

Raptor Lake Platform System Tools - Intel® Management Engine Firmware 17.0

User Guide

August 2021

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](https://www.intel.com), or from the OEM or retailer.

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

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

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

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

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

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

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

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

Intel and the Intel logo are trademarks of Intel Corporation in the U.S. and other countries.

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

Copyright © 2021, 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	Generating Config files or log file using tools	15
2.5	Intel® Management Engine Setting Checker Tool	15
2.6	Operating System Support	16
2.7	Generic System Requirements	16
2.8	Error Return	17
2.9	Usage of Double-Quote Character (")	17
2.10	Control Handler Support	18
3	Intel® Modular Flash Image Tool	19
3.1	System Requirements	19
3.2	Flash Image Details	20
3.2.1	Flash Space Allocation	20
3.3	Intel® Modular Flash Image Tool	20
3.3.1	Configuration Files	21
3.3.2	Introducing "Layouts" in MFIT	21
3.3.3	Starting with Layout selection	21
3.3.4	Creating New Configuration	22
3.3.5	Opening Existing Configuration	22
3.3.6	Path Variables	22
3.3.7	Build Results	23
3.3.8	Modifying the Flash Descriptor Region	24
3.3.9	Region Access Control	25
3.3.10	Modifying the Intel® Management Engine Region	27
3.3.11	Setting the Intel® Management Engine Region Binary File	27
3.3.12	Setting the Intel® PMC Binary File	28
3.3.13	Gbe (LAN) Region Settings	28
3.3.14	Modifying PDR Region	28
3.3.15	Modifying BIOS Region	29
3.3.16	Intel® Management Engine Section	29
3.3.17	Resiliency Configurations	30
3.3.18	Power	31
3.3.19	Manageability Application Section	31
3.3.20	Platform Protection	31
3.3.21	Provisioning Section	32
3.3.22	Building Flash Image	33
3.3.23	Decomposing Existing Flash Image	33
3.3.24	MFIT search feature	34
3.3.25	Command Line Interface	35
4	Flash Programming Tool	37
4.1	System Requirements	37
4.2	Flash Image Details	38



4.3	Microsoft Windows® Required Files.....	38
4.4	EFI Required Files	38
4.5	Programming Flash Device.....	38
4.5.1	Stopping Intel® ME SPI Operations	38
4.6	Programming NVARs	39
4.7	Usage.....	40
4.8	Examples.....	45
4.8.1	Complete SPI Flash Device with Binary File	45
4.8.2	Program Specific Region	45
4.8.3	Program SPI Flash from Specific Address.....	46
4.8.4	Dump Full Image.....	47
4.8.5	Dump Specific Region	47
4.8.6	Display SPI Information.....	48
4.8.7	Verify Image with Errors.....	48
4.8.8	Verify Image Successfully	49
4.8.9	Get Intel® ME settings.....	49
4.8.10	CVAR Configuration File Generation (-cfggen)	50
5	Intel® ME Manuf and ME ManufWin	53
5.1	Windows® PE Requirements	53
5.2	How to Use Intel® ME Manuf	53
5.3	Usage.....	54
5.4	Intel® ME Manuf –EOL Check.....	56
5.4.1	ErrorAction Field.....	56
5.4.2	MEManuf.xml File	57
5.4.3	MEManuf –EOL Variable Check	112
5.4.4	MEManuf –EOL Config Check.....	112
5.4.5	Output/Result	112
5.5	Examples.....	113
5.5.1	Example 1	113
6	Intel® ME Info	116
6.1	Windows® PE Requirements	116
6.2	Manageability configurations	116
6.3	Usage.....	116
6.4	Examples.....	131
6.4.1	Consumer Intel® ME FW SKU	131
6.4.2	Corporate Intel® ME FW SKU.....	131
6.4.3	Checking on Set-up and Configuration Process	137
7	Intel® ME Firmware Update.....	116
7.1	Requirements.....	139
7.2	Windows® PE Requirements	139
7.3	Enabling and Disabling Intel® FW Update.....	140
7.4	FW Update Flows	140
7.4.1	Full FW Update.....	140
7.4.2	Partial FW Update.....	141
7.5	Usage.....	141
7.6	Examples.....	143
7.6.1	Updates Intel® ME with Firmware Binary File.....	143
7.6.2	Partial Firmware Update	143
7.6.3	Display Supported Commands.....	143
7.6.4	Language Codes	144



8	UEFI Sample Application Leveraging FW Update API Library	146
8.1	Getting Started - FW Update Full Library	146
8.1.1	Introduction	146
8.1.2	Environment	146
8.1.3	Setup	146
8.1.4	Files in the Kit	146
8.2	Function Description	147
8.2.1	Full FW Update from Buffer (FS)(RS)	147
8.2.2	Partial FW Update from Buffer (FS)(RS)	148
8.2.3	Checking update progress (FS) (RS)	148
8.2.4	Get FW Update ability (FS)(RS)	148
8.2.5	Set FW Update enabling state (FS)(RS)	149
8.2.6	Retrieve OEM ID from Flash (FS)(RS)	149
8.2.7	Retrieve FW Type (FS)(RS)	149
8.2.8	Retrieve FW Type from Buffer (FS)(RS)	150
8.2.9	Retrieve FW Type from image file (FS)	150
8.2.10	Retrieve PCH SKU from the flash image (FS)(RS)	150
8.2.11	Retrieve PCH SKU from buffer (FS)(RS)	151
8.2.12	Retrieve PCH SKU from image file (FS)	151
8.2.13	Retrieves the product segment from the flash image (FS)(RS)	151
8.2.14	Retrieves the product segment from buffer (FS)(RS)	152
8.2.15	Retrieves the product segment from image file (FS)	152
8.2.16	Checking compatibility with update image from buffer (FS)(RS)	152
8.2.17	Checking compatibility with update image from image file (FS)	153
8.2.18	Get version of specific partition from flash image (FS)(RS)	153
8.2.19	Get version of specific partition from buffer (FS)(RS)	153
8.2.20	Get vendor ID for a specific partition (FS)(RS)	154
8.2.21	Performing a full FW Update (FS)	154
8.2.22	Performing a partial FW Update (FS)	154
8.2.23	Retrieving partition version from image file (FS)	155
8.2.24	Retrieving instance of a partition (FS)	155
8.2.25	Performing a partial FW Update with Instance ID from buffer (FS)	155
8.2.26	Performing a partial FW Update with Instance ID from file (FS)	156
8.2.27	Creating a restore point image into buffer (FS)(RS)	156
8.2.28	Creating a restore point image into file (FS)	156
8.2.29	Checking power source (FS)	157
8.2.30	Set ISH configuration file (RS Only)	157
8.2.31	Get PDT version and VDV version (RS Only)	157
9	Intel® Manifest Extension Utility (Intel® MEU)	158
9.1	Usage	158
Appendix A	: Intel® ME NVARs	159

Figures

Figure 3-1. SPI Flash Image Regions	Error! Bookmark not defined.
Figure 3-2. Path Variables	Error! Bookmark not defined.
Figure 3-3. Build Results	Error! Bookmark not defined.
Figure 3-5. Descriptor Region Length Parameter	Error! Bookmark not defined.
Figure 3-6. Flash Settings > Flash Components	Error! Bookmark not defined.
Figure 3-7. Flash Settings → Flash Configuration	Error! Bookmark not defined.



Figure 3-8. Descriptor Region → Master Access Section	Error! Bookmark not defined.
Figure 3-16. GbE Region Options	Error! Bookmark not defined.
Figure 3-17. PDR Region Options	Error! Bookmark not defined.
Figure 3-18. BIOS Region Parameters	Error! Bookmark not defined.
Figure 3-11. Intel® ME Kernel	Error! Bookmark not defined.
Figure 3-12. Power	Error! Bookmark not defined.
Figure 3-13. Manageability Application Section	Error! Bookmark not defined.
Figure 3-14. Provisioning Configuration Section	Error! Bookmark not defined.

