Definition
EC	Embedded Controller
EEPROM	Electrically Erasable Programmable Read Only Memory
EFI	Extensible Firmware Interface
EHCI	Enhanced Host Controller Interface
EID	Endpoint ID
End User	The person who uses the computer (either Desktop or Mobile). In corporate, the user usually does not have administrator privileges. The end user may not be aware to the fact that the platform is managed by Intel® AMT.
EOP	End Of Post
FCIM	Full Clock Integrated Mode
FCSS	Flex Clock Source Select
FDI	Flexible Display Interface
FLOCKDN	Flash Configuration Lock-Down
FMBA	Flash Master Base Address
FOV	Fixed Offset Variable
FPSBA	Flash PCH Strap Base Address
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
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	Hyper Text Transfer Protocol
HW	Hardware
IBEN	Input Buffer Enable

Acronym/Term	Definition
IBV	Independent BIOS Vendor
ICC	Integrated Clock Configuration
ID	Identification
IDER	Integrated Drive Electronics Redirection
INF	An information file (.inf) used by Microsoft operating systems that support the Plug and Play feature. When installing a driver, this file provides the OS with the necessary information about driver filenames, driver components, and supported hardware.
Intel® AMT	The Intel® AMT Firmware running on the embedded processor
Intel® DAL	Intel® Dynamic Application Loader (Intel® DAL)
Intel® FIT	Intel® Flash Image Tool
Intel® FPT	Intel® Flash Programming Tool
Intel® ME	Intel® Management Engine. The embedded processor residing in the chipset PCH.
Intel® MEBx	Intel® Management Engine BIOS Extensions
Intel® MEI driver	Intel® AMT host driver that runs on the host and interfaces between ISV Agent and the Intel® AMT HW.
Intel® MEINFO	Intel® Manageability Engine Information Tool to check whether ME is alive or not.
Intel® MEInfoWin	Windows® version of Intel® Manageability Engine Information Tool
Intel® MEManuf	Intel® Manageability Engine Manufacturing Tool validates Intel® ME functionality on the manufacturing line
Intel® MEManufWin	Windows® version of Intel® Manageability Engine Manufacturing Tool
ISV	Independent Software Vendor
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
MAC address	Media Access Control address
MCP	Multi-Chip Package (Central Processing Unit / Platform Controller Hub)

Acronym/Term	Definition
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
PCH	Peripheral Controller Hub
PCI	Peripheral Component Interconnect
PCIe	Peripheral Component Interconnect Express
PDR	Platform Descriptor Region
PHY	Physical Layer
PID	Provisioning ID
PKI	Public Key Infrastructure
PM	Power Management
PRTC	Protected Real Time Clock
PSK	Pre-Shared Key
PSL	PCH Strap Length
RCFG	Remote Configuration
RCS	Remote Connectivity Service
RNG	Random Number Generator

Acronym/Term	Definition
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.
SW	Software
Sx	All S states which are different than S0
System States	Operating System power states such as S0, S1, S2, S3, S4, and S5.
TCP/IP	Transmission Control Protocol/Internet Protocol.
TLS	Transport Layer Security
UEP	Unified Emulation Partition
UI	User Interface
UIM	User Identifiable Mark
UMA	Unified Memory Access
Un-configured state	The state of the Intel® ME FW when it leaves the OEM factory. At this stage the Intel® ME FW is not functional and must be configured.
UNS	User Notification Services

Acronym/Term	Definition
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
ARB SVN	Anti-Rollback Security Version Number

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 15.0	CDI document
PCH EDS	CDI
Rocket Lake 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, refer Tool's user guides in the other directories in the FW release.

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

- **Rocket Lake Platforms based on Intel® CSME 15.0 PCH** – All tools in the Intel® CSME 15.0 PCH FW release kit are designed for platforms using 11th Generation Intel® Core™ (Rocket Lake) processors and Intel® CSME 15.0 PCH platforms only. These tools will also work with Lewisburg PCH series platforms. These tools do not work properly on any other legacy platforms. Tools designed for other platforms also do not work properly on the 11th Generation Intel® Core™ Processor or Intel® CSME 15.0 PCH platform.

Note: Some of the information in this revision of the System Tools User Guide may be irrelevant to Rocket Lake platforms. In addition, some information relevant to Rocket Lake platforms may be missing. These issues will be resolved in a future revision of the document.

- **Intel® vPro™ platform** – All features listed in this document are available for Intel® vPro™ platforms using Intel® ME 15.0 Corporate firmware. There are some features that are specifically designed for the Intel® vPro™ platform and only work on it.
- **Intel® ME Firmware 15.0 SKU** – A common set of tools is provided for the following Intel® ME FW 15.0 SKUs: Consumer Intel® ME FW SKU and Corporate Intel® ME FW SKU. The following features are only available for Corporate Intel® ME FW SKUs and Consumer Intel® ME FW SKU users should generally ignore them:

Intel® AMT

Intel® ME BIOS Extension (Intel® MEBx)

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

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

2.2 Image Editing Tools

The following tools create and write flash images:

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

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

- **FPT:**
Programs the SPI flash memory of individual regions or the entire flash device.
Modifies some Intel® ME settings (NVAR), FPFs after Intel® ME is flashed on the SPI part.
- **FWUpdate** – updates the Intel® ME FW code region on a flash device that has already been programmed with a complete 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) allows the Intel® ME and Intel® AMT functionality to be tested immediately after the PCH chipset is generated. This tool is designed to be able to run quickly and is generally run on the manufacturing line to do manufacturing testing.

2.4 Intel® Management Engine Setting Checker Tool

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

2.5 Operating System Support

Table 2-1. OS Support for Tools

Intel® ME and Manufacturing Tools	UEFI (64 bit)	Windows® Server 2012 64 With the Latest SP	Windows® 10 DT 64 bit	OSX® (El Capitan / Yosemite)	Windows PE for Windows 10	Linux Kernel 4.1 and Higher
Intel® Flash Programing Tool	x	x	X		x	x
Intel® MEManuf Tool	x	x	x		x	x
Intel® ME Info Tool	x	x	x		x	x
Intel® Firmware Update Tool	x	x	x		x	x
Intel® Manifest Extension Utility Tool		x	x	x		x
Intel® Flash Image Tool		x	x	x		x
Intel® ICC CCT Tool	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. ISH is not supported on MEInfo/ MEManuf for Linux or Windows® Server. Also, Separate ISH tool needs to be used where functionalities are ported from MEInfo and MEManuf tool.
3. Currently the System Tools use the EDK Development Kit.

2.6 Generic System Requirements

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

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

Refer the description of each tool for its exact requirements.

Table 2-2. Tools Summary

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 and Programming NVARs / FPPs	X
Intel® FWUpdate	Updates the FW code while maintaining the previously set values	X

2.7 Error Return

Tools will return errors differently depending on the Operating System the tools run on:

- For Linux:
 - Tools return the error category only.
- For Windows*:
 - The first 8 bits will be the error category, and the rest of the bits will represent the error code.

For example, an error with *error code* = 29 (*i.e.* 0001 1101), and *error category* = 11 (*i.e.* 0000 1011). In Linux, tools will return the Error Category only which is 11. In Windows*, tools will return the combination of ErrorCode and the ErrorCategory (0001 1101 0000 1011) which is 7435.

2.8 Usage of Double-Quote Character (")

The EFI version of the tools handle multi-word argument differently than the Windows® version. If there is a single argument that consists of multiple words delimited 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 and Windows® has a built-in CLI.

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

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

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

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

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

2.9 PMX Driver Limitation

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

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

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

Warning: Running the PMX driver with the Windows® 7 (and higher) driver verifier turned on causes the OS to crash. Do not include PMX as part of the verifier driver list if the user is running Windows® 7 (and higher) with the driver verifier turned on.

2.10 Control Handler Support

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



3 Intel® Flash Image Tool

The Flash Image Tool (**FIT.exe**) creates and configures a complete SPI image file for Intel® CSME 15.0 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)
- IFWI: Intel® ME and PMC
- EC
- Platform Descriptor Region
- ISH

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 make file. 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

The tool does not have to run on an Intel® ME-enabled system.

3.2 Flash Image Details

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

Figure 3-1. SPI Flash Image Regions

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

Table 3-1. Flash Image Regions – Description

Region	Description
Descriptor	<p>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.</p> <p>Note: This region MUST be locked before the serial flash device is shipped to end users. Refer section 3.4.9 below for more information. Failure to lock the Descriptor Region leaves the Intel® ME device vulnerable to security attacks.</p>
IFWI: Intel® ME and PMC	This region contains code and configuration data for Intel® ME applications, such as Intel® AMT technology. It takes up a variable amount of space at the end of the Descriptor.
EC	This contains the Embedded Controller binary used for eSPI.
GbE	This region contains code and configuration data for an Intel Integrated LAN (Gigabit Ethernet). It takes up a variable amount of space at the end of the Intel® ME region.
BIOS	This region contains code and configuration data for the entire computer.
PDR	This region lets system manufacturers describe custom features for the platform.

3.2.1 Flash Space Allocation

Space allocation for each region is determined as follows:

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

3.3 Required Files

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

- vsccommn.bin
- .xml file

3.4 Intel® Flash Image Tool

Refer following for further information:

- General configuration information – Refer FW Bringup Guide from the appropriate Intel® ME FW kit.
- Detailed information on how to configure PCH Soft Straps and VSCC information – Refer Intel® CSME 15.0 PCH SPI Programming Guide and for C620 Lewisburg platforms refer LBG SPI Programming Guide within the kit.

3.4.1 Configuration Files

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

3.4.2 Creating New Configuration

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

3.4.3 Opening Existing Configuration

To open an existing configuration file:

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

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

3.4.4 Saving Configuration

To save the current configuration in an XML file:

Choose File → **Save** or File → **Save As**; the Save File dialog appears if the Configuration has not been given a name or if File → **save as** was chosen.

1. Select the path and enter the file name for the configuration.
2. Click Save.

3.4.5 Environment Variables

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

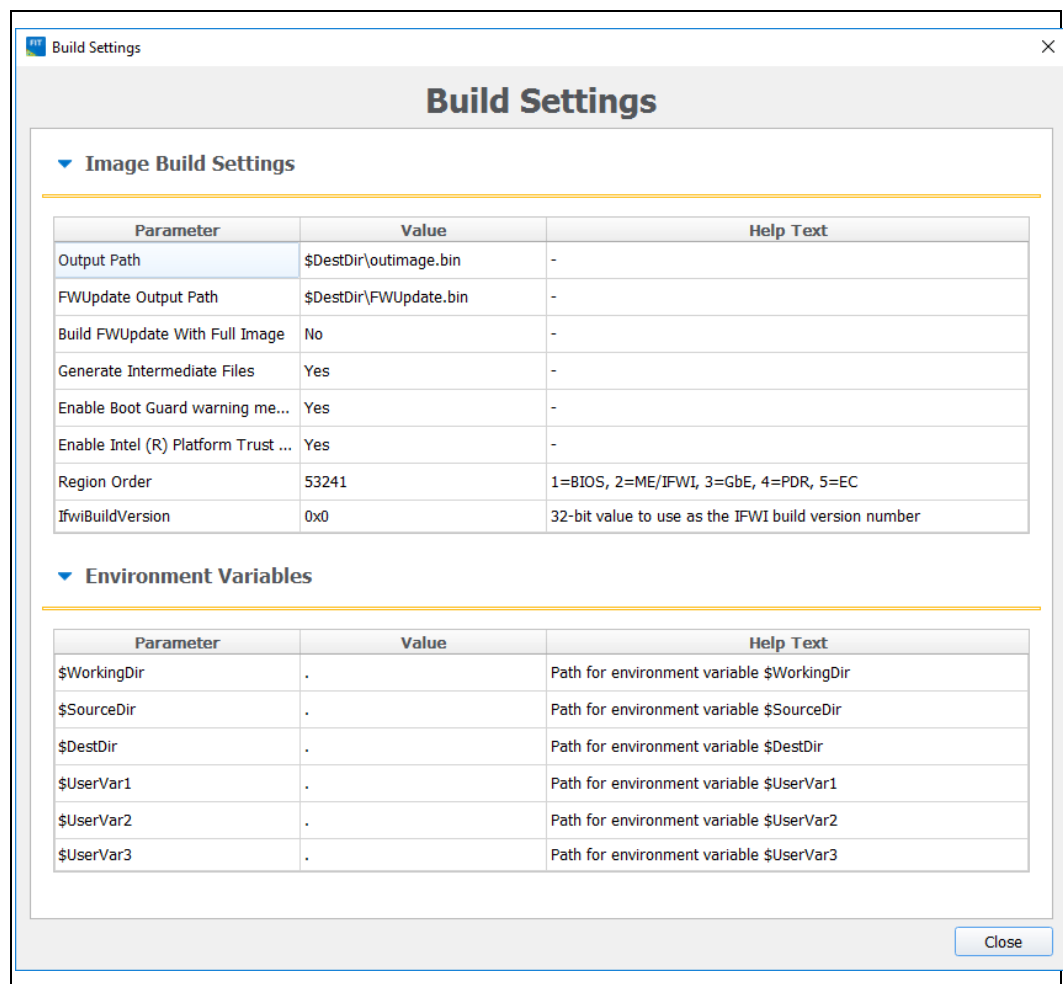
It is recommended that the environment variables be the first thing that the user sets when working with a new configuration. This ensures that FIT can properly substitute environment variables into paths to keep them relative. Doing this also speeds up configuration because many of the **Open File**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


To modify the environment variables:

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

- `$WorkingDir` – the directory functions as a basic path variable when modified in the GUI. If `$WorkingDir` CLI flag is used when launching FIT GUI, then the `fit.log` will be created in `$WorkingDir` directory.
- `$SourceDir` – the directory that contains the base image binary files from which a complete flash image is prepared. Usually these base image binary files are obtained from Intel® VIP on the Web, a BIOS programming resource, or another source.
- `$DestDir` – the directory in which the final combined image is saved, as well as intermediate files generated during the build. Also the directory where the components of an image are stored when an image is decomposed.
- `$UserVar1-3` – used when the above variables are not populated.

Figure 3-2. Environment Variables Dialog



2. Click  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.

Note: Build Settings

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

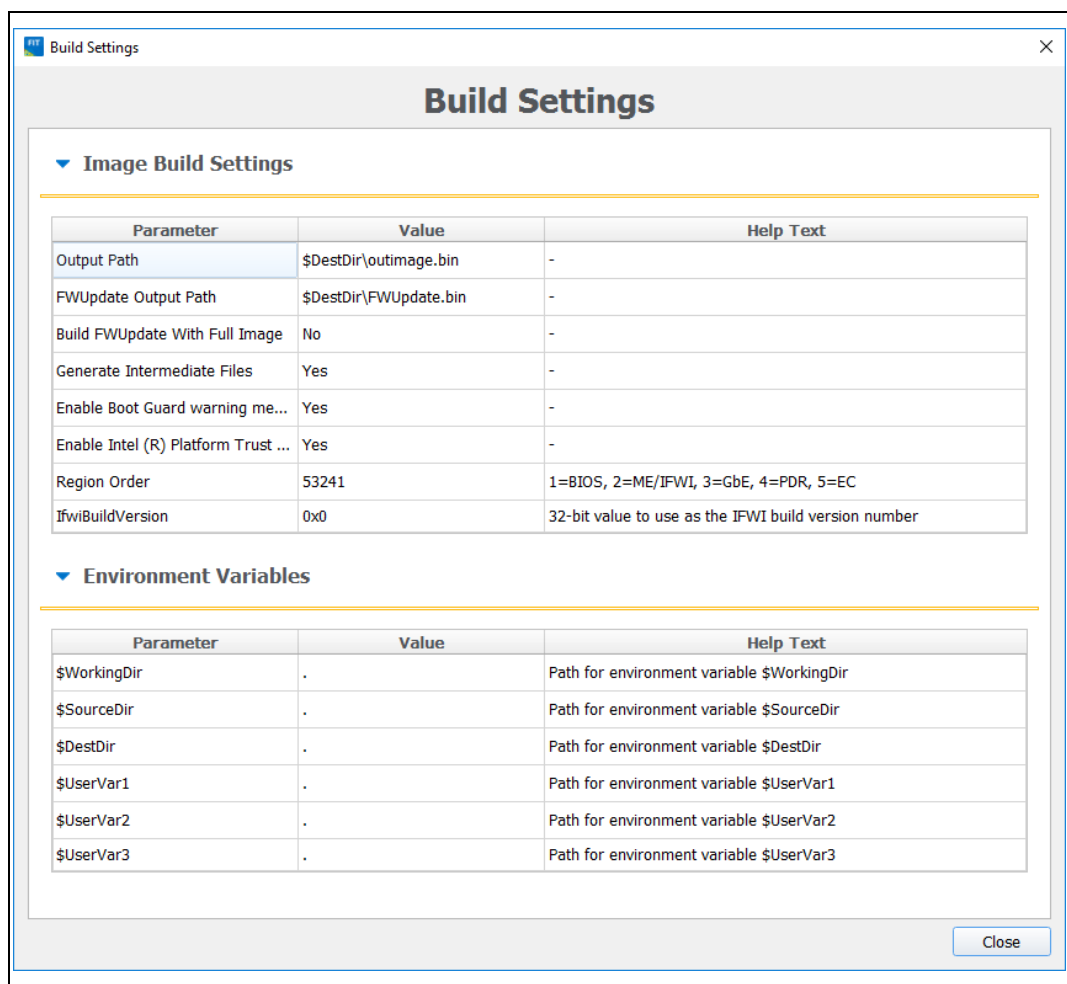
To modify the build setting:

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

Table 3-2. Build Settings Dialog Options

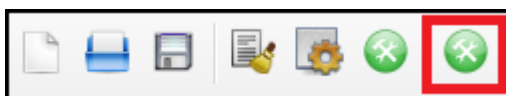
Option	Description
Output path	The path and filename where the final image should be saved after it is built. NOTE: Using the \$DestDir environment variable makes the configuration more portable.
FWUpdate Output Path	The path and filename where the FWUpdate image should be saved after it is built.
Build FWUpdate With Full Image	Allows for the FWUpdate image to be built along with the final full image. If set to no, then a full image will be created only.
Generate intermediate build files.	Causes the application to generate separate (intermediate) binary files for each region, in addition to the final image file (Refer Figure 3). These files are located in the specified output folder's INT subfolder. These image files can be programmed individually with the FPT.
Enable Boot Guard Warning message at build time.	Allows to enable boot guard warning messages at the build time.
Enable Intel® Platform Trust Technology messages at build time.	Allows to enable Intel® Platform Trust Technology warning messages at the build time
CPU Stepping	Which CPU stepping to use.
Environment Variables	

Figure 3-3. Build Settings Dialog



Note: Intel® FIT tool has the ability to build images meant for FWUpdate purposes. To do so, click on the build icon as marked below. This action would build a FWUpdate image only and save it in the earlier defined path in the Build Settings Dialog.

Figure 3-4. FWUpdate image build icon



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.

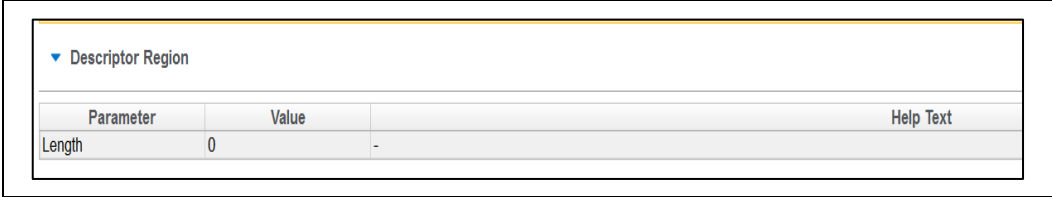
3.4.7 Descriptor Region Length

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**.

3Figure 3-5. Descriptor Region Length Parameter



The screenshot shows a dialog box titled "Descriptor Region". It contains a table with three columns: "Parameter", "Value", and "Help Text". The table has one row with the parameter "Length", a value of "0", and a hyphen "-" in the help text column.

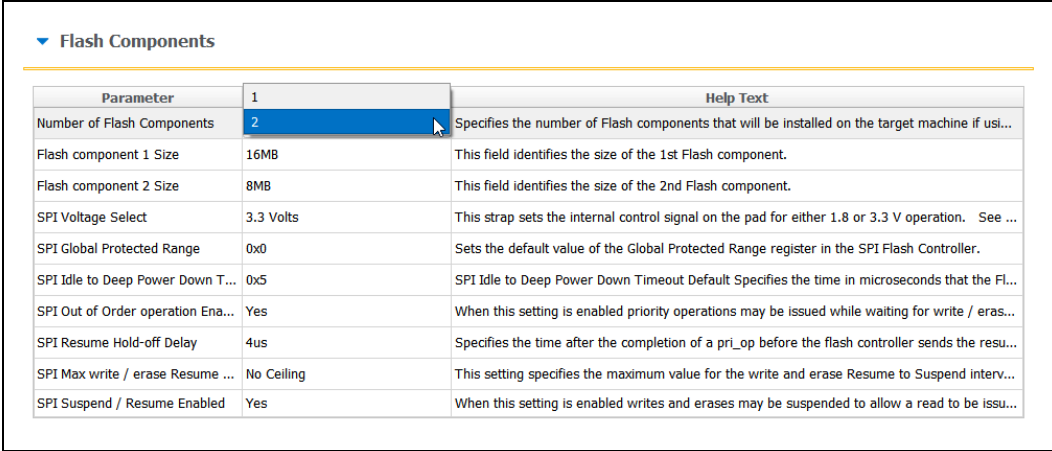
Parameter	Value	Help Text
Length	0	-

3.4.8 Setting the Number and Size of the Flash Components

To set the number of flash components:

1. Select **Flash Settings** in the left pane; expand the Flash Component node in the right pane.
2. Refer [3-](#) all the parameters in the Flash Component section are listed in the right pane.

Figure 3-6. Flash Settings > Flash Components



The screenshot shows a dialog box titled "Flash Components". It contains a table with three columns: "Parameter", "Value", and "Help Text". The "Number of Flash Components" parameter has a dropdown menu open showing the value "2".

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

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

To set the size of each flash component:

1. Double-click on the value of one of these parameters Flash Component 1 Size / Flash Component 2 Size.

2. Select the correct component size from the drop-down list; that parameter is updated.
3. Repeat steps 2-3 for the other parameter.

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

Figure 3-7. 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 3-3. Region Access Control Table

Master Read/Write Access				
Region (#)	CPU and BIOS	ME/PCH	GbE Controller	EC
Descriptor (0)	Not Accessible	Not Accessible	Not Accessible	Not Accessible

Master Read/Write Access				
Region (#)	CPU and BIOS	ME/PCH	GbE Controller	EC
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: <ol style="list-style-type: none"> 1. Descriptor and PDR region is not a master, so they will not have Master R/W access. 2. Descriptor should NOT have write access by any master in production systems. 3. PDR region should only have read and/or write access by CPU/Host. GbE and ME should NOT have access to PDR region. 				

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

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

Key:

0 – Denied access

1 – Allowed access

NC –Bit may be either 0 or 1 since it is unused.

Table 3-4. CPU/BIOS Access

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

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

Example:

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

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

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

To set these access values in FIT:

1. Select **Flash Settings Tab → Host CPU/BIOS Master Access, Intel ME Master Access, Gbe Master Access and EC Master Access** in the right pane; the access parameters are listed in the right pane.
2. Double-click on each parameter and set its access value in one of the following ways:
 - To generate an image for debug purposes or to leave the SPI region open: select 0xFF for both read and write access in all 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 settings for production (e.g., select 0x0C for Intel® ME read access and 0x0D for Intel® ME write access).

Figure 3-8. Descriptor Region → Master Access Section

▼ Host CPU / BIOS Master Access		
Parameter	Value	
Host CPU / BIOS Write ...	0xFFFF	-
Host CPU / BIOS Read ...	0xFFFF	-
▼ Intel(R) ME Master Access		
Parameter	Value	
Intel(R) ME Write Access	0xFFFF	-
Intel(R) ME Read Access	0xFFFF	-
▼ GbE Master Access		
Parameter	Value	
GbE Write Access	0xFFFF	-
GbE Read Access	0xFFFF	-

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, refer Intel® CSME 15.0 PCH-LP SPI Programming Guide, Section 6.4 and for Lewisburg C620 family platform, refer to LBG SPI Programming Guide, Section 4.4.)

VSCC Table can be accessed:

- 1. Select Flash Settings Tab on the left pan
- 2. Expand VSCC Entries on the right pan as shown in [Figure3-9](#) below:

3.4.11 Adding New Table

To add a new table:

- 1. Choose [Add VSCC Entry](#) on top left → **VSCC Entry**.

Figure3-9. Add VSCC Table Entry Dialog

Parameter	Value	Help Text
VscEntryName	Vsc Entry	-
Vendor ID	0x1F	-
Device ID 0	0x47	-
Device ID 1	0x00	-

- 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. ([Figure3-9](#), which shows the parameters of a new VSCC table.)

