

# **Tiger Lake Platform System Tools - Intel® Management Engine Firmware 15.0**

**User Guide**

---

***December 2019***

***Revision 1.0***

**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](http://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](http://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](http://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 © 2019, 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 .....	15
2.6	Generic System Requirements.....	16
2.7	Error Return .....	16
2.8	Usage of Double-Quote Character (") .....	17
2.9	Control Handler Support .....	17
3	Intel® Flash Image Tool .....	18
3.1	System Requirements .....	18
3.2	Flash Image Details.....	18
3.2.1	Flash Space Allocation.....	19
3.3	Required Files.....	19
3.4	Intel® Flash Image Tool.....	19
3.4.1	Configuration Files .....	20
3.4.2	Creating New Configuration .....	20
3.4.3	Opening Existing Configuration.....	20
3.4.4	Saving Configuration.....	20
3.4.5	Environment Variables .....	20
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 .....	27
3.4.10	VSCC Table.....	31
3.4.11	Adding New Table.....	32
3.4.12	Removing Existing VSCC Table .....	32
3.4.13	Modifying the Intel® Management Engine Region .....	33
3.4.14	Setting the Intel® Management Engine Region Binary File.....	33
3.4.15	Setting the Intel® PMC Binary File.....	33
3.4.16	Intel® Management Engine Section .....	34
3.4.17	Power .....	35
3.4.18	Manageability Application Section .....	36
3.4.19	Platform Protection .....	38
3.4.20	Provisioning Section .....	39
3.4.21	Gbe (LAN) Region Settings .....	40
3.4.22	Setting Gbe Region Length Option .....	41
3.4.23	Setting Gbe Region Binary File .....	41
3.4.24	Enabling/Disabling GbE Region.....	41
3.4.25	Modifying PDR Region .....	41
3.4.26	Setting PDR Region Length Option .....	42
3.4.27	Setting PDR Region Binary File .....	42
3.4.28	Enabling/Disabling PDR Region.....	42
3.4.29	Modifying BIOS Region.....	42



	3.4.30	Setting BIOS Region Length Parameter .....	43
	3.4.31	Setting the BIOS Region Binary File .....	43
	3.4.32	Enabling/Disabling the BIOS Region .....	43
	3.4.33	Building Flash Image.....	43
	3.4.34	Decomposing Existing Flash Image .....	44
	3.4.35	Command Line Interface .....	45
	3.4.36	Example – Decomposing Image and Extracting Parameters .....	46
	3.4.37	More Examples of FIT CLI .....	46
4		Flash Programming Tool .....	48
	4.1	System Requirements .....	48
	4.2	Flash Image Details.....	49
	4.3	Microsoft Windows® Required Files.....	49
	4.4	EFI Required Files .....	49
	4.5	Programming Flash Device.....	49
	4.5.1	Stopping Intel® ME SPI Operations .....	49
	4.6	Programming NVARs .....	50
	4.7	Usage.....	50
	4.8	Examples.....	57
	4.8.1	Complete SPI Flash Device with Binary File .....	57
	4.8.2	Program Specific Region .....	57
	4.8.3	Program SPI Flash from Specific Address.....	58
	4.8.4	Dump Full Image.....	59
	4.8.5	Dump Specific Region .....	59
	4.8.6	Display SPI Information.....	60
	4.8.7	Verify Image with Errors.....	60
	4.8.8	Verify Image Successfully .....	61
	4.8.9	Get Intel® ME settings.....	61
	4.8.10	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
	5.4	Intel® MEManuf –EOL Check.....	68
	5.4.1	ErrorAction Field.....	68
	5.4.2	MEManuf.xml File .....	69
	5.4.3	MEManuf –EOL Variable Check .....	132
	5.4.4	MEManuf –EOL Config Check.....	132
	5.4.5	Output/Result .....	132
	5.5	Examples.....	132
	5.5.1	Example 1 .....	132
6		Intel® MEInfo .....	136
	6.1	Windows® PE Requirements .....	136
	6.2	Usage.....	136
	6.3	Examples.....	151
	6.3.1	Consumer Intel® ME FW SKU .....	151
	6.3.2	Corporate Intel® ME FW SKU.....	154
	6.3.3	Checks Whether Computer Has Completed Set-up and Configuration Process .....	157
7		Intel® ME Firmware Update.....	158
	7.1	Requirements.....	158



7.2	Windows® PE Requirements .....	158
7.3	Enabling and Disabling Intel® FWUpdate .....	159
7.4	FWUpdate Flows .....	159
7.4.1	Full FWUpdate .....	159
7.4.2	Partial FWUpdate .....	159
7.5	Usage .....	160
7.6	Examples .....	162
7.6.1	Updates Intel® ME with Firmware Binary File .....	162
7.6.2	Partial Firmware Update .....	162
7.6.3	Display Supported Commands .....	163
8	UEFI Sample Application Leveraging FWUpdate API Library .....	165
8.1	Getting Started - FWUpdate Library .....	165
8.1.1	Introduction .....	165
8.1.2	Environment .....	165
8.1.3	Setup .....	165
8.1.4	Sample App .....	165
8.2	Function Description .....	176
8.2.1	Get Interfaces .....	176
8.2.2	Get Last Status .....	176
8.2.3	Check Policy .....	177
8.2.4	Check Policy Buffer .....	177
8.2.5	Verify OEM Id .....	178
8.2.6	Get Ipu Partition Attributes .....	178
8.2.7	Get FW Update Info Status .....	178
8.2.8	FW Update Query Status Get Response .....	179
8.2.9	FW Update Full – Using Buffer .....	179
8.3	FW Update Partial Buffer .....	180
8.3.1	PDT Data (Sensor Calibration Data) Update .....	181
8.3.2	ISH Firmware Version .....	181
9	Intel® Manifest Extension Utility (Intel® MEU) .....	182
9.1	Usage .....	182
Appendix A	: Intel® ME NVARs .....	183

## 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
Figure 3-5. Descriptor Region Length Parameter .....	25
Figure 3-6. Flash Settings > Flash Components .....	26
Figure 3-7. Flash Settings → Flash Configuration .....	27
Figure 3-8. Descriptor Region → Master Access Section .....	31
Figure 3-9. Add VSCC Table Entry Dialog .....	32
Figure 3-10. Deleting VSCC Table Entry Dialog .....	33
Figure 3-11. Intel® ME Kernel .....	35
Figure 3-12. Power .....	36
Figure 3-13. Manageability Application Section .....	37
Figure 3-14. Provisioning Configuration Section .....	39



Figure 3-15. Provisioning Configuration Section (Cont..)	40
Figure 3-16. GbE Region Options	41
Figure 3-17. PDR Region Options	42
Figure 3-18. BIOS Region Parameters	43

## Tables

Table 2-1. OS Support for Tools	15
Table 2-2. Tools Summary	16
Table 3-1. Flash Image Regions – Description	19
Table 3-2. Build Settings Dialog Options	23
Table 3-3. Region Access Control Table	28
Table 3-4. CPU/BIOS Access	29
Table 3-5. FIT Command Line Options	45
Table 4-1. Named Variables Options	50
Table 4-2. Command Line Options for fpt.efi, fpt.exe and fptw.exe	52
Table 4-3. FPT–closemnf Behavior	56
Table 4-4. Intel-Recommend Access Settings	56
Table 5-1. Options for Tool	66
Table 5-2. Intel® MEmanuf Test Matrix	68
Table 6-1. Intel® MEInfo Command Line Options	136
Table 6-2. List of Components that Intel® MEINFO Displays	138
Table 7-1. Image File Update Options	161



# Revision History

Revision Number	Description	Date
0.3	<ul style="list-style-type: none"><li>Initial Release.</li></ul>	March 2018
0.5	<ul style="list-style-type: none"><li>Added new <a href="#">Chapter 8 "UEFI Sample Application Leveraging FWUpdate API Library"</a></li><li>Updated <a href="#">Appendix B</a> with <a href="#">Appendix B.3 "FWUpdate API Library Errors"</a></li><li>Updated <a href="#">Appendix A</a>: Added the FPFs list</li><li>Updated <a href="#">Appendix A</a>: Added OEM Secure Boot Policy bit mapping</li><li>Updated <a href="#">Appendix A</a>: Updated NVARs naming</li></ul>	June 2018
0.7	<ul style="list-style-type: none"><li>Added details regarding the <a href="#">ErrorAction Field</a>.</li><li>Updated <a href="#">Error Return</a> sub-chapter.</li><li>Updated <a href="#">NVARs Descriptions</a>.</li><li>Updated <a href="#">Appendix B</a> with new system level error codes.</li><li>Updated <a href="#">Intel® ME Firmware Update</a> introduction paragraph.</li></ul>	February 2019
0.71	<ul style="list-style-type: none"><li>Removed "FWUpdLcl -generic" command from FWUpdate tool.</li><li>Updated Appendix B with new command line tools errors.</li></ul>	February 2019
0.72	<ul style="list-style-type: none"><li>Updated supported -CLOSEMNF arguments in FPT tool.</li></ul>	April 2019
0.73	<ul style="list-style-type: none"><li>Updated <a href="#">"Intel-Recommended Access Settings"</a> table.</li><li>Updated Command Line Tools Errors with new errors.</li></ul>	May 2019
0.8	<ul style="list-style-type: none"><li>Removed PMX Section</li><li>Removed -p -list -spibar -hashed -comparefpf Commands</li><li>Updated <a href="#">MEInfo Consumer Intel(R) ME FW SKU XML</a></li><li>Updated <a href="#">MEManuf. XML File</a></li><li>Removed fparts.txt</li><li>Updated Config Server FQDN length value</li><li>Split error list into a separated guide</li></ul>	September 2019
0.9	<ul style="list-style-type: none"><li>Updated the following tools outputs, configs, examples, and tables descriptions to reflect new changes:<ul style="list-style-type: none"><li>Intel® MEINFO</li><li>Intel® MEManuf</li><li>Intel® FPT</li><li>Intel® FWUpdateLcl.exe</li></ul></li><li>Updated Intel® Flash Image Tool chapter</li><li>Update Appendix A – Added/Modified NVARs</li></ul>	December 2019
1.0	<ul style="list-style-type: none"><li>Align Revision Number</li></ul>	December 2019





# 1 Introduction

---

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

## 1.1 Terminology

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

- Tiger Lake PCH platform – All tools in the Tiger Lake PCH FW release kit are designed for 9<sup>th</sup> Generation Intel® Core™ Processor and Tiger Lake 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 9<sup>th</sup> Generation Intel® Core™ Processor or Tiger Lake PCH platform.
- Intel® vPro® platform – All features listed in this document are available for Intel® vPro® platforms with Intel® ME FW 15.0. 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 are provided for the following Intel® ME FW 15.0 SKUs: Consumer Intel® ME FW SKU and Corporate Intel® ME FW SKU.
  - 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:**
  - a. Combines the Descriptor, GbE, BIOS, PDR, ISH and Intel® ME FW binaries into one image.
  - b. Configures soft straps and NVARs for Intel® ME settings and another for Outputs that can be programmed by a flash programming device (FPT Tool).
- **Intel® FPT:**
  - a. Programs the SPI flash memory of individual regions or the entire flash device.
  - b. 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 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 Programming 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

**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	Connectivity between Intel® ME Devices	X
Intel® MEInfo	Firmware Aliveness – outputs certain Intel® ME parameters	X
Intel® FPT	Programs the image onto the flash memory and Programming NVARs / FPFs	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 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 Tiger Lake 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:

- Descriptor region
- BIOS
- Intel integrated LAN (GbE)
- IFWI: Intel® ME and PMC
- EC
- Platform Descriptor Region
- ISH
- Sub-Partitions
  - i. IUnit- Sub-Partition
  - ii. PCH Configuration Sub-Partition
  - iii. GBST Configuration Sub-Partition

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

### 3.2.1 Flash Space Allocation

Space allocation for each region is determined as follows:

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

### 3.3 Required Files

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

- vsccommn.bin
- .xml file

### 3.4 Intel® Flash Image Tool

Refer following for further information:

- General configuration information – Refer FW Bringup Guide from the appropriate Intel® ME FW kit.
- Detailed information on how to configure PCH Soft Straps and VSCC information – Refer Tiger Lake 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**

▼ Image Build Settings

Parameter	Value	Help Text
Output Path	\$DestDir\outimage.bin	-
FWUpdate Output Path	\$DestDir\FWUpdate.bin	-
Build FWUpdate With Full Image	No	-
Generate Intermediate Files	Yes	-
Enable Boot Guard warning me...	Yes	-
Enable Intel (R) Platform Trust ...	Yes	-
Region Order	53241	1=BIOS, 2=ME/IFWI, 3=GbE, 4=PDR, 5=EC
IfwiBuildVersion	0x0	32-bit value to use as the IFWI build version number
Redundancy Enabled	false	Enable Redundancy support for critical layout components
MRP Enabled	false	Enable MRP support for critical layout components
Read-Only protection for Minim...	No	Set this to Yes if you wish to have the start of CSE region up to boot par
Default Data Partition Enabled	false	Enable CSE Default Data partition
Intel(R) Manifest Extension Utili...		-
Signing Tool Path		-
Signing Tool	OpenSSL	-

▼ Environment Variables

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

- 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.
- Repeat Step 2 until the directories of all relevant environment variables have been defined.
- 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.