Tables

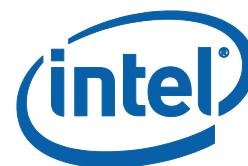
Table 2-1. OS directories unsupported by tools	15
Table 2-2. OS Support for Tools	16
Table 2-3. Tools Summary	17
Table 3-1. Flash Image Regions – Description	Error! Bookmark not defined.
Table 3-2. Build Results Dialog Options	Error! Bookmark not defined.
Table 3-3. Region Access Control Table	Error! Bookmark not defined.
Table 3-4. CPU/BIOS Access	Error! Bookmark not defined.
Table 3-5. MFIT Command Line Options	Error! Bookmark not defined.
Table 4-1. Named Variables Options	39
Table 4-2. Command Line Options for fpt.efi, fpt.exe and fptw.exe	40
Table 4-3. FPT-closemef Behavior	44
Table 4-4. Intel-Recommend Access Settings	44
Table 5-1. Options for Tool	54
Table 5-2. Intel® ME Manuf Test Matrix	56
Table 6-1. Intel® ME Info Command Line Options	117
Table 6-2. List of Components that Intel® ME INFO Displays	118
Table 7-1. Image File Update Options	142



Revision History

Revision Number	Description	Date
0.5	<ul style="list-style-type: none">• Initial Release	May 2021
1.0	<ul style="list-style-type: none">• Updated the MEInfo consumer output• Updated MFIT command line options• Updated FWUpdate tool and Functions under chapters 7 & 8• Update Appendix A: Intel® ME NVARs	August 2021

§ §



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
FW Update	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® Modular-FIT	Intel® Modular 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® ME INFO	Intel® Manageability Engine Information Tool to check whether ME is alive or not.
Intel® ME Info	Windows® version of Intel® Manageability Engine Information Tool
Intel® ME Manuf	Intel® Manageability Engine Manufacturing Tool validates Intel® ME functionality on the manufacturing line
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)
NM	Number of Masters
NVAR	Named Variable

Acronym/Term	Definition
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. A 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
ROM	Read Only Memory
RPAS	Remote Connectivity Service

Acronym/Term	Definition
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
UPDPARAM	Update Parameter Tool
USB	Universal Serial Bus

Acronym/Term	Definition
USB _r	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 17.0	CCL
PCH EDS	CCL
Raptor Lake SPI Programming Guide	Release kit



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:

- Raptor Lake PCH platform – All tools in the Raptor Lake PCH FW release kit are designed for 9th Generation Intel® Core™ Processor and Raptor 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 9th Generation Intel® Core™ Processor or Raptor Lake PCH platform.
- Intel® vPro® platform – All features listed in this document are available for Intel® vPro® platforms with Intel® ME FW 17.0. There are some features that are specifically designed for the Intel® vPro® platform and only work on it.
- Intel® ME Firmware 17.0 SKU – A common set of tools are provided for the following Intel® ME FW 17.0 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® MFIT:**
 - 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.
- **FW Update** – 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 FW Update feature.

2.3 Manufacturing Line Validation Tool

The manufacturing line validation tool (Intel® ME Manuf) 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 Generating Config files or log file using tools

To ensure no impact to OS system directories or files, Intel Tools prevents generating or creating config files and log files into OS System directories. Files generated from tools can be Intel® FPT & Intel® ME Manuf Configurations files as well as log files created by running -verbose command in Intel® ME Info, Intel® ME Manuf, and Intel® FPT.

The below table displays the directories that tools will not permit specifying them as path targets for generating any sort of file:

Table 2-1. OS directories unsupported by tools

Windows® OS	Linux
C:\Program Files	"/sbin"
C:\Program Files (x86)	"/bin"
C:\Windows	"/etc"
C:\ProgramData	"/boot"
	"/lib"
	"/srv"
	"/sys"
	"/usr"
	"/var" (except "/var/tmp")

2.5 Intel® Management Engine Setting Checker Tool

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

2.6 Operating System Support

Table 2-2. OS Support for Tools

Intel® ME and Manufacturing Tools	UEFI (64 bit)	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
Intel® ME Manuf Tool	x	x		x	x
Intel® ME Info Tool	x	x		x	x
Intel® Firmware Update Tool	x	x		x	x
Intel® Manifest Extension Utility Tool		x	x		x
Intel® Modular Flash Image Tool		x	x		x

Notes:

1. 64 bit support does NOT mean that a tool is compiled as a 64 bit application – but that it can run as a 32 bit application on a 64 bit platform.
2. ISH is not supported on ME Info/ ME Manuf for Linux or Windows® Server. Also, Separate ISH tool needs to be used where functionalities are ported from ME Info and ME Manuf tool.
3. Currently the System Tools use the EDK Development Kit.

2.7 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 2-3. Tools Summary

Tool Name	Feature Tested	Runs on Intel® ME device
Intel® ME Manuf	Connectivity between Intel® ME Devices	X
Intel® ME Info	Firmware Aliveness – outputs certain Intel® ME parameters	X
Intel® FPT	Programs the image onto the flash memory and Programming NVARs/FPFs	X
Intel® FW Update	Updates the FW code while maintaining the previously set values	X

2.8 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.

2.9 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.

2.10 Control Handler Support

Intel® ME Info and Intel® FPT and Intel® ME Manuf 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® Modular Flash Image Tool*

The Modular Flash Image Tool (**MFIT.exe**) creates and configures a complete SPI image file for Raptor Lake PCH-LP platforms in the following way:

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

The following binary fields:

- Descriptor region
- BIOS
- Intel integrated LAN (GbE)
- IFWI: Intel® ME, PMC, & PCHC
- EC
- Platform Descriptor Region
- ISH
- IUnit- Sub-Partition
- PCH Configuration Sub-Partition
- GBST Configuration Sub-Partition
- TCSS Components

The Flash Descriptor Region created by MFIT

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.

MFIT 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: MFIT 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, PMC, & PCHC Intel® ME Applications	EC	GbE	PDR	BIOS
------------	--	----	-----	-----	------

Table 3-1. Flash Image Regions – Description

Region	Description
Descriptor	This region contains information such as the space allocated for each region of the flash image, read-write permissions for each region, and a space which can be used for vendor-specific data. It takes up a fixed amount of space at the beginning of the flash memory. Note: This region MUST be locked before the serial flash device is shipped to end users. Refer section 3.3.9 below for more information. Failure to lock the Descriptor Region leaves the Intel® ME device vulnerable to security attacks.
IFWI: Intel® ME, PMC, & PCHC	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 the regions, the Intel® ME region expands to fill the remaining space.

3.3 Intel® Modular Flash Image Tool

Refer following for further information:

- General configuration information – Refer FW Bring-up Guide from the appropriate Intel® ME FW kit.

- Detailed information on how to configure PCH Soft Straps and VSCC information – Refer Raptor Lake PCH SPI Programming Guide.

3.3.1 Configuration Files


The flash image can be configured in many ways, depending on the target hardware and the required FW options. MFIT 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 later and used to build subsequent flash images.

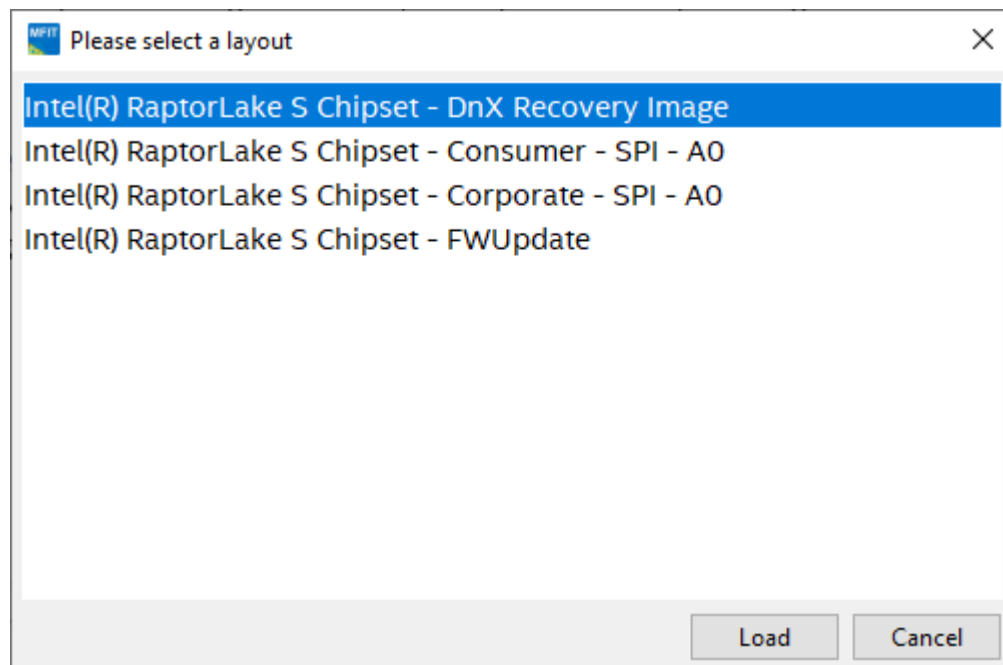
3.3.2 Introducing “Layouts” in MFIT

MFIT introduces the “Layout” feature; upon starting MFIT, user will be prompted to choose a layout to work with. Upon choosing the layout MFIT will adjust its configurations and structure to accommodate the chosen layout. E.g., when choosing a consumer layout, Intel® Manageability configurations will not be loaded by the tool as these configurations are not relevant to the specified layout.