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

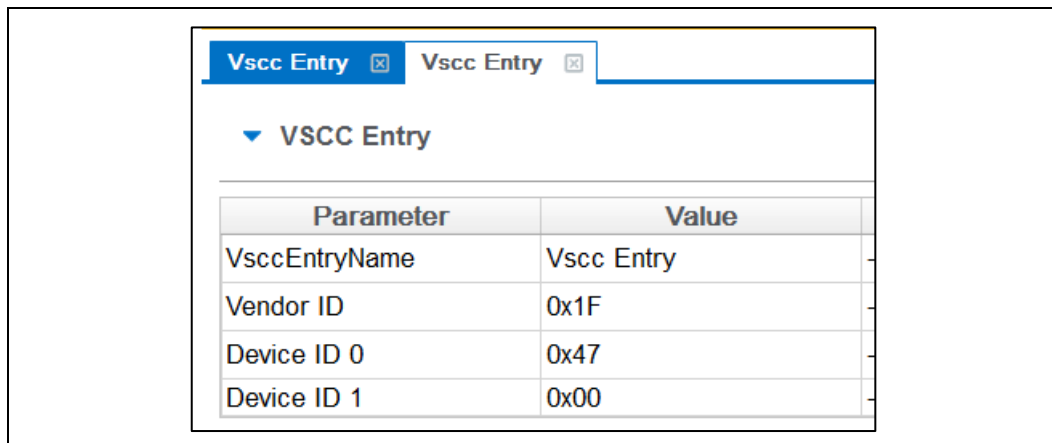
Note: If the descriptor region is being built manually the user will need to reference the VSCC table information for the parts being supported from the manufacturers’ serial flash data sheet. Intel® CSME 15.0 PCH-LP SPI Programming Guide should be used to calculate the VSCC values. For C620 family of workstation systems, use the LBG SPI Programming Guide for further reference concerning the VSCC table definitions.

3.4.12 Removing Existing VSCC Table

To remove an existing table:

1. Click on the name of the table in the top tab that the user wants to remove as shown in Figure 12.

Figure 3-10. Deleting VSCC Table Entry Dialog



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

3.4.13 Modifying the Intel® Management Engine Region

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

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

3.4.14 Setting the Intel® Management Engine 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 **OK** to update the parameter; when the flash image is built, the contents of this file is copied into the Intel® ME Region.

3.4.15 Setting the Intel® PMC Binary File

To select the Intel® PMC binary file:

1. Select the Intel® ME & PMC Region available under Flash Layout tab on the left pane.
2. Double-click on the PMC Binary file parameter in the list; select the Intel® PMC file to be used.

3. Click OK to update the parameter; when the flash image is built, the contents of this file will be merged into the output image generate by the Intel® FIT tool.

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

3.4.16 Intel® Management Engine Section

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

Click on 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 3-11](#).

Figure 3-11. Intel® ME Kernel

The screenshot displays the Intel® ME Kernel configuration interface. It features several expandable sections, each containing a table of parameters with their values and help text.

- Processor**

Parameter	Value	Help Text
Processor Emulation	No Emulation	-
ProcMissing	No onboard glue logic	-
- Intel (R) ME Firmware Update**

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

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

Parameter	Value	Help Text
OEM Tag	0x00000000	-
- MCTP Configuration**

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

Parameter	Value	Help Text
Reserved	No	-

3.4.17 Power

This section describes the platform power configuration settings.

Click on the Power tab on the left pane to configure power parameters.
(Refer Figure 12)

Figure 3-12. 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

3.4.18 Power Delivery PD Controller Configuration

This section resides under the Flex I/O tab. It displays PMC-PD configuration parameters for configuration for each Type-C port supported via PD Controllers per PCH SKU.

Figure 3-13. Power Delivery PD Controller Configuration

▼ Power Delivery PD Controller Configuration

Parameter	Value	
PMC-PD controller USB-C Mode...	PMC / SMBus	This bit defines how the PMC interfaces with the type C components on t
Re-timer Power Gating Enabled	Yes	Indicates whether platform Re-timer power gating is enabled.
Type-C port 1 Enabled	Yes	Indicates whether the associated Type-C port is enabled.
USB2 Port Number associated f...	USB2 Port 4	USB2 port number for the associated Type-C port
USB3 Port Number associated f...	Type-C Port 1	USB3 port number for the associated Type-C port
Type-C Port 1 Re-Timer Present	Yes	Indicates whether a re-timer is present for the associated Type-C port.
Type-C Port 1 Re-timer Configu...	No	Indicates whether the associated re-timer requires configuration. Enable
Type-C Port 1 Re-timer SMBus ...	0x40	SMBus address for the associated re-timer.
Type C Port 1 SMBus Address	0x50	SMBus address for the associated Type-C port.
Type-C port 2 Enabled	Yes	Indicates whether the associated Type-C port is enabled

3.4.19 Manageability Application Section

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

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

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

Figure 3-14. 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	-

3.4.20 Platform Protection

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

Figure 3-15. Platform Protection Section

▶ Boot Guard Configuration

▼ Intel(R) PTT Configuration

Parameter	Value	Help Text
Intel(R) PTT Supported	Yes	This setting permanently disables Intel(R) PTT in the firmware...
Intel(R) PTT initial power-up state	Enabled	-
Intel(R) PTT Supported [FPF]	Yes	This setting will permanently disable Intel(R) PTT through platf...
Intel(R) PTT RPMC Supported	No	This setting determines if RPMC is enabled for Intel(R) PTT. N...
Intel(R) PTT RPMC Rebinding Enabled	No	This setting determines if Rebinding of RPMC enabled SPI part...

▼ TPM Over SPI Bus Configuration

Parameter	Value	Help Text
TPM Clock Frequency	17MHz	This setting determines the clock frequency setting to be used for the TPM over SPI bus.
TPM Over SPI Bus Enabled	Yes	This setting determines the clock frequency setting to be used for the TPM over SPI bus.

These options control the availability and visibility of FW features.

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

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

- Intel® Platform Trusted Technology
- Intel® Content Protection

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

Figure 3-15. 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.

Figure 3-16. Provisioning Configuration Section (Cont..)

▼ OEM Default Certificate 5		
Parameter	Value	Help Text
Certificate Enabled	No	-
Certificate Friendly Name		Enter Hash Name. Maximum of 32 characters.
Certificate Stream		Enter raw hash string or certificate file.

3.4.22 Gbe (LAN) Region Settings

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

Figure 3-17. GbE Region Options

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

3.4.23 Setting Gbe Region Length Option

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

3.4.24 Setting Gbe Region Binary File

To select the Gbe Region binary file:

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

3.4.25 Enabling/Disabling GbE Region

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

To disable the GbE Region:

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

To enable the GbE Region:

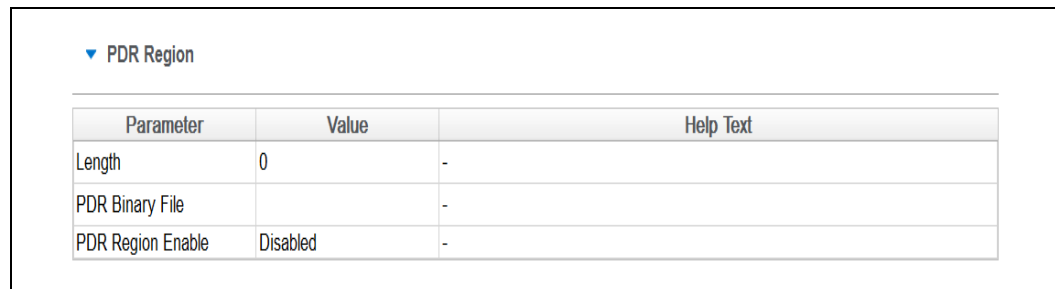
1. Click on Flash Layout tab on the left pane to load the binary file for Gbe region

2. Choose **Enable Region** from the drop down menu.

3.4.26 Modifying PDR Region

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

Figure 3-18. PDR Region Options



▼ PDR Region

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

3.4.27 Setting PDR Region Length Option

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

3.4.28 Setting PDR Region Binary File

To select the PDR region binary file:

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

3.4.29 Enabling/Disabling PDR Region

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

To disable the PDR Region:

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

Note: This region is disabled by default.

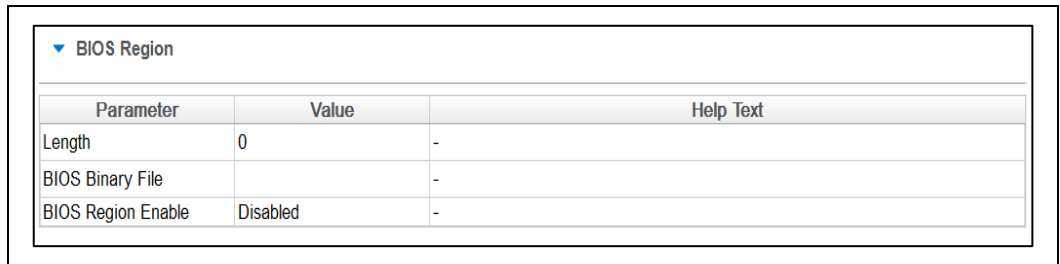
To enable the PDR Region:

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

3.4.30 Modifying BIOS Region

The BIOS Region contains the BIOS code run by the host processor. By placing the BIOS Region at the end there is a chance the system will still boot. It is also important to note that the BIOS binary file is aligned with the end of the BIOS Region so that the reset vector is in the correct place. This means that if the binary file is smaller than the BIOS Region, the region is padded at the beginning instead of at the end.

Figure 3-19. BIOS Region Parameters



The screenshot shows a software window titled "BIOS Region" with a table of parameters. The table has three columns: "Parameter", "Value", and "Help Text". The parameters listed are "Length" with a value of "0", "BIOS Binary File" with a value of "-", and "BIOS Region Enable" with a value of "Disabled".

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

3.4.31 Setting BIOS Region Length Parameter

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

3.4.32 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.

3.4.33 Enabling/Disabling the BIOS Region

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

To disable the BIOS Region:

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

To enable the BIOS Region:

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

3.4.34 Building Flash Image

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

To build a flash image with the currently loaded configuration:

- Choose **Build > Build Image**.
- OR –
- Specify an XML file with the /b option in the command line.

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

- Binary file representing the image
- Text file detailing the various regions in the image
- Optional set of intermediate files (refer 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.

3.4.35 Decomposing Existing Flash Image

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

To decompose an image:

8. Chose **File → Open**.
9. Change the file type filter to the appropriate file type.
10. 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.

3.4.36 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.exe [-exp] [-h|?] [-version|ver] [-b] [-bfwu] [-ofwu] [-o] [-f]
        [-me] [-bios] [-pdr] [-sku] [-ec] [-gbe] [-iunit] [-rombypass]
        [-pmcp] [-ish] [-sd_token] [-iom] [-nphy] [-tbt] [-sam] [-pphy]
        [-oem_km] [-w] [-s] [-d] [-u1] [-u2] [-u3] [-i] [-flashcount]
        [-flashsize1] [-flashsize2] [-save] [-set] [-cloHelp]
```

Table 3-5. FIT Command Line Options

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

Option	Description
-S <path>	Overrides the source file directory environment variable \$SourceDir. It is recommended that the user set these environmental variables before starting a project.
-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.

3.4.37 Example – Decomposing Image and Extracting Parameters

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

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

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

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

The xml file contains the current Intel® ME parameters.

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

3.4.38 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\\ME15.0_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.
- Provision HDCP
- Provided FPF's Access
- Helps doing Closemfn

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

4.1 System Requirements

FPT requires that the platform is bootable (i.e. working BIOS) and has an operating system available to run on. It is designed to deliver a custom image to a computer that is already able to boot and is not a means to get a blank system up and running. FPT must be run on the system with the flash memory to be programmed.

One possible workflow for using FPT is:

1. A pre-programmed flash with a bootable BIOS image is plugged into a new computer.
2. The computer boots.
3. FPT is run and a new BIOS/Intel® ME/GbE image is written to flash.
4. The computer powers down.
5. The computer powers up, boots, and is able to access its Intel® ME/GbE capabilities as well as any new custom BIOS features.

4.2 Flash Image Details

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

4.3 Microsoft Windows® Required Files

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

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

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

Note: Supported drivers are required to reside on the running OS for Intel® FPT tool to function; idrvdll.dll & pmxdll.dll (or with the proper suffix of 32e when running 64bit OS)

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

4.4 EFI Required Files

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

- fparts.txt – contains a comma-separated list of attributes for supported flash devices. The text in the file explains each field. An additional entry may be required in this file to describe the flash part which is on the target system. Examine the target board before adding the appropriate attribute values. The supplied file is already populated with default values for SPI devices used with Intel CRBs.
- fpt.efi – the executable used to program the final image file into the flash. Before running fpt.efi, all the required files must be placed at root directory of the disk otherwise error like "FPT is unable to find FPARTS.TXT "might be displayed.

4.5 Programming Flash Device

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

4.5.1 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 HMRFP0 ENABLE Intel® MEI command to Intel® ME (for more information refer PCH Intel® ME BIOS writer's guide).

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

4.6 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 getFileEx/setfileEx/CommitFiles APIs.

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

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

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

Table 4-1. Named Variables Options

Option	Description
fpt.exe -CLEAR	Overwrites a pending NVAR value update request with the file system's current.
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 setfileEx.

Option	Description
fpt.exe -U -N <NVAR name>	Accept for updating UEP values using SetFile API
fpt.exe -U -IN <Text file>	Accepts cfggen file with values set and will use setfile to update

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

4.7 Usage

The EFI 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 for windows* OS, and the '-h' option for EFI.

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

```
FPTW64.exe [-H|?] [-VER] [-EXP] [-VERBOSE] [-NORESET] [-Y] [-I]
           [-F] [-ERASE] [-VERIFY] [-NOVERIFY] [-D] [-DESC] [-BIOS]
           [-ME] [-GBE] [-PDR] [-EC] [-SAVEMAC] [-SAVESXID] [-B] [-E]
           [-REWRITE] [-ADDRESS|A] [-LENGTH|L] [-CVARS] [-MASTERACCESSGEN]
           [-CFGGEN] [-U] [-CLEAR] [-O] [-IN] [-N] [-V] [-CLOSEMNF] [-GRESET]
           [-PAGE] [-R] [-VARS] [-COMMIT] [-DISABLEME] [-FPFS] [-PROVHDCP]
           [-READHDCP] [-GETPID] [-WRITETOKEN] [-ERASETOKEN] [-PROVKB]
           [-COMMITARBSVN]
```

Table 4-2. 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.
-CLEAR	Overwrites a pending NVAR value update request with the file system's current.
-VER	Shows the version of the tools.
-EXP	Shows examples of how to use the tools.
-VERBOSE [<file>]	Displays the tool's debug information or stores it in a log file.
-Y	Bypasses Prompt. FPT does not prompt user for input. This confirmation will automatically be answered with "y".
-P <file>	Flash parts file. Specifies the alternate flash definition file which contains the flash parts description that FPT has to read. By default, FPT reads the flash parts definitions from fparts.txt.
-LIST	Supported Flash Parts. Displays all supported flash parts. This option reads the contents of the flash parts definition file and displays the contents on the screen.
-I	Info. Displays information about the image currently used in the flash.

Option	Description
-F <file> [NOVERIFY]	Flash. Programs a binary file into an SPI flash. The user needs to specify the binary file to be flashed. FPT reads the binary, and then programs the binary into the flash. After a successful flash, FPT verifies that the SPI flash matches the provided image. Without specify the length with -L option, FPT will use the total SPI size instead of an image size. The NOVERIFY sub-option <i>*must*</i> follow the file name. This will allow flashing the SPI without verifying the programming was done correctly. The user will be prompted before proceeding unless '-y' is used.
-ERASE	Block Erase. Erases all the blocks in a flash. This option does not use the chip erase command but instead erases the SPI flash block by block. This option can be used with a specific region argument to erase that region. This option cannot be used with the -f, -b, -c, -d or -verify options.
-VERIFY <file>	Verify. Compares a binary to the SPI flash. The image file name has to be passed as a command line argument if this flag is specified.
-NOVERIFY	Suboption of -F, see -F for details.
-D <file>	Dump. Reads the SPI flash and dumps the flash contents to a file or to the screen using the STDOUT option. The flash device must be written in 4KB sections. The total size of the flash device must also be in increments of 4KB.
-DESC	Read/Write Descriptor region. Specifies that the Descriptor region is to be read, written, or verified. The start address is the beginning of the region.
-BIOS	Read/Write BIOS region. Specifies that the BIOS region is to be read, written, or verified. Start address is the beginning of the region.
-ME	Read/Write Intel® ME region. Specifies that the Intel® ME region is to be read, written, or verified. The start address is the beginning of the region.
-EC	Read/Write EC region. Specifies that the EC region is to be read, written, or verified. The start address is the beginning of the region.
-GBE	Read/Write GbE region. Specifies that the GbE region is to be read, written, or verified. The start address is the beginning of the region.
-PDR	Read/Write PDR region. Specifies that the PDR region is to be read, written, or verified. The start address is the beginning of the region.
-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.

Option	Description
-B	Blank Check. Checks whether the SPI flash is erased. If the SPI flash is not empty, the application halts as soon as contents are detected. The tool reports the address at which data was found.
-E	Skip Erase. Does not erase blocks before writing. This option skips the erase operation before writing and should be used if the part being flashed is a blank SPI flash device.
-A<value>, -ADDRESS <value>	Write/Read Address. Specifies the start address at which a read, verify, or write operation must be performed. The user needs to provide an address. This option is not used when providing a region since the region dictates the start address.
-L <value>, -LENGTH <value>	Write/Read Length. Specifies the length of data to be read, written, or verified. The user needs to provide the length. This option is not used when providing a region since the region/file length determines this.
-CVARS	Lists all the current manufacturing line configurable variables.
-MASTERACCESSGEN	Generates a Manufacturing Line Configurable Master Access Input File.
-CFGGEN	NVAR Input file generation option. This creates a file which can be used to update the line configurable NVARS.
-U	Update. Updates the NVARS and FPFs in the flash. The user can update 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 <code>-cfggen</code> command). The <code>-in <file></code> option is used to specify the input file.
-O <file>	Output File. The file used by FPT to output NVAR information.
-IN <file>	Input File. The file used by FPT for NVAR input. This option flag must be followed by a text file (i.e., <code>fpt -u -in FPT.cfg</code>). The tool updates the NVARS contained in the text file with the values provided in the input file. User can also use <code>FPT -cfggen</code> 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.

Option	Description
-CLOSEMNF [NO] [PDR] [EC] [BIOS] <file>	<p>End of Manufacturing. This option is executed at the end of manufacturing phase. This option does the following:</p> <p>Sets the Intel® ME manufacturing mode done bit (Global Locked bit).</p> <p>Verifies that the Intel® ME manufacturing mode done bit (Global Locked) is set.</p> <p>Sets the master region access permission in the Descriptor region to its Intel-recommended value</p> <p>Verifies that flash regions are locked.</p> <p>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.</p> <p>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.</p> <p>The PDR, BIOS, EC, or GBE addition will allow CPU\BIOS Read and Write access to the PDR region of flash.</p> <p>It is now supported to run -closemnf in master_access.xml</p> <p>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.</p> <p>Important:</p> <p>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.</p>
-GRESET	Global Reset. FPT performs a global reset.
-PAGE	Pauses the screen when a page of text has been reached. Hit any key to continue.
-SPIBAR	Display SPI BAR. FPT uses this option to display the SPI Base Address Register.
-R <name>	NVAR and FPFs 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	<p>Display Supported Variables. FPT uses this option to display all variables supported for the -R and -COMPARE commands.</p> <p>Note: This will no longer display UEP based values which are tied to configuring iFPF's.</p>

Option	Description
-COMMIT	Commit. FPT uses this option to commit all setfile commands NVARs changes to NVAR and cause relevant reset accordingly. If no pending variable changes are present, Intel® ME does not reset and the tool displays the status of the commit operation.
-HASHED	Hash Variable Output. FPT uses this option to distinguish whether the displayed output is hashed by the FW. For variables that can only be returned in hashed form (such as the Intel® MEBx password), this option has no effect – the data displayed is hashed regardless.
-DISABLEME	Disable the Management Engine.
-COMPAREFPF<name>	Compare the FPF with a value passed in by the user.
-FPFS	Displays a list of the FPFs.
-COMMITFPF <name>	Commits NVAR values to FPF via firmware and prevents further modification of FPFs.
-PROVHDCP <file><file>	Provision platform with the key and cert provided.
-READHDCP	Displays the HDCP Rx provisioning status.
-GETPID <file>	Retrieve the part id.
-REWRITE	Allows to rewrite the SPI with file data even if flash is identical.
-WRITETOKEN <file>	Write the token where the file name is the token name.
-ERASETOKEN	Delete the token.
-PROVKB <iv_and_keybox.bin>	Provision Widevine using IV (Initialization Vector) and encrypted KeyBox file.
-COMMITARBSVN	Commits ARB SVN to FPFs. This would commit the Anti Rollback SVN to the FPFs.

Table 4-3. FPT-closemnf Behavior

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

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

Table 4-4. Intel-Recommend Access Settings

	ME	GBE	BIOS	EC
Read	0b 0000 0000 1101 = 0x00d	0b 0000 0000 1000 = 0x009	0b 0000 000† 000‡ 1011 = 0x0†‡F	0b 0000 0001 0000 00*1 = 0x0101 or 0x0103
Write	0b 0000 0000 1100 = 0x004	0b 0000 0000 1000 = 0x008	0b 000† 000‡ 1010 = 0x†‡A	0b 0000 0001 0000 0x100
Note: 1. ‡ = Value dependent on if PDR is implemented and if Host access is desired. 2. † = Optional BIOS access to the EC region. 3. * = Optional EC Read access to BIOS.				

Notes:

1. Case **A** depends on platform design if optional BIOS access to PDR, add PDR parameter after -closemnf; BIOS Read = 0x1F, BIOS Write = 0x1A.
2. Case **B** depends on platform design if optional BIOS access to the EC region, add EC parameter after -closemnf; BIOS Read = 0x10F, BIOS Write = 0x10A.
3. Case **C** depends on platform design if optional enable EC read access to BIOS, add BIOS parameter after -closemnf; EC Read = 0x103.

4.8 Updating Hash Certificate through NVAR

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

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

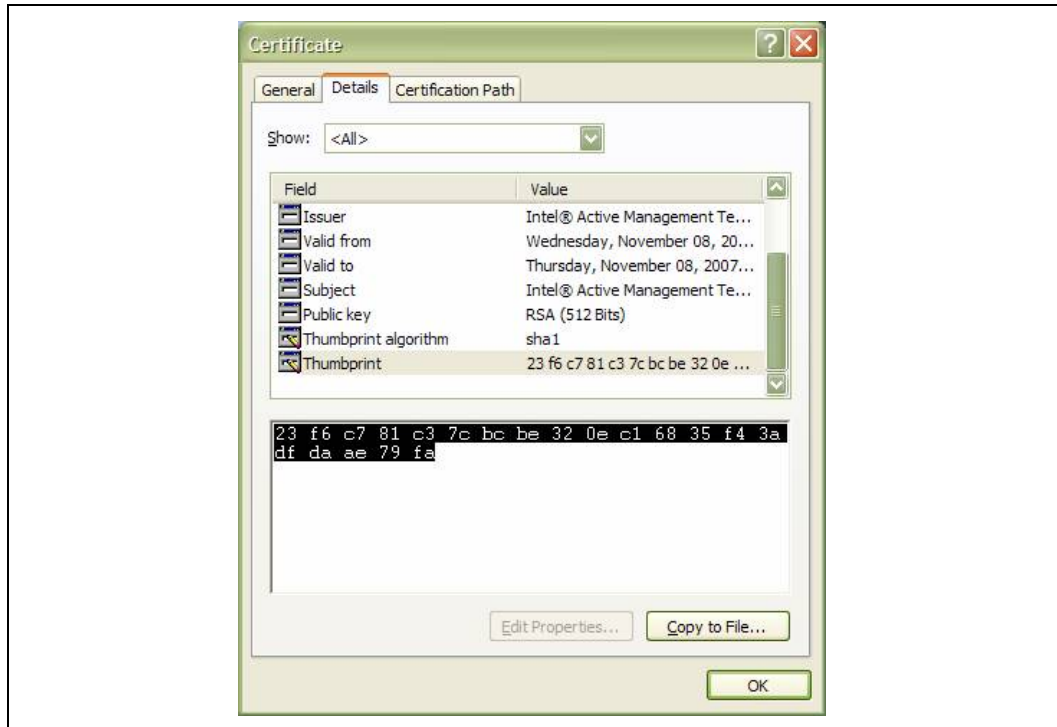
- The OEM Customizable Certificates 1-3 are not default certificates and are deleted after a full un-provisioning.

- The OEM Customizable Certificates 1-3 are configurable by NVAR (with FPT or other flash programming methods) or FIT.

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

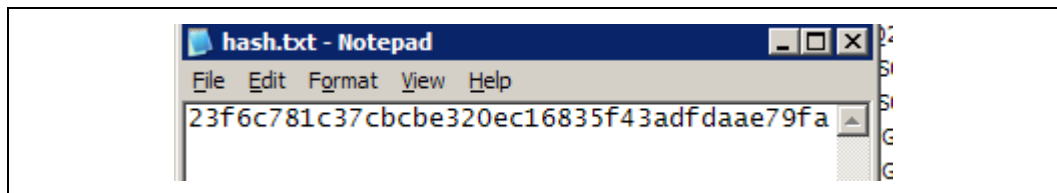
6. Copy the raw hash values from a valid certificate file.

Figure 4-1. Raw Hash Values from Certificate File



7. Paste the raw hash values into a text file
8. Remove all the spaces from the text file.

Figure 4-2. Sample Hash.txt File



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

EXAMPLE:

```
; OEMCustomCert1 Certificate
; All data is required to update the certificate.
; See the Tools Users Guide for detailed explanation
; Of required data and format.
```

```
OEMCustomCert1 IsActive      = 0x01
OEMCustomCert1 FriendlyName = MyCert
OEMCustomCert1 RawHashFile   = 23f6c781c37cbcbe320ec16835f43adfdaae79fa
```

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

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

4.9 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) 2005-2019, 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 Intel® CSME 15.0 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

4.10 Examples

The following examples illustrate the usage of the EFI 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.

4.10.1 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 Menu → CPU CONFIGURATION → BIOS GUARD : Disabled
2. BIOS MENU → INTEL ADVANCED Menu → PCH I/O CONFIGURATION → SECURITY CONFIGURATION → BIOS LOCK : Disabled
3. BIOS MENU → INTEL ADVANCED Menu → 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.

4.10.2 Program Specific Region

```
fpt.exe -f bios.rom -BIOS

EFI:
fpt.efi -f bios.rom -BIOS

Intel (R) Flash Programming Tool Version: xx.x.x.xxxx
Copyright (C) 2005 - 2019, Intel Corporation. All rights reserved.