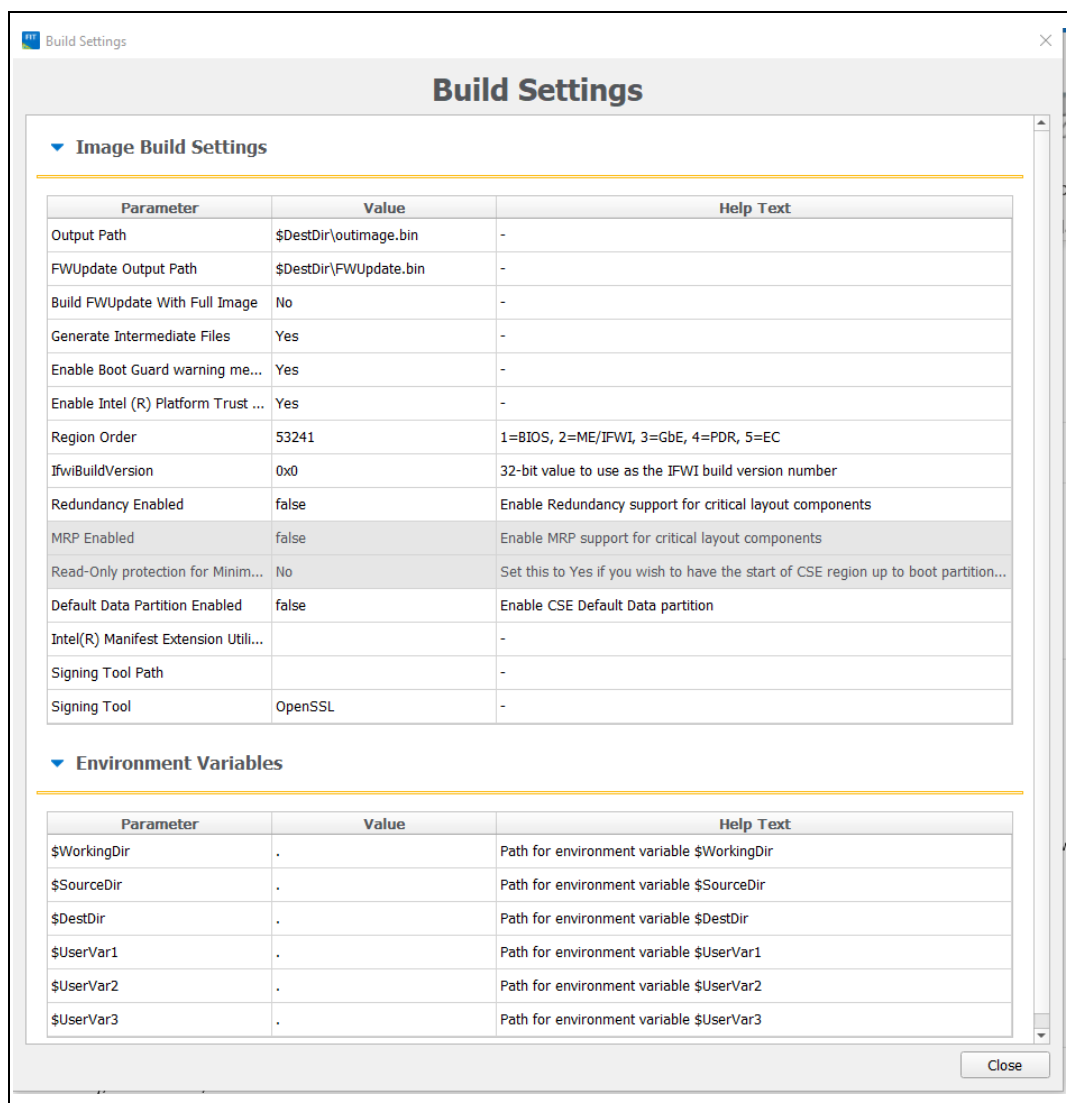
**To modify the build setting:**

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**

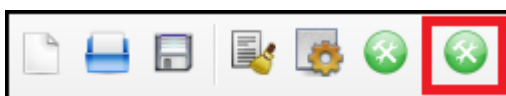
Option	Description
Output path	The path and filename where the final image should be saved after it is built. <b>NOTE:</b> Using the \$DestDir environment variable makes the configuration more portable.
FWUpdate Output Path	The path and filename where the FWUpdate image should be saved after it is built.
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
Region Order	1=BIOS, 2=ME/IFWI, 3=GbE, 4=PDR, 5=EC
IFWIBuildVersion	32-bit value to use as the IFWI build version number
Redundancy Enabled	Enable Redundancy support for critical layout components
MRP Enabled	Enable MRP support for critical layout components
Read-Only protection for Minimal Recovery code	Set this to Yes if you wish to have the start of CSE region up to boot partition 1 to be read only
Default Data Partition Enabled	Enable CSE Default Data partition
Intel(R) Manifest Extension Utility Path	The path and filename where the final image should be saved
Signing Tool Path	The path and filename where the final Signing Tool
Signing Tool	If the tool is disabled or open to SSL

**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 **OEM Section Binary** 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

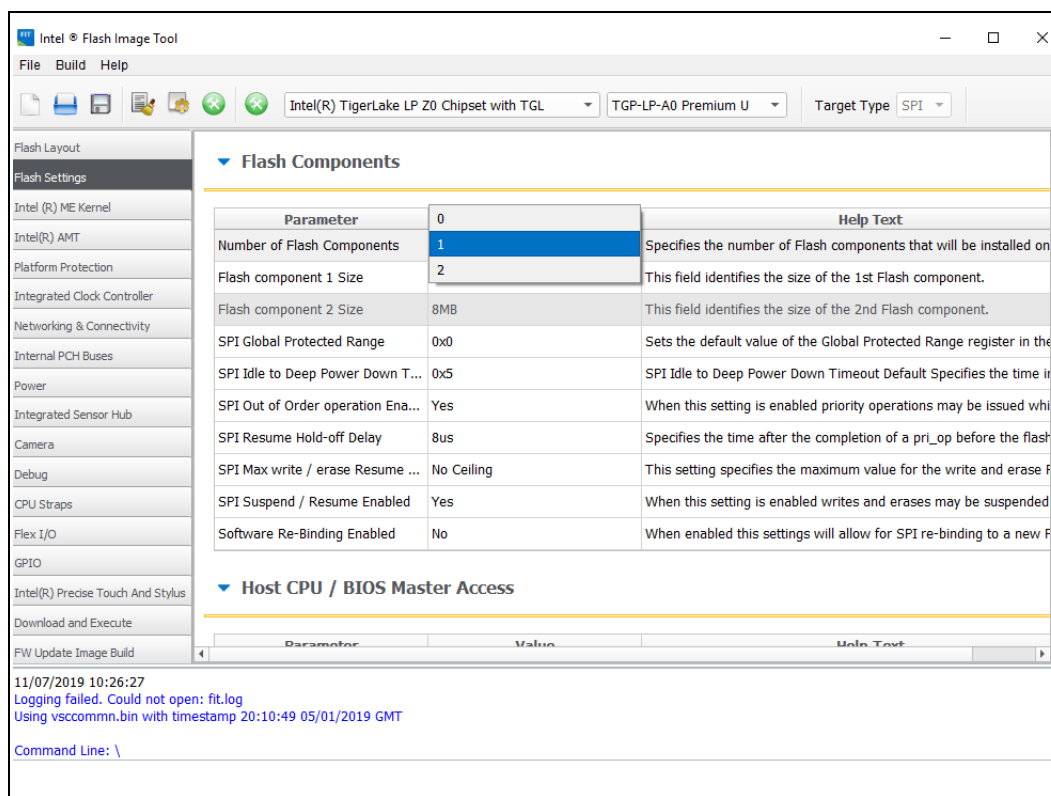
▼ Descriptor Region		
Parameter	Value	Help Text
OEM Section Binary		This loads the OEM Section binary that will be merged into the o

### 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- 6](#) all the parameters in the Flash Component section are listed in the right pane.

**Figure 3-6. Flash Settings > Flash Components**



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 0 ,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 0 /1 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 Enable	Yes	This soft-strap only has effect if Du
Dual Output Read Enable	Yes	This soft-strap only has effect if Du
Fast Read Clock Frequency	50MHz	This setting allows customers to co
Fast Read Supported	Yes	This setting allows customers to en
Invalid Instruction 0	0x21	This setting allows customers to co
Invalid Instruction 1	0x42	This setting allows customers to co
Invalid Instruction 2	0x60	This setting allows customers to co
Invalid Instruction 3	0xAD	This setting allows customers to co
Invalid Instruction 4	0xB7	This setting allows customers to co
Invalid Instruction 5	0xB9	This setting allows customers to co
Invalid Instruction 6	0xC4	This setting allows customers to co
Invalid Instruction 7	0xC7	This setting allows customers to co
Quad I/O Read Enable	Yes	This soft-strap only has effect if Qu
Quad Output Read Enable	Yes	This soft-strap only has effect if Qu
Read ID and Read Status Clock ...	50MHz	This setting allows customers to co
Write and Erase Clock Frequency	50MHz	This setting allows customers to co

### 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**

<b>Master Read/Write Access</b>				
<b>Region (#)</b>	<b>CPU and BIOS</b>	<b>ME/PCH</b>	<b>GbE Controller</b>	<b>EC</b>
Descriptor (0)	Not Accessible	Not Accessible	Not Accessible	Not Accessible
BIOS (1)	CPU and BIOS can always read from and write to BIOS region	Read / Write	Read / Write	Read / Write
ME (2)	Read / Write	ME can always read from and write to ME region	Read / Write	Read / Write
GbE (3)	Read / Write	Read / Write	GbE software can always read from and write to GbE region	Read / Write
PDR (4)	Not Accessible	Not Accessible	Not Accessible	Not Accessible
EC - Embedded Controller (Optional) (8)	Read / Write	Read / Write	Read / Write	EC can always read from and write to EC region
SubPartitions	Read / Write	PCH Configuration can always read from and write to SubPartition region	Read / Write	Read / Write
<b>NOTES:</b> 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 0xFFFF 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 0x00F / 0x01F 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 Access I...	0xFFFF	This setting determines write access con
Host CPU / BIOS Write Access ...	0x0000	This setting determines write access con
Host CPU / BIOS Read Access I...	0xFFFF	This setting determines read access conl
Host CPU / BIOS Read Access C...	0x0000	This setting determines read access conl
▼ Intel(R) ME Master Access		
Parameter	Value	
Intel(R) ME Write Access Intel ...	0xFFFF	This setting determines write access con
Intel(R) ME Write Access Custom	0x0000	This setting determines read access conl
Intel(R) ME Read Access Intel R...	0xFFFF	This setting determines read access conl
Intel(R) ME Read Access Custom	0x0000	This setting determines read access conl
▼ GbE Master Access		
Parameter	Value	
GbE Write Access Intel Recom...	0xFFFF	This setting determines read access conl
GbE Write Access Custom	0x0000	This setting determines read access conl
GbE Read Access Intel Recomm...	0xFFFF	This setting determines read access conl
GbE Read Access Custom	0x0000	This setting determines read access conl

### 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 Tiger Lake 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. The Tiger Lake PCH-LP SPI Programming Guide should be used to calculate the VSSC values. For C620 family of workstation systems, use the LBG SPI Programming Guide for further reference concerning the VSCC table definitions.

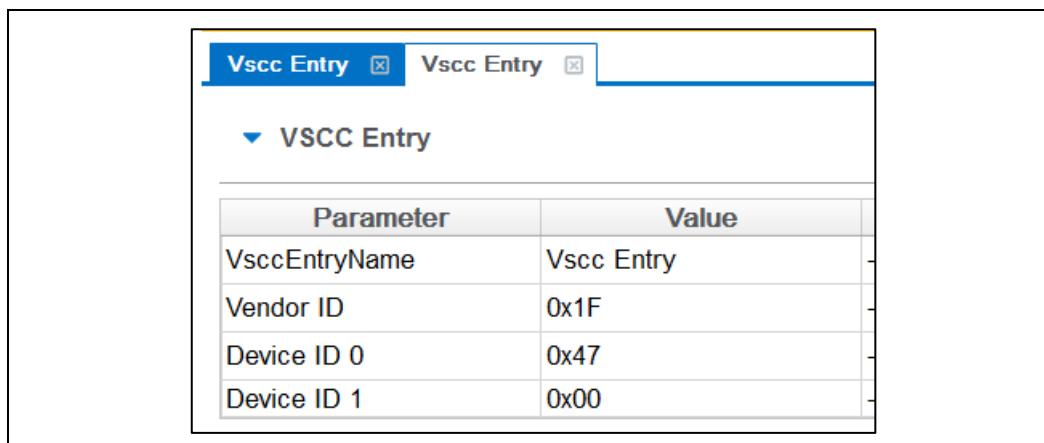
### 3.4.12 Removing Existing VSCC Table

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**

▼ Processor		
Parameter	Value	
Processor Emulation	No Emulation	-
▼ Intel (R) ME Firmware Update		
Parameter	Value	
Firmware Update OEM ID	00000000-0000-0000-0000-000...	This setting allows configuration of an OEM unique ID to ensure that custo
Hide MEBx Firmware Update Co...	No	This setting allows customers to hide the Firmware Update option in the I
Intel(R) ME Region Flash Protec...	Yes	This setting enables descriptor unlock of the ME Region when the HMRFPV
▼ Image Identification		
Parameter	Value	
OEM Tag	0x00000000	-
▼ Firmware Diagnostics		
Parameter	Value	
Automatic Built in Self Test	Disabled	This setting enables the firmware Automatic Built in Self Test which is ex
▼ End of Manufacturing Configuration		
Parameter	Value	
EOM on First Boot Enabled	No	This setting detremines if End of Manufacturing will be triggered on first b
Flexible EOM setting options	Lock Descriptor and OEM Configs	This setting deteremines which settings will be automatically committed d
▼ MCTP Configuration		
Parameter	Value	
MCTP Stack Configuration	0x920030	Defines the ME's 8-bits MCTP Endpoint IDs for each SMBus physical interf
MctpDevicePortEc	0x02	-
MctpDevicePortSio	0x00	-
MctpDevicePortIsh	0x00	-
MctpDevicePortBmc	0x00	-
▼ Intel (R) ME Boot Configuration		