Furthermore, FWUpdate layout is also available to be selected in MFIT. Upon selecting FWUpdate layout, MFIT will only display relevant fields and parameters to the FWUpdate layout.


3.3.3 Starting with Layout selection

To start using MFIT, the user must first select the desired image layout. Upon clicking on “**New Image from Layout**” button , the user will be prompted to choose a layout from the list as seen below.




3.3.4 Creating New Configuration

MFIT supports creating an XML configuration file template that will help the user create their own configuration XML. This template configuration XML file can be

created by clicking **"Save Config"** . It can also be created from the command line using `-s` or `--saveconfig` options.

3.3.5 Opening Existing Configuration

To open an existing configuration file:

3. Click on the **"Load Config"** button .
4. Select the XML file to load.
5. Click Open.

The above can be achieved using the command line option `"-loadconfig"`.

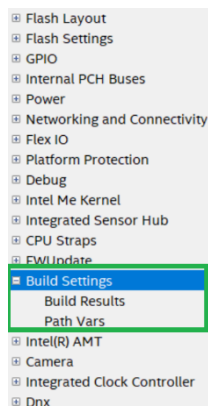
Note: The user can also open a file by dragging and dropping a configuration file into the main window of the application.

3.3.6 Path Variables

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

It is recommended that the path variables be the first thing that the user sets when working with a new configuration. This ensures that MFIT can properly substitute path variables into paths to keep them relative.

To modify the path variables, navigate to the "Build Settings" in the tree of sections and then to the path Vars.



Under the Path Vars, the following variables are available for modification:

- \$WorkingDir\$ – the directory functions as a basic path variable when modified in the GUI. If \$WorkingDir\$ CLI flag is used when launching MFIT GUI, then the MFIT.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. Path Variables

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

3.3.7 Build Results

MFIT lets the user set several options that control how the image is built.

To modify the build results, choose “Build Results” under the “Build Settings” tree option and modify the relevant settings.

Table 3-2. Build Results Dialog Options

Option	Description
SKU	SKU Emulation
MRP Enabled	Enable MRP support for critical layout components
Output path	The path and filename where the final image should be saved after it is built. NOTE: Using the \$DestDir\$ environment variable makes the configuration more portable.
Output Config XML path	The path and filename where the Config XML should be saved after building the image. NOTE: Using the \$DestDir\$ environment variable makes the configuration more portable.

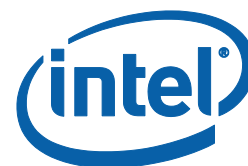
Output Binaries Number	Specifies the number of binaries built upon image creation
Default Data Partition Enabled	Enable CSE Default Data partition
Redundancy Enabled	Enable Redundancy support for critical layout components
Read-Only protection for Minimal Recovery code	Set this to Yes if you wish to have the start of CSE region up to boot partition 1 to be read only
IFWI Image Version	32-bit value to use as the IFWI build version number
Intel(R) Manifest Extension Utility Path	The path and filename where the used Intel® MEU tool resides
OpenSSL path	The path and filename where the OpenSSL tool resides

Figure 3-3. Build Results

The screenshot shows a 'Build Results' configuration window. It contains a list of settings on the left and their corresponding values or input fields on the right. The settings include paths for the Intel(R) Manifest Extension Utility and Open SSL Signing Tool, a dropdown for Signing Enabled (set to Disabled), a field for Descriptor Debug Signing Key, a dropdown for Sku (set to No Emulation), a dropdown for Factory Defaults Restoration Status (set to Disabled), a field for Region Order (set to \$324), fields for Output Path and Output Config XML Path (both set to \$DestDir\$\image.bin and \$DestDir\$\Untitled.xml respectively), fields for Number of Flash Components (set to 1), Flash Components Sizes (set to 32), a dropdown for Flash Components Sizes Unit (set to MB), a field for Total Image Size (set to 32), a dropdown for Default Data Partition Enabled (set to false), a dropdown for Redundancy Enabled (set to false), and a field for Ifwi Image Version (set to 0x0).

3.3.8 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 to ensure that the system is secure.



3.3.9 Region Access Control

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

Table 3-3. Region Access Control Table

Master Read/Write Access				
Region (#)	CPU and BIOS	ME/PCH	GbE Controller	EC
Descriptor (0)	Not Accessible	Not Accessible	Not Accessible	Not Accessible
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
NOTES: <ol style="list-style-type: none"> 1. Descriptor and PDR region is not a master, so they will not have Master R/W access. 2. Descriptor should NOT have write access by any master in production systems. 3. PDR region should only have read and/or write access by CPU/Host. GbE and ME should NOT have access to PDR region. 				

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

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

Key:

0 – Denied access

1 – Allowed access

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

Table 3-4. CPU/BIOS Access

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

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

Example:

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

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

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

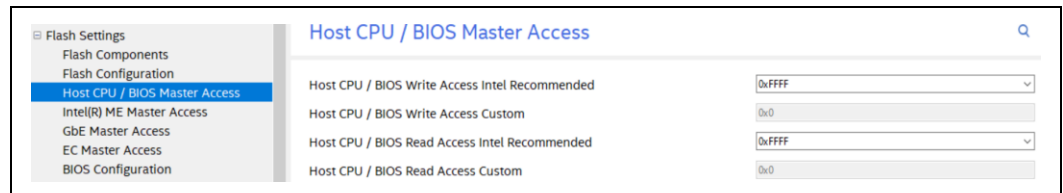
To set these access values in MFIT:

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 3-4. Descriptor Region → Master Access Section



3.3.10 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® MFIT.

3.3.11 Setting the Intel® Management Engine Region Binary File

To select the Intel® ME region binary file:

1. Select the "IFWI: Intel® ME and PMC Region" available under Flash Layout in the tree pane.
2. Either type in the path or click on the dotted button to select the path.
3. Click **OK** to update the parameter; when the flash image is built, the contents of this file is copied into the Intel® ME Region.

3.3.12 Setting the Intel® PMC Binary File

To select the Intel® PMC binary file:

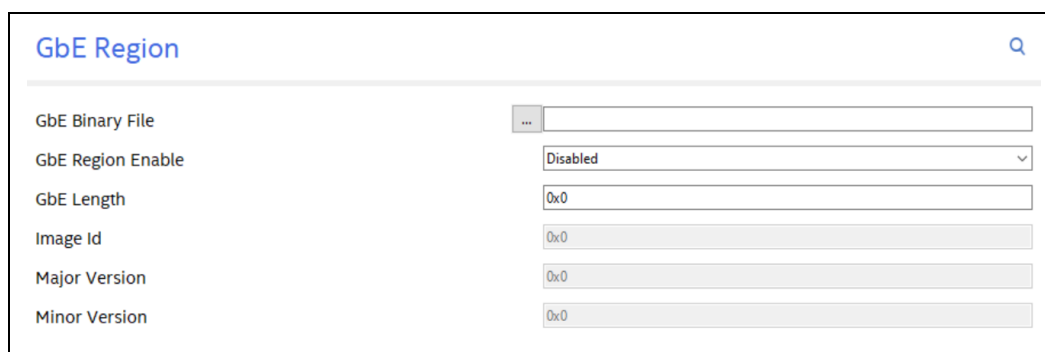
1. Select the "IFWI: Intel® ME and PMC Region" available under Flash Layout in the tree pane.
2. Either type in the path or click on the dotted button to select the path.
3. When the image is built, contents of this file is copied into the Intel® PMC Region.

Note: Intel® MFIT returns a build error in case wrong PMC binary is selected for stitching.

3.3.13 Gbe (LAN) Region Settings

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

Figure 3-5. GbE Region Options



The screenshot shows a window titled "GbE Region" with a search icon in the top right corner. The window contains the following settings:

Setting	Value
GbE Binary File	...
GbE Region Enable	Disabled
GbE Length	0x0
Image Id	0x0
Major Version	0x0
Minor Version	0x0

3.3.13.1 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.

3.3.13.2 Enabling/Disabling GbE Region

The GbE Region can be excluded from the flash image by disabling it in the MFIT. To enable/disable GbE region, the user can set the "**GbE Region Disable**" option to "Enabled"/"Disabled"

3.3.14 Modifying PDR Region

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

Figure 3-6. PDR Region Options

PDR Region

PDR Binary File

PDR Region Enable

Disabled

PDR Length

0x0

3.3.14.1 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.