Reading HSFSTS register... Flash Descriptor: Valid

    --- Flash Devices Found ---
    W25Q256FV    ID:0xEF4019    Size: 32768KB (262144Kb)

Processing Flash memory block 950 from 2559.
- Erasing Flash Block [0x9B7000] - 100 percent complete.
- Programming Flash [0x09B7000] 2332KB of 2332KB - 100 percent
complete.
Processing Flash memory block 1550 from 2559.
- Erasing Flash Block [0xC0F000] - 100 percent complete.
- Programming Flash [0x0C0F000] 1916KB of 1916KB - 100 percent
complete.
Processing Flash memory block 1591 from 2559.
- Erasing Flash Block [0xC38000] - 100 percent complete.
- Programming Flash [0x0C38000] 160KB of 160KB - 100 percent
complete.
Processing Flash memory block 1748 from 2559.
- Erasing Flash Block [0xCD5000] - 100 percent complete.
- Programming Flash [0x0CD5000] 532KB of 532KB - 100 percent
complete.
Processing Flash memory block 1805 from 2559.
- Erasing Flash Block [0xD0E000] - 100 percent complete.
- Programming Flash [0x0D0E000] 188KB of 188KB - 100 percent
complete.
Processing Flash memory block 1816 from 2559.
- Erasing Flash Block [0xD19000] - 100 percent complete.
- Programming Flash [0x0D19000] 36KB of 36KB - 100 percent
complete.
Processing Flash memory block 1908 from 2559.
- Erasing Flash Block [0xD75000] - 100 percent complete.
- Programming Flash [0x0D75000] 344KB of 344KB - 100 percent
complete.
Processing Flash memory block 2042 from 2559.
- Erasing Flash Block [0xDFB000] - 100 percent complete.
- Programming Flash [0x0DFB000] 364KB of 364KB - 100 percent
complete.
```

```

Processing Flash memory block 2324 from 2559.
- Erasing Flash Block [0xF15000] - 100 percent complete.
- Programming Flash [0x0F15000] 596KB of 596KB - 100 percent
complete.
Processing Flash memory block 2540 from 2559.
- Erasing Flash Block [0xFED000] - 100 percent complete.
- Programming Flash [0x0FED000] 52KB of 52KB - 100 percent
complete.
Processing Flash memory block 2559 from 2559.
- Erasing Flash Block [0x1000000] - 100 percent complete.
- Programming Flash [0x1000000] 20KB of 20KB - 100 percent
complete.

RESULT: The data is identical.10240KB of 10240KB - 100 percent
complete.

FPT Operation Successful.

```

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

4.10.3 Program SPI Flash from Specific Address

```

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

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

Intel (R) Flash Programming Tool Version: xx.x.x.xxxx

Copyright (C) 2005 - 2019, Intel Corporation. All rights reserved.

Reading HSFSTS register... Flash Descriptor: Valid

--- Flash Devices Found ---

W25Q256FV ID:0xEF4019 Size: 32768KB (262144Kb)

Warning: Not all of the file data will be written to flash because
the file is longer than the flash area to be written to!
File: "image.bin"
File Length: 16777216
Write Length: 2048.

Do you want to continue? <Y/N>: y

- Reading Flash [0x0001000] 4KB of 4KB - 100 percent complete.
- Erasing Flash Block [0x001000] - 100 percent complete.
- Programming Flash [0x0001000] 4KB of 4KB - 100 percent
complete.

RESULT: The data is identical. 2KB of 2KB - 0 percent
complete.

Flash device was programmed. It is recommended to perform

```

G3 power cycle to complete the flashing process.
FPT Operation Successful.

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.

4.10.4 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) 2005-2019, Intel Corporation. All rights reserved.

Platform: Intel(R) Qxx Express Chipset
Reading HSFSTS register... Flash Descriptor: Valid

--- Flash Devices Found ---
      W25Q256FV      ID: 0xEF4019      Size: 32768KB (262144Kb)

- Reading Flash [0x1000000] 16384KB of 16384KB - 100% complete.
Writing flash contents to file "imagedump.bin"...
Memory Dump Complete

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

FPT Operation Successful
```

4.10.5 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) 2005-2019, Intel Corporation. All rights reserved.

Reading HSFSTS register... Flash Descriptor: Valid

--- Flash Devices Found ---
      W25Q256FV      ID: 0xEF4019      Size: 32768KB (262144Kb)

- Reading Flash [0x0001000]... 4KB of 4KB - 100% complete.
Writing flash contents to file "descdump.bin"...
Memory Dump Complete
```

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

```
FPT Operation Successful
```

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

4.10.6 Display SPI Information

```
fptw.exe -I  
-----  
Intel (R) Flash Programming Tool. Version: XX.X.X.XXXX  
Copyright (c) 2007 - 2017, Intel Corporation. All rights reserved.  
  
Reading HSFSTS register... Flash Descriptor: Valid  
  
    --- Flash Devices Found ---  
    W25Q256FV ID: 0xEF4019      Size: 32768KB (262144Kb)  
  
Warning: There are some addresses that are not defined in any regions.  
Read/Write/Erase operations are not possible on those addresses.  
  
    --- Flash Image Information ---  
    Signature: VALID  
    Number of Flash Components: 1  
        Component 1 - 32768KB (262144Kb)  
    Regions:  
        DESC      - Base: 0x00000000, Limit: 0x00000FFF  
        BIOS      - Base: 0x01183000, Limit: 0x01B82FFF  
        CSME      - Base: 0x00083000, Limit: 0x01082FFF  
        GbE       - Base: 0x00081000, Limit: 0x00082FFF  
        PDR       - Not present  
        EC        - Base: 0x00001000, Limit: 0x00080FFF  
    Master Region Access:  
        BIOS      - ID: Read: 0xFFFF, Write: 0xFFFF  
        CSME      - ID: Read: 0xFFFF, Write: 0xFFFF  
        GbE       - ID: Read: 0xFFFF, Write: 0xFFFF  
        EC        - ID: Read: 0xFFFF, Write: 0xFFFF  
  
Total Accessible SPI Memory: 28172KB, Total Installed SPI Memory: 32768KB  
  
FPT Operation Successful.
```

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

4.10.7 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) 2005-2019, Intel Corporation. All rights reserved.

Reading HSFSTS register... Flash Descriptor: Valid

--- Flash Devices Found ---
      W25Q256FV      ID: 0xEF4019      Size: 32768KB (262144Kb)

-Verifying Flash [0x0000000]      4KB of 16384KB - 0 percent complete
Error 207: Data verify mismatch found.

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

```

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.

4.10.8 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) 2005-2019, Intel Corporation. All rights reserved.
Platform: Intel(R) Qxx Express Chipset
Reading HSFSTS register... Flash Descriptor: Valid
--- Flash Devices Found ---
      W25Q256FV      ID: 0xEF4019      Size: 32768KB (65536Kb)
-Verifying Flash [0x800000] 32768KB of 32768KB - 100% complete.
RESULT: The data is identical.
FPT Operation Successful

```

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.

4.10.9 Get Intel® ME settings

```

fpt.exe -r "Privacy/SecurityLevel"
fpt.efi -r "^"Privacy/SecurityLevel"^^"

-----
Intel (R) Flash Programming Tool. Version:  x.x.x.xxxx
Copyright (c) 2007-2014, Intel Corporation. All rights reserved.
Platform: Intel(R) Qxx Express Chipset
Reading HSFSTS register... Flash Descriptor: Valid

```

```

--- Flash Devices Found ---
W25Q64BV      ID: 0xEF4017      Size: 8192KB (65536Kb)
Variable: "Privacy/SecurityLevel"
Value: True / 01
Retrieve Operation: Successful

```

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

4.10.10 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 initial Power-up State =

        Integrated Sensor Hub Initial Power-up State =

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 =

```

Note: In order to configure Example: FeatureShipState using Cfg file, remove the default value first otherwise Tool will not accept the new values that needs to be configured. Same method applies for OEMSkuRule as well.

Before: FeatureShipState = 0xA1FE7C47

Manageability Application initial Power-up State =

Integrated Sensor Hub Initial Power-up State =

For Configuring, Config file should like:

FeatureShipState =

Manageability Application initial Power-up State = disable.

Integrated Sensor Hub Initial Power-up State =



5 *Intel® MEmanuf and MEmanufWin*

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

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

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

5.1 **Windows® PE Requirements**

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

5.2 **How to Use Intel® MEmanuf**

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

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

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

VSCCMMN.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 Windows® version of the tool can be executed by:

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

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

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

Table 5-1. Options for Tool

Option	Description
No option	<p>There are differences depending on the firmware SKU type the system is running on:</p> <p>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 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.</p> <p>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.</p> <p>If running on a Consumer SKU image, the tool will request the FW to run a complete BIST which does not involve any power transition at the end of the test. Test result will be reported back right after the test is done and cleared.</p> <p>If BIST test result is not displayed after BIST test is done, the tool needs to be run again (with or without any BIST related argument combinations) to retrieve the result, once test result is displayed, it will be cleared.</p> <p>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.</p>
-EXP	Shows examples of how to use the tools.
-H or -?	Displays the help screen.
-VER	Shows the version of the tools.

Option	Description
-S0	The same as No option, except that there is no power reset/hibernation performed at the end of the BIST test including Intel® AMT SKU. The test result is reported back right after the test is done and cleared.
-F <filename>	Load customer defined .cfg file
-TEST	Run full test
-NOWLAN	<p>Note: This option is not applicable for Consumer Intel® ME FW SKU.</p> <p>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"</p> <p>Note:</p> <p>-S0 can only be used on the platform which Intel® AMT is present and can be enabled in the field.</p>
-WLAN	Force wireless LAN test
-BLOCKNET	<p>Note: This option is not applicable for Consumer Intel® ME FW SKU.</p> <p>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.</p>
-ALLOWNET	<p>Note: This option is not applicable for Consumer Intel® ME FW SKU.</p> <p>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.</p>
-BISTRESULT	Returns last BIST results.
-ERRLIST <test name>	Return a list of available codes.

Option	Description
-EOL <Var Config> - F <filename>	<p>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.</p> <p>The Full BIST test for ME15.0 is a combination of M0_HW, Live_HW and M0_Config. The Runtime BIST is a combination of M0_HW and M0_Config.</p> <p>Intel® MEManuf Sx test will require system is capable to enter sleep state, keep ping the platform with network package and keep the system up will make the test failed.</p> <p>Host based Tests</p> <p>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.</p> <p>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.</p> <p>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).</p> <p>Intel® MEManuf -EOL Check.)</p> <p>When -f flag is used along with a file name (<filename>), the tool will load the file as the configuration file, instead of using MEManuf.xml.</p>
-NEXTREBOOT	<p>Upon successful platform reboot CM3 Autotest will be performed.</p> <p>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.</p>
-CFGGEN <filename>	<p>Use this option along with a filename to generate a default configuration file. This file (with or without modification) can be used for the -EOL option. Rename it MEManuf.xml before using it. It is highly recommended to use this option to generate a new MEManuf.xml with an up-to-date variable names list before using the Intel® MEManuf End-Of-Line check feature.</p>
-ALL	<p>Use this option to generate all possible tests for configuration file. All BIST, EOLConfig, and EOLVAR types of tests will be included in the generated XML.</p> <p>Note: Intel recommended tests will be enabled regardless of -all parameter to meet corresponding dependencies</p>
-VERBOSE <file>	<p>Displays the debug information of the tool or stores it in a log file.</p>

Option	Description
-PAGE	When it takes more than one screen to display all the information, this option lets the user pause the display and then press any key to continue on to the next screen.
-NOGFX	This option will skip KVM related test.
-GFX	This option will force KVM related test.
-NOLAN	<p>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.</p> <p>Note: -S0 can only be used on the platform which Intel® AMT is present and can be enabled in the field.</p>
-LAN	This option will force LAN test

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

Table 5-2. Intel® MEManuf Test Matrix

		CM3 Supported SKU	Consumer SKU
BIST Disabled in the ME BOOT	No option	-1st time: Run full BIST test (with 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 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 and host triggered hibernation in Windows® - Save the CM3 test result in SPI .	Run runtime BIST test (with no reset)
	-S0	Run runtime BIST test (with no reset)	Same as CM3 Supported SKU

Note: ICC data check is performed for all options.

Note: The Full BIST test for ME15.0 is a combination of M0_HW, Live_HW and M0_Config. The Runtime BIST is a combination of M0_HW and M0_Config.

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

5.3.1 Host based Tests

1. ME/BIOS VSCC validation, Intel® MEManuf verifies that flash SPI ID on the system is described in VSCC table. If found, VSCC entry for relevant SPI part should match the known good values that pre-populated in the file.
2. Intel® ME state check, Intel® MEManuf verifies Intel® ME is in normal state. This is done by checking the value of 4 fields (initialization state, mode of operation, current operation state, and error state) in FW status register1. If any of these fields indicates Intel® ME is in abnormal state, Intel® MEManuf will report error without running BIST test.
3. ICC data check, Intel® MEManuf verifies that valid OEM ICC data is present and programmed accordingly. This is done by checking FW status register2 ICC bits (which are bit 1 and 2 equal to 3).

5.4 Intel® MEManuf –EOL Check

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

5.4.1 ErrorAction Field

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

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

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

5.4.2 MEManuf.xml File

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

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

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). -->
```

```

<!-- 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="IP Loading - TCSS IPL Tests : TCSS IPL Health Test -
IOM">
    <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
    <!-- Description>TCSS Loading Health Test - IOM</Description -->
    <!-- IntelRequired>True</IntelRequired -->
    <!-- TestType>M0_CONFIG</TestType -->
    <!-- End of uneditable fields -->
    <!-- Please edit the fields below ONLY with the State or ErrAction
-->
    <State>Enabled</State>
    <ErrAction>ErrorContinue</ErrAction>
  </csmebist>
  <csmebist name="Policy Kernel - Boot Guard : Self Test">
    <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
    <!-- Description>Get test result from NVAR
SECURE_BOOT_SELF_TEST_RESULT.</Description -->
    <!-- IntelRequired>True</IntelRequired -->
    <!-- TestType>M0_HW</TestType -->
    <!-- End of uneditable fields -->
    <!-- Please edit the fields below ONLY with the State or ErrAction
-->
    <State>Enabled</State>
    <ErrAction>ErrorContinue</ErrAction>
  </csmebist>
  <csmebist name="Policy Kernel - ME Configuration : PROC_MISSING">
    <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
    <!-- Description>Only on mobile. Test fails if rule is not set to
MEFWCAPS_NO_ONBOARD_GLUE_LOGIC.</Description -->
    <!-- IntelRequired>True</IntelRequired -->
    <!-- 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="VDM - General : VDM engine">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Test VDM.</Description -->
        <!-- IntelRequired>True</IntelRequired -->
        <!-- 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="PAVP - General : Verify Edp and Lspcon
Configurations">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Check if LSPCON and 5K ports are
overlapped</Description -->
        <!-- IntelRequired>True</IntelRequired -->
        <!-- 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="PAVP - General : Set Lspcon Port">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Test the validity of the 5K port
configuration</Description -->
        <!-- IntelRequired>True</IntelRequired -->
        <!-- 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="PAVP - General : Set Edp Port">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Test the validity of the LSPCON port
configuration</Description -->
        <!-- IntelRequired>True</IntelRequired -->
        <!-- 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="Secure boot KM SVN">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Check fpf Secure boot KM SVN against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Hex number with 0x prefix." example="0x00">
</RequiredValue>
    </eolconfig>
    <eolconfig name="Secure boot BSMM SVN">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Check fpf Secure boot BSMM SVN against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Hex number with 0x prefix." example="0x00">
</RequiredValue>
    </eolconfig>
    <eolconfig name="Secure boot ACM SVN">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Check fpf Secure boot ACM SVN against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Hex number with 0x prefix." example="0x00">
</RequiredValue>
    </eolconfig>

```

```

    <eolconfig name="ROT KM SVN">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Check fpf ROT KM SVN against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Hex number with 0x prefix." example="0x00">
</RequiredValue>
    </eolconfig>
    <eolconfig name="PMC SVN">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Check fpf PMC SVN against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Hex number with 0x prefix." example="0x00">
</RequiredValue>
    </eolconfig>
    <eolconfig name="OEM KM SVN">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Check fpf OEM KM SVN against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Hex number with 0x prefix." example="0x00">
</RequiredValue>
    </eolconfig>
    <eolconfig name="IDLM SVN">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->

```

```

        <!-- Description>Check fpf IDLM SVN against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Hex number with 0x prefix." example="0x00">
</RequiredValue>
    </eolconfig>
    <eolconfig name="DNX SVN">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Check fpf DNX SVN against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Hex number with 0x prefix." example="0x00">
</RequiredValue>
    </eolconfig>
    <eolconfig name="USB Port ID">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Check fpf USB Port ID against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Hex number with 0x prefix."
example="0x00000000"> </RequiredValue>
    </eolconfig>
    <eolconfig name="UFS Boot Source">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Check fpf UFS Boot Source against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->

```

```

        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Enabled/00/Disabled/01" example="Enabled">
</RequiredValue>
    </eolconfig>
    <eolconfig name="TXT Supported">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Check fpf TXT Supported against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Disabled/00/Enabled/01" example="Disabled">
</RequiredValue>
    </eolconfig>
    <eolconfig name="Secure boot KM Anti Rollback">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Check fpf Secure boot KM Anti Rollback against
expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Disabled/00/Enabled/01" example="Disabled">
</RequiredValue>
    </eolconfig>
    <eolconfig name="SPI Boot Source">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Check fpf SPI Boot Source against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->

```

```

        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Enabled/00/Disabled/01" example="Enabled">
</RequiredValue>
    </eolconfig>
    <eolconfig name="SOC Config Lock State">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Check fpf SOC Config Lock State against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->

        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Disabled/00/Enabled/01" example="Disabled">
</RequiredValue>
    </eolconfig>
    <eolconfig name="ROT Anti Rollback">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Check fpf ROT Anti Rollback against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->

        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Disabled/00/Enabled/01" example="Disabled">
</RequiredValue>
    </eolconfig>
    <eolconfig name="RBE Anti Rollback">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Check fpf RBE Anti Rollback against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->

        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Disabled/00/Enabled/01" example="Disabled">
</RequiredValue>

```

```

    </eolconfig>
    <eolconfig name="Persistent PRTC Backup Power">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Check fpf Persistent PRTC Backup Power against
expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->

        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Enabled/00/Disabled/01" example="Enabled">
</RequiredValue>
    </eolconfig>
    <eolconfig name="PMC Anti Rollback">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Check fpf PMC Anti Rollback against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->

        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Disabled/00/Enabled/01" example="Disabled">
</RequiredValue>
    </eolconfig>
    <eolconfig name="OEM key Hash RSA key size">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Check fpf OEM key Hash RSA key size against
expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->

        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Disabled/00/Enabled/01" example="Disabled">
</RequiredValue>
    </eolconfig>
    <eolconfig name="Force Boot Guard ACM">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->

```

```

        <!-- Description>Check fpf Force Boot Guard ACM against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Disabled/00/Enabled/01" example="Disabled">
</RequiredValue>
    </eolconfig>
    <eolconfig name="Key Manifest ID">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Check fpf Key Manifest ID against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Hex number with 0x prefix." example="0x00">
</RequiredValue>
    </eolconfig>
    <eolconfig name="OEM Secure Boot Policy">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Check fpf OEM Secure Boot Policy against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Hex number with 0x prefix."
example="0x0000"> </RequiredValue>
    </eolconfig>
    <eolconfig name="OEM Platform ID">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Check fpf OEM Platform ID against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->

```

```

        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Hex number with 0x prefix."
example="0x0000"> </RequiredValue>
    </eolconfig>
    <eolconfig name="OEM Key Revocation State">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Check fpf OEM Key Revocation State against
expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Disabled/00/Enabled/01" example="Disabled">
</RequiredValue>
    </eolconfig>
    <eolconfig name="OEM Key Manifest">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Check fpf OEM Key Manifest against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Disabled/00/Enabled/01" example="Disabled">
</RequiredValue>
    </eolconfig>
    <eolconfig name="OEM KM Anti Rollback">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Check fpf OEM KM Anti Rollback against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->

```



```

        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Disabled/00/Enabled/01" example="Disabled">
</RequiredValue>
    </eolconfig>
    <eolconfig name="OEM ID">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Check fpf OEM ID against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Hex number with 0x prefix."
example="0x0000"> </RequiredValue>
    </eolconfig>
    <eolconfig name="Intel(R) PTT">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Check fpf Intel(R) PTT against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Disabled/00/Enabled/01" example="Disabled">
</RequiredValue>
    </eolconfig>
    <eolconfig name="Intel(R) Manageability HW Fuse Status">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Check fpf Intel(R) Manageability HW Fuse Status
against expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Enabled/00/Disabled/01" example="Enabled">
</RequiredValue>

```

```

    </eolconfig>
    <eolconfig name="IDLM Anti Rollback">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Check fpf IDLM Anti Rollback against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->

        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Disabled/00/Enabled/01" example="Disabled">
</RequiredValue>
    </eolconfig>
    <eolconfig name="Glitch Detection Enabled">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Check fpf Glitch Detection Enabled against
expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->

        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Disabled/00/Enabled/01" example="Disabled">
</RequiredValue>
    </eolconfig>
    <eolconfig name="Glitch Detection Disabled">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Check fpf Glitch Detection Disabled against
expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->

        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Enabled/00/Disabled/01" example="Enabled">
</RequiredValue>
    </eolconfig>
    <eolconfig name="Glitch Detection Cfg Done">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->

```

```

        <!-- Description>Check fpf Glitch Detection Cfg Done against
expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Disabled/00/Enabled/01" example="Disabled">
</RequiredValue>
    </eolconfig>
    <eolconfig name="Flash Descriptor Verification">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Check fpf Flash Descriptor Verification against
expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Disabled/00/Enabled/01" example="Disabled">
</RequiredValue>
    </eolconfig>
    <eolconfig name="Error Enforcement Policy 1">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Check fpf Error Enforcement Policy 1 against
expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Disabled/00/Enabled/01" example="Disabled">
</RequiredValue>
    </eolconfig>
    <eolconfig name="Error Enforcement Policy 0">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Check fpf Error Enforcement Policy 0 against
expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->

```

```

        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Disabled/00/Enabled/01" example="Disabled">
</RequiredValue>
    </eolconfig>
    <eolconfig name="EMMC Boot Source">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Check fpf EMMC Boot Source against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Enabled/00/Disabled/01" example="Enabled">
</RequiredValue>
    </eolconfig>
    <eolconfig name="DNX Anti Rollback">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Check fpf DNX Anti Rollback against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Disabled/00/Enabled/01" example="Disabled">
</RequiredValue>
    </eolconfig>
    <eolconfig name="DAL OEM Signing">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Check fpf DAL OEM Signing against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->

```

```

        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Disabled/00/Enabled/01" example="Disabled">
</RequiredValue>
    </eolconfig>
    <eolconfig name="BSMM Anti Rollback">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Check fpf BSMM Anti Rollback against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->

        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Disabled/00/Enabled/01" example="Disabled">
</RequiredValue>
    </eolconfig>
    <eolconfig name="2nd OEM RSA Key size">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Check fpf 2nd OEM RSA Key size against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->

        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Disabled/00/Enabled/01" example="Disabled">
</RequiredValue>
    </eolconfig>
    <eolconfig name="2nd OEM Public Key Hash">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Check fpf 2nd OEM Public Key Hash against
expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->

        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="48 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 04 AB F3 45 03 1D EF A2 B7 E8 98 79 10
45 AB DE"> </RequiredValue>
    </eolconfig>
    <eolconfig name="2nd OEM Key Hash size">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Check fpf 2nd OEM Key Hash size against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Disabled/00/Enabled/01" example="Disabled">
</RequiredValue>
    </eolconfig>
    <eolconfig name="1st OEM RSA Key size">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Check fpf 1st OEM RSA Key size against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Disabled/00/Enabled/01" example="Disabled">
</RequiredValue>
    </eolconfig>
    <eolconfig name="1st OEM Public Key Hash">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Check fpf 1st OEM Public Key Hash against
expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="48 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 04 AB F3 45 03 1D EF A2 B7 E8 98 79 10
45 AB DE"> </RequiredValue>

```

```

    </eolconfig>
    <eolconfig name="Attestation KeyBox test">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Check Attestation KeyBox data
validity</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->

        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
    </eolconfig>
    <eolconfig name="HW Glitch Detection State">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Check HW Glitch Detection state</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>3</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->

        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Disabled/Enabled" example="Disabled">
</RequiredValue>
    </eolconfig>
    <eolconfig name="Manageability Hardware Support">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Check Manageability Hardware Support</Description
-->