### 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	
SLP_S5# / GPD10 Signal Config...	Enable as SLP_S5#	This setting allows the user to assign
SLP_S3# / GPD4 Signal Configu...	Enable as SLP_S3#	This setting allows the user to assign
SLP_S4# / GPD5 Signal Configu...	Enable as SLP_S4#	This setting allows the user to assign
SLP_A# / GPD6 Signal Configur...	Enable as SLP_A#	This setting allows the user to assign
SLP_S0# Tunnel	Disabled	This setting Enables / Disables the tun
▼ Deep Sx		
Parameter	Value	
Deep Sx Enabled	Yes	This setting enables / disables suppor
▼ PchThermalReporting		
Parameter	Value	
Thermal Power Reporting Enab...	Yes	This setting enabled a once-per-secor

### 3.4.18 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 3-13. Manageability Application Section**

▼ Intel(R) AMT Configuration		
Parameter	Value	
Intel(R) AMT Supported	Yes	This setting allows customers to disable Intel(R) AMT on the platform and force the platform into Standalone mode.
Manageability Hardware Status	Enabled	This setting will permanently disable Intel(R) AMT hardware through platform FPFs. At End-of-Manufacture, the hardware status is set to Disabled.
Intel(R) ME Network Services Support	Yes	This setting allows customers to enable / disable Intel(R) ME Network Services on the platform. Note: This setting is only applicable when the hardware status is Enabled.
Manageability Application Support	Yes	This setting allows customers to permanently disable Intel(R) AMT and Standard Manageability mode.
Manageability Application Initial State	Enabled	This setting allows customers to determine the power up state for Intel(R) AMT or Standard Manageability mode.
Intel(R) AMT Idle Timeout	0xFFFF	This setting configures the idle timeout value before Intel(R) AMT enters into an off state.
Intel(R) AMT Watchdog Automatic	No	This setting allows customers to enable the Intel (R) ME firmware to trigger an automatic platform reset when the watchdog timer expires.
▼ KVM Configuration		
Parameter	Value	
Firmware KVM Screen Blanking	No	This setting enables KVM Screen blanking capabilities in the firmware image. Note: This feature is disabled by default.
KVM Redirection Supported	Yes	This setting allows customers to enable / disable the KVM Redirection capabilities of the firmware. No redirection is supported by default.
▼ Provisioning Configuration		
Parameter	Value	
Embedded Host Based Configuration	No	This setting allows customers to enable / disable Embedded Host Based Configuration. EHBC is primarily used for provisioning the platform.
PKI Domain Name Suffix		This setting allows OEMs to pre-configure the Domain Name Suffix used for PKI provisioning in their firmware.
▶ 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	
Redirection Localized Language	English	This setting allows customers to configure which localized language will be used initially by the redirection server.
Redirection Privacy / Security Level	Default	This setting allows customers to configure the Privacy and Security level for redirection operation.
▼ TLS Configuration		
Parameter	Value	
Transport Layer Security Support	Yes	This setting allows customers to enable / disable firmware Transport Layer Security support.

### 3.4.19 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**

▼ Content Protection		
Parameter	Value	
PAVP Supported	Yes	This setting determines if the Protected Audio Video Path (PAVP) feature will be permanently
HDCP Internal Display Port 1 - 5K	PortA	This setting determines which port is connected for 5K output on Internal Display 1. Note:
HDCP Internal Display Port 2 - 5K	None	This setting determines which port is connected for 5K output on Internal Display 2. Note: E
▶ Graphics uController		
▶ Hash Key Configuration for Bootguard / ISH		
▶ Boot Guard Configuration		
▶ Type-C Firmware Anti-Rollback Configuration		
▼ Intel(R) PTT Configuration		
Parameter	Value	
Intel(R) PTT Supported	Yes	This setting permanently disables Intel(R) PTT in the firmware image.
Intel(R) PTT initial power-up st...	Enabled	-
Intel(R) PTT Supported [FPF]	Yes	This setting will permanently disable Intel(R) PTT through platform FPFs. Caution: Using thi
▶ TPM Over SPI Bus Configuration		
▶ BIOS Guard Configuration		
▼ TXT Configuration		
Parameter	Value	
TXT Supported	No	This setting determines is enabled for the platform.
▶ Crypto Hardware Support		
▼ Platform Trusted Device Setup Support		
Parameter	Value	
Enable TDS Capabilities	No	This setting enables Intel(R) Trusted Device Setup on the platform

These options control the availability and visibility of FW features.

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

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

- Intel® Platform Trusted Technology
- Intel® Content Protection

### 3.4.20 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 3-14. Provisioning Configuration Section**

▼ Provisioning Configuration

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

▼ OEM Customizable Certificate 1

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

▼ OEM Customizable Certificate 2

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

▼ OEM Customizable Certificate 3

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

▼ OEM Default Certificate 1		
Parameter	Value	Help Text
Certificate Enabled	No	-
Certificate Friendly Name		Enter Hash Name. Maximum of 32 characters.
Certificate Stream		Enter raw hash string or certificate file.
▼ OEM Default Certificate 2		
Parameter	Value	Help Text
Certificate Enabled	No	-
Certificate Friendly Name		Enter Hash Name. Maximum of 32 characters.
Certificate Stream		Enter raw hash string or certificate file.
▼ OEM Default Certificate 3		
Parameter	Value	Help Text
Certificate Enabled	No	-
Certificate Friendly Name		Enter Hash Name. Maximum of 32 characters.
Certificate Stream		Enter raw hash string or certificate file.
▼ OEM Default Certificate 4		
Parameter	Value	Help Text
Certificate Enabled	No	-
Certificate Friendly Name		Enter Hash Name. Maximum of 32 characters.
Certificate Stream		Enter raw hash string or certificate file.

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

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

### 3.4.21 Gbe (LAN) Region Settings

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



**Figure 3-16. GbE Region Options**

▼ GbE Region		
Parameter	Value	
Length	0	-
GbE Binary File		This loads the Intel(R) Integrated LAN binary t
GbE Region Enable	Disabled	This option allows the user to enable or disabl
Image Id	0	This displays Image ID of the currently loaded
Major Version	0	This displays Major revision number of the cur
Minor Version	0	This displays Minor revision number of the cur
- - - - -		

### 3.4.22 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.

### 3.4.23 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.

### 3.4.24 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.

### 3.4.25 Modifying PDR Region

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

**Figure 3-17. PDR Region Options**

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

### 3.4.26 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.

### 3.4.27 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.

### 3.4.28 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.

### 3.4.29 Modifying BIOS Region

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

**Figure 3-18. BIOS Region Parameters**

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

### 3.4.30 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.

### 3.4.31 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.

### 3.4.32 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.

### 3.4.33 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.

### 3.4.34 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.

### 3.4.35 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.)
-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.

Option	Description
-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 <0-7>	Overrides the size of the 1st flash component (0=512KB, 1=1MB, 2=2MB, 3=4MB, 4=8MB, 5=16MB, 6=32MB, 7=64MB).
-flashsize2 <0-7>	Overrides the size of the 2nd flash component (0=512KB, 1=1MB, 2=2MB, 3=4MB, 4=8MB, 5=16MB, 6=32MB, 7=64MB).
-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.
-exp	Displays example usage of this tool.
-tbt <file>	Overrides the binary source file for the TBT region.
-sam <file>	Overrides the binary source file for the SAMF region.
-pphy <file>	Overrides the binary source file for the PPHY region.
-iom <file>	Overrides the binary source file for the IOM region.

### 3.4.36 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.

### 3.4.37 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 Closemfnf

**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:

11. A pre-programmed flash with a bootable BIOS image is plugged into a new computer.
12. The computer boots.
13. FPT is run and a new BIOS/Intel® ME/GbE image is written to flash.
14. The computer powers down.
15. 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**:

- fptw.exe – the executable used to program the final image file into the flash.
- 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 (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**:

- 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 -CVARS	Displays a list of the supported manufacturing configurable named variables (NVARs) [Intel(R) AMT HW Status].
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.
fpt.exe -U -N <NVAR name> -V <NVAR Value>	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.
-NORESET	NoReset option can follow FPT commands that either trigger or promote a reset. It delays the reset and allows users to bundle the resets into one
-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".
-I	Info. Displays information about the image currently used in the flash.
-F <file> [NOVERIFY]	Flash. Programs a binary file into an SPI flash. The user needs to specify the binary file to be flashed. FPT reads the binary, and then programs the binary into the flash. After a successful flash, FPT verifies that the SPI flash matches the provided image. Without specify the length with -L option, FPT will use the total SPI size instead of an image size.  The 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.

Option	Description
-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.
-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.

Option	Description
-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 &lt;file&gt;</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 FPT <code>-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 ( <code>-v</code> ).
-V <value>	Value. Specifies the value for the NVAR variable. The name of variable is specified in the Name flag. The Value flag must follow the Name flag.
-CLOSEMNF [-in <file>] [-noReset]	<p>End of Manufacturing. This option is executed at the end of manufacturing phase. This option does the following:</p> <ol style="list-style-type: none"> <li>1. Sets the Intel® ME manufacturing mode done bit (Global Locked bit).</li> <li>2. Verifies that the Intel® ME manufacturing mode done bit (Global Locked) is set.</li> <li>3. Sets the master region access permission in the Descriptor region to its Intel-recommended value</li> <li>4. Verifies that flash regions are locked.</li> </ol> <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 "NoReset" 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>It is now supported to run <code>-closemnf</code> in <code>master_access.xml</code></p> <p>Note: Running FPT-closemnf also sets the default value for any unprovisioning process. Run FPT <code>-closemnf</code> 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>If no Flexibility in EOM has been configured in either FIT/FPT, the closemnf will perform all 4 steps mentioned above. In case flexibility of EOM has been set, the closemnf will behave as configured.</p>

Option	Description
-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.
-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 <code>-hashed</code> option can be used to display the hash of a value instead of the clear-text value.
-VARS	Display Supported Variables. FPT uses this option to display all variables supported for the <code>-R</code> and <code>-COMPARE</code> commands. Note: This will no longer display UEP based values which are tied to configuring iFPF's
-COMMIT [-noReset]	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.
-DISABLEME	Disable the Management Engine.
-FPFS [Intel(R) AMT HW Status]	Displays a list of the FPFs [Intel(R) AMT HW Status]
-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		
EOM Settings	Flash Protection Mode	NVARs Configuration State	FPFs committed
<i>Lock(Flash, Config)</i> <i>** (Legacy EOM flow)</i>	<i>Protected</i>	<i>Unlocked</i>	<i>Yes</i>
<i>Lock (Config)</i>	<i>Unprotected</i>	<i>Locked</i>	<i>Yes</i>
<i>Lock(Flash)</i>	<i>Protected</i>	<i>Unlocked</i>	<i>Yes</i>
<i>Lock(none)</i>	<i>Unprotected</i>	<i>Unlocked</i>	<i>Yes</i>

\*\* Return value 0 indicates successful completion. In the second case, FPT –closemnf returns the following :

```
Intel (R) Flash Programming Tool Version: 15.0.0.xxxx
Copyright (C) 2005 - 2019, Intel Corporation. All rights reserved.
```

```
Reading HSFSTS register... Flash Descriptor: Valid
```

```
Warning: Proceeding with EOM flow will result in a global reset.
Are you sure you want to continue? <Y/N>: n
```

```
Warning: Close Manufacturing process was canceled by user.
```

```
FPT Operation Successful.
```

**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
<b>Note:</b> <ol style="list-style-type: none"> <li>‡ = Value dependent on if PDR is implemented and if Host access is desired.</li> <li>† = Optional BIOS access to the EC region.</li> <li>* = Optional EC Read access to BIOS.</li> </ol>				

**Notes:**

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



## 4.8 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.

### 4.8.1 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.

### 4.8.2 Program Specific Region

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

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

Intel (R) Flash Programming Tool Version: 15.x.x.xxxx
Copyright (C) 2005 - 2019, Intel Corporation. All rights reserved.

Reading HSFSTS register... Flash Descriptor: Valid

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.

### 4.8.3 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: 15.x.x.xxxx

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

Reading HSFSTS register... Flash Descriptor: Valid

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.

#### 4.8.4 Dump Full Image

```

fpt.exe -d imagedump.bin

EFI:
fpt.efi -d imagedump.bin

-----
Intel (R) Flash Programming Tool. Version:  15.x.x.xxxx
Copyright (c) 2005-2019, Intel Corporation. All rights reserved.

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

- 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

```

#### 4.8.5 Dump Specific Region

```

fpt.exe -d descdump.bin -desc

EFI:
fpt.efi -d descdump.bin -desc

-----
Intel (R) Flash Programming Tool. Version:  15.x.x.xxxx

```

```

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

Reading HSFSTS register... Flash Descriptor: Valid

- 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**.

## 4.8.6 Display SPI Information

```

fptw.exe -I
-----
Intel (R) Flash Programming Tool Version: 15.X.X.XXXX
Copyright (C) 2005 - 2019, Intel Corporation. All rights reserved.