3.3.14.2 Enabling/Disabling PDR Region

The PDR Region can be excluded from the flash image by disabling it in the MFIT. To enable/disable PDR region, the user can set the **"PDR Region Disable"** option to "Enabled"/"Disabled"

3.3.15 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 3-7. BIOS Region Parameters

BIOS Region

BIOS Binary File

BIOS Length

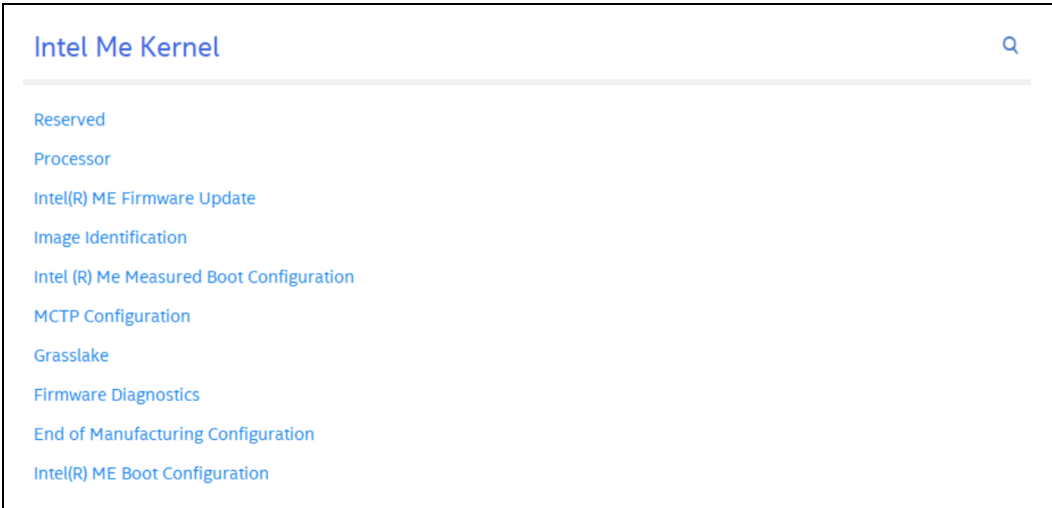
3.3.15.1 Setting BIOS Region Length Parameter

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.

3.3.16 Intel® Management Engine Section

This section describes Intel® ME FW Kernel parameters. (Refer FW Bring-up guide for general information and refer Appendix for more details.) Click on Intel® ME Kernel option in the tree pane to configure Intel® ME parameters.

Figure 3-8. Intel® ME Kernel



3.3.17 Resiliency Configurations

Intel® CSME supports the creation of backup copies for both critical data and code residing within Intel® CSME.

3.3.17.1 Code Resiliency

Enabling code resiliency creates an extra backup copy of the boot critical (BC) partitions. Upon detecting corruption in any of the boot critical partitions (BC1), Intel® CSME will boot from BC2 and indicate the need to update BC1.

To enable code resiliency, users can modify the “Redundancy Enabled” field in MFIT under build settings.

Redundancy Enabled	true
--------------------	------

3.3.17.2 Data Resiliency

Enabling data resiliency creates a small region within the data region that contains a backup copy of the critical data of Intel® CSME. The created region will be populated with the data upon triggering EOM allowing for users to configure the data and include configured data within the backup region.

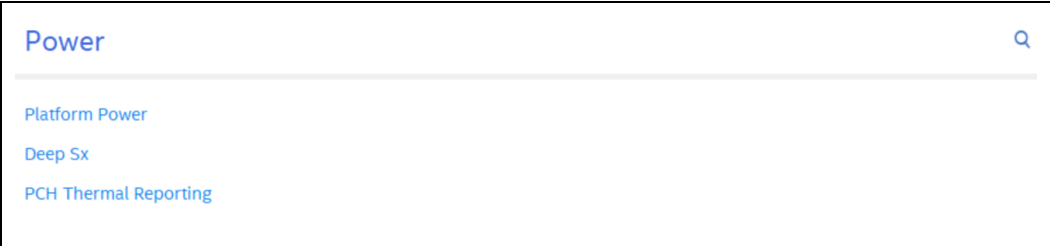
To enable data resiliency, users can modify the “Factory Defaults Restoration Status” field in MFIT under build settings.

Factory Defaults Restoration Status	Enabled
-------------------------------------	---------

3.3.18 Power

This section describes the platform power configuration settings. Click on the Power option on the tree pane to configure power parameters as seen in figure 12.

Figure 3-9. Power



3.3.19 Manageability Application Section

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

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

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

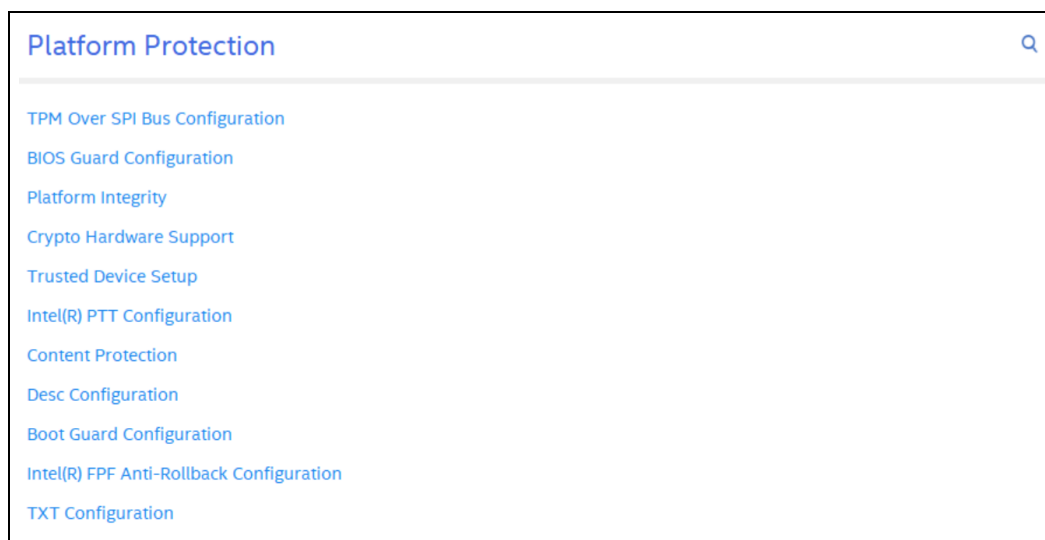
Figure 3-10. Manageability Application Section



3.3.20 Platform Protection

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

Figure 3-15. Platform Protection Section



These options control the availability and visibility of FW features.

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

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

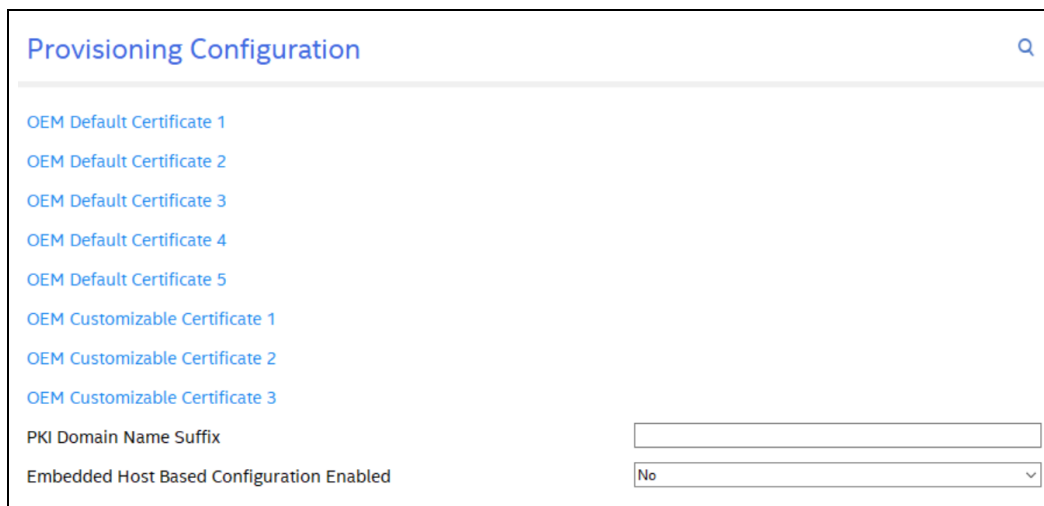
- Intel® Platform Trusted Technology
- Intel® Content Protection

3.3.21 Provisioning Section

The Provisioning section allows the end user to specify the configuration settings, Intel® Upgrade Service, and Intel® DAL. (See the FW Bring up guide for general information and see Appendix E for more details.