        <!-- IntelRequired>True</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>3</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->

        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
    </eolconfig>
    <eolconfig name="Enforce RPMC Support">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Check if RPMC configuration is
enabled</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>SPI_DEP</Dependencies -->

```

```

        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
    </eolconfig>
    <eolconfig name="PCHC FW version">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Check PCHC FW version against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>PCHC_PARTITION</Dependencies -->
        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="major_ver.minor_ver.hotfix_ver.build_num"
example="1.2.3.0004"> </RequiredValue>
    </eolconfig>
    <eolconfig name="Boot Guard status">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Verifies validity of Boot Guard FW status. As a
RequiredValue provide Profile Level for profile dependent
checks</Description -->
        <!-- IntelRequired>True</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Hex number with 0x prefix." example="0x00">
</RequiredValue>
    </eolconfig>
    <eolconfig name="FW status">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Verifies validity of FW status</Description -->
        <!-- IntelRequired>True</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>

```



```

    </eolconfig>
    <eolconfig name="Checking NVM for fatal flash logs">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Inspection of NVM found fatal flash
logs</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->

        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
    </eolconfig>
    <eolconfig name="Confirm ARB SVN value">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Confirms that the minimum ARB SVN saved in the
PCH fuses matches the ARB SVN of the FW image</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->

        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
    </eolconfig>
    <eolconfig name="PCH Unlocked state">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Verifies that PCH is locked</Description -->
        <!-- IntelRequired>True</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->

        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
    </eolconfig>
    <eolconfig name="HW Binding Disabled">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Verifies that HW binding is disabled</Description
-->

        <!-- IntelRequired>True</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->

```

```

-->      <!-- Please edit the fields below ONLY with the State or ErrAction
-->      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
    </eolconfig>
    <eolconfig name="SOC Config Lock">
      <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
      <!-- Description>Check that SOC Config Lock FPF is
set.</Description -->
      <!-- IntelRequired>True</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>1</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or ErrAction
-->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
    </eolconfig>
    <eolconfig name="FPFs in UEP Committed">
      <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
      <!-- Description>Check that FPFs in UEP are committed to
Hardware.</Description -->
      <!-- IntelRequired>True</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>1</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or ErrAction
-->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
    </eolconfig>
    <eolconfig name="Validate Keybox Provisioning">
      <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
      <!-- Description>Check to see if Keybox is
provisioned</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or ErrAction
-->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
    </eolconfig>
    <eolconfig name="Firmware Update OEM ID">
      <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
      <!-- Description>Check Firmware Update OEM ID value</Description -
->

```

```

        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>3</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="UUID" example="00000000-0000-0000-0000-
000000000000"> </RequiredValue>
        </eolconfig>
        <eolconfig name="GBE version">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Check Gbe Version against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>SPI_DEP</Dependencies -->
        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="major_ver.minor_ver" example="0.6">
</RequiredValue>
        </eolconfig>
        <eolconfig name="BIOS version">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Check BIOS Version against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Customer specific"
example="HSQLPTU1.86C.0117.R00.1303102001"> </RequiredValue>
        </eolconfig>
        <eolconfig name="ME FW version">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Check Firmware Version against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->

```

```

-->      <!-- Please edit the fields below ONLY with the State or ErrAction
-->
-->      <State>Enabled</State>
-->      <ErrAction>ErrorContinue</ErrAction>
-->      <RequiredValue format="major_ver.minor_ver.hotfix_ver.build_num
H|LP|ULT Corporate|Consumer|Slim" example="12.0.0.1040 LP Consumer">
-->      </RequiredValue>
-->      </eolconfig>
-->      <eolconfig name="Security Descriptor Override (SDO) check">
-->          <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
-->          <!-- Description>Check SDO pin</Description -->
-->          <!-- IntelRequired>True</IntelRequired -->
-->          <!-- Dependencies>SPI_DEP</Dependencies -->
-->          <!-- Level>1</Level -->
-->          <!-- End of uneditable fields -->
-->          <!-- Please edit the fields below ONLY with the State or ErrAction
-->
-->      <State>Enabled</State>
-->      <ErrAction>ErrorContinue</ErrAction>
-->      </eolconfig>
-->      <eolconfig name="RPMC Configuration">
-->          <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
-->          <!-- Description>Check RPMC configuration</Description -->
-->          <!-- IntelRequired>True</IntelRequired -->
-->          <!-- Dependencies>SPI_DEP</Dependencies -->
-->          <!-- Level>1</Level -->
-->          <!-- End of uneditable fields -->
-->          <!-- Please edit the fields below ONLY with the State or ErrAction
-->
-->      <State>Enabled</State>
-->      <ErrAction>ErrorContinue</ErrAction>
-->      </eolconfig>
-->      <eolconfig name="EC Write Access Permissions">
-->          <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
-->          <!-- Description>Check EC write access</Description -->
-->          <!-- IntelRequired>True</IntelRequired -->
-->          <!-- Dependencies>SPI_DEP</Dependencies -->
-->          <!-- Level>2</Level -->
-->          <!-- End of uneditable fields -->
-->          <!-- Please edit the fields below ONLY with the State or ErrAction
-->
-->      <State>Enabled</State>
-->      <ErrAction>ErrorContinue</ErrAction>
-->      <RequiredValue format="Hex number with 0x prefix."
example="0x0101. Value left empty will result in checking against Intel
recommended values."> </RequiredValue>
-->      </eolconfig>
-->      <eolconfig name="EC Read Access Permissions">

```

```

        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Check EC read access</Description -->
        <!-- IntelRequired>True</IntelRequired -->
        <!-- Dependencies>SPI_DEP</Dependencies -->
        <!-- Level>2</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Hex number with 0x prefix."
example="0x0101. Value left empty will result in checking against Intel
recommended values."> </RequiredValue>
    </eolconfig>
    <eolconfig name="BIOS Write Access Permissions">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Check BIOS write access</Description -->
        <!-- IntelRequired>True</IntelRequired -->
        <!-- Dependencies>SPI_DEP</Dependencies -->
        <!-- Level>2</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Hex number with 0x prefix."
example="0x0101. Value left empty will result in checking against Intel
recommended values."> </RequiredValue>
    </eolconfig>
    <eolconfig name="BIOS Read Access Permissions">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Check BIOS read access</Description -->
        <!-- IntelRequired>True</IntelRequired -->
        <!-- Dependencies>SPI_DEP</Dependencies -->
        <!-- Level>2</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Hex number with 0x prefix."
example="0x0101. Value left empty will result in checking against Intel
recommended values."> </RequiredValue>
    </eolconfig>
    <eolconfig name="GBE Write Access Permissions">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Check GBE write access</Description -->
        <!-- IntelRequired>True</IntelRequired -->

```

```

        <!-- Dependencies>SPI_DEP</Dependencies -->
        <!-- Level>2</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Hex number with 0x prefix."
example="0x0101. Value left empty will result in checking against Intel
recommended values."> </RequiredValue>
    </eolconfig>
    <eolconfig name="GBE Read Access Permissions">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Check GBE read access</Description -->
        <!-- IntelRequired>True</IntelRequired -->
        <!-- Dependencies>SPI_DEP</Dependencies -->
        <!-- Level>2</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Hex number with 0x prefix."
example="0x0101. Value left empty will result in checking against Intel
recommended values."> </RequiredValue>
    </eolconfig>
    <eolconfig name="ME Write Access Permissions">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Check ME write access</Description -->
        <!-- IntelRequired>True</IntelRequired -->
        <!-- Dependencies>SPI_DEP</Dependencies -->
        <!-- Level>2</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Hex number with 0x prefix."
example="0x0101. Value left empty will result in checking against Intel
recommended values."> </RequiredValue>
    </eolconfig>
    <eolconfig name="ME Read Access Permissions">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Check ME read access</Description -->
        <!-- IntelRequired>True</IntelRequired -->
        <!-- Dependencies>SPI_DEP</Dependencies -->
        <!-- Level>2</Level -->
        <!-- End of uneditable fields -->

```

```

-->      <!-- Please edit the fields below ONLY with the State or ErrAction
-->
-->      <State>Enabled</State>
-->      <ErrAction>ErrorContinue</ErrAction>
-->      <RequiredValue format="Hex number with 0x prefix."
example="0x0101. Value left empty will result in checking against Intel
recommended values."> </RequiredValue>
-->      </eolconfig>
-->      <eolconfig name="ME Manufacturing Mode status">
-->          <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
-->          <!-- Description>Check End of Manufacturing Mode against Intel
recommended value</Description -->
-->          <!-- IntelRequired>True</IntelRequired -->
-->          <!-- Dependencies></Dependencies -->
-->          <!-- Level>1</Level -->
-->          <!-- End of uneditable fields -->
-->          <!-- Please edit the fields below ONLY with the State or ErrAction
-->
-->          <State>Enabled</State>
-->          <ErrAction>ErrorContinue</ErrAction>
-->      </eolconfig>
-->      <eolconfig name="EOP status check">
-->          <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
-->          <!-- Description>Check that EOP was sent/recieved</Description -->
-->          <!-- IntelRequired>True</IntelRequired -->
-->          <!-- Dependencies></Dependencies -->
-->          <!-- Level>1</Level -->
-->          <!-- 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="EOM Settings">
-->          <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
-->          <!-- Description>Test variable against expected value</Description
-->
-->          <!-- IntelRequired>False</IntelRequired -->
-->          <!-- Dependencies></Dependencies -->
-->          <!-- Level>3</Level -->
-->          <!-- End of uneditable fields -->
-->          <!-- Please edit the fields below ONLY with the State or ErrAction
-->
-->          <State>Disabled</State>
-->          <ErrAction>ErrorContinue</ErrAction>
-->          <RequiredValue format="Lock(Flash,Config)/00/Lock(Flash,Config) on
1st Boot/01/Lock(Config)/02/Lock(Config) on 1st

```

```

Boot/03/Lock(Flash)/04/Lock(Flash) on 1st
Boot/05/Lock(none)/06/Lock(none) on 1st Boot/07"
example="Lock(Flash,Config)"> </RequiredValue>
</eolvar>
<eolvar name="SMx State">
  <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
  <!-- Description>Test variable against expected value</Description
-->
  <!-- IntelRequired>False</IntelRequired -->
  <!-- Dependencies></Dependencies -->
  <!-- Level>3</Level -->
  <!-- End of uneditable fields -->
  <!-- Please edit the fields below ONLY with the State or ErrAction
-->
  <State>Disabled</State>
  <ErrAction>ErrorContinue</ErrAction>
  <RequiredValue format="Disabled/01/Enabled/00" example="Disabled">
</RequiredValue>
</eolvar>
<eolvar name="CSME Measured Boot to TPM">
  <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
  <!-- Description>Test variable against expected value</Description
-->
  <!-- IntelRequired>False</IntelRequired -->
  <!-- Dependencies></Dependencies -->
  <!-- Level>3</Level -->
  <!-- End of uneditable fields -->
  <!-- Please edit the fields below ONLY with the State or ErrAction
-->
  <State>Disabled</State>
  <ErrAction>ErrorContinue</ErrAction>
  <RequiredValue format="Disabled/00/Enabled/01" example="Disabled">
</RequiredValue>
</eolvar>
<eolvar name="Intel(R) ME Region Flash Protection Override">
  <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
  <!-- Description>Test variable against expected value</Description
-->
  <!-- IntelRequired>False</IntelRequired -->
  <!-- Dependencies></Dependencies -->
  <!-- Level>3</Level -->
  <!-- End of uneditable fields -->
  <!-- Please edit the fields below ONLY with the State or ErrAction
-->
  <State>Disabled</State>
  <ErrAction>ErrorContinue</ErrAction>
  <RequiredValue format="False/00/True/01" example="False">
</RequiredValue>
</eolvar>

```



```

    <eolvar name="FW Update State">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Test variable against expected value</Description
-->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>3</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Disabled/00/Enabled/01/Full and Partial
disabled/03" example="Disabled"> </RequiredValue>
    </eolvar>
    <eolvar name="Intel(R) ME CLINK Signal">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Test variable against expected value</Description
-->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>3</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Disabled/00/Enabled/01" example="Disabled">
</RequiredValue>
    </eolvar>
    <eolvar name="OEM Tag">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Test variable against expected value</Description
-->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>3</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Hex number with 0x prefix."
example="0x00000000"> </RequiredValue>
    </eolvar>
    <eolvar name="Processor Emulation">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->

```

```

-->      <!-- Description>Test variable against expected value</Description>
-->      <!-- IntelRequired>False</IntelRequired -->
-->      <!-- Dependencies></Dependencies -->
-->      <!-- Level>3</Level -->
-->      <!-- End of uneditable fields -->
-->      <!-- Please edit the fields below ONLY with the State or ErrAction
-->
-->      <State>Disabled</State>
-->      <ErrAction>ErrorContinue</ErrAction>
-->      <RequiredValue format="No
Emulation/00/vPro/01/Core/02/Celerno/03/Pentium/04/Xeon/05/Xeon
Manageability Capable/06" example="No Emulation"> </RequiredValue>
-->      </eolvar>
-->      <eolvar name="PROC_MISSING">
-->          <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
-->          <!-- Description>Test variable against expected value</Description>
-->          <!-- IntelRequired>False</IntelRequired -->
-->          <!-- Dependencies>MOBILE</Dependencies -->
-->          <!-- Level>3</Level -->
-->          <!-- End of uneditable fields -->
-->          <!-- Please edit the fields below ONLY with the State or ErrAction
-->
-->          <State>Disabled</State>
-->          <ErrAction>ErrorContinue</ErrAction>
-->          <RequiredValue format="No onboard glue logic/ff" example="No
onboard glue logic"> </RequiredValue>
-->          </eolvar>
-->          <eolvar name="LAN Power Well">
-->              <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
-->              <!-- Description>Test variable against expected value</Description>
-->              <!-- IntelRequired>False</IntelRequired -->
-->              <!-- Dependencies></Dependencies -->
-->              <!-- Level>3</Level -->
-->              <!-- End of uneditable fields -->
-->              <!-- Please edit the fields below ONLY with the State or ErrAction
-->
-->              <State>Disabled</State>
-->              <ErrAction>ErrorContinue</ErrAction>
-->              <RequiredValue
format="CoreWell/00/SusWell/01/MEWell/02/SLP_LAN# (MGPIO3) /03"
example="CoreWell"> </RequiredValue>
-->              </eolvar>
-->              <eolvar name="Unconfigure On RTC">
-->                  <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
-->                  <!-- Description>Test variable against expected value</Description>
-->

```

```

        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>3</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Enabled/00/Disabled/01" example="Enabled">
</RequiredValue>
    </eolvar>
    <eolvar name="Intel(R) PTT initial power-up state">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Test variable against expected value</Description
-->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>3</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Disabled/00/Enabled/01" example="Disabled">
</RequiredValue>
    </eolvar>
    <eolvar name="FeatureShipState">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Test variable against expected value</Description
-->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>3</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Hex number with 0x prefix."
example="0x00000000"> </RequiredValue>
    </eolvar>
    <eolvar name="Integrated Sensor Hub Supported">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Test variable against expected value</Description
-->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>3</Level -->
        <!-- End of uneditable fields -->

```

```

-->      <!-- Please edit the fields below ONLY with the State or ErrAction
-->      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Disabled/00/Enabled/01" example="Disabled">
</RequiredValue>
    </eolvar>
    <eolvar name="Intel(R) PTT Supported">
      <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
      <!-- Description>Test variable against expected value</Description
-->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or ErrAction
-->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Disabled/00/Enabled/01" example="Disabled">
</RequiredValue>
    </eolvar>
    <eolvar name="Firmware Dynamic Application Loader Supported">
      <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
      <!-- Description>Test variable against expected value</Description
-->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or ErrAction
-->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="No/00/Yes/01" example="No">
</RequiredValue>
    </eolvar>
    <eolvar name="Intel(R) ME Network Services Supported">
      <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
      <!-- Description>Test variable against expected value</Description
-->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or ErrAction
-->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>

```

```

        <RequiredValue format="Yes/00/No/01" example="Yes">
</RequiredValue>
    </eolvar>
    <eolvar name="TLS Supported">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Test variable against expected value</Description
-->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>3</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="No/00/Yes/01" example="No">
</RequiredValue>
    </eolvar>
    <eolvar name="PAVP Supported">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Test variable against expected value</Description
-->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>3</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="No/00/Yes/01" example="No">
</RequiredValue>
    </eolvar>
    <eolvar name="OEMSKURule">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Test variable against expected value</Description
-->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>3</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Hex number with 0x prefix."
example="0x00000000"> </RequiredValue>
    </eolvar>
    <eolvar name="Automatic Built in Self Test">

```

```

        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Test variable against expected value</Description
-->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>3</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Disabled/00/Enabled/01" example="Disabled">
</RequiredValue>
    </eolvar>
    <eolvar name="LSPCON Port Config">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Test variable against expected value</Description
-->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>3</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="0x0/00/0x2/02/0x4/04/0x8/08" example="0x0">
</RequiredValue>
    </eolvar>
    <eolvar name="eDP Port Config">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Test variable against expected value</Description
-->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>3</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue
format="None/None/00/A/None/01/B/None/02/A/B/03/C/None/04/C/A/05/C/B/06"
example="None/None"> </RequiredValue>
    </eolvar>
    <eolvar name="MCTP Device Ports">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->

```

```

-->      <!-- Description>Test variable against expected value</Description>
-->      <!-- IntelRequired>False</IntelRequired -->
-->      <!-- Dependencies></Dependencies -->
-->      <!-- Level>3</Level -->
-->      <!-- End of uneditable fields -->
-->      <!-- Please edit the fields below ONLY with the State or ErrAction
-->
-->      <State>Disabled</State>
-->      <ErrAction>ErrorContinue</ErrAction>
-->      <RequiredValue format="Hex number with 0x prefix."
example="0x00000000"> </RequiredValue>
-->      </eolvar>
-->      <eolvar name="End of Manufacturing Enable">
-->      <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
-->      <!-- Description>Test variable against expected value</Description>
-->
-->      <!-- IntelRequired>False</IntelRequired -->
-->      <!-- Dependencies></Dependencies -->
-->      <!-- Level>3</Level -->
-->      <!-- End of uneditable fields -->
-->      <!-- Please edit the fields below ONLY with the State or ErrAction
-->
-->      <State>Disabled</State>
-->      <ErrAction>ErrorContinue</ErrAction>
-->      <RequiredValue format="No/00/Yes/01" example="No">
</RequiredValue>
-->      </eolvar>
-->      <eolvar name="Delayed Authentication Mode Config">
-->      <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
-->      <!-- Description>Test variable against expected value</Description>
-->
-->      <!-- IntelRequired>False</IntelRequired -->
-->      <!-- Dependencies></Dependencies -->
-->      <!-- Level>3</Level -->
-->      <!-- End of uneditable fields -->
-->      <!-- Please edit the fields below ONLY with the State or ErrAction
-->
-->      <State>Disabled</State>
-->      <ErrAction>ErrorContinue</ErrAction>
-->      <RequiredValue format="Disabled/00/Enabled/01" example="Disabled">
</RequiredValue>
-->      </eolvar>
-->      <eolvar name="Firmware Update OEM ID">
-->      <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
-->      <!-- Description>Test variable against expected value</Description>
-->
-->      <!-- IntelRequired>False</IntelRequired -->
-->      <!-- Dependencies></Dependencies -->

```

```

        <!-- Level>3</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Hex" example="00000000-0000-0000-0000-
000000000000"> </RequiredValue>
    </eolvar>
    <eolvar name="Intel(R) Manageability HW Status">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Test variable against expected value</Description
-->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>3</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Enabled/00/Disabled/01" example="Enabled">
</RequiredValue>
    </eolvar>
    <eolvar name="Debug Override Production Silicon">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Test variable against expected value</Description
-->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>3</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Hex number with 0x prefix."
example="0x00000000"> </RequiredValue>
    </eolvar>
    <eolvar name="Debug Override Pre-Production Silicon">
        <!-- The commented fields below CANNOT be edited. Any edits will
be ignored by the tool -->
        <!-- Description>Test variable against expected value</Description
-->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>3</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State or ErrAction
-->

```



```

        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Hex number with 0x prefix."
example="0x00000000"> </RequiredValue>
    </eolvar>
    <!-- END OF EOL VAR TESTS -->
</memanuf_config>

```

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

To select which test items to run: Modify the State item as <State> Enabled </State> to enable the subtest

Wherever there is a section for Required Value, Example: <RequiredValue format="major_ver.minor_ver" example="0.6"> </RequiredValue>, Please enter the required values in the xml file which will be used by MEManuf for testing.

Here is the example that explain how to use this feature:

```

<eolconfig name="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 -->
    <!-- edit the fields below ONLY with the State or ErrAction -->
    <State>Enabled</State>
    <ErrAction>ErrorContinue</ErrAction>
    <RequiredValue format="major_ver.minor_ver" example="0.6">
</RequiredValue>
</eolconfig>

```

5.4.3 MEManuf –EOL Variable Check

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

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

5.4.4 MEManuf –EOL Config Check

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

Table 5-3. MEManuf - EOL Config Tests

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

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

5.4.5 Output/Result

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

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

5.5 Examples

5.5.1 Example 1

5.5.1.1 Example for Consumer Intel® ME FW SKU

```
MEManuf -verbose
```

```
Intel (R) MEManuf Version: 15.0.0.1202  
Copyright (C) 2005 - 2020, Intel Corporation. All rights reserved.
```

```
LPC Device Id: A087.  
Platform: Rocketlake Platform
```

```

General FW Information
  FW Status Register1      0x90000245
  FW Status Register2      0x62000506
  FW Status Register3      0x00000020
  FW Status Register4      0x00004000
  FW Status Register5      0x00001F03
  FW Status Register6      0x84400308

  Current FW State          Normal
  Flash Partition Table     Valid
  FW Memory State           CM0 with UMA
  FW Initialization         Complete
  BUP Loading state         Success
  FW Error Code             No Error
  FW Mode Of Operation      Normal
  SPI Flash Log             Not Present
  FW Loading Phase          HOSTCOMM Module
  FW Loading Phase Status   UNKNOWN
  ME File System Corrupted  No
  RPMC status               OK
Feature enablement is 0x31309640
ME initialization state valid
ME operation mode valid
Current operation state valid
ME error state valid
MFS is not corrupted
PCH SKU Emulation is correct

Request Intel(R) ME BIST status command... done

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

Get Intel(R) ME test data command... done
Total of 7 Intel(R) ME test result retrieved


Policy Kernel - Boot Guard : Self Test - Passed

VDM - General : VDM engine - Passed

PAVP - General : Verify Edp and Lspcon Configurations - Passed

PAVP - General : Set Lspcon Port - Passed

PAVP - General : Set Edp Port - Passed

IP Loading - TCSS IPL Tests : TCSS IPL Health Test - IOM - Passed

Policy Kernel - ME Configuration : PROC_MISSING - Passed

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


MEManuf Operation Passed

```

5.5.1.2 Example for Corporate Intel® ME FW SKU

Intel (R) MEManuf Version: 15.0.0.1202

Copyright (C) 2005 - 2020, Intel Corporation. All rights reserved.

LPC Device Id: A082.

Platform: Rocketlake Platform

General FW Information

FW Status Register1	0x90000245
FW Status Register2	0x62000506
FW Status Register3	0x000000B0
FW Status Register4	0x00004000
FW Status Register5	0x00001F03
FW Status Register6	0x84400308
Current FW State	Normal
Flash Partition Table	Valid
FW Memory State	CM0 with UMA
FW Initialization	Complete
BUP Loading state	Success
FW Error Code	No Error
FW Mode Of Operation	Normal
SPI Flash Log	Not Present
FW Loading Phase	HOSTCOMM Module
FW Loading Phase Status	UNKNOWN
ME File System Corrupted	No
RPMC status	OK

Found an Intel vPro Wireless device

Feature enablement is 0x7DF6D645

ME initialization state valid

ME operation mode valid

Current operation state valid

ME error state valid

MFS is not corrupted

PCH SKU Emulation is correct

Request Intel(R) ME BIST status command... done

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

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

Total of 19 Intel(R) ME test result retrieved

Policy Kernel - Boot Guard : Self Test - Passed

VDM - General : VDM engine - Passed

USBr - General : Storage - Passed

USBr - General : KVM - Passed

Common Services - LAN : Connectivity to NIC in M0 - Passed

AMT - KVM : Compression engine - Passed

AMT - KVM : Compare engine - Passed

AMT - EC : Basic connectivity - Passed

PAVP - General : Verify Edp and Lspcon Configurations - Passed

PAVP - General : Set Lspcon Port - Passed

PAVP - General : Set Edp Port - Passed

IP Loading - TCSS IPL Tests : TCSS IPL Health Test - IOM - Passed

Policy Kernel - ME Password : Validate MEBx password - Passed

Policy Kernel - ME Configuration : PROC_MISSING - Passed

Common Services - EHBC State : EHBC and Privacy Level states compatibility - Passed

Common Services - EHBC State : Valid Embedded Host Based Configuration (EHBC) state - Passed

Common Services - Privacy Level : Valid Privacy Level settings - Passed

AMT - Power : Valid WLAN power well (Mobile) - Passed

AMT - Power : Valid LAN power well - Passed

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

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® 10.

6.1 Windows® PE Requirements

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

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

6.2 Anti-rollback SVN

The following 32 bit counter HW SVNs were added to harden the ARB security:

- RBE_M_SVN.
- IDLM_M_SVN.
- DNX_M_SVN.
- OEM_KM_SVN.

Note: in order to validate all the previously mentioned SVNs, the appropriate parameter located in Intel® FIT tool > Platform Protection > Intel® FPF Anti-Rollback Configuration needs to be enabled.

6.3 Usage

The executable can be invoked by:

```
MEInfoWin64.exe [-EXP] [-H|?] [-VER] [-FITVER] [-FEAT]
                 [-VALUE] [-FWSTS] [-VERBOSE] [-PAGE]

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

Table 6-1. Intel® MEInfo Command Line Options

Option	Description
-FEAT <name> <column> -VALUE <value>	Compares the value of the given feature name (and optional column name for features displayed in a table) with the value in the command line. If the feature name or value is more than one word, the entire name or value must be enclosed in quotation marks (together with the optional column name). For example -feat "PTT FPF". If the values are identical, a message indicating success appears. If the values are not identical, the actual value of the feature is returned. Only one feature may be requested in a command line.
-FITVER	Displays FIT version information
-FEAT <name> <column>	Retrieves the current value for the specified feature (and optional column name for features displayed in a table). If the feature name is more than one word, the entire feature name (and optional column name) must be enclosed in quotation marks. For example -feat "PTT FPF". The feature name entered must be the same as the feature name displayed by Intel® MEINFO. Intel® MEINFO can retrieve all of the information detailed below. However, depending on the SKU selected, some information may not appear. Note: For the EFI shell version you need to add additional "^" to enclose the text string in order for it to be properly parsed. Example: MEINFO.efi -feat "^"BIOS boot state"^"
-FWSTS	Decodes the Intel® ME FW status register value field and breaks it down into the following bit definitions for easy readability: FW Status Register1: 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
No option:	If the tool is invoked without parameters, it reports information for all components listed in Table 6-2 below for full SKU FW.