Reading HSFSTS register... Flash Descriptor: Valid

--- Flash Image Information ---
Signature: VALID
Number of Flash Components: 1
    Component 1 - 32768KB (262144Kb)
Regions:
    DESC      - Base: 0x00000000, Limit: 0x00000FFF
    BIOS      - Base: 0x01400000, Limit: 0x01FFFFFF
    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: 32768KB, 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".

## 4.8.7 Verify Image with Errors

```
fpt.exe -verify outimage.bin
```

```

EFI:
fpt.efi -verify outimage.bin

-----

Intel(R) Flash Programming Tool. Version: 15.x.x.xxxx
Copyright (c) 2005-2019, Intel Corporation. All rights reserved.

Reading HSFSTS register... Flash Descriptor: Valid

-Verifying Flash [0x00000000]      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.

#### 4.8.8 Verify Image Successfully

```

fpt.exe -verify outimage.bin

EFI:
fpt.efi -verify outimage.bin

-----

Intel (R) Flash Programming Tool. Version: 15.x.x.xxxx
Copyright (c) 2005-2019, Intel Corporation. All rights reserved.
Platform: Intel(R) Qxx Express Chipset
Reading HSFSTS register... Flash Descriptor: Valid
-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.

#### 4.8.9 Get Intel® ME settings

```

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

-----

Intel (R) Flash Programming Tool. Version: 15.x.x.xxxx
Copyright (c) 2007-2014, Intel Corporation. All rights reserved.
Platform: Intel(R) Qxx Express Chipset

```

```
Reading HSFSTS register... Flash Descriptor: Valid
Variable: "Privacy/SecurityLevel"
Value: True / 01
Retrieve Operation: Successful
```

#### 4.8.10 CVAR Configuration File Generation (-cfggen)

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

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

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

##### Example FPT.CFG output:

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

CSME Measured Boot to TPM = 0x01

Config Server FQDN =

Config Server IPv6/IPv4 Address =
```

```

Config Server IPv6/IPv4 Port = 0x26F3
Delayed Authentication Mode Config = 0x00
Disable All Pre-Installed Cert Hashes = 0x01
Discrete vPro NIC on-board State = 0x00
Domain Name =
EOM Settings = 0x00
Embedded Host Based Config = 0x00
FW Update State = 0x01
Firmware KVM Screen Blanking = 0x00
GPIO = ""
Host Name =
Integrated Sensor Hub Supported = 0x01
Intel(R) AMT Idle Timeout = 0xFFFF
Intel(R) AMT Supported = 0x01
Intel(R) AMT WD Auto Reset = 0x00
Intel(R) ME Network Services Supported = 0x00
Intel(R) PTT Supported = 0x01
Intel(R) PTT initial power-up state = 0x01
Intel(R) Precise Touch Technology Supported = 0x00
KVM = 0x01
KVM Redirection Supported = 0x01
LSPCON Port Config = 0x00
;   MEBxPassword NVAR value is not displayed because it is stored
encrypted.
MEBxPassword =
Manageability App Supported = 0x01
Manageability App initial power-up state = 0x01
;   OEM Custom Cert 1 Certificate
;   All data is required to update the certificate.
;   See the Tools Users Guide for detailed explanation

```

```

; of required data and format.
OEM Custom Cert 1 Active      =
OEM Custom Cert 1 Friendly Name =
OEM Custom Cert 1 Stream      =

; OEM Custom Cert 2 Certificate
; All data is required to update the certificate.
; See the Tools Users Guide for detailed explanation
; of required data and format.
OEM Custom Cert 2 Active      =
OEM Custom Cert 2 Friendly Name =
OEM Custom Cert 2 Stream      =

; OEM Custom Cert 3 Certificate
; All data is required to update the certificate.
; See the Tools Users Guide for detailed explanation
; of required data and format.
OEM Custom Cert 3 Active      =
OEM Custom Cert 3 Friendly Name =
OEM Custom Cert 3 Stream      =

OEM Tag = 0x00000000

On Board Discrete vPro NIC SMBus address = 0x00

On dock vPro NIC SMBus address = 0x00

Opt-in Policy = 0x11

PAVP Supported = 0x01

PKI Domain Name Suffix =

RCFG/ZTC = 0x01

Redirection Privacy / Security Level = 0x01

SAM Configuration =

```

**§ §**



## 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, KVM, etc. This tool is meant to be run on the manufacturing line.

### 5.1 **Windows® PE Requirements**

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

### 5.2 **How to Use Intel® MEmanuf**

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

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

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

## 5.3 Usage

The Windows® version of the tool can be executed by:

```
MEManufWin64.exe [-EXP] [-H|?] [-VER] [-BLOCKNET] [-ALLOWNET]
                  [-TEST] [-S0] [-BISTRESULT] [-NEXTREBOOT] [-EOL]
                  [-CFGGEN] [-F] [-VERBOSE] [-PAGE] [-ALL] [-LEVEL]
```

**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.
-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

Option	Description
-BLOCKNET	<p><b>Note:</b> 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><b>Note:</b> 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.
-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.</p> <p>When -f flag is used along with a file name (&lt;filename&gt;), 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><b>Note:</b> 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>	Use this option along with a filename to generate a default configuration file. This file (with or without modification) can be used for the -EOL option. Rename it MEManuf.xml before using it. It is highly recommended to use this option to generate a new MEManuf.xml with an up-to-date variable names list before using the Intel® MEManuf End-Of-Line check feature.
-ALL	<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><b>Note:</b> Intel recommended tests will be enabled regardless of -all parameter to meet corresponding dependencies</p>
-VERBOSE <file>	Displays the debug information of the tool or stores it in a log file.
-PAGE	When it takes more than one screen to display all the information, this option lets the user pause the display and then press any key to continue on to the next screen.
-LEVEL[level between 1 & 3]	Select EOL tests level

**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:** 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.

## 5.4 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="Policy Kernel - Power Package : Live Heap
Test">
    <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
    <!-- Description>Allocate memory in live heap in M0,
write in M3, read back in M0.</Description -->
    <!-- IntelRequired>True</IntelRequired -->
    <!-- TestType>LIVE_HW</TestType -->
    <!-- End of uneditable fields -->
    <!-- Please edit the fields below ONLY with the State
or ErrAction -->
```

```

        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
    </csmebist>
    <csmebist name="Common Services - General : WLAN enabled
only on mobile or desktop">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>If platform is not desktop or mobile
(server) there should be no WLAN (value 0x80).</Description
-->
        <!-- IntelRequired>True</IntelRequired -->
        <!-- TestType>M0_CONFIG</TestType -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State
or ErrAction -->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
    </csmebist>
    <csmebist name="Common Services - Wireless LAN :
Connectivity to NIC">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Runs the Wlan test WLAN access
through Clink 1.</Description -->
        <!-- IntelRequired>True</IntelRequired -->
        <!-- TestType>M0_HW</TestType -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State
or ErrAction -->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
    </csmebist>
    <csmebist name="Policy Kernel - ME Configuration : Wlan
Power Well">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>WLAN power well setting.</Description
-->
        <!-- IntelRequired>True</IntelRequired -->
        <!-- TestType>M0_CONFIG</TestType -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State
or ErrAction -->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
    </csmebist>
    <csmebist name="Policy Kernel - ME Password : Validate
MEBx password">

```

```

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

```



```

        <!-- Please edit the fields below ONLY with the State
or ErrAction -->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
    </csmebist>
    <csmebist name="Common Services - LAN : Connectivity to
NIC in M0">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>LAN test runs only if AMT is not
permanently disabled and mDNSProxy is not
disabled.</Description -->
        <!-- IntelRequired>True</IntelRequired -->
        <!-- TestType>M0_HW</TestType -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State
or ErrAction -->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
    </csmebist>
    <csmebist name="Common Services - EHBC State : EHBC and
Privacy Level states compatibility">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Check while both EHBC and privacy
level are available, (PrivLevel != Default) && (EHBCState
== EHBC_STATE_ENABLE).</Description -->
        <!-- IntelRequired>True</IntelRequired -->
        <!-- TestType>M0_CONFIG</TestType -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State
or ErrAction -->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
    </csmebist>
    <csmebist name="Common Services - EHBC State : Valid
Embedded Host Based Configuration (EHBC) state">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Check if EHBC state is
available.</Description -->
        <!-- IntelRequired>True</IntelRequired -->
        <!-- TestType>M0_CONFIG</TestType -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State
or ErrAction -->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>

```

```

    </csmebist>
    <csmebist name="Common Services - Privacy Level : Valid
Privacy Level settings">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Check if privacy level is
available.</Description -->
        <!-- IntelRequired>True</IntelRequired -->
        <!-- TestType>M0_CONFIG</TestType -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State
or ErrAction -->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
    </csmebist>
    <csmebist name="AMT - KVM : Compression engine">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Test KVM compression
engine.</Description -->
        <!-- IntelRequired>True</IntelRequired -->
        <!-- TestType>M0_HW</TestType -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State
or ErrAction -->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
    </csmebist>
    <csmebist name="AMT - KVM : Compare engine">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Test KVM compare engine.</Description
-->
        <!-- IntelRequired>True</IntelRequired -->
        <!-- TestType>M0_HW</TestType -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State
or ErrAction -->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
    </csmebist>
    <csmebist name="AMT - EC : Basic connectivity">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Only on mobile, if power source is
DC, test fails.</Description -->
        <!-- IntelRequired>True</IntelRequired -->
        <!-- TestType>M0_HW</TestType -->

```

```

        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State
or ErrAction -->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
    </csmebist>
    <csmebist name="AMT - Power : Valid WLAN power well
(Mobile)">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Run the tests verifying the internal
variables.</Description -->
        <!-- IntelRequired>True</IntelRequired -->
        <!-- TestType>M0_CONFIG</TestType -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State
or ErrAction -->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
    </csmebist>
    <csmebist name="AMT - Power : Valid LAN power well">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Run the tests verifying the internal
variables.</Description -->
        <!-- IntelRequired>True</IntelRequired -->
        <!-- TestType>M0_CONFIG</TestType -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State
or ErrAction -->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
    </csmebist>
    <csmebist name="PAVP - General : Verify Edp and Lspcon
Configurations">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Check if LSPCON and 5K ports are
overlapped</Description -->
        <!-- IntelRequired>True</IntelRequired -->
        <!-- 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="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 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 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="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="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 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="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 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="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 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="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) AMT HW Status">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Check fpf Intel(R) AMT HW 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 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="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="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 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="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="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="Hex" 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="HswLPTU1.86C.0117.R00.1303102001">
</RequiredValue>
    </eolconfig>
    <eolconfig name="ME FW version">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Check Firmware Version against
expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- 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="System UUID">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Check System UUID against programmed
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>VPRO</Dependencies -->
        <!-- 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="See example" example="550e8400-
e29b-41d4-a716-446655440000"> </RequiredValue>
    </eolconfig>
    <eolconfig name="MAC address">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Check MAC address</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>VPRO|LAN|IPV4_LAN_HW</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="6 hex pairs separated by ':'"
example="00:01:12:A2:3B:45"> </RequiredValue>
    </eolconfig>
    <eolconfig name="Security Descriptor Override (SDO)
check">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Check SDO pin</Description -->
        <!-- IntelRequired>True</IntelRequired -->
        <!-- Dependencies>SPI_DEP</Dependencies -->
        <!-- 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="vPro TBT I2C Re-timer 4 address">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- 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="0x00"> </RequiredValue>
    </eolvar>
    <eolvar name="vPro TBT I2C Re-timer 3 address">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- 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="0x00"> </RequiredValue>
    </eolvar>
    <eolvar name="vPro TBT I2C Re-timer 2 address">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- 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="0x00"> </RequiredValue>
    </eolvar>
    <eolvar name="vPro TBT I2C Re-timer 1 address">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- 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="0x00"> </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="WLAN Power Well">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- 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/80/CoreWell/81/SusWell/82/MEWell/83/WLAN
Sleep via SLP_WLAN#/86" example="Disabled">
</RequiredValue>
    </eolvar>
    <eolvar name="Unconfigure On RTC">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- 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="Trusted Device Setup Supported">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- 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="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="TDS Signing Policy">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- 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="Seal Signing Requested/00/PMF
Signing Only/01/No Signing/02" example="Seal Signing
Requested"> </RequiredValue>
    </eolvar>
    <eolvar name="TDS Seal State">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- 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 And
Intact/01/Enabled And Broken/02" example="Disabled">
</RequiredValue>
    </eolvar>
    <eolvar name="TDS Reseal Timeout">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- 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="0x00"> </RequiredValue>
    </eolvar>
    <eolvar name="StorageState">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- 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="SOL">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->

```

```

        <!-- 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="Redirection Privacy / Security Level">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- 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="Default/01/Enhanced/02/Extreme/03"
example="Default"> </RequiredValue>
    </eolvar>
    <eolvar name="Redirection">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- 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="RCFG/ZTC">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->

```

```

        <!-- Dependencies>CORP</Dependencies -->
        <!-- 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="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="PKI Domain Name Suffix">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->

```

```

        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- 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="String" example="Any">
</RequiredValue>
    </eolvar>
    <eolvar name="PAVP Supported">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- 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="Opt-in Policy">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- 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="0x00"> </RequiredValue>
    </eolvar>
    <eolvar name="On dock vPro NIC SMBus address">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->

```

```

        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- 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="0x00"> </RequiredValue>
    </eolvar>
    <eolvar name="On Board Discrete vPro NIC SMBus address">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- 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="0x00"> </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="OEM Tag">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->