Click Intel® AMT option in the tree pane to specify the OEM settings.

Figure 3-11. Provisioning Configuration Section



Provisioning Configuration

OEM Default Certificate 1

OEM Default Certificate 2

OEM Default Certificate 3

OEM Default Certificate 4

OEM Default Certificate 5

OEM Customizable Certificate 1

OEM Customizable Certificate 2

OEM Customizable Certificate 3

PKI Domain Name Suffix

Embedded Host Based Configuration Enabled

3.3.22 Building Flash Image

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

To build a flash image with the currently loaded configuration:

- Click on the **"Build"** button .
- Or Specify an XML file with the /b option in the command line.

MFIT 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.

3.3.23 Decomposing Existing Flash Image

MFIT 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:

6. Click on the **"Decompose image"** button .
7. Change the file type filter to the appropriate file type.
8. 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 MFIT. MFIT will use Intel default value instead. User might want to check the log file to find out which NVARs were not parsed.

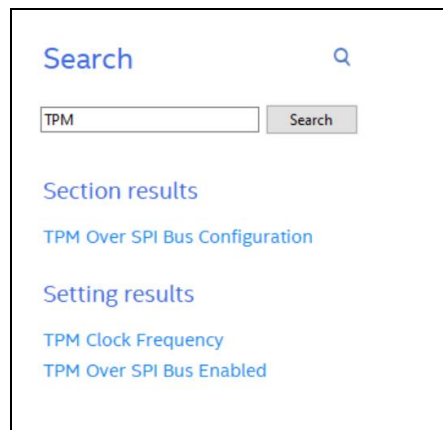
Note: The ME region binary contained in INT folder after image generation only contains the firmware default base settings for ME region no MFIT customization is applied.

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

3.3.24 MFIT search feature

MFIT supports the search ability to quickly locate desired features.

Searching is conducted throughout the entire tree of sections, sub-sections, & configurations' names.



3.3.25 Command Line Interface

MFIT supports command line options.

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

The command line syntax for MFIT is:

MFIT.exe [--help] [--version] [--gui] [--decompose in_bin_path | --layout in_layout_xml_path][--loadingconfig in_config_xml_path][--build [out_bin_path]][--saveconfig out_config_xml_path]

Table 3-5. MFIT Command Line Options

Option	Description
--version, -v	This command displays the current version of M-FIT
--help, -h	This command displays the applicable commands of M-FIT In addition, possible layouts are displayed when running the help command
--gui, -g	This command will start a new instance of the M-FIT GUI
--decompose, -d [in_bin_path]	This command will cause M-FIT to break down the given image to its original ingredient. It receives as argument the image file path
--layout, -l [in_layout_xml_path]	This command receives as argument the name of the layout. By another CMD flag M-FIT will display a list of all the available layouts
--loadconfig, -c [in_config_xml_path]	This command will receive the XML file path containing all the configurations, as input to the image
--build, -b [out_bin_path]	Build is the main command to build an image. It receives as argument the file name to produce
--saveconfig, -s [out_config_xml_path]	M-FIT will dump the latest configurations stored in its internal data base. It receives as argument the file name to produce
--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
--iunit <file>	Overrides the binary source file for the iUnit region
--EC <file>	Overrides the binary source file for the EC region
--PMCP <file>	Overrides the binary source file for the PMCP region
--ISH <file>	Overrides the binary source file for the ISH region
--OEM_KM <file>	Overrides the binary source file for the OEM KM
--SPHY	Overrides the binary source file for the SPHY
--NPHY	Overrides the binary source file for the NPHY
--TBT	Overrides the binary source file for the TBT

Option	Description
--IOM	Overrides the binary source file for the IOM
--RVP	Specifies platform RVP for retrieving the default values of the relevant Soft Straps.

Command Line examples:

1. Choosing layout & Saving a config file:
 - ❖ `mfit.exe --layout "Intel(R) RaptorLake P Chipset - Consumer - SPI" --saveconfig "config.xml"`
2. Commands with IUPs:
 - ❖ `mfit.exe --layout "Intel(R) RaptorLake P Chipset - Consumer - SPI" --build "output_image.bin" --loadconfig "loaded_config.xml" --bios "bios.bin" --me "me.bin" --ec "ec.bin"`
3. Build command with MEU and OpenSSL paths
 - ❖ `mfit.exe --layout "Intel(R) RaptorLake P Chipset - Consumer - SPI" --build "output_image.bin" --loadconfig "loaded_config.xml" --meu_path "meu.exe" --openssl_path "openssl.exe"`
4. Build an image with signing enabled
 - ❖ `mfit.exe --layout "Intel(R) RaptorLake P Chipset - Consumer - SPI" --build "output_image.bin" --loadconfig "loaded_config.xml" --signing_enabled`

Note: Command in example #4 will require all mandatory IUPs to be present in the command line statement.

4 *Flash Programming Tool*

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

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

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

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

4.1 System Requirements

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

One possible workflow for using FPT is:

9. A pre-programmed flash with a bootable BIOS image is plugged into a new computer.
10. The computer boots.
11. FPT is run and a new BIOS/Intel® ME/GbE image is written to flash.
12. The computer powers down.
13. 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 MFIT 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 always runs. Intel® ME is capable of writing to the flash device at any time, even when the management mode is set to none and it may appear that no writing would occur.

Note: Programming the flash or running any of the FPT commands has a limitation on the path length to 128 Characters.

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).
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>	Used 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] [-VERIFY] [-NOVERIFY] [-D] [-DESC] [-BIOS]
[-ME] [-GBE] [-PDR] [-EC] [-SAVEMAC] [-SAVESXID] [-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 NOVERFY 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.
-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.
-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. The config file can handle up to 36 NVARs.
-U	Update. Updates the NVARs and FPFs in the flash. The user can update by specifying their names and values in the parameter file. The parameter file must be in an INI file format (the same format generated by the -cfggen command). The -in <file> option is used to specify the input file.

Option	Description
-O <file>	Output File. The file used by FPT to output NVAR information.
-IN <file>	Input File. The file used by FPT for NVAR input. This option flag must be followed by a text file (i.e., <code>fpt -u -in FPT.cfg</code>). The tool updates the NVARs contained in the text file with the values provided in the input file. User can also use <code>FPT -cfggen</code> to generate this file. The provided config file can handle up to 36 NVARs.
-N <value>	Name. Specifies the name of the NVAR that the user wants to update in the image file or flash. The name flag must be used with Value (-v).
-V <value>	Value. Specifies the value for the NVAR variable. The name of variable is specified in the Name flag. The Value flag must follow the Name flag.
-CLOSEMNF [-in <file>] [-noReset]	End of Manufacturing. This option is executed at the end of manufacturing phase. This option does the following: <ol style="list-style-type: none"> 1. Sets the Intel® ME manufacturing mode done bit (Global Locked bit). 2. Verifies that the Intel® ME manufacturing mode done bit (Global Locked) is set. 3. Sets the master region access permission in the Descriptor region to its Intel-recommended value 4. Verifies that flash regions are locked. <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 <code>FPT-closemnf</code> also sets the default value for any unprovisioning process. Run <code>FPT -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 MFIT/FPT, the <code>closemnf</code> will perform all 4 steps mentioned above. In case flexibility of EOM has been set, the <code>closemnf</code> will behave as configured.</p>
-GRESET	Global Reset. FPT performs a global reset.
-PAGE	Pauses the screen when a page of text has been reached. Hit any key to continue.