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

Feature Name	Feature Data Source (Intel® ME Kernel/ Intel® AMT/SW / Other)	Consumer SKU	Corporate SKU	Specific Feature Dependency	Field Value
Tools Version	SW (Intel® MEInfo)	X	X	N/A	Version string Example: 14.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 14.x.y.ZZZZ; where x=minor, y = HF/MR, ZZZZ = Build Number.
GbE Version	Other (Directly reading from SPI)	X	X	GbE Region to be present in the image	A version string
PMC FW Version	Intel® ME Kernel	X	X	PMC Region to be present in the image	Version string
Descriptor Version	Intel® ME Kernel	X	X	Descriptor Region to be present in the image	Version string
IOMP FW Version	Intel® ME Kernel	X	X	IO Manageability Engine binary version	Version string
MGPP FW Version	Intel® ME Kernel	X	X	MG PHY Binary version	Version string
TBTP FW Version	Intel® ME Kernel	X	X	Thunderbolt™ Binary	Version string
VendorID	Intel® ME Kernel	X	X	N/A	A number (in Hex)

Feature Name	Feature Data Source (Intel® ME Kernel/ Intel® AMT/SW / Other)	Consumer SKU	Corporate SKU	Specific Feature Dependency	Field Value
FW Version	Intel® ME Kernel	X	X	N/A	Version string 14.x.y.ZZZZ A B; where x=minor, y = HF/MR, ZZZZ = Build Number, A=LP/H, B=SKU type [Consumer/Corporate].
LMS version*	Other (Reading Windows® registry entries)	X	X	Only when Windows® LMS driver is installed	A version string
Intel® MEI Driver version*	Other (Reading Windows® registry entries)	X	X	Only when Windows® Intel® MEI driver is installed	A version string
Wireless Driver/ Hardware Version*	Other (Reading Windows® registry entries)	X	X	Only when wireless HW is present, and wireless Windows® driver is installed	A version string
PCH Information	Intel® ME Kernel	X	X	N/A	Display of PCH Information including: Version Device ID Step Data SKU Type Replacement Counter
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.
FW Type	Intel® ME Kernel	X	X	N/A	Pre-Production/Production

Feature Name	Feature Data Source (Intel® ME Kernel/ Intel® AMT/SW / Other)	Consumer SKU	Corporate SKU	Specific Feature Dependency	Field Value
AMT Global State	Intel® ME Kernel	N/A	X	N/A	Enabled/Disabled
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
TCSS FW partial update	Intel® ME Kernel	X	X	FIT TCSS enable feature set to 'Enabled'	Enabled/Disabled
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 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

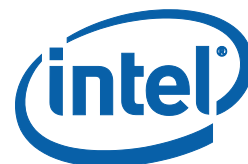
Feature Name	Feature Data Source (Intel® ME Kernel/ Intel® AMT/SW / Other)	Consumer SKU	Corporate SKU	Specific Feature Dependency	Field Value
OEM Id	Intel® ME Kernel	X	X	Only if fw image supports OEM Id	UUID for OEM to check during FW Update
Capability Licensing Service	Intel® ME Kernel	X	X	Not available 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
M3 Autotest	Intel® ME Kernel	X	X	FIT CM3 Autotest Enabled set to 'true'	Enabled/Disabled
C-Link Status	Intel® ME Kernel	X	X	Intel® Wireless LAN	Enabled/Disabled
RPMC Replay Protection	Intel® ME Kernel	X	X	FIT PTT RPMC Supported feature set to 'Yes'	Supported/Not Supported
RPMC Replay Protection Bind Counter	Intel® ME Kernel	X	X	N/A	Counter indicating the number that SPI flash has been rebound
RPMC Replay Protection Bind Status	Intel® ME Kernel	X	X	N/A	Pre-Bind/Bound
RPMC Protection Rebind	Intel® ME Kernel	X	X	FIT PTT RPMC Rebinding Enabled feature set to 'Yes'	Supported/Not Supported
RPMC Replay Protection Max Rebind	Intel® ME Kernel	X	X	N/A	Counter indicating the maximum number of rebinds
Storage Device Type	Intel® ME Kernel	X	X	N/A	SPI/UFS

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

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

Feature Name	Feature Data Source (Intel® ME Kernel/ Intel® AMT/SW / Other)	Consumer SKU	Corporate SKU	Specific Feature Dependency	Field Value
Provisioning Mode	Intel® AMT	N/A	X	Only shown when AMT is enabled	N/A
FWSTS	Intel® ME Kernel	X	X	N/A	Two 32bit Hexadecimal numbers and their bit definition breakdown
Wireless Micro-code Mismatch	FWU	N/A	X	N/A	Yes: FW has detected a ucode mismatch, and partial FWupdate needs to be performed
Wireless LAN in Firmware	FWU	N/A	X	N/A	The “friendly name” matching the WLAN ucode in FW
Wireless Micro-code ID in Firmware	FWU	N/A	X	N/A	The current WLAN ucode in FW
Wireless LAN Hardware	PCI address	N/A	X	N/A	The “friendly name” of the Wireless LAN hardware installed on the system
Wireless Hardware ID	PCI address	N/A	X	N/A	The WLAN DeviceID read from PCI space of the installed WLAN on the system
EPID Group ID					
Keybox					
Intel® PTT Supported	Intel® ME Kernel	X	X	FIT PTT is set to 'Enable'	Yes/No
Intel® PTT initial power-up state	Intel® ME Kernel	X	X	N/A	Enabled/Disabled
PAVP Supported	Intel® ME Kernel	X	X	N/A	Yes/No
ISH initial power state	Intel® ME Kernel	X	X	N/A	Yes/No
End of Manufacturing Enable	Intel® ME Kernel	X	X	Running closemanuf on platform	Yes/No
OEM Public Key Hash FPF	Intel® ME Kernel	X	X	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	X	X	BIOS	SHA-256bit Hash entry
Minimum Allowed Anti Rollback SVN	Intel® ME Kernel	X	X	BIOS	Counter indicating the minimum allowed ARB SVN value
Image Anti Rollback SVN	Intel® ME Kernel	X	X	BIOS	Counter indicating the ARB SVN existing in the FW Image
Trusted Computing Base SVN	Intel® ME Kernel	X	X	BIOS	Counter indicating TCB SVN
Crypto HW Support	Intel® ME Kernel	X	X	BIOS	Enabled/Disabled
BSMM SVN FPF	Intel® ME Kernel	X	X	BIOS	Hash of Public Key to verify Boot Policy Manifest
Protect BIOS Environment	Intel® ME Kernel	X	X	BIOS	Yes / No
CPU Debugging	Intel® ME Kernel	X	X	BIOS	Enabled / Disabled
BSP Initialization	Intel® ME Kernel	X	X	BIOS	Enabled / Disabled
Measured Boot	Intel® ME Kernel	X	X	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
iTouch	SW (Intel® MEInfo)	All	All	iTouch	iTouch information includes: <ul style="list-style-type: none"> • Device ID • HW Revision ID • FW Revision ID • Frame Size • Feedback Size • Sensor Mode • Maximum Number of Touch Point • SPI Frequency SPI I/O Mode



6.4 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.

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

6.4.1 Corporate Intel® ME FW SKU

```

Intel (R) MEInfo Version: 15.0.0.1192
Copyright (C) 2005 - 2020, Intel Corporation. All rights reserved.

General FW Information
FW Status Register1      0x900000245
FW Status Register2      0x690000506
FW Status Register3      0x000000030
FW Status Register4      0x00004004
FW Status Register5      0x00001F03
FW Status Register6      0x84400308

Current FW State          Normal
Flash Partition Table     Valid
FW Memory State           CM0 with UMA
FW Initialization         Complete
BUP Loading state         Success
FW Error Code             No Error
FW Mode Of Operation      Normal
SPI Flash Log             Present
FW Loading Phase          HOSTCOMM Module
FW Loading Phase Status   UNKNOWN
ME File System Corrupted  No
RPMC status               OK

Found an Intel vPro Wireless device

Platform Type             Mobile
FW Image Type             Pre-Production
Last ME Reset Reason      Other
BIOS Boot State           Post Boot
Boot Critical Code Redundancy Disabled
Current Boot Partition    1
Factory Defaults Restoration Status Disabled
BIOS Recovery State       Disabled
CSME Measured Boot to TPM Enabled
Capability Licensing Service State Enabled
Crypto HW Support         Enabled
FW Update State           Enabled
Firmware Update OEM ID    00000000-0000-0000-0000-000000000000
Intel(R) ISH Power State  Enabled
OEM Tag                   0x00
TCSS FW partial update    Disabled
TLS State                 Enabled
Transactional FW Supported No

Intel(R) ME Code Versions
BIOS Version              RKLSFWI1.R00.2527.A14.2002202322
MEBx Version              15.0.0.0001

```


GbE Version	0.3
MEI Driver Version	2012.15.0.1209
FW Version	15.0.0.1192 LP Corporate
LMS Version	2012.15.0.1209
Wireless Hardware Version	2.1.77
Wireless Driver Version	21.80.0.4
IUPs Information	
PMC FW Version	150.1.10.1015
OEM FW Version	15.0.0.7075
ISHC FW Version	5.41.14364.0105
IUNT FW Version	15.0.0.7054
LOCL FW Version	15.0.0.1192
WCOD FW Version	15.0.0.1192
IOM FW Version	16.16.0.4112
NPHY FW Version	11.220.115.1030
TBT FW Version	13.0.0.0604
PCHC FW Version	15.0.0.7009
PCH Information	
PCH Name	RKL
PCH Device ID	A082
PCH Revision ID	B0
PCH SKU Type	Pre-Production ES
PCH Replacement State	Disabled
PCH Replaceable Counter	0
PCH Unlocked State	Disabled
Flash Information	
Storage Device Type	SPI
SPI Flash ID 1	EF4019
RPMC	Unsupported
RPMC Bind Counter	0
RPMC Bind Status	Pre-bind
RPMC Rebind	Unsupported
RPMC Replay Protection Max Rebind	1
BIOS Read Access	0x000F
BIOS Write Access	0x000A
GBE Read Access	0x0009
GBE Write Access	0x0008
ME Read Access	0x000D
ME Write Access	0x0004
EC Read Access	0x0101
EC Write Access	0x0100
FW Capabilities	
	0x7DF6D645
Intel(R) Active Management Technology	Present/Enabled
Intel(R) Protected Audio Video Path	Present/Enabled
Intel(R) Dynamic Application Loader	Present/Enabled
Intel(R) Platform Trust Technology	Present/Enabled
Service Advertisement & Discovery	Present/Enabled
Persistent RTC and Memory	Present/Enabled
End Of Manufacturing	
EOM Settings	Lock(Flash,Config)
NVAR Configuration State	Locked
HW Binding State	Disabled
Flash Protection Mode	Protected
FPF Committed	No

Intel(R) Active Management Technology	
Intel(R) AMT State in FW	Present/Enabled
MAC Address	88-88-88-88-87-88
IPv4 Address	192.168.168.10
IPv6 Enablement	Disabled
Configuration State	Completed
Provisioning Mode	PKI
Auto-BIST State	Disabled
Wired AMT Link Status	Link Up
WLAN uCode Config ID Installed	0xA0F0
Wireless LAN in Firmware	Intel(R) Wi-Fi 6 AX201
WLAN uCode Config ID Required	0xA0F0
Wireless LAN Hardware	Intel(R) Wi-Fi 6 AX201
Localized Language	English
Wireless C-Link Status	Enabled
Intel(R) SMLink0 MCTP Address	0x00
System UUID	4c545a46-3339-3056-b030-3535465a544c
Wireless MAC Address	84-fd-d1-e9-2e-1a
Wireless IPv4 Address	192.168.168.11
Wireless IPv6 Enablement	Disabled
AMT Global State	Enabled
Discrete vPro NIC on-board State	Disabled
Intel(R) Manageability HW Status	Enabled
On Board Discrete vPro NIC SMBus address	0x00
On dock vPro NIC SMBus address	0x00
Redirection Privacy / Security Level	Default
Reseal Timeout	0x06
Seal State	Disabled
Signing Policy	Seal Signing Required
Trusted Device Setup Supported	Disabled
vPRO TBT Dock State	Disabled
vPro TBT I2C Re-timer 1 address	0x20
vPro TBT I2C Re-timer 2 address	0x21
vPro TBT I2C Re-timer 3 address	0x22
vPro TBT I2C Re-timer 4 address	0x23
Intel(R) Protected Audio Video Path	
Widevine provisioning state	Not Provisioned
Attestation KeyBox	Not Provisioned
EPID Group ID	0x4DC
EPID Re-key needed	False
PAVP State	Yes
Security Version Numbers	
Trusted Computing Base SVN	1
Anti Rollback SVN	
PMC	1 [minimum allowed: 1]
CSE	1 [minimum allowed: 1]
ROT KM	1 [minimum allowed: 1]
IDLM	1 [minimum allowed: 1]
SECURE BOOT BSMM	1 [minimum allowed: 1]
OEM KM	1 [minimum allowed: 1]
SECURE BOOT KM	1 [minimum allowed: 1]
SECURE BOOT ACM	1 [minimum allowed: 1]
HW Glitch Detection	
TRC Polarity	0x08 Rising Trans

TRC Mode	Full-cycle polarity trans		
TRC State	Disabled		
Intel(R) Platform Trust Technology			
Intel(R) PTT State	Enabled		
Intel(R) PTT initial power-up state	Enabled		
SMx state	Enabled		
FW Supported FPFs	FPF	UEP	
		*In Use	
	---	---	
1st OEM Key Hash Revoked	Not set	Disabled	# Disabled=0,
Enabled=1			
1st OEM Key Hash size	Not set	Enabled	# Disabled=0,
Enabled=1			
1st OEM RSA Key size	Not set	Enabled	# Disabled=0,
Enabled=1			
2nd OEM Key Hash Revoked	Not set	Disabled	# Disabled=0,
Enabled=1			
2nd OEM Key Hash size	Not set	Enabled	# Disabled=0,
Enabled=1			
2nd OEM RSA Key size	Not set	Disabled	# Disabled=0,
Enabled=1			
BSMM Anti Rollback	Not set	Enabled	# Disabled=0,
Enabled=1			
DAL OEM Signing	Not set	Disabled	# Disabled=0,
Enabled=1			
DNX Anti Rollback	Not set	Enabled	# Disabled=0,
Enabled=1			
Error Enforcement Policy 0	Not set	Disabled	# Disabled=0,
Enabled=1			
Error Enforcement Policy 1	Not set	Disabled	# Disabled=0,
Enabled=1			
Glitch Detection Disabled	Not set	Enabled	# Enabled=0,
Disabled=1			
Glitch Detection Enabled	Not set	Disabled	# Disabled=0,
Enabled=1			
IDLM Anti Rollback	Not set	Enabled	# Disabled=0,
Enabled=1			
Intel PTT Anti Hammering	Not set	0x00	
Intel PTT Encryption Key	Not set	Not Revoked	# Not
Revoked=0, Revoked=1			
Intel(R) Manageability HW Fuse Status	Not set	Enabled	# Enabled=0,
Disabled=1			
Intel(R) PTT	Not set	Enabled	# Disabled=0,
Enabled=1			
OEM ID	Not set	0x00	
OEM KM Anti Rollback	Not set	Enabled	# Disabled=0,
Enabled=1			
OEM Key Manifest	Not set	Enabled	# Disabled=0,
Enabled=1			
OEM Key Revocation State	Not set	Disabled	# Disabled=0,
Enabled=1			
OEM Platform ID	Not set	0x00	
OEM Secure Boot Policy	Not set	0x78	
CPU Debugging	Not set	Enabled	# Enabled=0,
Disabled=1			
BSP Initialization	Not set	Enabled	# Enabled=0,
Disabled=1			

[illegible]

6.4.2 Retrieve Current Value of Flash Version

```
C:\ MEINFO.exe -feat "BIOS boot state"
```

```
Intel(R) MEINFO Version: 15.0.0.xxxx  
Copyright(C) 2005 - 2020, Intel Corporation. All rights reserved.
```

BIOS boot State: Post Boot

```
> MEINFO.efi -feat ""BIOS boot state""  
Intel(R) MEINFO Version: 15.0.0.xxxx  
Copyright(C) 2005 - 2017, Intel Corporation. All rights reserved.
```

BIOS boot State: Post Boot

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

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

```
Intel(R) MEINFO Version: 15.0.0.xxxx  
Copyright(C) 2005 - 2017, Intel Corporation. All rights reserved.
```

Local FWUpdate: Success - Value matches FW value.

```
> MEINFO.efi -feat ""Setup and Configuration"" -value ""Not  
Completed""
```

```
Intel(R) MEINFO Version: 15.0.0.xxxx  
Copyright(C) 2005 - 2017, Intel Corporation. All rights reserved.
```

Local FWUpdate: Success - Value matches FW value.

§ §

7 *Intel® ME Firmware Update*

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

FWUpdate does not update the BIOS, GbE, or Descriptor Regions. It updates the FW code portion along with the WCOD 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.

It is important to note that WCOD & LOCL are part of Intel® CSME and therefore included in the *_base*.bin file.

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 '**ME15.0_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.

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

7.1 Requirements

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

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

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

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

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

7.2 Windows® PE Requirements

In order for tools to work under Windows® PE environment, the user will need to manually load a driver by using the .inf file in the Intel® MEI driver installation files. Once the .inf file located, the user will need to use Windows® PE command `drvload`

*.inf to load it into the running system each time Windows® PE reboots. Failure to do so causes a tools reporting error.

7.3 Enabling and Disabling Intel® FWUpdate

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

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

Disabled – does not allow FW to be updated

Enabled – allows FW to be updated

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

For more details, refer Intel® MEBx user guide.

7.4 FWUpdate Flows

7.4.1 Full FWUpdate

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

Global Reset will be required to complete the FWUpdate operation.

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

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

7.4.2 Partial FWUpdate

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

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

7.5 Usage

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

```
FWUpdLcl64.exe [-H|?] [-VER] [-EXP] [-VERBOSE] [-F] [-Y]
               [-ALLOWSV] [-SILENT] [-FWVER] [-SAVE] [-PARTID] [-INSTID]
               [-FORCERESET] [-OEMID] [-PARTVER] [-PARTVENDOR]

FWUpdLcl.efi  FWUpdLcl64.exe [-H|?] [-VER] [-EXP] [-VERBOSE] [-F] [-Y]
               [-ALLOWSV] [-SILENT] [-FWVER] [-SAVE] [-PARTID] [-INSTID]
               [-FORCERESET] [-OEMID] [-PARTVER] [-PARTVENDOR]
```

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

Table 7-1. Image File Update Options

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

7.6 Examples

7.6.1 Updates Intel® ME with Firmware Binary File

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

This command updates Intel® ME with FW.BIN file. If the firmware on current platform is newer than the 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.

```
FWUpdLcl.exe -f FW.BIN
```

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

7.6.2 Partial Firmware Update

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

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

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

Non-Verbose Mode

```
...\FWUpdLcl.exe -f FW.BIN.bin -partid WCOD
```

```
Intel (R) Firmware Update Utility version 15.0.0.xxxx  
Copyright (C) 2007-2017, Intel Corporation. All rights reserved.
```

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

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

Verbose Mode

```
...\FWUpdLcl.exe -f FW.BIN.bin -partid WCOD -verbose
```

```
Intel (R) Firmware Update Utility version 15.0.0.xxxx  
Copyright (C) 2007-2017, Intel Corporation. All rights reserved.
```

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

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

FW Update is completed successfully.

7.6.3 Display Supported Commands

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

The arguments relevant for this usage are any of the command line options with the prefix '-' 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

```
...\FWUpdLcl.exe -exp partid
```

```
Intel (R) Firmware Update Utility version 15.0.0.xxxx  
Copyright (C) 2007-2017, Intel Corporation. All rights reserved.
```

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

1. F<file> PARTID[<Partition ID>] [FORCERESET] [VERBOSE[<file>]] [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:  
> FWUpdLcl.efi -exp partid
```

```
Intel (R) Firmware Update Utility version 15.0.0.xxxx  
Copyright (C) 2007-2017, Intel Corporation. All rights reserved.
```

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

1. F<file> PARTID[<Partition ID>] [FORCERESET] [VERBOSE[<file>]] [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.

7.6.4 Language Codes

Language	Language Code
English	0x01
French	0x02
German	0x03

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

8 *UEFI Sample Application Leveraging FWUpdate API Library*

8.1 Getting Started - FWUpdate Library

8.1.1 Introduction

This chapter will describe the Firmware Update Libraries that will be used for Intel® Management Engine (Intel® ME) update. It contains a description of the various APIs to be used.

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

8.1.2 Environment

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

8.1.3 Setup

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

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

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

8.1.4 Sample App

The sample code provides you with an example of how to integrate the UEFI FWupdate lib into your BIOS or UEFI application. Error handling, command line processing and loading the update image into memory is left to the customers.

Example – Developing FWUpdate Sample App

Note: Please Refer to the Actual Sample App Source Code under EFI/SampleSource/ Provided in the Kit for Proper Details.