```



[illegible]





```

        <!-- Please edit the fields below ONLY with the State
or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="String" example="Any">
</RequiredValue>
    </eolvar>
    <eolvar name="OEM Default Cert 4 Active">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- 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/NotActive/00/True/01/Active/01"
example="False"> </RequiredValue>
    </eolvar>
    <eolvar name="OEM Default Cert 3 Stream">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- 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="0x000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000
000000000000000000000000"> </RequiredValue>
    </eolvar>
    <eolvar name="OEM Default Cert 3 Friendly Name">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->

```











```

        <!-- Please edit the fields below ONLY with the State
or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Disabled/00/Enabled/01"
example="Disabled"> </RequiredValue>
    </eolvar>
    <eolvar name="Manageability App Supported">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- 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="MEBxPassword">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- 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="32 hex pairs with space between
pairs" example="04 AB F3 45 03 1D EF A2 B7 E8 98 79 10 45
AB DE F2 35 49 A0 01 35 78 29 37 AB DE EF FA 10 EF 33">
</RequiredValue>
    </eolvar>
    <eolvar name="MCTP Device Ports">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->

```

```

        <!-- 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="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="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)/0
3" example="CoreWell"> </RequiredValue>
    </eolvar>
    <eolvar name="KVM Redirection Supported">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->

```



```

        <!-- Dependencies>CORP</Dependencies -->
        <!-- 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="KVM">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- 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) Precise Touch Technology
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) 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="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="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="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="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="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="Intel(R) AMT WD Auto Reset">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->

```

```

        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- 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) AMT Supported">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- 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) AMT Idle Timeout">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- 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="0x0000"> </RequiredValue>
    </eolvar>
    <eolvar name="Integrated Sensor Hub Supported">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->

```

```

        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- 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="Host Name">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- 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="String" example="Any">
</RequiredValue>
    </eolvar>
    <eolvar name="Firmware Update OEM ID">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- 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="Firmware KVM Screen Blanking">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->

```

```

        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- 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="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="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="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="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="Embedded Host Based 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>CORP</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="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="Domain Name">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- 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="String" example="Any">
</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="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>
    <eolvar name="Config Server IPv6/IPv4 Port">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- 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="0x0000"> </RequiredValue>
    </eolvar>
    <eolvar name="Config Server IPv6/IPv4 Address">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- 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="String" example="Any">
</RequiredValue>
    </eolvar>
    <eolvar name="Config Server FQDN">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- 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="String" example="Any">
</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="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>
    <!-- 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.

**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

```
Intel (R) MEManuf Version: 15.x.x.xxxx
Copyright (C) 2005 - 2019, Intel Corporation. All rights reserved.

LPC Device Id: A082.
Platform: Tigerlake Platform
General FW Information
    FW Status Register1          0x94000255
    FW Status Register2          0x39850106
    FW Status Register3          0x00000020
    FW Status Register4          0x00004004
    FW Status Register5          0x80000133
```

FW Status Register6	0x80400308
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	BringUp
FW Loading Phase Status	CM0_MKHI_HANDLER_STOP
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 6 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	
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.x.x.xxxx
Copyright (C) 2005 - 2019, Intel Corporation. All rights reserved.

LPC Device Id: A082.
Platform: Tigerlake Platform
```

```

General FW Information
  FW Status Register1      0x94000255
  FW Status Register2      0x88110106
  FW Status Register3      0x00000030
  FW Status Register4      0x00004004
  FW Status Register5      0x80000133
  FW Status Register6      0x80400308

  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         Maestro
  FW Loading Phase Status  UNKNOWN
  ME File System Corrupted No
  RPMC status              OK
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

Get Intel(R) ME test data command... done
Total of 19 Intel(R) ME test result retrieved


Policy Kernel - Power Package : Live Heap Test - Passed

Common Services - LAN : Connectivity to NIC in M3 - Passed

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
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 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 (MEInfoWin64.exe) requires administrator privileges to run under Windows® OS. The user needs to use the Run as Administrator option to open the CLI in Windows® 10.

### 6.1 Windows® PE Requirements

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

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

### 6.2 Usage

The executable can be invoked by:

```
EInfoWin64.exe [-EXP] [-H|?] [-VER] [-FITVER] [-FEAT]
               [-VALUE] [-FWSTS] [-VERBOSE] [-PAGE]
```

**Table 6-1. Intel® MEInfo Command Line Options**

Option	Description
-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 <code>-feat "PTT FPF"</code> .  If the values are identical, a message indicating success appears. If the values are not identical, the actual value of the feature is returned. Only one feature may be requested in a command line.
-FITVER	Displays FIT version information



Option	Description																																				
-FEAT <name> <column>	<p>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.</p> <p>Intel® MEINFO can retrieve all of the information detailed below. However, depending on the SKU selected, some information may not appear.</p> <p><b>Note:</b> For the EFI shell version you need to add additional "^" to enclose the text string in order for it to be properly parsed.</p>																																				
-FWSTS	<p>Decodes the Intel® ME FW status register value field and breaks it down into the following bit definitions for easy readability:</p> <p>Intel (R) MEInfo Version: 15.x.x.xxxx Copyright (C) 2005 - 2019, Intel Corporation. All rights reserved.</p> <p>General FW Information</p> <table> <tr> <td>FW Status Register1</td><td>0x94000255</td></tr> <tr> <td>FW Status Register2</td><td>0x39850106</td></tr> <tr> <td>FW Status Register3</td><td>0x00000020</td></tr> <tr> <td>FW Status Register4</td><td>0x00004004</td></tr> <tr> <td>FW Status Register5</td><td>0x80000133</td></tr> <tr> <td>FW Status Register6</td><td>0x80400308</td></tr> </table> <table> <tr> <td>Current FW State</td><td>Normal</td></tr> <tr> <td>Flash Partition Table</td><td>Valid</td></tr> <tr> <td>FW Memory State</td><td>CM0 with UMA</td></tr> <tr> <td>FW Initialization</td><td>Complete</td></tr> <tr> <td>BUP Loading state</td><td>Success</td></tr> <tr> <td>FW Error Code</td><td>No Error</td></tr> <tr> <td>FW Mode Of Operation</td><td>Normal</td></tr> <tr> <td>SPI Flash Log</td><td>Present</td></tr> <tr> <td>FW Loading Phase</td><td>BringUp</td></tr> <tr> <td>FW Loading Phase Status</td><td>CM0_MKHI_HANDLER_STOP</td></tr> <tr> <td>ME File System Corrupted</td><td>No</td></tr> <tr> <td>RPMC status</td><td>OK</td></tr> </table>	FW Status Register1	0x94000255	FW Status Register2	0x39850106	FW Status Register3	0x00000020	FW Status Register4	0x00004004	FW Status Register5	0x80000133	FW Status Register6	0x80400308	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	BringUp	FW Loading Phase Status	CM0_MKHI_HANDLER_STOP	ME File System Corrupted	No	RPMC status	OK
FW Status Register1	0x94000255																																				
FW Status Register2	0x39850106																																				
FW Status Register3	0x00000020																																				
FW Status Register4	0x00004004																																				
FW Status Register5	0x80000133																																				
FW Status Register6	0x80400308																																				
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	BringUp																																				
FW Loading Phase Status	CM0_MKHI_HANDLER_STOP																																				
ME File System Corrupted	No																																				
RPMC status	OK																																				
-VERBOSE <filename>	<p>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".</p> <p>This option works with no option and -feat.</p>																																				
-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	Consumer SKU	Corporate SKU	Specific Feature Dependency	Field Value
Tools Version	SW (Intel® MEInfo)	X	X	N/A	Version string Example: 15.x.y.ZZZZ; where x=minor, y = HF/MR, ZZZZ = Build Number.
<b>General FW Information</b>					
Platform Type	Intel® ME Kernel	X	X	Indicating the type of the platform	Indicating the type of the platform (Mobile, desktop, etc.)
FW Image Type	Intel® ME Kernel	X	X	N/A	Indicating the type of the FW image (Pre-Production, Production, etc.)
Last ME Reset Reason	Intel® ME Kernel	X	X	N/A	Power up/ Firmware reset/ Global system reset/ Unknown
BIOS Boot State	Intel® ME Kernel	X	X	N/A	Pre Boot/ In Boot/ Post Boot
Boot Critical Code Redundancy	Intel® ME Kernel	X	X	N/A	Enabled/Disabled
Current Boot Partition	Intel® ME Kernel	X	X	N/A	String
BIOS Recovery State	Intel® ME Kernel	X	X	N/A	N/A
CSME Measured Boot to TPM	Intel® ME Kernel	X	X	BIOS	Enabled/Disabled

Feature Name	Feature Data Source	Consumer SKU	Corporate SKU	Specific Feature Dependency	Field Value
Capability Licensing Service State	Intel® ME Kernel	X	X	Not available on Corporate SKU. N/A unless supported by FW feature capability	Enabled/Disabled
Crypto HW Support	Intel® ME Kernel	X	X	BIOS	Enabled/Disabled
FW Update State	Intel® ME Kernel	X	X	N/A	Enabled/Disabled/ Password Protected
Firmware Update OEM ID	Intel® ME Kernel	X	X	Only if fw image supports OEM Id	UUID for OEM to check during FW Update
Intel(R) ISH Power State	Intel® ME Kernel	X	X	N/A	Enabled/Disabled
Intel(R) PTT State	Intel® ME Kernel	X	X	FIT PTT is set to 'Enable'	Enabled/Disabled
Intel(R) PTT initial power-up state	Intel® ME Kernel	X	X	N/A	Enabled/Disabled
OEM Tag	Intel® ME Kernel	X	X	N/A	A 32bit Hexadecimal number
TCSS FW partial update	Intel® ME Kernel	X	X	FIT TCSS enable feature set to 'Enabled'	Enabled/Disabled
TLS State	Intel® ME Kernel	X	X	N/A	Enabled/Disabled
<b>Intel(R) ME Code Versions</b>					
BIOS Version	Intel® ME Kernel	X	X	MEBx needs to be present	Version String
MEBx Version	Intel® ME Kernel	X	X	MEBx needs to be present. Not available on Consumer SKU	Version String
GbE Version	Other (Directly reading from SPI)	X	X	GbE Region to be present in the image	A version string

Feature Name	Feature Data Source	Consumer SKU	Corporate SKU	Specific Feature Dependency	Field Value
MEI Driver Version	Other (Reading Windows® registry entries)	X	X	Only when Windows® Intel® MEI driver is installed	A version string
FW Version	Intel® ME Kernel	X	X	N/A	Version string 15.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
<b>IUPs Information</b>					
PMC FW Version	Intel® ME Kernel	X	X	PMC Region to be present in the image	Version string
OEM FW Version	Intel® ME Kernel	X	X	N/A	Version string
ISHC FW Version	Intel® ME Kernel	X	X	ISH region to be present in the image	Version string
IUN FW Version	Intel® ME Kernel	X	X	IUnit region to be present in the image	Version string
LOCL FW Version	Intel® ME Kernel	X	X	N/A	Version string
WCOD FW Version	Intel® ME Kernel	X	X	N/A	Version string
IOM FW Version	Intel® ME Kernel	X	X	IO Manageability Engine binary version	Version string
NPHY FW Version	Intel® ME Kernel	X	X	NPHY Binary version	Version string
TBT FW Version	Intel® ME Kernel	X	X	Thunderbolt™ Binary	Version string
PCHC FW Version	Intel® ME Kernel	X	X	N/A	Version string

Feature Name	Feature Data Source	Consumer SKU	Corporate SKU	Specific Feature Dependency	Field Value
<b>PCH Information</b>					
PCH Name	Intel® ME Kernel	X	X	N/A	String (e.g. TGL)
PCH Device ID	Intel® ME Kernel	X	X	N/A	String
PCH Revision ID	Intel® ME Kernel	X	X	N/A	PCH Stepping string
PCH SKU Type	Intel® ME Kernel	X	X	N/A	String (e.g. Pre-Production ES)
PCH Replacement State	Intel® ME Kernel	X	X	Should be enabled in Intel® FIT	Disabled/Enabled
PCH Replaceable Counter	Intel® ME Kernel	X	X	PCH Replacement should be enabled	Counter indicating the number that PCH has been replaced
PCH Unlocked State	Intel® ME Kernel	X	X	N/A	Enabled/Disabled
<b>Flash Information</b>					
Storage Device Type	Other (Directly reading from SPI)	X	X	Only when there are flash parts HW installed	SPI, eMMC, UFS
SPI Flash ID 1	Other (Directly reading from SPI)	X	X	Only when there are flash parts HW installed	A JEDEC ID number (in Hex)
RPMC	Intel® ME Kernel	X	X	FIT PTT RPMC Supported feature set to 'Yes'	Supported/Not Supported
RPMC Bind Counter	Intel® ME Kernel	X	X	N/A	Counter indicating the number that SPI flash has been rebound
RPMC Bind Status	Intel® ME Kernel	X	X	N/A	Pre-Bind/Bound

Feature Name	Feature Data Source	Consumer SKU	Corporate SKU	Specific Feature Dependency	Field Value
RPMC 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
BIOS Read Access	Other (Directly reading from SPI)	X	X	N/A	32 bits controlling read access permission of the CPU/BIOS
BIOS Write Access	Other (Directly reading from SPI)	X	X	N/A	32 bits controlling read write permission of the CPU/BIOS
GBE Read Access	Other (Directly reading from SPI)	X	X	N/A	32 bits controlling read access permission of the GbE
GBE Write Access	Other (Directly reading from SPI)	X	X	N/A	32 bits controlling write access permission of GbE
ME Read Access	Other (Directly reading from SPI)	X	X	N/A	32 bits controlling read access permission of ME
ME Write Access	Other (Directly reading from SPI)	X	X	N/A	32 bits controlling write access permission of ME
EC Read Access	Other (Directly reading from SPI)	X	X	N/A	32 bits controlling read access permission of EC
EC Write Access	Other (Directly reading from SPI)	X	X	N/A	32 bits controlling write access permission of EC

Feature Name	Feature Data Source	Consumer SKU	Corporate SKU	Specific Feature Dependency	Field Value
<b>FW Capabilities</b>					
Intel(R) Active Management Technology	Intel® ME Kernel	N/A	X	AMT status in the image (Corporate only)	Present/Enabled Present/Disabled Not Present/Disabled
Intel(R) Protected Audio Video Path	Intel® ME Kernel	X	X	PAVP status in the image	Present/Enabled Present/Disabled Not Present/Disabled
Intel(R) Dynamic Application Loader	Intel® ME Kernel	X	X	DAL status in the image	Present/Enabled Present/Disabled Not Present/Disabled
Intel(R) Platform Trust Technology	Intel® ME Kernel	X	X	PTT status in the image	Present/Enabled Present/Disabled Not Present/Disabled
Service Advertisement & Discovery	Intel® ME Kernel	X	X	N/A	Present/Enabled Present/Disabled Not Present/Disabled
Persistent RTC and Memory	Intel® ME Kernel	X	X	PRTC status in the image	Present/Enabled Present/Disabled Not Present/Disabled
<b>End of Manufacturing</b>					
EOM Settings	Intel® ME Kernel	X	X	Set in FIT or FPT. Defines the behavior of EOM triggering	Lock(Flash, Config) Lock(Flash, Config) on 1st Boot Lock(Config) Lock(Config) on 1st Boot Lock(Flash) Lock(Flash) on 1st Boot Lock(none) Lock(none) on 1st Boot
NVAR Configuration State	Intel® ME Kernel	X	X	Changes after triggering EOM	Locked/Unlocked
HW Binding State	Intel® ME Kernel	X	X	Changes after triggering EOM	Enabled/Disabled
Flash Protection Mode	Intel® ME Kernel	X	X	Changes after triggering EOM	Protected/Unprotected

Feature Name	Feature Data Source	Consumer SKU	Corporate SKU	Specific Feature Dependency	Field Value
FPF Committed	Intel® ME Kernel	X	X	Changes after triggering EOM	Yes/No
<b>Intel(R) Active Management Technology</b>					
Intel(R) AMT State in FW	Intel® ME Kernel	N/A	X	AMT status in the image (Corporate only)	Present/Enabled Present/Disabled Not Present/Disabled
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 "=")
IPv4 Address	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 ".")
AMT State	Intel® ME Kernel	N/A	X	N/A	Enabled/Disabled
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
Provisioning Mode	Intel® AMT	N/A	X	Only shown when AMT is enabled	N/A
Auto-BIST State	Intel® ME Kernel	X	X	FIT CM3 Autotest Enabled set to 'true'	Enabled/Disabled



Feature Name	Feature Data Source	Consumer SKU	Corporate SKU	Specific Feature Dependency	Field Value
Wired AMT 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
Localized Language	Intel® AMT	N/A	X	Intel® AMT CEM (a.k.a. Common Service) is used. Not available on Corporate SKU	Language (e.g. English)
Wireless C-Link Status	Intel® ME Kernel	X	X	Intel® Wireless LAN	Enabled/Disabled
Intel(R) SMLink0 MCTP Address	Intel® AMT	N/A	X	Intel® AMT CEM (a.k.a. Common Service) is used. Not available on Corporate Sku	String Address
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
AMT Global State	Intel® ME Kernel	N/A	X	N/A	Enabled/Disabled
Discrete vPro NIC on-board State	Intel® ME Kernel	N/A	X	vPro enabled	Enabled/Disabled
Intel(R) Manageability HW Status	Intel® ME Kernel	N/A	X	Manageability HW to be present on platform	Enabled/Disabled
On Board Discrete vPro NIC SMBus address	Intel® ME Kernel	N/A	X	vPro enabled	String Address

Feature Name	Feature Data Source	Consumer SKU	Corporate SKU	Specific Feature Dependency	Field Value
On dock vPro NIC SMBus address	Intel® ME Kernel	N/A	X	vPro enabled	String Address
Redirection Privacy / Security Level	Intel® AMT	N/A	X	Only shown when AMT is enabled	Default/Enhanced/Extreme /Unknown
TDS Reseal Timeout	Intel® AMT	N/A	X	Only on TDS capable platforms	Hex for timeout
TDS Seal State	Intel® AMT	N/A	X	Only on TDS capable platforms	Enabled/Disabled
TDS Signing Policy	Intel® AMT	N/A	X	Only on TDS capable platforms	Policy defined by Enterprise IT
Trusted Device Setup Supported	Intel® AMT	N/A	X	Only on TDS capable platforms	Enabled/Disabled
vPRO TBT Dock State	Intel® ME Kernel	N/A	X	vPro enabled	Enabled/Disabled
vPro TBT I2C Re-timer 1 address	Intel® ME Kernel	N/A	X	vPro enabled	A 32bit Hexadecimal number
vPro TBT I2C Re-timer 2 address	Intel® ME Kernel	N/A	X	vPro enabled	A 32bit Hexadecimal number
vPro TBT I2C Re-timer 3 address	Intel® ME Kernel	N/A	X	vPro enabled	A 32bit Hexadecimal number
vPro TBT I2C Re-timer 4 address	Intel® ME Kernel	N/A	X	vPro enabled	A 32bit Hexadecimal number
<b>Intel(R) Protected Audio Video Path</b>					
Widevine provisioning state	Intel® ME Kernel	N/A	X	N/A	Provisioned/Not Provisioned
Attestation KeyBox	Intel® ME Kernel	N/A	X	N/A	Provisioned/Not Provisioned
EPID Group ID	Intel® ME Kernel	N/A	X	N/A	A 32bit Hexadecimal number

Feature Name	Feature Data Source	Consumer SKU	Corporate SKU	Specific Feature Dependency	Field Value
EPID Re-key needed	Intel® ME Kernel	N/A	X	N/A	True/False
PAVP State	Intel® ME Kernel	N/A	X	N/A	Yes/No
<b>Security Version Numbers</b>					
Trusted Computing Base SVN	Intel® ME Kernel	X	X	BIOS	Counter indicating TCB SVN
PMC	Intel® ME Kernel	X	X	BIOS	Counter indicating PMC SVN
CSE	Intel® ME Kernel	X	X	BIOS	Counter indicating CSE SVN
ROT KM	Intel® ME Kernel	X	X	BIOS	Counter indicating ROT KM SVN
IDL M	Intel® ME Kernel	X	X	BIOS	Counter indicating IDLM SVN
OEM KM	Intel® ME Kernel	X	X	BIOS	Counter indicating OEM KM SVN
<b>FW Supported FPFs</b>					
1st & 2nd OEM Key Hash Revoked	Intel® ME Kernel	X	X	BIOS	Enabled/Disabled
1st & 2nd OEM Key Hash size	Intel® ME Kernel	X	X	BIOS	Enabled/Disabled
1st & 2nd OEM RSA Key size	Intel® ME Kernel	X	X	BIOS	Enabled/Disabled
BSMM Anti Rollback	Intel® ME Kernel	X	X	BIOS	Enabled / Disabled
DAL OEM Signing	Intel® ME Kernel	X	X	BIOS DAL should be enabled in the image	Enabled / Disabled
DNX Anti Rollback	Intel® ME Kernel	X	X	BIOS	Enabled / Disabled
DNX SVN	Intel® ME Kernel	X	X	BIOS	Hash of Public Key to verify Boot Policy Manifest
Error Enforcement Policy 0	Intel® ME Kernel	X	X	BIOS	Enabled / Disabled

Feature Name	Feature Data Source	Consumer SKU	Corporate SKU	Specific Feature Dependency	Field Value
Error Enforcement Policy 1	Intel® ME Kernel	X	X	BIOS	Enabled / Disabled
Glitch Detection Disabled	Intel® ME Kernel	X	X	BIOS	Enabled / Disabled
Glitch Detection Enabled	Intel® ME Kernel	X	X	BIOS	Enabled / Disabled
IDLM Anti Rollback	Intel® ME Kernel	X	X	BIOS	Enabled / Disabled
IDLM SVN	Intel® ME Kernel	X	X	BIOS	Hash of Public Key to verify Boot Policy Manifest
Intel PTT Anti Hammering	Intel® ME Kernel	X	X	BIOS	Hash of Public Key to verify Boot Policy Manifest
Intel PTT Encryption Key	Intel® ME Kernel	X	X	BIOS	Hash of Public Key to verify Boot Policy Manifest
Intel(R) AMT HW Status	Intel® ME Kernel	X	X	BIOS	Enabled / Disabled
Intel(R) PTT	Intel® ME Kernel	X	X	BIOS	Enabled / Disabled
OEM ID	Intel® ME Kernel	X	X	BIOS	Hash of Public Key to verify Boot Policy Manifest
OEM KM Anti Rollback	Intel® ME Kernel	X	X	BIOS	Enabled / Disabled
OEM KM SVN	Intel® ME Kernel	X	X	BIOS	Hash of Public Key to verify Boot Policy Manifest
OEM Key Manifest	Intel® ME Kernel	X	X	BIOS	Enabled / Disabled
OEM Key Revocation State	Intel® ME Kernel	X	X	BIOS	Enabled / Disabled
OEM Platform ID	Intel® ME Kernel	X	X	BIOS	Hash of Public Key to verify Boot Policy Manifest
OEM Secure Boot Policy	Intel® ME Kernel	X	X	BIOS	Hash of Public Key to verify Boot Policy Manifest
CPU Debugging	Intel® ME Kernel	X	X	BIOS	Enabled / Disabled
BSP Initialization	Intel® ME Kernel	X	X	BIOS	Enabled / Disabled

Feature Name	Feature Data Source	Consumer SKU	Corporate SKU	Specific Feature Dependency	Field Value
Protect BIOS Environment	Intel® ME Kernel	X	X	BIOS	Enabled / Disabled
Measured Boot	Intel® ME Kernel	X	X	BIOS	Enabled / Disabled
Verified Boot	Intel® ME Kernel	X	X	BIOS	Enabled / Disabled
Key Manifest ID	Intel® ME Kernel	X	X	BIOS	Hash of Public Key to verify Boot Policy Manifest
Force Boot Guard ACM	Intel® ME Kernel	X	X	BIOS	Enabled / Disabled
OEM key Hash RSA key size	Intel® ME Kernel	X	X	BIOS	Enabled / Disabled
PMC Anti Rollback	Intel® ME Kernel	X	X	BIOS	Enabled / Disabled
PMC SVN	Intel® ME Kernel	X	X	BIOS	Hash of Public Key to verify Boot Policy Manifest
Persistent PRTC Backup Power	Intel® ME Kernel	X	X	BIOS	Enabled / Disabled
RBE Anti Rollback	Intel® ME Kernel	X	X	BIOS	Enabled / Disabled
ROT Anti Rollback	Intel® ME Kernel	X	X	BIOS	Enabled / Disabled
ROT KM SVN	Intel® ME Kernel	X	X	BIOS	Hash of Public Key to verify Boot Policy Manifest
RPMB Monotonic Counters	Intel® ME Kernel	X	X	BIOS	Counter indication to RPMB
SOC Config Lock State	Intel® ME Kernel	X	X	BIOS	Enabled / Disabled
SPI Boot Source	Intel® ME Kernel	X	X	BIOS	Enabled / Disabled
Secure boot ACM SVN	Intel® ME Kernel	X	X	BIOS	Hash of Public Key to verify Boot Policy Manifest
Secure boot BSMM SVN	Intel® ME Kernel	X	X	BIOS	Hash of Public Key to verify Boot Policy Manifest
Secure boot KM Anti Rollback	Intel® ME Kernel	X	X	BIOS	Enabled / Disabled
Secure boot KM SVN	Intel® ME Kernel	X	X	BIOS	Hash of Public Key to verify Boot Policy Manifest

Feature Name	Feature Data Source	Consumer SKU	Corporate SKU	Specific Feature Dependency	Field Value
TXT Supported	Intel® ME Kernel	X	X	BIOS	Enabled / Disabled
UFS Boot Source	Intel® ME Kernel	X	X	BIOS	Enabled / Disabled
USB Port ID	Intel® ME Kernel	X	X	BIOS	Hash of Public Key to verify Boot Policy Manifest
1st & 2nd OEM Public Key Hash	Intel® ME Kernel	X	X	BIOS	SHA-256bit Hash entry
iTouch	SW (Intel® MEInfo)	X	X	iTouch	iTouch information includes: <ul style="list-style-type: none"> <li>• Device ID</li> <li>• HW Revision ID</li> <li>• FW Revision ID</li> <li>• Frame Size</li> <li>• Feedback Size</li> <li>• Sensor Mode</li> <li>• Maximum Number of Touch Point</li> <li>• SPI Frequency</li> </ul> SPI I/O Mode



## 6.3 Examples

This is a simple test that indicates whether the FW is alive. If the FW is alive, the test returns device-specific parameters. The output is from the Windows® version.

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

### 6.3.1 Consumer Intel® ME FW SKU

```

Intel (R) MEInfo Version: 15.x.x.xxxx
Copyright (C) 2005 - 2019, Intel Corporation. All rights reserved.