Option	Description
-R <name>	NVAR and FPFs Read. FPT uses this option to retrieve NVAR value for a specific NVAR file name. The value of the variable is displayed. By default, all non-secure variables are displayed in clear-text and secure NVAR will be displayed in HASH. The -hashed option can be used to display the hash of a value instead of the clear-text value.
-VARS	Display Supported Variables. FPT uses this option to display all variables supported for the -R and -COMPARE commands. Note: This will no longer display UEP based values which are tied to configuring 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	Displays a list of the FPFs
-PROVHDCP <file><file>	Provision platform with the key and cert provided.
-READHDCP	Displays the HDCP Rx provisioning status.
-GETPID <file>	Retrieve the part id.
-REWRITE	Allows to rewrite the SPI with file data even if flash is identical.
-WRITETOKEN <file>	Write the token where the file name is the token name.
-ERASETOKEN	Delete the token.
-PROVKB <iv_and_keybox.bin>	Provision Widevine using IV (Initialization Vector) and encrypted KeyBox file.
-COMMITARBSVN	Commits ARB SVN to FPFs. This would commit the Anti Rollback SVN to the FPFs

Table 4-3. FPT–closemnf Behavior

	Condition after FPT -closemnf		
EOM Settings	Flash Protection Mode	NVARs Configuration State	FPFs committed
Lock(Flash, Config) **(Legacy EOM flow)	Protected	Locked	Yes
Lock (Config)	Unprotected	Locked	Yes
Lock(Flash)	Protected	Unlocked	Yes
Lock(none)	Unprotected	Unlocked	Yes

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

```
Intel (R) Flash Programming Tool Version: 17.0.0.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
Note: <ol style="list-style-type: none"> ‡ = Value dependent on if PDR is implemented and if Host access is desired. † = Optional BIOS access to the EC region. * = Optional EC Read access to BIOS. 				

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: 17.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: 17.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:  17.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:  17.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: 17.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: 17.x.x.xxxx
Copyright (c) 2005-2019, Intel Corporation. All rights reserved.
Reading HSFSTS register... Flash Descriptor: Valid
-Verifying Flash [0x0000000] 4KB of 16384KB - 0 percent complete
Error 207: Data verify mismatch found.
Warning: There are some addresses that are not defined in any
regions.Read/Write/Erase operations are not possible on those addresses.
```

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

4.8.8 Verify Image Successfully

```
fpt.exe -verify outimage.bin
EFI:
fpt.efi -verify outimage.bin
-----
Intel (R) Flash Programming Tool. Version: 17.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: 17.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® ME Manuf and ME ManufWin*

Intel® ME Manuf 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® ME ManufWin (Intel® ME ManufWin) 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® ME Manuf 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® ME Manuf**

Intel® ME Manuf 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® ME Manuf 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, ME Manuf calls the test. Otherwise, it reports the result or the failure message from the previous test.

Intel® ME Manuf tools report the result or cause a reboot. If there is a reboot, Intel® ME Manuf 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® ME Manuf, 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>Note: This option is not applicable for Consumer Intel® ME FW SKU.</p> <p>This option blocks any network traffic that goes in/out of the integrated GbE wired/wireless LAN interface. If Intel® AMT is disabled, "Error 9257: Cannot run the command since Intel® AMT is not available" is returned.</p>
-ALLOWNET	<p>Note: This option is not applicable for Consumer Intel® ME FW SKU.</p> <p>This option allows any network traffic that goes in/out of the integrated GbE wired/wireless LAN interface. If Intel® AMT is disabled, "Error 9257: Cannot run the command since Intel® AMT is not available" is returned.</p>
-BISTRESULT	Returns last BIST results.
-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 <code>config</code> or <code>var</code> is optional. Using <code>-EOL</code> without a sub option is equivalent to the <code>-EOL config</code>.</p> <p>When <code>-f</code> flag is used along with a file name (<filename>), the tool will load the file as the configuration file, instead of using MEManuf.xml.</p>
-NEXTREBOOT	<p>Upon successful platform reboot CM3 Autotest will be performed.</p> <p>Note: This is a standalone command and will only work if CM3 Autotest has been enabled in the firmware image. CM3 Autotest will be executed on the next CMoff – CM0 transition (example: Cold Reset), Global Reset or G3. The option itself will not trigger any platform reboots.</p>
-CFGGEN <filename>	Use this option along with a filename to generate a default configuration file. This file (with or without modification) can be used for the <code>-EOL</code> 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® ME Manuf 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>Note: Intel recommended tests will be enabled regardless of <code>-all</code> 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® ME Manuf 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 ME17.0 is a combination of M0_HW, Live_HW and M0_Config. The Runtime BIST is a combination of M0_HW and M0_Config.

Intel® ME Manuf 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® ME Manuf –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® ME Manuf 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 ME Manuf is run. If there is no MEManuf.xml file on that folder, MEManuf -EOL config runs the Intel recommended default check only.

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

</eolconfig>
<eolconfig name="Persistent PRTC Backup Power">
  <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
  <!-- Description>Check fpf Persistent PRTC Backup
Power against expected value</Description -->
  <!-- IntelRequired>False</IntelRequired -->
  <!-- Dependencies></Dependencies -->
  <!-- Level>1</Level -->
  <!-- End of uneditable fields -->
  <!-- Please edit the fields below ONLY with the State
or ErrAction -->
  <State>Enabled</State>
  <ErrAction>ErrorContinue</ErrAction>
  <RequiredValue format="Enabled/00/Disabled/01"
example="Enabled"> </RequiredValue>
</eolconfig>
<eolconfig name="PMC Anti Rollback">
  <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
  <!-- Description>Check fpf PMC Anti Rollback against
expected value</Description -->
  <!-- IntelRequired>False</IntelRequired -->
  <!-- Dependencies></Dependencies -->
  <!-- Level>1</Level -->
  <!-- End of uneditable fields -->
  <!-- Please edit the fields below ONLY with the State
or ErrAction -->
  <State>Enabled</State>
  <ErrAction>ErrorContinue</ErrAction>
  <RequiredValue format="Disabled/00/Enabled/01"
example="Disabled"> </RequiredValue>
</eolconfig>
<eolconfig name="PID Refurbish Counter">
  <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
  <!-- Description>Check fpf PID Refurbish Counter
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 key Hash RSA key size">
  <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
  <!-- Description>Check fpf OEM key Hash RSA key size
against expected value</Description -->
  <!-- IntelRequired>False</IntelRequired -->
  <!-- Dependencies></Dependencies -->
  <!-- Level>1</Level -->
  <!-- End of uneditable fields -->
  <!-- Please edit the fields below ONLY with the State
or ErrAction -->
  <State>Enabled</State>
  <ErrAction>ErrorContinue</ErrAction>
  <RequiredValue format="Disabled/00/Enabled/01"
example="Disabled"> </RequiredValue>
</eolconfig>
<eolconfig name="Force Boot Guard ACM">
  <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
  <!-- Description>Check fpf Force Boot Guard ACM
against expected value</Description -->
  <!-- IntelRequired>False</IntelRequired -->
  <!-- Dependencies></Dependencies -->
  <!-- Level>1</Level -->
  <!-- End of uneditable fields -->
  <!-- Please edit the fields below ONLY with the State
or ErrAction -->
  <State>Enabled</State>
  <ErrAction>ErrorContinue</ErrAction>
  <RequiredValue format="Disabled/00/Enabled/01"
example="Disabled"> </RequiredValue>
</eolconfig>
<eolconfig name="Key Manifest ID">
  <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
  <!-- Description>Check fpf Key Manifest ID against
expected value</Description -->
  <!-- IntelRequired>False</IntelRequired -->
  <!-- Dependencies></Dependencies -->
  <!-- Level>1</Level -->
  <!-- End of uneditable fields -->
  <!-- Please edit the fields below ONLY with the State
or ErrAction -->
  <State>Enabled</State>
  <ErrAction>ErrorContinue</ErrAction>
  <RequiredValue format="Hex number with 0x prefix."
example="0x00"> </RequiredValue>

```

```

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

```

```

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

```

```

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

```

```

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

```

```

    </eolconfig>
    <eolconfig name="DNX Anti Rollback">
        <!-- The commented fields below CANNOT be edited. Any
        edits will be ignored by the tool -->
        <!-- Description>Check fpf DNX Anti Rollback against
        expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State
        or ErrAction -->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Disabled/00/Enabled/01"
        example="Disabled"> </RequiredValue>
    </eolconfig>
    <eolconfig name="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="UPID OEM Platform ID">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Check UPID OEM Platform ID is
configured</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>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
    </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="Firmware Update OEM ID">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Check Firmware Update OEM ID
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>3</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State
or ErrAction -->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="UUID" example="00000000-0000-
0000-0000-000000000000"> </RequiredValue>
    </eolconfig>
    <eolconfig name="GBE version">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Check Gbe Version against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>SPI_DEP</Dependencies -->
        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State
or ErrAction -->
        <State>Enabled</State>

```

```

        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="major_ver.minor_ver"
example="0.6"> </RequiredValue>
    </eolconfig>
    <eolconfig name="BIOS version">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Check BIOS Version against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State
or ErrAction -->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Customer specific"
example="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="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="Verify ME EOM status">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>The test checks the status of the
platforms EOM against configured EOM behavior. The test
starts by verifying the type of EOM that was configured and
runs its check against the configuration
accordingly</Description -->
        <!-- IntelRequired>True</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State
or ErrAction -->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
    </eolconfig>
    <eolconfig name="EOP status check">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Check that EOP was
sent/recieved</Description -->
        <!-- IntelRequired>True</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- Level>1</Level -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State
or ErrAction -->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
    </eolconfig>
    <!-- END OF EOL CONFIG TESTS -->
    <!-- EOL VAR TESTS -->
    <eolvar name="EOM Settings">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Test variable against expected
value</Description -->