```
/*++
```

```
Copyright (c) 2014-2016 Intel Corporation
```

```
Module Name:
```

```
    FwUpdLcl.c
```

```
Abstract:
```

```
    Sample application demonstrating the usage of the FWU Client UEFI interface
```

```
Revision History
```

```
--*/
```

```
#include "efi.h"
```

```
#include "efilib.h"
```

```
#include "Fwu_Common.h"
```

```
#include "Common.h"
```

```
#include "me_status.h"
```

```
#include "FWUpdateLib.h"
```

```
#include "cse_basic_types.h"
```

```
#include "typedef.h"
```

```
// This function handles the callback from the FWU library for displaying
```

```
// the percentage of completeness of the FW update
```

```
void DisplaySendStatus(float BytesSent, float BytestobeSent)
```

```
{
```

```
    float Value = BytesSent/BytestobeSent * 100;
```

```
    UINT32 pValue = (UINT32)Value;
```

```
    if (pValue != 100)
```

```
    {
```

```
        Print (L"Sending the update image to FW for verification: [ %d%% ]\r",pValue);
```

```
    }else
```

```
    {
```

```
        Print (L"Sending the update image to FW for verification: [ COMPLETE ] \n");
```

```
    }
```

```
}}
```

```
// This is the main entry point for the FW Update application.
```

```
// It handles the initialization of the required libraries and
```

```
// interfaces to the FW Update Library.
```

```
EFI_DRIVER_ENTRY_POINT (InitializeFwUpdLclApplication)
```

```
EFI_STATUS
```

```
InitializeFwUpdLclApplication (
```

```
IN EFI_HANDLE      ImageHandle,
```

```

    IN EFI_SYSTEM_TABLE    *SystemTable
)
{
    EFI_STATUS      Status;
    CHAR16          ImageName[256];
    UINTN           ImageLength = 0;
    UINT8           *ImageBuffer = NULL;
    BOOLEAN         bAllowSV;
    BOOLEAN         bUsePassword;
    BOOLEAN         bPid;
    BOOLEAN         bF;
    BOOLEAN         bPdt;
    BOOLEAN         bIshVer;
    //CHAR          Password[9];
    char            *Password = NULL;
    UINT32          FWUpdateStatus;
    DWORD           loops = 500;
    BOOLEAN         done = FALSE;
    UINT32          lastStatus = 0;
    UINT32          platCheck = 0;
    FWVersion       fwVersion;
    INT32           platCheckReturn = 0;
    UINT32          CheckPolicyStatus = 0;
    UPDATE_TYPE     Upd_Type;
    VersionLib      ver;
    UINT32          index = 0;
    UINT32          status;
    UINT32          UpdateStatus = 0;
    UINT32          TotalStages = 0;
    UINT32          PercentWritten = 0;
    CHAR8           symbol;
    UINT32          lastResetType;
    UPDATE_FLAGS_LIB update_flags;
    UINT16          interfaces;
    int             timer30s = 0;
    unsigned int    indexMod;
    int             percentage0s = 0;
    int             percentdiff = 0;
    UINT32          ComparePartID = 0;
    UINT32          hexValueInstId = 0;
    IPU_UPDATED_INFO IpuUpdatedInfo;
    UINT32          PartId = 0;
    UINT32          fwuError;
    FWU_GET_IPU_PT_ATTRB_MSG_REPLY FwuGetIpuAttrbMsgInfo;
    bool            found = false;
    UINT32          i, j = 0;
    UINT16 major = 0;
    UINT16 minor = 0;
    UINT16 hotfix = 0;
    UINT16 build = 0;
    UINT32 itr = 0;

    // Zero out the update flag structure
    ZeroMem(&update_flags, sizeof(UPDATE_FLAGS_LIB));
    ZeroMem((char*)&IpuUpdatedInfo, sizeof(IPU_UPDATED_INFO));

```

```

// Initialize the EFI Toolkit Library. Set BS, RT, &ST globals
// BS = Boot Services RT = RunTime Services
// ST = System Table

InitializeLib (ImageHandle, SystemTable);

Print (L"\n Intel (R) Firmware Update Utility Sample Application \n");
Print (L"\n Intel (R) Firmware Update Utility Version: %d.%d.%d.", VER_MAJOR,
VER_MINOR, VER_HOTFIX);
Print (L"%d\n", VER_BUILD);

Print (L"\n");

Print (ID_INFO_1);

// Determine the command line arguments

Status = ParseCommandLine (ImageHandle, ImageName, &bAllowSV,
&bUsePassword, &bPid, &bF, &bPdt, &bIshVer);
if (EFI_ERROR (Status))
{
    DEBUG ((D_ERROR, "Unable to process command line - %r\n", Status));
    return Status;
}

// Display ISH FW Version with '/g' option

if (bIshVer){
    Status = GetPartVersion(FPT_PARTITION_NAME_ISHC, &major, &minor, &hotfix,
&build);

    if (EFI_ERROR (Status))
    {
        DEBUG ((D_ERROR, "GetPartVersion Error %r\n", Status));
        return Status;
    }

    Print(L"ISH FW Version: %d.%d.%d.%d \n\n", major, minor, hotfix, build);
}

//
// Load image into memory buffer
//
Print (L"\n Loading image into memory : ... \n");

Status = GetUpdateImage (ImageHandle, ImageName, &ImageLength,
&ImageBuffer);
if (EFI_ERROR (Status)) {
    Print(L" %r ",Status);
    return Status;
}

if (bPdt)

```



```

    {
        Print(L"Sending Image for Executing PDT Update. \n");

        Status = HeciPdt((char *)ImageBuffer, (unsigned int)ImageLength);
        if (EFI_ERROR(Status)) {
            Print(L"Send Failed. \n");
        }
        else {
            Print(L"Send Succeeded. \n");
        }
    }
    else
    {
        //
        // Get the current status of the ME FWUpdate Client - verifies if the
client is
        // installed
        //

        if (GetLastStatus(&lastStatus))
        {
            Print (ID_ERROR_19, FWU_LAST_STATUS);
            return EFI_SUCCESS;
        }
        //
        // Is there a pending reset?
        //

        if (GetLastUpdateResetType (&lastResetType))
        {
            Print (ID_ERROR_19, FWU_LAST_STATUS);
            return EFI_SUCCESS;
        }
        if (STATUS_UPDATE_HOST_RESET_REQUIRED == lastStatus)
        {
            Print (ID_ERROR_51, FWU_REBOOT_NEEDED);
            return EFI_SUCCESS;
        }

        if (IsUpdateStatusPending (lastStatus))
        {
            Print (ID_ERROR_20, FWU_UPD_PROCESS);
            return EFI_SUCCESS;
        }

        switch (lastResetType)
        {
        case MFT_PART_INFO_EXT_UPDATE_ACTION_HOST_RESET:

        case MFT_PART_INFO_EXT_UPDATE_ACTION_GLOBAL_RESET:
            Print (ID_ERROR_51, FWU_REBOOT_NEEDED);
            return EFI_SUCCESS;
            break;
        default:

```

```

        break;
    }

    Print (ID_INFO_3);

    //
    // Is update supported?
    //
    if (GetInterfaces (&interfaces))
    {
        Print (ID_ERROR_19, FWU_LAST_STATUS);
        return EFI_SUCCESS;
    }

    switch (interfaces)
    {
    case FW_UPDATE_DISABLED:
        Print (L"Local FWUpdate is Disabled\n");
        return EFI_SUCCESS;
    case FW_UPDATE_PASSWORD_PROTECTED:
        Print (L"Local FWUpdate is Password Protected\n");
        break;
    case FW_UPDATE_ENABLED:
        break;
    default:
        break;
    }

    Print (L"\n Checking Firmware Parameters ... \n \n");
    CheckPolicyStatus = CheckPolicyBuffer((char *)ImageBuffer,
(INT32)ImageLength, (INT32)bAllowSV, &Upd_Type, &ver);

    switch (Upd_Type)
    {
    case DOWNGRADE_SUCCESS:

    case SAMEVERSION_SUCCESS:

    case UPGRADE_SUCCESS:
        break;

    case DOWNGRADE_FAILURE:
        Print (L"FW Update downgrade not allowed\n");
        return EFI_SUCCESS;
        break;
    case SAMEVERSION_FAILURE:
        Print (L"FW Update same version not allowed, specify /s on
command line\n");
        return EFI_SUCCESS;
        break;
    default:
        break;
    }

```

```

        if(bPid)
        {
            Print(L"\n Executing ISH Partial FWUpdate");

            ComparePartID = FPT_PARTITION_NAME_ISHC;
            //Print(L"\n compareID: 0x%x",ComparePartID);

            //Get Partition Attribute from Firmware
            if (FWU_ERROR_SUCCESS != (fwuError =
GetExtendedIpuPartitionAttributes(&FwuGetIpuAttrbMsgInfo,
FWU_IPU_UPDATE_OPERATION)))
            {
                DisplayTextForReturnErrorCode(fwuError);
                return fwuError;
            }

            PartId = 0;
            //Loop through expected partitions from FW to find partition
requested
            for(j=0;j<FwuGetIpuAttrbMsgInfo.NumOfPartition;j++)
            {
                if(ComparePartID ==
FwuGetIpuAttrbMsgInfo.PtAttribute[j].PtNameId)
                {
                    PartId =
FwuGetIpuAttrbMsgInfo.PtAttribute[j].PtNameId;
                    found = true;
                    break;
                }
            }

            if(!found)
            {
                DisplayTextForReturnErrorCode(FWU_PID_NOT_EXPECTED);
                //Print(L"ParID: 0x%x\tInstId: 0x%x
\n",ComparePartID,hexValueInstId);
                return FWU_PID_NOT_EXPECTED;
            }
            Print(L"%s", ID_WARN_0);
            ///Actual Partial FW update
            //
            // Password hack for testing - replace with OEM version if password
required
            //
            if (!bUsePassword)
            {
                ZeroMem (Password, sizeof (Password));
            }
            if (bUsePassword)
            {
                FWUpdateStatus = FwUpdatePartialBuffer ((char
*)ImageBuffer, (unsigned
int)ImageLength,PartId,0,&IpuUpdatedInfo,"P@ssw0rd",FWU_ENV_MANUFACTURING,
mOemId, update_flags, &DisplaySendStatus);
            }
        }
    }

```

```

else
{
    FWUpdateStatus = FwUpdatePartialBuffer ((char
*)ImageBuffer, (unsigned int)ImageLength,PartId,0,&IpuUpdatedInfo,Password,
FWU_ENV_MANUFACTURING, mOemId, update_flags, &DisplaySendStatus);
}

if (FWU_ERROR_SUCCESS != FWUpdateStatus)
{
    DisplayTextForReturnErrorCode(FWUpdateStatus);
    if (ImageBuffer)
    {
        FreePool (ImageBuffer);
    }
    return EFI_SUCCESS;
}

if (ImageBuffer)
{
    FreePool (ImageBuffer);
}
}else{
    //
    // Password hack for testing - replace with OEM
    version if password required
    //
    Print(L"\n");
    Print(L"%s \n", ID_WARN_0);
    /*if (!bUsePassword)
    {
        ZeroMem (Password, sizeof (Password));
    }*/

    //if (bUsePassword)
    //{
        // FWUpdateStatus = FwUpdateFullBuffer
        ((char *)ImageBuffer, (unsigned int)ImageLength, "P@ssw0rd", 0,
        FWU_ENV_MANUFACTURING, mOemId, update_flags, &DisplaySendStatus);
        //}
        //else
        //{
            FWUpdateStatus = FwUpdateFullBuffer ((char
*)ImageBuffer, (unsigned int)ImageLength, Password, 0,
FWU_ENV_MANUFACTURING, mOemId, update_flags, &DisplaySendStatus);
            //}

            if (FWU_ERROR_SUCCESS != FWUpdateStatus)
            {
                DisplayTextForReturnErrorCode(FWUpdateStatus);
                if (ImageBuffer)
                {
                    FreePool (ImageBuffer);
                }
                return EFI_SUCCESS;
            }

```

```

        //return FWUpdateStatus;
    }

    if (ImageBuffer)
    {
        FreePool (ImageBuffer);
    }
}

//
// Image downloaded to FW Update Client
// Now query the status of the update being applied
//
Print(L"\n FW Update: [ 0 %% ]\r");

index = 0;
//
// Loop through Polling for Fw Update Stages
//
ProgressDot();
do {
    //We mod4 the index to determine which ascii animation
frame to display for this iteration.
    indexMod = (++index % 4);
    //symbol = (++index % 2 == 0)?'|': '-';
    switch(indexMod)
    {
        //loop through (|) (/) (-) (\) (|) (/) ...
        {
            case CMD_LINE_STATUS_UPDATE_1: symbol = '|'; break;
            case CMD_LINE_STATUS_UPDATE_2: symbol = '/'; break;
            case CMD_LINE_STATUS_UPDATE_3: symbol = '-'; break;
            case CMD_LINE_STATUS_UPDATE_4: symbol = '\\'; break;
        }
        Status =
FWUpdate_QueryStatus_Get_Response(&UpdateStatus, &TotalStages,
&PercentWritten, &lastStatus, &lastResetType);
        if(PercentWritten > 100)
        {
            break;
        }
        if (Status == FWU_ERROR_SUCCESS)
        {
            Print (L"FW Update: [ %d%% (%c)]\r"
,PercentWritten, symbol);
        } else if (lastStatus != STATUS_UPDATE_NOT_READY)
        {
            Print (L"\n");
            break; //break out of the loop
        }

        BS->Stall(100000); // Wait 1 sec before polling again
        if(timer30s >= 30)
        {
            percentdiff = PercentWritten - percentage0s;
            if(percentdiff < 1)

```

```

        {
            //TODO: Add timeout when add cmdline option
            //Status = FWU_UPDATE_TIMEOUT;
        } else
        {
            percentage0s = PercentWritten;
            timer30s = 0;
        }
    } else
    {
        timer30s++;
    }
} while ((PercentWritten != 100) && (Status ==
FWU_ERROR_SUCCESS));

switch (Status)
{
case FWU_NO_MEMORY:

case FWU_IME_NO_DEVICE:
    Print (ID_ERROR_68, FWU_UPDATE_POLLING_FAILED);
    return EFI_SUCCESS;
case FWU_IME_NOT_READY:
    DisplayTextForReturnErrorCode(status);
    return EFI_SUCCESS;
case FWU_ERROR_FW:
    Print (ID_ERROR_69, FWU_ERROR_FW, UpdateStatus);
    return EFI_SUCCESS;
default:
    break;
}

switch (lastStatus)
{
case STATUS_SUCCESS:
    switch (lastResetType)
    {
        case MFT_PART_INFO_EXT_UPDATE_ACTION_NONE:

        case MFT_PART_INFO_EXT_UPDATE_ACTION_CSE_RESET:
            Print (L"\nFW Update is completed successfully.\n");
            break;
        case MFT_PART_INFO_EXT_UPDATE_ACTION_HOST_RESET:

        case
MFT_PART_INFO_EXT_UPDATE_ACTION_GLOBAL_RESET:
            Print (L"\nFW Update is complete and a reboot will run
the new FW.\n");
            break;
        default:
            Print (L"\nFW Update is complete and a reboot will run
the new FW.\n");
            break;
    }
    fwuError = FWU_ERROR_SUCCESS;

```

```

        break;
    case STATUS_UPDATE_IMAGE_INVALID:
        DisplayTextForReturnErrorCode(FWU_IMG_HEADER);
        break;
    case STATUS_UPDATE_INTEGRITY_FAILURE:
        DisplayTextForReturnErrorCode(FWU_SGN_MISMATCH);
        break;
    case STATUS_UPDATE_SKU_MISMATCH:
        DisplayTextForReturnErrorCode(FWU_SKU_MISMATCH);
        break;
    case STATUS_UPDATE_FW_VERSION_MISMATCH:
        DisplayTextForReturnErrorCode(FWU_VER_MISMATCH);
        break;
    case STATUS_UPDATE_GENERAL_FAILURE:
        DisplayTextForReturnErrorCode(FWU_GENERAL);
        break;
    case STATUS_UPDATE_OUT_OF_RESOURCES:
        DisplayTextForReturnErrorCode(FWU_NO_MEMORY);
        break;
    case STATUS_UPDATE_AUDIT_POLICY_FAILURE:

DisplayTextForReturnErrorCode(FWU_AUDIT_POLICY_FAILURE);
        break;
    case STATUS_UPDATE_ERROR_CREATING_FT:
        DisplayTextForReturnErrorCode(FWU_ERROR_CREATING_FT);
        break;
    case STATUS_UPDATE_SAL_NOTIFICATION_ERROR:

DisplayTextForReturnErrorCode(FWU_SAL_NOTIFICATION_ERROR);
        break;
    case STATUS_INVALID_OEM_ID:
        DisplayTextForReturnErrorCode(FWU_INVALID_OEM_ID);
        break;
    case STATUS_DOWNGRADE_NOT_ALLOWED_VCN_RESTRICTION:
        DisplayTextForReturnErrorCode(FWU_IMAGE_UNDER_VCN);
        break;
    case STATUS_DOWNGRADE_NOT_ALLOWED_SVN_RESTRICTION:
        Print("FW downgrade is not allowed due to SVN
restriction.\n");
        break;
    case STATUS_UPDATE_IMAGE_BLACKLISTED:
        Print("FW update/downgrade is not allowed to the supplied FW
image.\n");
        break;
    default:
        DEBUG ((D_ERROR, "lastStatus = %d\n",lastStatus));
        DisplayTextForReturnErrorCode(FWU_GENERAL);
        break;
    }
    return EFI_SUCCESS;
}
}

```

8.2 Function Description

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

8.2.1 Get Interfaces

unsigned int GetInterfaces(unsigned short *interfaces);

Purpose: This function gets the local FW update settings from Intel® Management Engine BIOS Extension (Intel® MEBX) to determine whether or not Firmware can be updated.

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

8.2.2 Get Last Status

unsigned int GetLastStatus(unsigned int *lastStatus);

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

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

1.2.3 Get Last Update Reset Type

unsigned int GetLastUpdateResetType(unsigned int *lastResetType);

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

Arguments	LastResetType - The last FWUpdate reset type
-----------	---

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

8.2.3 Check Policy

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

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

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

8.2.4 Check Policy Buffer

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

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

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

8.2.5 Verify OEM Id

```
bool VerifyOemId(_UUID id);
```

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

Arguments	Id - OEM id
Returns	True=OEMID matched False = OEM id mismatch

8.2.6 Get Ipu Partition Attributes

```
unsigned int GetIpuPartitionAttributes(FWU_GET_IPU_PT_ATTRB_MSG_REPLY  
*FwuGetIpuAttrbMsgInfo);
```

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

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

8.2.7 Get FW Update Info Status

```
unsigned int GetFwUpdateInfoStatus(FWU_INFO_FLAGS *StatusFlags);
```

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

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

Arguments	StatusFlags - BITS 0:1 (2 bits) 0 = No recovery; 1 = Full Recovery Mode; 2 = Partial Recovery Mode (unused at present). BIT2; IPU_NEEDED bit, if set we are in IPU_NEEDED state. BIT3; FW_INIT_STATUS done. BIT4; FWU_IN_PROGRESS
-----------	---

Returns	0 = Success 8193 = Heci Device not found 8204 = Heci message has incorrect message type 8213 = Heci Buffer Size is Small Error 8710 = Insufficient memory Error 8777 = Failure in Send or Receive of the Get Info Status Command. Or even when FW returns an error status after receiving command
---------	--

8.2.8 FW Update Query Status Get Response

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

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

Arguments	UpdateStatus - indicates the current FW Update stage being executed. TotalStages - indicates the total number of FW Update stages available. PercentWritten - indicates the percentage complete of the FW Update process LastUpdateStatus - indicates the status of the fwupdate process just completed LastResetType - indicates Reset type required for the fwupdate process just completed
Returns	0= Success 1 = Invalid Manifest Data in partition 8193 = Heci Device not found 8204 = Heci message has incorrect message type 8213 = Heci Buffer Size is Small Error 8710 = Insufficient memory Error 8724 = Failure to send or receive messages to heci to get Status Info 8741 = FW returns incorrect Message Type

8.2.9 FW Update Full – Using Buffer

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

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

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

8.3 FW Update Partial Buffer

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

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

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

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

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

Arguments	Buffer - Buffer
	Buffer Length – Length of buffer
Returns	Partition ID - denotes the partition ID, which could be WLAN (wcod) or language (locl).
	WCOD ID = 0x244f4357 and LOCL ID = 0x4C434F4C
	Flags: Bit 0 of the flags is used to set allow same version update. Other bits are reserved and can be used in the future.

IpuUpdatedInfo - Contain the information that is actually used to update the IPU partition.

0 = Success

Non-zero value = Failure

8.3.1 PDT Data (Sensor Calibration Data) Update

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

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

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

Example for Usage:

```
FwUpdLclApp.efi -d <Pdt Data Binary>
FWUpdLclApp.efi -d INTC_pdt_SPT_RR3_BOM1_SENSORS
```

Arguments **Buffer** - Buffer

Buffer Length - Length of buffer

Returns **If Payload is sent to CSME successfully then Send Succeeded Message will be seen.**

8.3.2 ISH Firmware Version

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

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

9 **Intel® Manifest Extension Utility (Intel® MEU)**

The Intel® Manifest Extension Utility (MEU) inputs a firmware binary created by a 3rd party and outputs an independent-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® 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 OpenSSL tool as the signing infrastructure tool to sign the partition.

9.1 **Usage**

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

§ §

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 A-1. NVARs Descriptions

Fixed Offset Name	Description	Data Length (in Bytes)	Expected Value	Reset Type	Mfg. Post EOM/Pre EOP
Non-Application Specific Fixed Offset Item Descriptions					
MEBxPassword	<p>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:</p> <p>ASCII(32) <= char <= ASCII(126)</p> <p>Cannot contain these characters: , : "</p> <p>Must contain for complexity:</p> <ul style="list-style-type: none"> 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 <p>See section 2.7 for format and strong password requirements.</p>	8<=N<=32	Password	ME	Yes

Fixed Offset Name	Description	Data Length (in Bytes)	Expected Value	Reset Type	Mfg. Post EOM/Pre EOP																																																															
OEMSkuRule	<p>UINT32 (little endian) value. This controls what features are permanently disabled by OEM.</p> <p>Notes:</p> <p>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.</p> <p>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.</p>	4	<p>Feature Capable: 1 Feature Permanently disabled: 0</p> <table><thead><tr><th>Bit</th><th>Description</th><th>Notes</th></tr></thead><tbody><tr><td>31</td><td>Reserved</td><td></td></tr><tr><td>30</td><td>Reserved</td><td></td></tr><tr><td>29</td><td>PTT</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></tbody></table> <p>1. For corporate SKUs bits 0 and 2 need to be both set to '1' to allow for Intel® AMT to work.</p> <p>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'.</p>	Bit	Description	Notes	31	Reserved		30	Reserved		29	PTT		29:22	Reserved		21	TLS		20	DAL		19	Reserved		18	KVM	2	17	Reserved		16	ME Network Disable		15:13	Reserved		12	PAVP		11	Reserved		10	ISH		9:6	Reserved		5	Reserved		4:3	Reserved		2	Manageability and Security Application	1	1	Reserved		0	Manageability Full	1	Global	No
Bit	Description	Notes																																																																		
31	Reserved																																																																			
30	Reserved																																																																			
29	PTT																																																																			
29:22	Reserved																																																																			
21	TLS																																																																			
20	DAL																																																																			
19	Reserved																																																																			
18	KVM	2																																																																		
17	Reserved																																																																			
16	ME Network Disable																																																																			
15:13	Reserved																																																																			
12	PAVP																																																																			
11	Reserved																																																																			
10	ISH																																																																			
9:6	Reserved																																																																			
5	Reserved																																																																			
4:3	Reserved																																																																			
2	Manageability and Security Application	1																																																																		
1	Reserved																																																																			
0	Manageability Full	1																																																																		

FeatureShipState	<p>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.</p> <p>Notes:</p> <p>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.</p>	4	<div>Feature Enabled: 1 Feature Disabled: 0</div> <table><thead><tr><th>Bit</th><th>Description</th><th>Notes</th></tr></thead><tbody><tr><td>31:30</td><td>Reserved</td><td></td></tr><tr><td>29</td><td>PTT</td><td></td></tr><tr><td>28:11</td><td>Reserved</td><td></td></tr><tr><td>10</td><td>ISH</td><td></td></tr><tr><td>3:9</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></tbody></table> <p>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.</p>	Bit	Description	Notes	31:30	Reserved		29	PTT		28:11	Reserved		10	ISH		3:9	Reserved		2	Manageability Full		1:0	Reserved		Global	Yes
Bit	Description	Notes																											
31:30	Reserved																												
29	PTT																												
28:11	Reserved																												
10	ISH																												
3:9	Reserved																												
2	Manageability Full																												
1:0	Reserved																												

Fixed Offset Name	Description	Data Length (in Bytes)	Expected Value	Reset Type	Mfg. Post EOM/Pre EOP
WLAN Power Well	Sets which power well the board uses for WLAN cards	4	0x80 = Disabled 0x81 = CoreWell 0x82 = Primary Well 0x83 = ME Well 0x86 = WLAN Sleep via SLP_WLAN#	Global	No
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	No

GPIO	GPIO	60	<p>GPIO groups and pad range for each</p> <p>grp pad#</p> <p>For LP platforms:</p> <p>GPP_A 0-23</p> <p>GPP_B 0-23</p> <p>GPP_C 0-23</p> <p>GPP_D 0-19</p> <p>GPP_E 0-23</p> <p>GPP_F 0-19</p> <p>GPP_G 0-7</p> <p>GPP_H 0-23</p> <p>GPD 0-11</p> <p>For H platforms:</p> <p>GPP_A 0-23</p> <p>GPP_B 0-23</p> <p>GPP_C 0-23</p> <p>GPP_D 0-15</p> <p>GPP_E 0-12</p> <p>GPP_F 0-23</p> <p>GPP_G 0-15</p> <p>GPP_H 0-23</p> <p>GPP_I 0-14</p> <p>GPP_J 0-9</p> <p>GPP_K 0-11</p> <p>GPD 0-11</p> <p>Example read of GPIO:</p> <p>Variable: "gpio"</p> <p>Value:</p> <p>0x0000 : 00 00 00 00 04 00 00 00 06 00 00 00 01 00 00 00</p> <p>0x0010 : 00 00 00 00 01 00 00 00 04 00 00 00 0C 00 00 00</p> <p>0x0020 : 01 00 00 00 00 00 00 00 08 00 00 00 01 00 00 00</p> <p>0x0030 : 0F 00 00 00 01 00 00 00 00 00 00 00</p> <p>Note: the only locations that can be modified are underlined above.</p> <p>The format for updating the GPIO is as follows...</p> <p>Gpio =</p> <p>0x000000000030000000110000000010000 000000000001000000002000000170000 000100000000000000008000000030000 0013000000010000000000000000</p>	ME	No
------	------	----	---	----	----

Fixed Offset Name	Description	Data Length (in Bytes)	Expected Value	Reset Type	Mfg. Post EOM/Pre EOP
			RST = GPP_D_17 IRQ = GPP_C_23 DFU = GPP_D_19		
FWUpdLcl	Enabled Firmware Update Capability	1	0 = disabled 1 = enabled	Global	Yes
eDP Port Config	EDP Port Configuration. Up to two ports can be enabled 0x00 - 0x01 - A 0x02 - B 0x04 - C 0x08 - D 0x10 - E 0x20 - F	1	0x0:None/None 0x1:A/None 0x2:B/None 0x3:A/B 0x4:C/None 0x5:C/A 0x6:C/B 0x8:D/None 0x9:D/A 0xa:D/B 0xC:D/C 0x10:E/None 0x11:E/A 0x12:E/B 0x14:E/C 0x18:E/D 0x20:F/None 0x21:F/A 0x22:F/B 0x24:F/C 0x28:F/D 0x30:F/E	Global	No
LSPCON Port Config	LSPCON Port Configuration.	1		Global	No
Delayed Authentication Mode Config	Enables Delayed Authentication Mode on the platform	1	0 = Disabled 1 = Enabled	ME	Yes
Unconfigure On RTC	Enables platform to unconfigure on RTC removal	1	0 = Disabled 1 = Enabled	ME	Yes
AMT Related NVARs					
OEM Custom Cert 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

Fixed Offset Name	Description	Data Length (in Bytes)	Expected Value	Reset Type	Mfg. Post EOM/Pre EOP
OEM Custom Cert 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 Custom Cert 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
Redirection Privacy / Security Level	Redirection (KVM, SOL, Storage Redirection) privacy level and configuration (RCFG, CCM) settings.	1	Default 0x01 Enhanced 0x02 Extreme 0x03 Default: SOL enabled = true Storage Redirection 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 Storage Redirection enabled = true KVM enabled = true Opt-in can be disabled= false Opt-in configurable remotely = true RCFG and CCM = true Extreme SOL enabled = false Storage Redirection enabled = false KVM enabled = false Opt-in can be disabled= false KVM opt-in configurable remotely = N/A RCFG and CCM = false	ME	No
Embedded Host Based Config	Embedded Host Based Configuration State.	1	0 = Disabled 1 = Enabled	ME	No
Firmware KVM Screen Blanking	Screen Blanking Enabled	1	0 = Disabled 1 = Enabled	ME	No
PKI Domain Name Suffix	PKI DNS Suffix. Null terminated string	32	PKI DNS Suffix in dotted string format	ME	Yes
Config Server FQDN	Configuration Server FQDN (Fully Qualified Domain Name)	256	Example: "intelFVE.com"	ME	Yes

Fixed Offset Name	Description	Data Length (in Bytes)	Expected Value	Reset Type	Mfg. Post EOM/Pre EOP
RCFG/ZTC	R Configuration	1	0 = Disabled 1 = Enabled Note: This is update only NVAR. Tool won't be able to read expected value.	ME	Yes
*Redirection	This is a bit-field Indicating the enable/disable status of Storage Redirection, SOL, and KVM features in Intel® AMT. bit[0]: 1 – Storage Redirection enabled, 0 – disabled bit[1]: 1 – SOL enabled, 0 – disabled bit[2]: 1 – KVM enabled, 0 – disabled	1	Range: 0-7 Example: Value of 4 (100b) indicates that KVM is enabled. Value of 3 (011b) indicates that Storage Redirection, and SOL are enabled. Value of 7 (111b) indicates that Storage Redirection, SOL, and KVM are enabled. Note: This is update only NVAR. Tool won't be able to read expected value.	ME	Yes

Fixed Offset Name	Description	Data Length (in Bytes)	Expected Value	Reset Type	Mfg. Post EOM/Pre EOP
*Opt-in Policy	Change User Opt-in (lower nibble). NONE = 0, KVM = 1, ALL = F Disable Opt-In Configurable from Remote IT (upper nibble). 0 - Opt-in is NOT Configurable from Remote IT 1 - Opt-in is Configurable from Remote IT	1	0x00 0x10 0x01 0x11 0x0F 0x1F Examples: In addition to the following, the values may not be configured remotely: Value of 0x00 indicates User Consent is not required. Value of 0x01 indicates User Consent is required for KVM only. Value of 0x0F indicates User Consent is required for (ALL). In addition to the following, the values may be configured remotely: Value of 0x10 indicates User Consent is not required. Value of 0x11 indicates User Consent is required for KVM only. Value of 0x1F indicates User Consent is required for (ALL).	ME	Yes
Host Name	Set Host Name Only	64	SkyLake KabyLake SunrisePoint	ME	Yes
Domain Name	Set Domain Name Only	192	myserver.intel.com amr.corp.intel.com www.intel.com mymail.somecollege.edu	ME	Yes
Config Server IPv6/IPv4 Address	Set Provisioning Server (IPv4/IPv6) Address	60	Example of IPV4: 192.168.1.200 255.255.255.0	ME	Yes
Config Server IPv6/IPv4 Port	Set Provisioning Server (IPv4/IPv6) Port	2	Within Range: 0 – 0xFFFF	ME	Yes

Fixed Offset Name	Description	Data Length (in Bytes)	Expected Value	Reset Type	Mfg. Post EOM/Pre EOP
Disable All Pre-Installed Cert Hashes	Disable all Pre-Installed Certificate Hashes	1	0 = Disabled 1 = Enabled Note: This is update only NVAR. Tool won't be able to read expected value.	ME	Yes
Intel(R) AMT Idle Timeout	Change the Idle Timeout in minutes	2	Within Range: 1 – 0xFFFF	ME	Yes
Intel(R) AMT WD Auto Reset	Intel ® AMT Watchdog Automatic Reset enabled	1	0 = disabled 1 = Enabled	ME	No
Field Programming Fuses					
Intel(R) PTT	Enables / Disables the fTPM / PTT FPFs	1	0 = Disabled 1 = Enabled	ME	No
BSP Initialization	Indicating the BSP initialization on boot	1	0 = Disabled 1 = Enabled	ME	No
CPU Debugging	Indication CPU debug capabilities	1	0 = Disabled 1 = Enabled	ME	No
Error Enforcement Policy 0	Error Enforcement Policy 0	1	0 = Disabled 1 = Enabled	ME	No
Error Enforcement Policy 1	Error Enforcement Policy 1	1	0 = Disabled 1 = Enabled	ME	No
Force Boot Force Boot Guard ACM	Indicates Boot Guard ACM is enforced or not	1	0 = Disabled 1 = Enabled	ME	No
Key Manifest ID	Contains key manifest required for authentication	1	0 = Disabled 1 = Enabled	ME	No
Measured Boot	One of the applicable profiles for Boot Guard	1	0 = Disabled 1 = Enabled	ME	No
OEM ID	OEM ID	1	0 = Disabled 1 = Enabled	ME	No
OEM Platform ID	OEM Platform ID	1	0 = Disabled 1 = Enabled	ME	No
OEM Secure Boot Policy	OEM Secure Boot Policy	1	0 = Disabled 1 = Enabled	ME	No
Persistent PRTC Backup Power	Persistent PRTC Backup Power	1	0 = Disabled 1 = Enabled	ME	No

Fixed Offset Name	Description	Data Length (in Bytes)	Expected Value	Reset Type	Mfg. Post EOM/Pre EOP
Protect BIOS Environment	Indicated if BIOS environment protection is enforced or not	1	0 = Disabled 1 = Enabled	ME	No
S3 Optimization	S3 optimization for Boot Guard	1	0 = Disabled 1 = Enabled	ME	No
Txt Supported	Txt Supported	1	0 = Disabled 1 = Enabled	ME	No
Verified Boot	One of the applicable profiles for Boot Guard	1	0 = Disabled 1 = Enabled	ME	No
OEM Public Key Hash	Hash of the provided OEM public key	32	32 Hex Pairs with space between pairs	ME	No

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

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

§ §

Appendix B : Tool Detail Error Codes

Errors are reported in the following format for all tools:

<ErrorTypeNumber>*256+<ErrorNameNumber>

B.1 Common Error Code for FIT tool

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

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

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

Error Code	Error Message
0	No Error
1	Initialize Error
2	Failed to build
3	Build general error
4	Build enumeration error
5	Error building attribute
6	Error Building Resources
7	Decompose Error.
8	Failed to decompose SMIP data
9	Failed to decompose Image
10	Failed to decompose Region
11	Error Decomposing attribute
12	Error calling decomposition actions
13	Decomposition node not found
14	Error decomposing class
15	Decomposition not ready
16	Failed to detect and configure ROM Bypass partition
17	Failed to decompose boot partition
18	Failed to generate decomposed files
19	Failed to decompose boot partition entry
20	Error executing pre-build actions
21	Error executing post-build actions
22	Error generating intermediate build output
23	Invalid object alignment value
24	Unable to resolve parent for attribute
25	Buffer offset out of bounds
26	Unresolved native type
27	Data Conversion error
28	Invalid H file output path
29	Error rounding attribute
30	Error updating attribute
31	Error executing post-decomp actions
32	Missing XML attribute
33	Failed to sign SMIP data

34	MEU config file Error
35	Bad command line options
36	Invalid PCH SKU specified
37	Error setting the log file
38	File not found
39	File name with the PERIODS at the end is not supported
40	Could not find VSCC value for given JEDEC code
41	Failed to open new image
42	Failed to open with processed commands
43	Failed to parse XML
44	Failed to parse XML settings
45	Full image could not be written
46	Could not locate input region file
47	Invalid input file type
48	Failed to populate known good VSCC database in memory
49	User did not accept the license agreement
50	Failed to launch GUI as requested
51	Failed to build CSE region file
52	Failed to process layout file
53	Failed to process configuration file
54	Unable to open layout file
55	Invalid root name detected in configuration file
56	Unknown root node found in XML
57	Invalid XML tag
58	Invalid XML child tag association
59	XML tag exception
60	Missing XML tag
61	Error resolving data dependencies
62	Failed to generate dependency map for XML tag
63	Failed to load native type definitions
64	Invalid XML value attribute
65	Error overriding CSE version Major number
66	Error overriding CSE version Minor number
67	Error overriding CSE version Hotfix number
68	Error overriding CSE version Build number
69	Error overriding CSE Internal Build Version
70	Error overriding signing key

71	Error overriding MFS BinObject
72	Error overriding the active LOCL Instance Id
73	Error overriding the active WCOD Instance Id
74	Error overriding the active MDMV Instance Id
75	Error enabling partition from the command line
76	Error disabling partition from the command line
77	Error setting partition length
78	Error overriding the LOCL UPV version
79	Error overriding the WCOD UPV version
80	Error overriding the MDMV UPV version
81	Error overriding the TCB SVN number
82	Error overriding the ARB SVN number
83	Error overriding the VCN number
84	Error Overriding the Sku Attributes value
85	Error overriding the Uma size
86	Error overriding compression mode on all modules
87	Error overriding CSE region length
88	Error enabling RomBypass partition
89	Error overriding PCH version
90	Error overriding DataFormatVersion
91	No XML version was specified
92	The XML version specified is not in the proper format: x.x
93	The version of the XML file you are loading is not supported
94	The version of the XML you are loading is greater than any version this application knows about
95	Unable to open config file
96	Error overriding header files output directory
97	Error overriding the AFS SKU ID
98	Error opening file
99	Error writing to file
100	Size mismatch in file write
101	Error reading text file
102	Setting not found
103	Invalid type specified for setting
104	Unable to resolve action's target attribute
105	Unable to resolve action's source attribute
106	Failed to write data to image buffer

107	Unable to resolve attribute used in action
108	Failed to open file
109	Failed to update image buffer offset
110	Source value of length 0 must be in hex string format
111	Failed to load NVARS from path
112	Detected overflow in CalcOffset operation
113	String length too long
114	Empty input string
115	Failed attribute length limit validation
116	Failed to write buffer to file
117	Invalid JSON Parameter(s)
118	Invalid region size
119	Unable to calculate hash
120	Invalid action parameters
121	Buffer overflow detected
122	Buffer overflow detected
123	Invalid Checksum Action Parameters
124	Error Calculating round to function
125	Missing data class
126	Invalid signing key
127	Failed to generate signature
128	Unable to generate intel.cfg file
129	Unable to generate intel.cfg SHA2
130	Failed to encrypt module
131	Error generating manifest independent partition
132	Error generating feature permissions extension
133	Error generating thread attributes extension
134	Error generating device attributes extension
135	Error generating mmio ranges extension
136	Error generating file producer extension
137	Failed to add group IDs to process extension
138	Error generating user info extension
139	Failed to adjust the FTUP partition length and offset
140	Found IUP partition (WCOD,MDMV,LOCL,ISH) before NFTP. This is not allowed
141	Found unexpected IUP partition. All IUP must be allocated in a contiguous block

142	Calculated FTUP partition size is smaller than FTUP size, this will break FWUpdate
143	Number of FPT entries does not fit in current FPT area supported by FTOOL
144	Unable to resolve user name
145	Unable to update partition offsets in database
146	Missing partition parameters
147	Missing partition instance
148	Unable to update partition offset
149	Error building Partial Firmware Update image
150	Failed to configure firmware runnable region
151	Unable to disable attribute
152	Invalid runnable region configuration
153	Make Module Failed
154	Get elf info failed
155	Make Module Failed
156	Parse Module metadata Failed.
157	Elf to Bin failed
158	Get Section Data failed
159	Invalid ModuleType. Module is not Process or Shared Library type
160	Failed to build shared library
161	Invalid TotalThreadStackSize value
162	Unable to get CM0HeapSize configuration parameter
163	Unable to get DefaultHeapSize configuration parameter
164	Invalid CM0Heap Value
165	Invalid DefaultHeap Value
166	Unable to find the FLREG layout entry
167	Failed to resolve region limit
168	Failed to resolve region base
169	Unable to find the Regions layout entry
170	Missing input region length configuration option
171	File Path could not be resolved
172	Invalid region size
173	Not enough flash space
174	Missing region data target
175	Unable to update region data target
176	Unable to load CSE region
177	Failed to allocate memory

178	Failed to parse CSE region
179	Not enough space to copy CSE region into image buffer
180	Unable to prepare CSE region
181	Invalid VSCC entry
182	Detected overflow in BPDT table
183	Invalid Descriptor offset
184	Invalid Descriptor size
185	Unable to parse ROM Bypass configuration
186	Unable to load ISH image
187	ISH image file size is too large
188	Failed to update ME Region
189	Invalid PKI Suffix:
190	Invalid Certificate Hash Format:
191	Invalid GUID format
192	Failed to parse GbE image
193	The file is not large enough to be a valid GbE
194	Invalid Region Order
195	Invalid settings combination
196	Unable to load Token
197	Unable to decompose Token
198	IDLM Binary is invalid or corrupt
199	Unable to decompose IDLM Binary
200	Failed to process VR profile selection
201	Failed to generate FW update image
202	String length is too large
203	Failed to generate CSE data partition
204	TBT Binary is invalid or corrupt
205	Chipset Init Binary is invalid or corrupt
206	Chipset Init Base Intel Recommendation table is invalid
207	Chipset Init Product version does not match the configured PCH SKU type
208	Failed to get image Metadata sub partition
209	Failed to load FITC binary to sub partition
210	Failed to generate Image Metadata partition
211	Failed to load FITC binary to sub partition
212	Failed to find child attribute
213	Failed to get Class Instance
214	Failed to map GPIOs

215	Invalid NVAR size
216	NVAR IO Error
217	Failed to set target IFWI configuration
218	Failed to load BIOS image from file
219	Failed to configure IFWI layout
220	Failed to prepare one or more IFWI components
221	Failed to load ME component
222	Detected invalid Sub-Partition
223	Detected build buffer overflow
224	Failed to update build buffer cursor offset
225	Failed to prepare ME BUP Sub-Partition
226	Failed to calculate boot partition sizes
227	Unable to update region data target
228	Failed to get ME Sub-Partitions
229	Failed to load OEM Key Manifest input file
230	Failed to add OEM Key Manifest to IFWI image
231	Failed to build SMIP data
232	Failed to load SMIP intermediate file
233	Failed to add SMIP Sub-Partition to IFWI image
234	Failed to add ROMB partition to IFWI image
235	Failed to load input file
236	Unable to determine image type
237	Detected invalid DNX image format.
238	Unable to detect number of flash components setting
239	Unable to resolve flash image size
240	Failed to validate Key Manifests
241	Failed to validate Public key hash
242	Failed to calculate BPDT Checksum
243	Failed to calculate and set required image padding
244	Invalid Manifest Extension Utility path
245	Utility to sign the SMIP data
246	Invalid signing key path
247	Invalid signing tool path
248	Failed to load data sub-partition
249	General error
250	Missing configuration attribute
251	Unable to set configuration value

252	IMR range value out of range
253	Unable to round up IMR value
254	Total IMR size exceeding maximum size
255	Invalid action parameters
256	Invalid attribute parameters
257	Missing JSON parameter in NVAR Action
258	Unable to convert NVAR index to U32
259	Unable to convert NVAR offset to U32
260	Unable to convert NVAR bitHi to U32
261	Unable to convert NVAR bitLo to U32
262	Unable to convert NVAR field size to U32
263	Unable to convert NVAR file size to U32
264	Failed to write NVAR
265	Failed to read NVAR
266	Invalid action parameter
267	Invalid target name
268	Invalid bitfield length specified
269	Error updating configuration variable
270	Could not load binary file
271	Could not resize NVAR for binary file
272	Specified variable size will not fit into fixed-size NVAR file
273	Specified variable size will not fit into cell
274	Specified offset is larger than NVAR size
275	Could not adjust NVAR params
276	Could not write NVAR value
277	Could not read NVAR value
278	Failed to write binary file for NVAR
279	Certificate NVAR size mismatch
280	Certificate NVAR name field size mismatch
281	Failed to save intermediate file
282	Unable to access data
283	Detected duplicate syscall id
284	Detected duplicate syscall name
285	Detected duplicate syscall group name
286	Detected loop in syscall group dependencies
287	Detected invalid syscall group name
288	Detected invalid syscall group raw value

289	Detected invalid syscall name in group definition
290	Detected invalid syscall id
291	Detected invalid syscall group id used in process module
292	Failed to generate header file definitions
293	Invalid Group Value
294	SystemResources Class has not been initialized
295	Internal error
296	Failed to get active module names
297	Unable to resolve type
298	Failed to generated system resources report
299	Failed to generate source code for bus driver
300	Detected duplicate process name
301	Detected duplicate process id
302	File write error
303	Detected duplicate user name
304	Detected duplicate special file label
305	Detected duplicate service name
306	Detected duplicate group id
307	Detected duplicate user id
308	Error opening file for read
309	Error reading file data
310	Size requested was too large
311	Error opening file for write
312	Error appending to file
313	Creating directory structure
314	Error running LZMA compression
315	Error running LZMA extraction
316	Wrong format found
317	Unknown Project
318	Invalid data pointer
319	Out of memory
320	Unable to remove file entry from FCS table
321	MFS was not initialized
322	FCS was not initialized
323	Failed to process FCS entries
324	Detected duplicate special file label
325	FileEntry already being used by another FCS table

326	Failed to create FCS handle
327	Failed to get file from FCS
328	Failed to get file attributes from FCS
329	Failed to add new file to FCS
330	Failed to delete file from FCS
331	Failed to flush FCS buffer into memory
332	Invalid FCS file
333	Failed to terminate MFS library
334	Failed to get file size from MFS
335	Failed to delete file from MFS
336	Failed to decompose CSE image
337	Failed to initialize MFS
338	Failed to load Intel.cfg table
339	Failed to load Fit.cfg table
340	Failed to create the current values table
341	Failed to generate the current values table
342	Failed to create new Fit.cfg table
343	NVAR Access error
344	FW Code Generation Error
345	Invalid ME Version
346	Module Not Found
347	Action not found
348	Action failed to execute
349	Failed to process input XML
350	Invalid command line options
351	Failed to save XML
352	Invalid XML template option specified
353	Invalid Manifest Version specified on CLI
354	Unable to load tool config xml
355	Unsupported signing tool specified
356	Invalid signing tool configuration
357	Invalid decomp binary type specified
358	File is not a valid XML file
359	Invalid manifest index value
360	Error finding manifests in file
361	Path provided is not a valid directory
362	Unable to find files

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

B.2 Common Error Code for All Tools

Error Code	Error Message
0	Success
1	Tool common error
2	Passed with warning
3	Internal Error. Unexpected error occurred
4	Unsupported OS.

Error Code	Error Message
5	Memory allocation error occurred.
6	Error accessing the function GetSystemFirmwareTable from kernel32.dll.
7	The function GetSystemFirmwareTable failed with Windows Error Code: %d.
8	Error accessing the kernel32.dll.
9	Commit Anti Rollback SVN failed.
10	Error occurred while reading the file.
11	Error getting current working directory path.
12	Error getting current working directory permission:
13	An unknown error occurred while opening the file.
14	An unknown error occurred while working with the file.
15	Error occurred while writing to the file.
16	Error while trying to read the signature of the file %S.
17	The file %s, is not signed by Intel(R) Embedded Subsystems and IP Blocks Group.
18	Invalid certificate information residing in file %s.
19	Failed to write 0x%02X to IO Port 0x%04X.
20	Cannot locate ME device.
21	Write register failure.
22	Circular buffer overflow.
23	Communication error between application and Intel(R) ME module.
24	Unsupported HECI bus message protocol version.
25	HECI Timeout.
26	Unexpected result in command response.
27	Cannot find host client.
28	Cannot find ME client.
29	Failure occurred during ME disconnect.
30	Client already connected.
31	No free connection available.
32	Flow control error.
33	No message.
34	Buffer size is too large.
35	Buffer is too small.
36	%s is too long.
37	Invalid command line option(s).
38	The following Parameter is not a valid option: %s.
39	PCH is not supported.

Error Code	Error Message
40	Internal Error (Safe function wrapper error: Invalid size).
41	Internal Error (Safe function wrapper error: compose string from list).
42	Internal Error (Safe function wrapper error: compose string).
43	Internal Error (Safe function wrapper error: memncpy).
44	Internal Error (Safe function wrapper error: strncpy).
45	Internal Error (Safe function wrapper error: strncat).
46	Internal Error (Safe function wrapper error: strtok).
47	Printf function failed.
48	Failed getting variable %s value.
49	The variable %s is supported on Corporate SKU only.
50	Unable to find matching LOCL.
51	Could not access PCI device.
52	Unable to load library.
53	Unable to change permission.
54	Unable to perform request due to permission failure.
55	Cannot find requested device.
56	Unable to perform CreateFile.
57	The FPF compare failed.
58	The CSE File Component requested, File Name is not valid for this operation.
59	Failed to read FPT NVARs config file. %s.
61	Fail to read FW Status Register value.
62	Fail to create verbose log file.
63	Unknown or unsupported hardware platform.
64	Failed to initialize SPI interface.
65	Could not update [%s].
66	Cannot update %s. Invalid data length.
67	Feature not found.
68	Feature not available.
69	%s actual value is - %s.
70	FW status test failed.
71	Boot Guard status test failed.
72	Parameter %s - %s.
73	The value of \'%s\' is missing.
74	Failed to communicate with CSME.This tool must be run from a privileged account (administrator/root).
75	Master Access config file value for %s format is invalid.

Error Code	Error Message
76	Failed to retrieve feature.
77	Master Access config file value for %s exceed maximum allowed value.
78	Failed to retrieve Intel (R) FIT version.
79	Failed to retrieve Intel (R) Internal Build Version.
80	Ambiguous Master Access value. Master Access config file region [%s] defined more than once.
81	MEManuf Operation Failed.
82	Invalid Access node name in Master Access configuration file.
83	Invalid RequiredValue node name in Master Access configuration file.
84	Invalid server address.
85	Intel(R) test failed to start, error 0x%X returned.
86	Intel(R) test timeout (exceeded 30 seconds).
87	Intel(R) ME test is currently running, try again later.
88	MEManuf EOL & BIST config file generation failed.