General FW Information

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
BIOS Recovery State           Unknown
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
Intel(R) PTT State            Enabled
Intel(R) PTT initial power-up state Enabled
OEM Tag                       0x00
TCSS FW partial update        Disabled
TLS State                     Enabled

Intel(R) ME Code Versions
BIOS Version                  TGLSFWI1.R00.2341.A00.1908181734
GbE Version                    0.3
MEI Driver Version             1932.15.0.1069
FW Version                     15.x.x.xxxx LP Consumer
LMS Version                    1932.15.0.1069

IUPs Information
PMC FW Version                 150.1.0.1008
OEM FW Version                 15.0.0.7062
ISHC FW Version                5.4.1.4307
IUN FW Version                 15.0.0.7054
IOM FW Version                 16.12.0.4108
NPHY FW Version                11.103.107.1018
TBT FW Version                 14.0.0.0116
PCHC FW Version                15.0.0.7008

PCH Information
PCH Name                       TGL
PCH Device ID                  A082
PCH Revision ID                A0

```

PCH SKU Type	Pre-Production ES
PCH Replacement State	Disabled
PCH Replaceable Counter	0
PCH Unlocked State	Enabled
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	0xFFFF
BIOS Write Access	0xFFFF
GBE Read Access	0xFFFF
GBE Write Access	0xFFFF
ME Read Access	0xFFFF
ME Write Access	0xFFFF
EC Read Access	0xFFFF
EC Write Access	0xFFFF
FW Capabilities	0x31309640
Intel(R) Protected Audio Video Path	Present/Enabled
Intel(R) Dynamic Application Loader	Present/Enabled
Intel(R) Platform Trust Technology	Present/Enabled
Persistent RTC and Memory	Present/Enabled
End Of Manufacturing	
EOM Settings	Lock(Flash,Config)
NVAR Configuration State	Unlocked
HW Binding State	Disabled
Flash Protection Mode	Unprotected
FPF Committed	No
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 SVNs	
PMC	1 [minimum allowed: N/A]
CSE	1 [minimum allowed: 1]
ROT KM	1 [minimum allowed: N/A]
IDL M	1 [minimum allowed: 1]
OEM KM	1 [minimum allowed: N/A]
FW Supported FPFs	
FPF	UEP
	*In Use
---	---



1st OEM Key Hash Revoked	Not set	Disabled
1st OEM Key Hash size	Not set	Enabled
1st OEM RSA Key size	Not set	Enabled
2nd OEM Key Hash Revoked	Not set	Disabled
2nd OEM Key Hash size	Not set	Disabled
2nd OEM RSA Key size	Not set	Disabled
BSMM Anti Rollback	Not set	Disabled
DAL OEM Signing	Not set	Unknown
DNX Anti Rollback	Not set	Disabled
DNX SVN	Not set	0x00
Error Enforcement Policy 0	Not set	Disabled
Error Enforcement Policy 1	Not set	Disabled
Glitch Detection Disabled	Not set	Enabled
Glitch Detection Enabled	Not set	Disabled
IDLM Anti Rollback	Not set	Enabled
IDLM SVN	Not set	0x00
Intel PTT Anti Hammering	Not set	0x00
Intel PTT Encryption Key	Not set	Not Revoked
Intel(R) AMT HW Status	Not set	Enabled
Intel(R) PTT	Not set	Enabled
OEM ID	Not set	0x00
OEM KM Anti Rollback	Not set	Disabled
OEM KM SVN	Not set	0x00
OEM Key Manifest	Not set	Enabled
OEM Key Revocation State	Not set	Disabled
OEM Platform ID	Not set	0x00
OEM Secure Boot Policy	Not set	0x78
CPU Debugging	Not set	Enabled
BSP Initialization	Not set	Enabled
Protect BIOS Environment	Not set	Enabled
Measured Boot	Not set	Enabled
Verified Boot	Not set	Enabled
Key Manifest ID	Not set	0x01
Force Boot Guard ACM	Not set	Disabled
OEM key Hash RSA key size	Not set	Enabled
PMC Anti Rollback	Not set	Disabled
PMC SVN	Not set	0x00
Persistent PRTC Backup Power	Not set	Enabled
RBE Anti Rollback	Not set	Enabled
ROT Anti Rollback	Not set	Disabled
ROT KM SVN	Not set	0x00
RPMB Monotonic Counters	Not set	0x00
SOC Config Lock State	Not set	Disabled
SPI Boot Source	Not set	Enabled
Secure boot ACM SVN	Not set	0x00
Secure boot BSMM SVN	Not set	0x00
Secure boot KM Anti Rollback	Not set	Disabled
Secure boot KM SVN	Not set	0x00
TXT Supported	Not set	Enabled
UFS Boot Source	Not set	Disabled
USB Port ID	Not set	0x00
1st OEM Public Key Hash FPF	Not set	
1st OEM Public Key Hash UEP		
F8F0E369158176990A549ED4C36D1A8639D8873DEFF7ED2DE34CB41BCCB30476		
2nd OEM Public Key Hash FPF	Not set	

```
2nd OEM Public Key Hash UEP
0000000000000000000000000000000000000000000000000000000000000000
```

## 6.3.2 Corporate Intel® ME FW SKU

```
Intel (R) MEInfo Version: 15.x.x.xxxx
Copyright (C) 2005 - 2019, Intel Corporation. All rights reserved.
```

### General FW Information

Platform Type	Mobile
FW Image Type	Pre-Production
Last ME Reset Reason	Global system reset
BIOS Boot State	Post Boot
Boot Critical Code Redundancy	Disabled
Current Boot Partition	1
BIOS Recovery State	Unknown
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
Intel(R) PTT State	Enabled
Intel(R) PTT initial power-up state	Enabled
OEM Tag	0x00
TCSS FW partial update	Disabled
TLS State	Enabled

### Intel(R) ME Code Versions

BIOS Version	TGLSFWI1.R00.2341.A00.1908181734
MEBx Version	15.0.0.0001
GbE Version	0.3
MEI Driver Version	1932.15.0.1069
FW Version	15.x.x.xxxx LP Corporate
LMS Version	1932.15.0.1069

### IUPs Information

PMC FW Version	150.1.0.1008
OEM FW Version	15.0.0.7062
ISHC FW Version	5.4.1.4307
IUN FW Version	15.0.0.7054
LOCL FW Version	15.0.0.7075
WCOD FW Version	15.0.0.7075
IOM FW Version	16.12.0.4108
NPHY FW Version	11.103.107.1018
TBT FW Version	14.0.0.0116
PCHC FW Version	15.0.0.7008

### PCH Information

PCH Name	TGL
PCH Device ID	A082
PCH Revision ID	A0
PCH SKU Type	Pre-Production ES
PCH Replacement State	Disabled
PCH Replaceable Counter	0

PCH Unlocked State	Enabled
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	0xFFFF
BIOS Write Access	0xFFFF
GBE Read Access	0xFFFF
GBE Write Access	0xFFFF
ME Read Access	0xFFFF
ME Write Access	0xFFFF
EC Read Access	0xFFFF
EC Write Access	0xFFFF
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	Unlocked
HW Binding State	Disabled
Flash Protection Mode	Unprotected
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	0.0.0.0
AMT State	Disabled
Configuration State	Not Started
Provisioning Mode	PKI
Auto-BIST State	Disabled
Wired AMT Link Status	Link Up
Localized Language	English
Wireless C-Link Status	Enabled
Intel(R) SMLink0 MCTP Address	0xFF
System UUID	88888888-8887-8888-8888-878888888888
AMT Global State	Enabled
Discrete vPro NIC on-board State	Disabled
Intel(R) Manageability HW Status	Unknown
On Board Discrete vPro NIC SMBus address	0x00
On dock vPro NIC SMBus address	0x00
Redirection Privacy / Security Level	Default
TDS Reseal Timeout	0x06
TDS Seal State	Disabled
TDS Signing Policy	Seal Signing Requested

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 SVNs		
PMC	1	[minimum allowed: N/A]
CSE	1	[minimum allowed: 1]
ROT KM	1	[minimum allowed: N/A]
IDLM	1	[minimum allowed: 1]
OEM KM	1	[minimum allowed: N/A]
FW Supported FPFs		
	FPP	UEP
		*In Use
	---	---
1st OEM Key Hash Revoked	Not set	Disabled
1st OEM Key Hash size	Not set	Enabled
1st OEM RSA Key size	Not set	Enabled
2nd OEM Key Hash Revoked	Not set	Disabled
2nd OEM Key Hash size	Not set	Disabled
2nd OEM RSA Key size	Not set	Disabled
BSMM Anti Rollback	Not set	Disabled
DAL OEM Signing	Not set	Unknown
DNX Anti Rollback	Not set	Disabled
DNX SVN	Not set	0x00
Error Enforcement Policy 0	Not set	Disabled
Error Enforcement Policy 1	Not set	Disabled
Glitch Detection Disabled	Not set	Enabled
Glitch Detection Enabled	Not set	Disabled
IDLM Anti Rollback	Not set	Enabled
IDLM SVN	Not set	0x00
Intel PTT Anti Hammering	Not set	0x00
Intel PTT Encryption Key	Not set	Not Revoked
Intel(R) AMT HW Status	Not set	Enabled
Intel(R) PTT	Not set	Enabled
OEM ID	Not set	0x00
OEM KM Anti Rollback	Not set	Disabled
OEM KM SVN	Not set	0x00
OEM Key Manifest	Not set	Enabled
OEM Key Revocation State	Not set	Disabled
OEM Platform ID	Not set	0x00
OEM Secure Boot Policy	Not set	0x78
CPU Debugging	Not set	Enabled

BSP Initialization	Not set	Enabled
Protect BIOS Environment	Not set	Enabled
Measured Boot	Not set	Enabled
Verified Boot	Not set	Enabled
Key Manifest ID	Not set	0x01
Force Boot Guard ACM	Not set	Disabled
OEM key Hash RSA key size	Not set	Enabled
PMC Anti Rollback	Not set	Disabled
PMC SVN	Not set	0x00
Persistent PRTC Backup Power	Not set	Enabled
RBE Anti Rollback	Not set	Enabled
ROT Anti Rollback	Not set	Disabled
ROT KM SVN	Not set	0x00
RPMB Monotonic Counters	Not set	0x00
SOC Config Lock State	Not set	Disabled
SPI Boot Source	Not set	Enabled
Secure boot ACM SVN	Not set	0x00
Secure boot BSMM SVN	Not set	0x00
Secure boot KM Anti Rollback	Not set	Disabled
Secure boot KM SVN	Not set	0x00
TXT Supported	Not set	Enabled
UFS Boot Source	Not set	Disabled
USB Port ID	Not set	0x00
1st OEM Public Key Hash FPF	Not set	
1st OEM Public Key Hash UEP		
F8F0E369158176990A549ED4C36D1A8639D8873DEFF7ED2DE34CB41BCCB30476		
2nd OEM Public Key Hash FPF	Not set	
2nd OEM Public Key Hash UEP		
00		

### 6.3.3 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.x.x.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.x.x.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.

```
FWUpdLc164.exe [-H|?] [-VER] [-EXP] [-VERBOSE] [-F] [-Y]
                [-SAVE] [-FWVER] [-PARTID] [-INSTID] [-ALLOWSV]
                [-FORCERESET] [-SILENT] [-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 <Partition ID>	<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 an FTP 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
-PARTVENDOR	Display the Vendor ID of a specific partition
-INSTID <Instance ID>	Provide a specific instance ID of a partition to perform partial update
-SILENT	Update without any display or user prompts

## 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.x.x.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.x.x.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

```
Intel (R) Firmware Update Utility Version: 15.0.0.7075
```

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

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

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

```
<file>]]
```

```
[-SILENT] [-Y] [-ALLOWSV]
```

```
2. -F <file> -PARTID <Partition ID> -INSTID <Instance ID> [-
```

```
FORCERESET]
```

```
[-VERBOSE [ <file>]] [-SILENT] [-Y]
```

#### Language Codes

Language	Language Code
English	0x01
French	0x02
German	0x03
Chinese Traditional	0x04
Japanese	0x05
Russian	0x06
Italian	0x07
Spanish	0x08
Brazilian Portuguese	0x09
Korean	0x0A
Chinese Simplified	0x0B
Arabic	0x0C

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

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

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

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

## 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	<b>Id</b> - 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 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	<b>StatusFlags</b> - 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	<b>UpdateStatus</b> - indicates the current FW Update stage being executed. <b>TotalStages</b> - indicates the total number of FW Update stages available. <b>PercentWritten</b> - indicates the percentage complete of the FW Update process <b>LastUpdateStatus</b> - indicates the status of the fwupdate process just completed <b>LastResetType</b> - 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	<p><b>Buffer</b> – Buffer with the update image</p> <p><b>Buffer Length</b> – Length of buffer</p> <p><b>Password</b> – MEBX Password</p> <p><b>ForceResetLib</b> – Flag to perform system reset</p> <p><b>UpdateEnvironment</b> – differentiates various firmware update process environment within the firmware (manufacturing/non-manufacturing)</p> <p><b>UUID OEMID</b> – OEM ID</p> <p><b>update_flags</b> – flag to indicate FW of recovery/rollback</p> <p><b>Func pointer</b> – (bytes of Binary</p>
Returns	<p>0 = Success</p> <p>Non-zero value = Failure</p>

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

	<b>IpuUpdatedInfo</b> - 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	<b>Buffer</b> - Buffer <b>Buffer Length</b> - Length of buffer
Returns	<b>If Payload is sent to CSME successfully then Send Succeeded Message will be seen.</b>

### 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 3<sup>rd</sup> 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.