```

```

        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- 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="Firmware KVM Screen Blanking">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State
or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="No/00/Yes/01" example="No">
</RequiredValue>
    </eolvar>
    <eolvar name="RSA1K Support">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State
or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Disabled/01/Enabled/00"
example="Disabled"> </RequiredValue>
    </eolvar>
    <eolvar name="SMx State">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->

```

```

        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State
or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Disabled/01/Enabled/00"
example="Disabled"> </RequiredValue>
    </eolvar>
    <eolvar name="Config Server IPv6/IPv4 Address">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State
or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="String" example="Any">
</RequiredValue>
    </eolvar>
    <eolvar name="PKI Domain Name Suffix">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State
or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="String" example="Any">
</RequiredValue>
    </eolvar>
    <eolvar name="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 -->
        <!-- 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="Config Server FQDN">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State
or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="String" example="Any">
</RequiredValue>
    </eolvar>
    <eolvar name="RCFG/ZTC">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State
or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Disabled/00/Enabled/01"
example="Disabled"> </RequiredValue>
    </eolvar>
    <eolvar name="Config Server IPv6/IPv4 Port">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- End of uneditable fields -->

```

```

        <!-- Please edit the fields below ONLY with the State
or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Hex number with 0x prefix."
example="0x0000"> </RequiredValue>
        </eolvar>
        <eolvar name="Redirection Privacy / Security Level">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State
or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue
format="Default/01/Enhanced/02/Extreme/03"
example="Default"> </RequiredValue>
        </eolvar>
        <eolvar name="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 -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State
or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Disabled/00/Enabled/01"
example="Disabled"> </RequiredValue>
        </eolvar>
        <eolvar name="Intel(R) ME Region Flash Protection
Override">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- End of uneditable fields -->

```

```

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

```



```

        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Disabled/00/Enabled/01"
example="Disabled"> </RequiredValue>
    </eolvar>
    <eolvar name="OEM Tag">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State
or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Hex number with 0x prefix."
example="0x00000000"> </RequiredValue>
    </eolvar>
    <eolvar name="Processor Emulation">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- 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="WLAN Power Well">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State
or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>

```

```

        <RequiredValue
format="Disabled/80/CoreWell/81/SusWell/82/MEWell/83/WLAN
Sleep via SLP_WLAN#/86" example="Disabled">
</RequiredValue>
    </eolvar>
    <eolvar name="LAN Power Well">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State
or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue
format="CoreWell/00/SusWell/01/MEWell/02/SLP_LAN# (MGPIO3) /0
3" example="CoreWell"> </RequiredValue>
    </eolvar>
    <eolvar name="Unconfigure On RTC">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State
or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Enabled/00/Disabled/01"
example="Enabled"> </RequiredValue>
    </eolvar>
    <eolvar name="Intel(R) PTT initial power-up state">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State
or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>

```

```

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

```

```

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

```

```

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

```

```

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

```

```

        <!-- Please edit the fields below ONLY with the State
or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Hex number with 0x prefix."
example="0x0000"> </RequiredValue>
    </eolvar>
    <eolvar name="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 -->
        <!-- 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 -->
        <!-- 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/D-None/08/D-A/09/D-
B/0a/D-C/0c/E-None/10/E-A/11/E-B/12/E-C/14/E-D/18/F-
None/20/F-A/21/F-B/22/F-C/24/F-D/28/F-E/30" example="None-
None"> </RequiredValue>
    </eolvar>
    <eolvar name="Domain Name">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- End of uneditable fields -->

```

```

        <!-- Please edit the fields below ONLY with the State
or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="String" example="Any">
</RequiredValue>
    </eolvar>
    <eolvar name="Host Name">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State
or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="String" example="Any">
</RequiredValue>
    </eolvar>
    <eolvar name="Thunderbolt Port4 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 -->
        <!-- 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="Thunderbolt Port3 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 -->
        <!-- 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="Thunderbolt Port2 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 -->
        <!-- 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="Thunderbolt Port1 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 -->
        <!-- 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 -->
        <!-- 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 -->
        <!-- 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="MCTP Device Ports">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State
or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Hex number with 0x prefix."
example="0x00000000"> </RequiredValue>
    </eolvar>
    <eolvar name="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 -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State
or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="No/00/Yes/01" example="No">
</RequiredValue>
    </eolvar>
    <eolvar name="Delayed Authentication Mode Config">

```

```

        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State
or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Disabled/00/Enabled/01"
example="Disabled"> </RequiredValue>
    </eolvar>
    <eolvar name="Firmware Update OEM ID">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- 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="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 -->
        <!-- 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="Intel(R) 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 -->
        <!-- 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="Intel(R) 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 -->
        <!-- 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 Required/00/PMF
Signing Required/01/No Signing Required/02" example="Seal
Signing Required"> </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 -->
        <!-- 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="Factory Defaults Recovery Status">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Test variable against expected
value</Description -->

```



```

        <!-- Dependencies>CORP</Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State
or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue
format="False/00/NotActive/00/True/01/Active/01"
example="False"> </RequiredValue>
    </eolvar>
    <eolvar name="Debug Override Production Silicon">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State
or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Hex number with 0x prefix."
example="0x00000000"> </RequiredValue>
    </eolvar>
    <eolvar name="Debug Override Pre-Production Silicon">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies></Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State
or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Hex number with 0x prefix."
example="0x00000000"> </RequiredValue>
    </eolvar>
    <eolvar name="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 -->
        <!-- End of uneditable fields -->

```

```

    <!-- Please edit the fields below ONLY with the State
or ErrAction -->
    <State>Disabled</State>
    <ErrAction>ErrorContinue</ErrAction>
    <RequiredValue format="No/00/Yes/01" example="No">
</RequiredValue>
    </eolvar>
    <eolvar name="Opt-in Policy">
    <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
    <!-- Description>Test variable against expected
value</Description -->
    <!-- IntelRequired>False</IntelRequired -->
    <!-- Dependencies>CORP</Dependencies -->
    <!-- End of uneditable fields -->
    <!-- Please edit the fields below ONLY with the State
or ErrAction -->
    <State>Disabled</State>
    <ErrAction>ErrorContinue</ErrAction>
    <RequiredValue format="Hex number with 0x prefix."
example="0x00"> </RequiredValue>
    </eolvar>
    <eolvar name="KVM">
    <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
    <!-- Description>Test variable against expected
value</Description -->
    <!-- IntelRequired>False</IntelRequired -->
    <!-- Dependencies>CORP</Dependencies -->
    <!-- End of uneditable fields -->
    <!-- Please edit the fields below ONLY with the State
or ErrAction -->
    <State>Disabled</State>
    <ErrAction>ErrorContinue</ErrAction>
    <RequiredValue format="Disabled/00/Enabled/01"
example="Disabled"> </RequiredValue>
    </eolvar>
    <eolvar name="SOL">
    <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
    <!-- Description>Test variable against expected
value</Description -->
    <!-- IntelRequired>False</IntelRequired -->
    <!-- Dependencies>CORP</Dependencies -->
    <!-- End of uneditable fields -->
    <!-- Please edit the fields below ONLY with the State
or ErrAction -->
    <State>Disabled</State>

```

```

        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Disabled/00/Enabled/01"
example="Disabled"> </RequiredValue>
    </eolvar>
    <eolvar name="StorageState">
        <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
        <!-- Description>Test variable against expected
value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>CORP</Dependencies -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State
or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Disabled/00/Enabled/01"
example="Disabled"> </RequiredValue>
    </eolvar>
    <eolvar name="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 -->
        <!-- End of uneditable fields -->
        <!-- Please edit the fields below ONLY with the State
or ErrAction -->
        <State>Disabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="Hex number with 0x prefix."
example="0x00000000"> </RequiredValue>
    </eolvar>
    <!-- END OF EOL VAR TESTS -->
</memanuf_config>

```

Lines which start with <!-- --> are comments. They are also used to inform users of the available test group names and the names of specific checks that are included in each test that Intel® ME Manuf 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 ME Manuf 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) ME Manuf Version: 17.x.x.xxxx
Copyright (C) 2005 - 2019, Intel Corporation. All rights reserved.

LPC Device Id: A082.
Platform: Raptorlake 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               CMO 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        CMO_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) ME Manuf Version: 17.x.x.xxxx
Copyright (C) 2005 - 2019, Intel Corporation. All rights reserved.