89	M3 results are not available from SPI. Please run -test option to perform the BIST test.
90	Could not read M3 results from SPI.
91	SMBus hardware is not ready.
92	Internal error - SMBus Read Byte PEC failure.
93	SMBus encountered time-out.
94	Signature: invalid! No more information can be displayed.
95	Internal error - Failed to match.
96	Internal error - Out of memory.
97	Internal error - Unable to get current PP.
98	Failed to retrieve test result from SPI.
99	Failed to retrieve power package setting.
100	Failed to retrieve power rule from SPI.
101	WLAN power well setting is set incorrectly.
102	Failed to retrieve test result from SPI.
103	Internal error - Failed to retrieve Platform Attribute.
104	Failed to retrieve PROC_MISSING NVAR setting.
105	PROC_MISSING NVAR setting is set incorrectly.
106	Failed to retrieve password from SPI.
107	Internal error - Password length is incorrect.
108	Internal error - Modified local password.
109	Internal error - Invalid password.
110	Boot Guard Self Test Failed.

Error Code	Error Message
111	Intel integrated LAN setting is set incorrectly.
112	Intel LAN Connected Device (PHY) physical connectivity error with ME.
113	Internal error - Illegal data length.
114	Internal error - Illegal data value.
115	EHBC State Test Failed - Error while reading data from flash.
116	EHBC State Test Failed - Contradiction with current Privacy Level.
117	Current WLAN does not match micro-code, please update WLAN micro-code in FW.
118	Communication with WLAN device failed.
119	Length of OEM Customizable Certificate Friendly Name setting is set incorrectly.
120	OEM Customizable Certificate Stream setting is set incorrectly.
121	OEM Customizable Certificate Hash Algorithm setting is set incorrectly.
122	Length of OEM Customizable Certificate Stream is set incorrectly.
123	Internal error - Unable to compress.
124	The compressed data is incorrect.
125	USBr EHCI 1 Enabled and/or USBr EHCI 2 Enabled setting is set incorrectly.
126	KVM device is already in use by other components.
127	Failed to retrieve power source.
128	Power source is not AC.
129	LAN power well setting is set incorrectly.
130	WLAN power well setting is set incorrectly.
131	System UUID actual value is all 0x00.
132	System UUID actual value is all 0xFF.
133	Security Descriptor Override Strap (SDO) is enabled.
134	End-Of-Post message is not sent.
135	Unable to determine Intel(R) ME Manufacturing Mode status.
136	Intel(R) ME is still in Manufacturing Mode.
137	BIOS has granted Intel(R) Gbe and/or ME access to its region.
138	%s mismatch, actual value is - %s.
139	Generating file in System Folder is not allowed
140	Cannot run the command since Intel(R) AMT is not available.
141	MFS is corrupted.
142	Using wrong PCH SKU Emulation via Intel (R) FIT vs whats the actual HW Type.

Error Code	Error Message
143	Cannot perform hibernation. Please manually reboot the system.
144	MEManuf Test Failed.
145	Test is enabled by the user but is unknown by the platform - %s.
146	Attempting to add sibling to XML root node.
147	File size is zero.
148	XML parsing failed.
149	XML parsing encountered data overflow.
150	Invalid XML error code conversion.
151	XML parser - out of memory error.
152	Missing RequiredValue xml node in Master Access configuration file.
153	Incorrect region name in Master Access configuration file.
154	Failed to retrieve list of BIST tests to run from FW.
155	Unexpected failure when retrieving BIST results.
156	Retrieving the EOL Config list of tests failed.
157	Retrieving the EOL Var list of tests failed.
158	No name attribute specified for test: %s.
159	Failed to parse configuration file provided.
160	No output file path specified to write configuration file.
161	No data to write to configuration file.
162	Invalid ErrAction specified
163	The 2 SPI flash devices do not have compatible command sets.
164	No SPI flash device could be identified. Please verify if Fparts.txt has support.
165	Failed to allocate memory for the flash part definition file %s.
166	Parsing file failed.
167	Protected Range Registers are currently set by BIOS, preventing flash access. Please contact the target system BIOS vendor for an option to disable Protected Range Registers.
168	Hardware sequencing failed. Make sure that you have access to target flash area.
169	The host CPU does not have read access to the target flash area. To enable read access for this operation you must modify the descriptor settings to give host access to this region.
170	An attempt was made to read beyond the end of flash memory.
171	Software sequencing failed. Make sure that you have access to target flash area.

Error Code	Error Message
172	Invalid Block Erase Size.
173	Invalid Write Granularity value.
174	Invalid Enable Write Status Register Command value.
175	The supplied zero-based index of the SPI Device is out of range.
176	Invalid descriptor region.
177	Region does not exist.
178	An attempt was made to write beyond the end of flash memory.
179	An attempt was made to erase beyond the end of flash memory.
180	The address 0x%08X of the block to erase is not aligned correctly.
181	Hardware timeout occurred in SPI device.
182	There are no supported SPI flash devices installed. Please check connectivity.
183	Unrecognized value in the HSFSTS register.
184	AEL is not equal to zero.
185	FCERR is not equal to zero.
186	Checking variable %s failed.
187	Invalid value for %s CVAR.
188	Invalid Manufacturing Line Configurable variable name %s.
189	File does not exist.
190	End Of Manufacturing Operation failure - Verification failure on Descriptor Lock settings.
191	Unable to get master base address from the descriptor.
192	Password does not match the criteria.
193	Invalid length of Manufacturing Line Configurable value. Check configuration file for correct length.
194	Invalid hash certificate file.
195	End Of Manufacturing Operation failure - Verification failure on ME Manufacturing Mode Done settings.
196	cfg_rules: the requested rule change is not supported after end of manufacturing.
197	Invalid parameter value specified by user. Use -? option to see help.
198	ME disabled.
199	Failed to get information about the installed flash devices.
200	An error occurred reading the flash descriptor signature.
201	Flash descriptor does not have correct signature.
202	The attempt to commit the Manufacturing Line Configurables has failed.

Error Code	Error Message
203	Access was denied opening file.
204	Failed to read the entire file into memory. File: %s.
205	The address is outside the boundaries of the flash area.
206	Unable to write data to flash. Address 0x%x.
207	Data verify mismatch found.
208	Failed to write the entire flash contents to file.
209	An error occurred reading the flash mapping data.
210	System booted in Non-Descriptor mode, but the flash appears to contain a valid signature.
211	An error occurred reading the flash components data.
212	An error occurred reading the flash region base/limit data.
213	An error occurred reading the flash master access data.
214	Flash is not blank.
215	PAVP oem config data: invalid edp port value
216	Setting Global Reset Failed.
217	ME disable not needed.
218	ME already disabled.
219	The request to disable the ME failed.
220	There is a problem with the GbE binary which prevents saving the data.
221	A required parameter is missing.
222	Committing the FPF is not allowed at this time.
223	The FPF has already been committed.
224	PAVP oem config data: invalid lspcon port value.
225	Committing a specific FPF is not supported. Consider committing all the FPFs.
226	Keybox file size invalid.
227	Invalid all hashes state file.
228	Invalid idle timeout file.
229	Invalid provisioning state file.
230	CEK is invalid.
231	CEK is not available.
232	Cannot provision after EOM.
233	Invalid redirection state file.
234	Bad CRC.
235	Bad Magic.
236	Invalid EHBC state file.
237	Keybox is not provisioned.

Error Code	Error Message
238	The host CPU does not have write access to the target flash area. To enable write access for this operation you must modify the descriptor settings to give host access to this region.
239	User selected to cancel the operation.
240	Internal error - Invalid Heci response length.
241	Error determining possible system states.
242	Cannot locate MEI driver.
243	Unexpected internal FW error occurred. Object was not found.
244	Invalid State found for test - %s.
245	ISH Internal Error.
246	IUP Not Found.
247	Cannot locate HID device.
248	Incorrect Report ID received.
249	MCTP SMBUS test failed.
250	Invalid config file. State was not found for test - %s.
251	Invalid config file. RequiredValue was not found for test - %s.
252	Invalid config file. ErrAction was not found for test - %s.
253	Unable to validate address range.
254	Memory window not set or device is not armed for operation.
255	Sensor could not be found. Either no sensor is connected, the sensor has not yet initialized, or the system is improperly configured.
256	Not enough memory/storage for requested operation.
257	Used in TOUCH_SENSOR_HID_READY_FOR_DATA_RSP to indicate sensor has been disabled or reset and must be reinitialized.
258	Used to indicate compatibility revision check between sensor and ME failed, or protocol ver between ME/HID/Kernels failed.
259	Indicates sensor went through an unexpected reset.
260	Requested sensor reset failed to complete.
261	Operation timed out.
262	Test mode pattern did not match expected values.
263	Indicates sensor reported fatal error during reset sequence. Further progress is not possible.
264	Indicates sensor reported non-fatal error during reset sequence. HID/BIOS logs error and attempts to continue.
265	Indicates sensor reported invalid capabilities, such as not supporting required minimum frequency or I/O mode.

Error Code	Error Message
266	Indicates that command cannot be complete until ongoing Quiesce I/O flow has completed.
267	Cannot find the NVAR file; the system maybe in EOM.
268	Invalid cfg rule data.
269	Cannot access the NVAR file attributes.
270	Failed to hash CSE file data.
271	Operation is not allowed after EOM.
272	Used an invalid input parameter to access the NVAR file.
273	FPF is not written.
274	Invalid privacy level file.
275	File is invalid.
276	Can not provision after EOM.
277	Certificate verification failed.
278	HDCP Rx is not provisioned.
279	Invalid string value entered for the Manufacturing Line Configurable.
280	Detected ME in recovery mode.
281	FW returned status: Erase token failure.
282	Detected invalid data size.
283	Detected invalid hex value.
284	Failed to retrieve 5K port setting.
285	Failed to retrieve LSPCON Port setting.
286	Display port settings are not correct.
287	EC Region write access permissions don't match Intel recommended values.
288	Unexpected size found in the file %s. Expected: 0x%X. Received: 0x%X.
289	Unable to execute command in this Firmware State. Please reboot.
290	GBE Region write access permissions don't match Intel recommended values.
291	GPIO file contains GPIO pin assignments that are not multiples of the GPIO pin data structure.
292	ME Region write access permissions don't match Intel recommended values.
293	Mismatch on FPF file %s - UEP: %s, FPF HW: %s.
294	FPFs are not committed to HW.
295	BIOS Region write access permissions don't match Intel recommended values.
296	Failed to read FPF HW.
297	SOC Config Lock is not set.
298	Lock bit FPF is not set on file.

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

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

Error Code	Error Message
356	PSVN partition is full, so cannot update to higher SVN.
357	Restore Point Image was requested, but it is not allowed because CSE is in Recovery Mode.
358	Display Partition Version failed.
359	Restore Point Image was requested, but there was Full/Partial FW Update before without Restart after it.
360	Update to incompatible PMC: The PMC instance ID is different, which may be due to H/LP SKU incompatibility.
361	Update to incompatible H/LP SKU image.
362	Update Image length is bigger than the expected size of the image according to its size in the flash. For example: Error on updating from Consumer to Corporate.
363	Manifest size in Update Image is too large.
364	Firmware update failed due to an internal error 364.
365	Firmware update failed due to an internal error 365.
366	Failed to verify signature of OEM or RoT key manifests. For example: Error on update from Production to Pre-Production.
367	Firmware update failed due to an internal error 367.
368	Firmware update failed due to an internal error 368.
369	Firmware update failed due to an internal error 369.
370	Manifest not found in partition (in Update or Flash Image).
371	Firmware update failed due to an internal error 371.
372	Loader failed to verify manifest signature of FTPR. Production vs. Pre-Production.
373	Loader failed to verify manifest signature of NFTP.
374	Loader failed to verify manifest signature of IDLM.
375	Loader failed to verify manifest signature of RBE.
376	Loader failed to verify manifest signature of PMC.
377	Loader failed to verify manifest signature of OEM KM.
378	Loader failed to verify manifest signature of WCOD.
379	Loader failed to verify manifest signature of LOCL.
380	Loader failed to verify manifest signature of PCHC.
381	Loader failed to verify manifest signature of IOMP.
382	Loader failed to verify manifest signature of NPHY.
383	Loader failed to verify manifest signature of TBTP.
384	Loader failed to verify manifest signature of ISHC.
385	Loader failed to verify manifest signature of IUNIT.
386	Some manifest extension is missing in FTPR.
387	Some manifest extension is missing in NFTP.

Error Code	Error Message
388	Some manifest extension is missing in IDLM.
389	Some manifest extension is missing in RBE.
390	Some manifest extension is missing in PMC. Wrong MEU Tool was used to create the partition.
391	Some manifest extension is missing in OEM KM. Wrong MEU Tool was used to create the partition.
392	Some manifest extension is missing in WCOD.
393	Some manifest extension is missing in LOCL.
394	Some manifest extension is missing in PCHC. Wrong MEU Tool was used to create the partition.
395	Some manifest extension is missing in IOMP. Wrong MEU Tool was used to create the partition.
396	Some manifest extension is missing in NPHY. Wrong MEU Tool was used to create the partition.
397	Some manifest extension is missing in TBTP. Wrong MEU Tool was used to create the partition.
398	Some manifest extension is missing in ISHC. Wrong MEU Tool was used to create the partition.
399	Some manifest extension is missing in IUNIT. Wrong MEU Tool was used to create the partition.
400	FTPR partition hash and calculated hash are not the same. If partition hash is zero - wrong MEU Tool was used to create the partition.
401	NFTP partition hash and calculated hash are not the same. If partition hash is zero - wrong MEU Tool was used to create the partition.
402	DLMP partition hash and calculated hash are not the same. If partition hash is zero - wrong MEU Tool was used to create the partition.
403	RBEP partition hash and calculated hash are not the same. If partition hash is zero - wrong MEU Tool was used to create the partition.
404	PMCP partition hash and calculated hash are not the same. If partition hash is zero - wrong MEU Tool was used to create the partition.
405	OEMP partition hash and calculated hash are not the same. If partition hash is zero - wrong MEU Tool was used to create the partition.
406	WCOD partition hash and calculated hash are not the same. If partition hash is zero - wrong MEU Tool was used to create the partition.
407	LOCL partition hash and calculated hash are not the same. If partition hash is zero - wrong MEU Tool was used to create the partition.
408	PCHC partition hash and calculated hash are not the same. If partition hash is zero - wrong MEU Tool was used to create the partition.
409	IOMP partition hash and calculated hash are not the same. If partition hash is zero - wrong MEU Tool was used to create the partition.
410	NPHY partition hash and calculated hash are not the same. If partition hash is zero - wrong MEU Tool was used to create the partition.

Error Code	Error Message
411	TBTP partition hash and calculated hash are not the same. If partition hash is zero - wrong MEU Tool was used to create the partition.
412	ISHC partition hash and calculated hash are not the same. If partition hash is zero - wrong MEU Tool was used to create the partition.
413	IUNP partition hash and calculated hash are not the same. If partition hash is zero - wrong MEU Tool was used to create the partition.
414	Place holder. This error code will not be returned by the FW.
415	Place holder. This error code will not be returned by the FW.
416	Place holder. This error code will not be returned by the FW.
417	Place holder. This error code will not be returned by the FW.
418	PMCP must have the same major API version as the version inside the list in FTPR, in the Update Image for Full Update.
419	OEMP must have the same major API version as the version inside the list in FTPR, in the Update Image for Full Update.
420	WCOD must have the same major API version as the version inside the list in FTPR, in the Update Image for Full Update, in the Flash Image for Partial Update.
421	LOCL must have the same major API version as the version inside the list in FTPR, in the Update Image for Full Update, in the Flash Image for Partial Update.
422	PCHC must have the same major API version as the version inside the list in FTPR, in the Update Image for Full Update, in the Flash Image for Partial Update.
423	IOMP must have the same major API version as the version inside the list in FTPR, in the Update Image for Full Update, in the Flash Image for Partial Update.
424	NPHY must have the same major API version as the version inside the list in FTPR, in the Update Image for Full Update, in the Flash Image for Partial Update.
425	TBTP must have the same major API version as the version inside the list in FTPR, in the Update Image for Full Update, in the Flash Image for Partial Update.
426	ISHC must have the same major API version as the version inside the list in FTPR, in the Update Image for Full Update, in the Flash Image for Partial Update.
427	IUNP must have the same major API version as the version inside the list in FTPR, in the Update Image for Full Update, in the Flash Image for Partial Update.
428	The size of an Update partition is bigger than the size of the Flash partition.
429	Location of partition to backup is not inside NFTP.
430	The number of IUPS in the Update/Flash Image is bigger than MAX_IUPS.

Error Code	Error Message
431	Partition name inside IUPs list (in FTPR manifest extension) is not IUP.
432	Non-optional IUP (like LOCL, WCOD) inside IUPs list (in FTPR manifest extension) is not in the Update Image.
433	PMC partition is not in the Update Image.
434	It is not allowed to do Partial Update on this partition.
435	It is not allowed to do Partial Update on Type-C partitions, according to NVAR.
436	RBEP and NFTP must have the same version as FTPR, in the Update Image.
437	RBEP and NFTP must have the same SVN as FTPR, in the Update Image.
438	RBEP and NFTP must have the same VCN as FTPR, in the Update Image.
439	Non-optional IUPs (like LOCL, WCOD) must have the same major build version as FTPR, in the Update Image for Full Update, in the Flash Image for Partial Update.
440	Update IUP must not have SVN smaller than SVN of Flash IUP.
441	Update Image length is not the same as Flash Image length.
442	Update IUP must not have VCN smaller than VCN of Flash IUP.
443	Update from PV bit ON to PV bit OFF is not allowed.
444	Update to PV bit OFF on Revenue platform is not allowed.
445	Update to higher SVN must be an upgrade - to higher build version.
446	Update to higher SVN must be to a higher Hot Fix number (the third number in the build version).
447	Non-optional IUP (like LOCL, WCOD) inside IUPs list (in FTPR manifest extension) is not in the Flash Image.
448	A partition that was searched in the Update Image is not in it.
449	Update between engineering build vs regular build is not allowed. Both builds have to be the same type: regular or engineering build. Engineering build is 7000 and above. Regular build is below 7000.
450	OEM KM partition is not in the Update Image, but ISHC/IUNP is in the Update Image, which is not allowed.
451	ISHC/IUNP do not exist in the same way in the Update Image and in the Flash Image.
452	OEM KM partition is not in the Flash Image, but it is in the Update Image, which is not allowed.
453	Partial FW Update: the Update Image contains IUP that is different than the one that was requested to be updated in the Partial Update command.
454	The Partial Update Image size is different than the size of the IUP in it (as it is in the manifest). This means that the Update Image contains more (or less) than the IUP partition.

Error Code	Error Message
455	Firmware update failed due to an internal error 455.
456	Firmware update failed due to an internal error 456.
457	Invalid FW Update enabled state.
458	Firmware update failed due to an internal error 458.
459	Firmware update failed due to an internal error 459.
460	Get Restore Point Image is not allowed, because a previous Get Restore Point operation already started. Both operations will be aborted. (Get Restore Point can be started again after this).
461	Firmware update failed due to an internal error 461.
462	Heci message length is not as expected.
463	FWU_START_MSG Heci message contains invalid value in UpdateEnvironment. Value should be FWU_ENV_MANUFACTURING. (Other possible value: FWU_ENV_IFU is obsolete).
464	FWU_DATA Heci command was sent, but the FW Update wasn't started with FWU_START Heci command before it.
465	Firmware update failed due to an internal error 465.
466	FW Update is not possible on UFS Flash after End Of Post (after the OS is running). It is possible only before the OS is running using Bios Capsule Update.
467	DPHY must have the same major API version as the version inside the list in FTPR, in the Update Image for Full Update, in the Flash Image for Partial Update.
468	DPHY partition hash and calculated hash are not the same. If partition hash is zero - wrong MEU Tool was used to create the partition.
469	Some manifest extension is missing in DPHY. Wrong MEU Tool was used to create the partition.
470	Loader failed to verify manifest signature of DPHY.
471	Update to higher TCB SVN must be also to higher ARB SVN.
472	Invalid Partition ID. Use a Partition ID which is on the Flash Image.
473	Display Partition Vendor ID failed.
474	Wrong structure of Update Image.
475	Flash Image content is invalid.
476	Firmware update failed due to an internal error 476.
477	Firmware update failed due to an internal error 477.
478	Firmware update failed due to an internal error 478.
479	FWU_END Heci command was sent, but there was no FWU_DATA command before it.

Error Code	Error Message
480	FWU_DATA Heci command has invalid data length (too big).
481	FW Update process received Heci command message with unknown command type.
482	Cannot obtain ME Mode.
484	BIOS does not support boot measurements.
485	BIOS does not support Trusted Device Setup boot.
486	This command is not supported on Slim SKU.
487	ODM ID \\ System Integrator ID \\ Reserved ID: value already set.
488	File already exists.
489	ME FW version mismatch, actual value is - %s.
490	Intel(R) Gbe version mismatch, actual value is - %s.
491	BIOS version mismatch, actual value is - %s.
492	System UUID mismatch, actual value is - %s.
493	Intel(R) Wired LAN MAC address mismatch, actual value is - %s.
494	Intel(R) Wireless LAN MAC address mismatch, actual value is - %s.
495	Wireless LAN micro-code mismatch, actual value is - %s.
496	Firmware Update OEM ID mismatch, actual value is - %s.
497	Touch - Vendor ID mismatch, actual value is - %s.
498	Invalid PKI suffix file.
499	Update to Image with lower ARB SVN is not allowed.
500	RBEP and NFTP must have the same unique build as FTPR, in the Update Image.
501	Firmware update failed due to an internal error 501.
502	RPMB fuse is set. Cannot commit FPFs.
503	PCHC partition is not in the Update Image.
504	Mismatch between FPF UEP and HW values.
505	Invalid Update Image length, size is smaller than required.
506	The internal structure of the Update Image is corrupted.
507	Update Image has wrong structure for Full Update operation.
508	Update Image has wrong structure for Partial Update operation.
509	Mandatory partitions (FTPR / NFTP / RBEP) were not found in the Update Image.
510	Number of IUPs in FW exceeds allowed maximum.
511	Invalid config file. Unknown test name found - %s.
512	Missing a required partition manifest in the Update Image.
513	Missing a required partition manifest extension in the Update Image.

Error Code	Error Message
514	The VAR invalid data size.
515	Update Image size exceeds allocated buffer.
516	FW failed to read FWSTS register.
517	Firmware update failed due to an internal error Read file failed.
518	Firmware update failed due to an internal error 518.
519	Full FW Update using same version is not allowed. Include /s in command line to allow it.
520	Invalid MPHY length.
521	FW failed to set ISH configuration file.
522	PCIe connectivity failure. Unable to connect to vPro NIC through designated bus.
523	SMBUS connectivity failure. Unable to connect to vPro NIC through designated bus.
524	Conflict in OEM Data: Overlapping values of LSPCON Port Config and eDP Port Config found.
525	Invalid configuration server FQDN value.
526	Invalid host FQDN file.
527	One or more GPIO pads provided in file have invalid ownership mode set.
528	One or more GPIO pads provided in file have invalid pad mode set.
529	Two or more GPIO pads provided in file have same feature field value set.
530	One or more GPIO pads provided in file have invalid feature field value set.
531	Invalid cert hash file.
532	Invalid host FQDN domain name.
533	Invalid host FQDN hostname.
534	ODM ID \\ System Integrator ID \\ Reserved ID: invalid size.
535	ODM ID \\ System Integrator ID \\ Reserved ID: invalid value.
536	One or more GPIO pads provided in file have invalid pad address set (group / pad number).
537	Two or more GPIO pads provided in file have same pad address set.
538	Update this var is not supported if AMT is provisioned.
539	Unsupported combination of EHBC state and privacy level files.
540	CSE is in Recovery Mode but FWSTS registers report Normal Mode.
541	The Flash Image that was burned on the platform was corrupted. CSE is in Recovery Mode at first boot.
542	Clear option is not supported for Hashed vars.
543	Update FTPR must have the same NVM compatibility (SPI/UFS) as Flash FTPR.

Error Code	Error Message
544	Incorrect API version.
545	Update RBEP must have the same NVM compatibility (SPI/UFS) as Flash RBEP.
546	Update NFTP must have the same NVM compatibility (SPI/UFS) as Flash NFTP.
547	Update IUP must have the same NVM compatibility (SPI/UFS) as Flash IUP.
548	Flash wear out violation.
549	Flash corruption.
550	Profile is not selectable in BIOS.
551	Profile index is too large.
552	No such profile in flash.
553	Command is not supported after EOP.
554	No such record.
555	File not found.
556	Invalid record format.
557	UOB record is too large.
558	Clock is not configurable
559	Register is locked.
560	No valid PRE UOB.
561	No valid PERM UOB.
562	No data for this clock.
563	Profile index is current.
564	No bclk adjustment found.
565	Warm reset ramp is not supported.
566	UOB record is already invalid.
567	No profile exists.
568	Authentication failure.
569	Pending file.
571	Frequency is too high.
572	Pending to revert to default.
573	Pending to set profile.
574	Invalid profile
575	Invalid OEM data.
576	Failed to read dynamic record.
578	Frequency is too low.
580	SSC mode change is not supported.
581	Range Violation: SSC is too high.
582	Survivability sync disabled.

Error Code	Error Message
583	Warm reset required is too low.
584	Specified target ID does not exist.
585	Specified register does not exist.
586	Invalidate successful.
588	Valid UOB already present.
589	Waiting for power cycle.
590	Survivability table access violation.
591	Survivability table is too large.
592	EID does not exist.
593	Success translate only.
594	Failure in reading PCIe Data.
595	Failure in writing PCIe Data.
596	Invalid PCIe configuration data.
597	CMD not supported before DID.
598	FIA MUX error - max config sku mismatch.
599	FIA MUX error - no config found.
600	FIA MUX error - getting laned limit.
601	FIA MUX error - reading configuration from file.
602	FIA MUX error - prompting to global reset.
603	Invalid FIA MUX configuration.
604	FIA MUX error - reading configuration to file.
605	FIA MUX error - reading configuration from straps.
606	Max bundles record reached.
607	Specified PLL is not supported
608	Data item unsupported.
609	Oem profile card violation.
612	Invalid argument.
613	AMT Ipv4 Interface is disabled.
614	Interface does not exists.
615	Invalid user consent policy file.
616	Anti Rollback SVN feature is disabled.
617	Input file is too big.
618	PCHC FW version mismatch, actual value is - %s
619	Update of partition between engineering build vs regular build is not allowed.
620	Unknown hardware platform.
621	Unsupported hardware platform. %s
622	Input configurable file contains cse file name duplicates.

Error Code	Error Message
623	ISIF must have the same major API version as the version inside the list in FTPR, Partial Update.
624	ISIF partition hash and calculated hash are not the same. If partition hash is zero - wrong MEU Tool was used to create the partition.
625	Some manifest extension is missing in ISIF. Wrong MEU Tool was used to create the partition.
626	Loader failed to verify manifest signature of ISIF.
627	ISIC must have the same major API version as the version inside the list in FTPR, in the Update Image for Full Update, in the Flash Image for Partial Update.
628	ISIC partition hash and calculated hash are not the same. If partition hash is zero - wrong MEU Tool was used to create the partition.
629	Some manifest extension is missing in ISIC. Wrong MEU Tool was used to create the partition.
630	Loader failed to verify manifest signature of ISIC.
631	have the same FW Type and Sub-Type as Flash FTPR
632	SAMF must have the same major API version as the version inside the list in FTPR, in the Update Image for Full Update, in the Flash Image for Partial Update.
633	SAMF partition hash and calculated hash are not the same. If partition hash is zero - wrong MEU Tool was used to create the partition.
634	Some manifest extension is missing in SAMF. Wrong MEU Tool was used to create the partition.
635	Loader failed to verify manifest signature of SAMF.
636	PPHY must have the same major API version as the version inside the list in FTPR, in the Update Image for Full Update, in the Flash Image for Partial Update.
637	PPHY partition hash and calculated hash are not the same. If partition hash is zero - wrong MEU Tool was used to create the partition.
638	Some manifest extension is missing in PPHY. Wrong MEU Tool was used to create the partition.
639	Loader failed to verify manifest signature of PPHY.
640	Using driverless mode. Not initializing PCI.
641	Using driverless mode. Not initializing SPI.
668	CSE is in Unlocked Mode. FW Update is not possible.