**Table A-1. NVARs Descriptions**

Fixed Offset Name	Description	Data Length (in Bytes)	Expected Value	Reset Type	Manufacturing Post EOM/Pre EOP
<b>Non-Application Specific Fixed Offset Item Descriptions</b>					
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) &lt;= char &lt;= ASCII(126)</p> <p>Cannot contain these characters: , : "</p> <p>Must contain for complexity:</p> <ol style="list-style-type: none"> <li>1. At least one Digit character (0 - 9)</li> <li>2. At least one 7-bit ASCII non alpha-numeric character above 0x20 (Example: ! \$ ;)</li> <li>3. Both lower-case and upper case Latin.</li> <li>4. Underscore and space are valid characters but are not used in determination of complexity.</li> </ol> <p>Refer <a href="#">Section Error! eference source not found.</a> for format and strong password requirements.</p>	8<=N<=32	Password	ME	Yes



OEMSKURule	<p>UINT32 (little endian) value. This controls which features are permanently disabled by OEM.</p> <p>There are reserved bits that 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 does not enable functionality that is not capable of working in the target hardware SKU. Refer 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	<div>Feature Capable: 1 Feature Permanently disabled: 0</div> <table><thead><tr><th>Bit</th><th>Description</th><th>Notes</th></tr></thead><tbody><tr><td>0</td><td>Intel(R) AMT Supported</td><td>1</td></tr><tr><td>1</td><td>Reserved</td><td></td></tr><tr><td>2</td><td>Manageability App Supported</td><td>1</td></tr><tr><td>3:9</td><td>Reserved</td><td></td></tr><tr><td>10</td><td>Integrated Sensor Hub Supported</td><td></td></tr><tr><td>11</td><td>Reserved</td><td></td></tr><tr><td>12</td><td>PAVP Supported</td><td></td></tr><tr><td>13:15</td><td>Reserved</td><td></td></tr><tr><td>16</td><td>Intel(R) ME Network Services Supported</td><td></td></tr><tr><td>17</td><td>Reserved</td><td></td></tr><tr><td>18</td><td>KVM</td><td>2</td></tr><tr><td>19:20</td><td>Reserved</td><td></td></tr><tr><td>21</td><td>TLS Supported</td><td></td></tr><tr><td>22:29</td><td>Reserved</td><td></td></tr><tr><td>29</td><td>Intel(R) PTT Supported</td><td></td></tr><tr><td>30:31</td><td>Reserved</td><td></td></tr></tbody></table> <div><b>NOTES:</b> 1. For corporate SKUs, both bits 0 and 2 need to be set to '1' to allow for Intel® AMT to work. 2. KVM (bit 18) should only be set to '1', when Manageability Application (bit 2) is set to '1'. If using a Corporate SKU, then Manageability Full (bit 0) must also be set to '1'.</div>	Bit	Description	Notes	0	Intel(R) AMT Supported	1	1	Reserved		2	Manageability App Supported	1	3:9	Reserved		10	Integrated Sensor Hub Supported		11	Reserved		12	PAVP Supported		13:15	Reserved		16	Intel(R) ME Network Services Supported		17	Reserved		18	KVM	2	19:20	Reserved		21	TLS Supported		22:29	Reserved		29	Intel(R) PTT Supported		30:31	Reserved		Global	No
Bit	Description	Notes																																																						
0	Intel(R) AMT Supported	1																																																						
1	Reserved																																																							
2	Manageability App Supported	1																																																						
3:9	Reserved																																																							
10	Integrated Sensor Hub Supported																																																							
11	Reserved																																																							
12	PAVP Supported																																																							
13:15	Reserved																																																							
16	Intel(R) ME Network Services Supported																																																							
17	Reserved																																																							
18	KVM	2																																																						
19:20	Reserved																																																							
21	TLS Supported																																																							
22:29	Reserved																																																							
29	Intel(R) PTT Supported																																																							
30:31	Reserved																																																							

Fixed Offset Name	Description	Data Length (in Bytes)	Expected Value	Reset Type	Manufacturing Post EOM/Pre EOP																		
FeatureShipState	<p>UINT32 (little endian) value. This controls which 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.</p> <p>This does not enable functionality that is not capable of working in the target hardware SKU. Refer the respective Firmware Bring-up Guide for a list of the features that are capable with what firmware bundle and Hardware SKU of Intel® 8 Series Chipset.</p>	4	<div>Feature Enabled: 1 Feature Disabled: 0</div> <table><thead><tr><th>Bit</th><th>Description</th><th>Note</th></tr></thead><tbody><tr><td>0:1</td><td>Reserved</td><td></td></tr><tr><td>2</td><td>Manageability App initial power-up state</td><td></td></tr><tr><td>3:28</td><td>Reserved</td><td></td></tr><tr><td>29</td><td>Intel(R) PTT initial power-up state</td><td></td></tr><tr><td>30:31</td><td>Reserved</td><td></td></tr></tbody></table> <p>When disabling PTT using Feature Shipment Time state NVAR, execute a reset after executing fpt.efi – commit to ensure PTT is disabled completely.</p>	Bit	Description	Note	0:1	Reserved		2	Manageability App initial power-up state		3:28	Reserved		29	Intel(R) PTT initial power-up state		30:31	Reserved		Global	Yes
Bit	Description	Note																					
0:1	Reserved																						
2	Manageability App initial power-up state																						
3:28	Reserved																						
29	Intel(R) PTT initial power-up state																						
30:31	Reserved																						

Fixed Offset Name	Description	Data Length (in Bytes)	Expected Value	Reset Type	Manufacturing Post EOM/Pre EOP
	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.				
WLAN Power Well	Sets, which power well the board uses for WLAN cards.	4	<b>0x80</b> = Disabled <b>0x81</b> = CoreWell <b>0x82</b> = Primary Well <b>0x83</b> = ME Well <b>0x86</b> = 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

Fixed Offset Name	Description	Data Length (in Bytes)	Expected Value	Reset Type	Manufacturing Post EOM/Pre EOP
GPIO	GPIO	1200	GPIO groups and pad range for each grp pad# For LP platforms: GPP_A 0-23 GPP_B 0-23 GPP_C 0-23 GPP_D 0-23 GPP_E 0-23 GPP_F 0-23 GPP_H 0-23 GPP_T 0-15 GPP_U 0-19 For H platforms: GPP_A 0-14 GPP_B GPP_C GPP_D 20-23 GPP_D Group Master GPP_E Group Master GPP_F Group Master GPP_G 0-15 GPP_H Group Master GPP_I Group Master GPP_K Group Master	<b>ME</b>	<b>No</b>
FW Update State	Enabled Firmware Update Capability	1	0 = Disabled 1 = Enabled 3 = Full and Partial disabled	<b>Global</b>	<b>Yes</b>
eDP Port Config	EDP Port Configuration. Up to two ports can be enabled 0x0 0x1 0x2 0x3 0x5 0x6	1	0x0:None/None    0x1:A/None 0x2:B/None        0x3:A/B 0x4:C/None        0x5:C/A 0x6:C/B	<b>Global</b>	<b>No</b>

Fixed Offset Name	Description	Data Length (in Bytes)	Expected Value	Reset Type	Manufacturing Post EOM/Pre EOP
LSPCON Port Config	LSPCON Port Configuration	0x0 0x2 0x4 0x8		Global	No
Attestation keyBox		16384	File		
CSME Measured Boot to TPM		1	0 = Disabled 1 = Enabled		
Discrete vPro NIC on-board State		1	0 = Disabled 1 = Enabled		
EOM Settings	FlexEom NVAR	3	{0, 1, 2, 3, 4, 5, 6, 7}, // PossibleValuesIDs {"Lock(Flash,Config)", "Lock(Flash,Config) on 1st Boot", "Lock(Config)", "Lock(Config) on 1st Boot", "Lock(Flash)", "Lock(Flash) on 1st Boot", "Lock(none)", "Lock(none) on 1st Boot"}, // possibleValuesStrings		
On Board Discrete vPro NIC SMBus address					
On dock vPro NIC SMBus address					
SAM Configuration		128			
TDS Reseal Timeout		8			
TDS Signing Policy		8	{0, 1, 2}, // PossibleValuesIDs {"Seal Signing Requested", "PMF Signing Only", "No Signing"}, // possibleValuesStrings		

Fixed Offset Name	Description	Data Length (in Bytes)	Expected Value	Reset Type	Manufacturing Post EOM/Pre EOP
Trusted Device Setup Supported		1	0 = Disabled 1 = Enabled		
vPRO TBT Dock State		1	0 = Disabled 1 = Enabled		
vPro TBT I2C Re-timer 1 address		8			
vPro TBT I2C Re-timer 2 address		8			
vPro TBT I2C Re-timer 3 address		8			
vPro TBT I2C Re-timer 4 address		8			
<b>AMT Related NVARs</b>					
OEM Custom Cert 1	Cert Hash Data. Refer Certificate Hash Entry Structure definition  If the platform is un-configured, the Certificate Hash is deleted.	55 => n >= 83	Valid Certificate Hash Entry (SHA1, SHA256 or SHA384)	<b>ME</b>	<b>Yes</b>
OEM Custom Cert 2	Cert Hash Data. Refer Certificate Hash Entry Structure definition  If the platform is un-configured, the Certificate Hash is deleted.	55 => n >= 83	Valid Certificate Hash Entry (SHA1, SHA256 or SHA384)	<b>ME</b>	<b>Yes</b>

Fixed Offset Name	Description	Data Length (in Bytes)	Expected Value	Reset Type	Manufacturing Post EOM/Pre EOP
OEM Custom Cert 3	Cert Hash Data. Refer Certificate Hash Entry Structure definition  If the platform is un-configured the Certificate Hash is deleted.	55 => n >= 83	Valid Certificate Hash Entry (SHA1, SHA256 or SHA384)	ME	Yes
Redirection Privacy / Security Level	Redirection (KVM, SOL, Storage Redirection) privacy level and configuration (RCFG, CCM) settings.	1	Default <b>0x01</b> Enhanced <b>0x02</b> Extreme <b>0x03</b> 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 = No 1 = Yes	ME	No

Fixed Offset Name	Description	Data Length (in Bytes)	Expected Value	Reset Type	Manufacturing Post EOM/Pre EOP
PKI Domain Name Suffix	PKI DNS Suffix. Null terminated string	223	PKI DNS Suffix in dotted string format	<b>ME</b>	<b>Yes</b>
Config Server FQDN	Configuration Server Fully Qualified Domain Name (FQDN)	256	Example: "intelFVE.com"	<b>ME</b>	<b>Yes</b>
RCFG/ZTC	R Configuration	1	0 = Disabled 1 = Enabled  This is update only NVAR. Tool won't be able to read expected value.	<b>ME</b>	<b>Yes</b>
*Redirection	<p>This is a bit-field indicating the enable/disable status of Storage Redirection, SOL, and KVM features in Intel® AMT.</p> <p>bit[0]: 1 – Storage Redirection enabled, 0 – disabled</p> <p>bit[1]: 1 – SOL enabled, 0 – disabled</p> <p>bit[2]: 1 – KVM enabled, 0 – disabled</p>	4	<p>Range: 0-7</p> <p>Example:</p> <p>Value of 4 (100b) indicates that KVM is enabled.</p> <p>Value of 3 (011b) indicates that Storage Redirection, and SOL are enabled.</p> <p>Value of 7 (111b) indicates that Storage Redirection, SOL, and KVM are enabled.</p> <p>This is update only NVAR. Tool won't be able to read expected value</p>	<b>ME</b>	<b>Yes</b>



Fixed Offset Name	Description	Data Length (in Bytes)	Expected Value	Reset Type	Manufacturing 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 <b>may not be</b> 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 <b>may be</b> 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	TigerLake	ME	Yes
Domain Name	Set Domain Name Only	192	<a href="http://myserver.intel.com">myserver.intel.com</a> <a href="http://amr.corp.intel.com">amr.corp.intel.com</a> <a href="http://www.intel.com">www.intel.com</a> <a href="mailto:mymail.somecollege.edu">mymail.somecollege.edu</a>	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
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
Disable All Pre-Installed Cert Hashes	Disable all Pre-installed Certificate Hashes	1	0 = Disabled 1 = Enabled  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	Manufacturing Post EOM/Pre EOP
Intel® AMT Idle Timeout	Change the Idle Timeout in minutes	2	Within Range: 1 – 0xFFFF	ME	Yes
Intel® AMT WD Auto Reset	Intel® AMT Watchdog Automatic Reset enabled	1	0 = Disabled 1 = Enabled	ME	No
<b>Field Programming Fuses</b>					
Glitch Detection	Enables/Disabled the Glitch detection FPF	1	0 = Disabled 1 = Enabled	ME	No
Intel® 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
Persistent PRTC Backup Power	Persistent PRTC Backup Power	1	0 = Disabled 1 = Enabled	ME	No
Protect BIOS Environment	Indicated if BIOS environment protection is enforced or not	1	0 = Disabled 1 = Enabled	ME	No

Fixed Offset Name	Description	Data Length (in Bytes)	Expected Value	Reset Type	Manufacturing Post EOM/Pre EOP
S3 Optimization	S3 optimization for Boot Guard	1	0 = Disabled 1 = Enabled	<b>ME</b>	<b>No</b>
Txt Supported	Txt Supported	1	0 = Disabled 1 = Enabled	<b>ME</b>	<b>No</b>
Verified Boot	One of the applicable profiles for Boot Guard	1	0 = Disabled 1 = Enabled	<b>ME</b>	<b>No</b>
OEM Public Key Hash	Hash of the provided OEM public key	32	32 Hex Pairs with space between pairs	<b>ME</b>	<b>No</b>
Second OEM Public Key Hash	Hash of the second provided OEM public key	32	32 Hex Pairs with space between pairs	<b>ME</b>	<b>No</b>

- 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.