LPC Device Id: A082.
Platform: Raptorlake 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® ME Info*

MEInfoWin and Intel® ME Info 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 ME Info (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.

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

6.2 Manageability configurations

It is important to note that upon disabling Manageability HW through the “Manageability Application Hardware Status” FPF or through the relevant NVARs (listed throughout this guide) on non-vPro PCHs/SKUs, Intel® MEInfo will display the image type as consumer, regardless if it is a corporate one, and therefore disable Network Interfaces limiting the functionality of features such as PCH thermal Measuring.

6.3 Usage

The executable can be invoked by:

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

Table 6-1. Intel® ME Info Command Line Options

Option	Description																																				
-VALUE <value>	<p>Compares the value of the given feature name (and optional column name for features displayed in a table) with the value in the command line. If the feature name or value is more than one word, the entire name or value must be enclosed in quotation marks (together with the optional column name). For example –feat “PTT FPF”.</p> <p>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.</p>																																				
-FITVER	Displays MFIT version information																																				
-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® ME INFO.</p> <p>Intel® ME INFO can retrieve all of the information detailed below. However, depending on the SKU selected, some information may not appear.</p> <p>Note: 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) ME Info Version: 17.x.x.xxxx</p> <p>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																																				

Option	Description
-VERBOSE <filename>	Turns on additional information about the operation for debugging purposes. This option has to be used together with the above mentioned option(s). Failure to do so generates the error: "Error 9254: Invalid command line option". This option works with no option and <code>-feat</code> .
-H or -?:	Displays the list of command line options supported by the Intel® ME INFO 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.
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® ME INFO Displays

Feature Name	Feature Data Source	Consumer SKU	Corporate SKU	Specific Feature Dependency	Field Value
Tools Version	SW (Intel® ME Info)	X	X	N/A	Version string Example: 17.x.y.ZZZZ; where x=minor, y = HF/MR, ZZZZ = Build Number.
General FW Information					
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

Feature Name	Feature Data Source	Consumer SKU	Corporate SKU	Specific Feature Dependency	Field Value
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
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	MFIT 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	MFIT TCSS enable feature set to 'Enabled'	Enabled/Disabled
TLS State	Intel® ME Kernel	X	X	N/A	Enabled/Disabled
CSME Measured Boot to TPM	Intel® ME Kernel	X	X	N/A	Enabled/Disabled

Feature Name	Feature Data Source	Consumer SKU	Corporate SKU	Specific Feature Dependency	Field Value
BIOS Recovery State	Intel® ME Kernel	X	X	N/A	Enabled/Disabled
Transactional FW Supported	Intel® ME Kernel	X	X	N/A	Yes/No
Intel(R) ME Code Versions					
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
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 17.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
IUPs Information					
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

Feature Name	Feature Data Source	Consumer SKU	Corporate SKU	Specific Feature Dependency	Field Value
IUnit 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
SPHY FW Version	Intel® ME Kernel	X	X	SPHY 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
PCH Information					
PCH Name	Intel® ME Kernel	X	X	N/A	String (e.g. RPL)
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® MFIT	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

Feature Name	Feature Data Source	Consumer SKU	Corporate SKU	Specific Feature Dependency	Field Value
Flash Information					
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	MFIT 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 Replay Protection Bind Status	Intel® ME Kernel	X	X	N/A	Pre-bind/Post-bind
RPMC Rebind	Intel® ME Kernel	X	X	MFIT 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

Feature Name	Feature Data Source	Consumer SKU	Corporate SKU	Specific Feature Dependency	Field Value
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
FW Capabilities					
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) 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
End of Manufacturing					
EOM Settings	Intel® ME Kernel	X	X	Set in MFIT or FPT. Defines the behavior of EOM triggering	Lock(Flash, Config) Lock(Flash, Config) on 1st Boot Lock(Config)

Feature Name	Feature Data Source	Consumer SKU	Corporate SKU	Specific Feature Dependency	Field Value
					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
FPF Committed	Intel® ME Kernel	X	X	Changes after triggering EOM	Yes/No
Intel(R) Active Management Technology					
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 ".")

Feature Name	Feature Data Source	Consumer SKU	Corporate SKU	Specific Feature Dependency	Field Value
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	MFIT CM3 Autotest Enabled set to 'true'	Enabled/Disabled
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

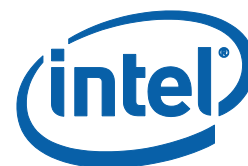
Feature Name	Feature Data Source	Consumer SKU	Corporate SKU	Specific Feature Dependency	Field Value
System UUID	Intel® AMT	N/A	X	AMT CEM (a.k.a. Common Service) is used. Not available on Corporate Sku	UUID of the system
AMT Global State	Intel® ME Kernel	N/A	X	N/A	Enabled/Disabled
Discrete vPro NIC on-board 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
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
Thunderbolt Port1 SMBus Address	Intel® ME Kernel	N/A	X	vPro enabled	A 32bit Hexadecimal number

Feature Name	Feature Data Source	Consumer SKU	Corporate SKU	Specific Feature Dependency	Field Value
Thunderbolt Port2 SMBus Address	Intel® ME Kernel	N/A	X	vPro enabled	A 32bit Hexadecimal number
Thunderbolt Port3 SMBus Address	Intel® ME Kernel	N/A	X	vPro enabled	A 32bit Hexadecimal number
Thunderbolt Port4 SMBus Address	Intel® ME Kernel	N/A	X	vPro enabled	A 32bit Hexadecimal number
Intel(R) Protected Audio Video Path					
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
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
Security Version Numbers					
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
IDLM	Intel® ME Kernel	X	X	BIOS	Counter indicating IDLM SVN
OEM KM	Intel® ME Kernel	X	X	BIOS	Counter indicating OEM KM SVN
FW Supported FPFs					
1st & 2nd OEM Key Hash Revoked	Intel® ME Kernel	X	X	BIOS	Enabled/Disabled

Feature Name	Feature Data Source	Consumer SKU	Corporate SKU	Specific Feature Dependency	Field Value
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
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
Error Enforcement Policy 1	Intel® ME Kernel	X	X	BIOS	Enabled / Disabled
Flash Descriptor Verification	Intel® ME Kernel	X	x	BIOS	Enabled / Disabled
Flexible EOM	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

Feature Name	Feature Data Source	Consumer SKU	Corporate SKU	Specific Feature Dependency	Field Value
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
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
PID Refurbish Counter	Intel® ME Kernel	X	X	BIOS	Hash of Public Key to verify Boot Policy Manifest
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
PTT Lockout Override Counter	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

Feature Name	Feature Data Source	Consumer SKU	Corporate SKU	Specific Feature Dependency	Field Value
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
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



6.4 Examples

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

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

6.4.1 Consumer Intel® ME FW SKU

```

Intel (R) ME Info Version: 17.x.x.xxxx
Copyright (C) 2005 - 2021, Intel Corporation. All rights reserved.

General FW Information
FW Status Register1      0x900000245
FW Status Register2      0x89110106
FW Status Register3      0x00000020
FW Status Register4      0x00004004
FW Status Register5      0x00001F43
FW Status Register6      0x046003C9

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   MAESTRO_CM0_ENTRY_COMPLETE_START
ME File System Corrupted  No
RPMC status               OK

Platform Type             Work Station
FW Image Type             Pre-Production
Last ME Reset Reason      Other
BIOS Boot State           Post Boot
Boot Critical Code Redundancy Disabled
Current Boot Partition    1
Factory Defaults Restoration Status Disabled
Factory Defaults Recovery Status Enabled
Firmware Update OEM ID    00000000-0000-0000-0000-000000000000
Intel(R) ICPS SW SKUing Eligible Disabled
Crypto HW Support         Enabled
Intel(R) ISH Power State  Enabled
OEM Tag                   0x00
FW Update State           Enabled
TLS State                 Disabled
CSME Measured Boot to TPM Disabled
BIOS Recovery State       Disabled

Intel(R) ME Code Versions
BIOS Version              RPLSFWI1.R00.2307.A00.2107260154
GbE Version               0.5
MEI Driver Version        2127.0.1.0
FW Version                 17.x.x.xxxx H Consumer

IUPs Information
PMC FW Version            160.2.0.1029
OEM FW Version            15.0.0.1237
ISHC FW Version           5.4.2.4555
NPHY FW Version           13.30.206.7114
SPHY FW Version           13.0.1.7075
PCHC FW Version           16.0.0.1011

PCH Information
PCH Name                  RPL
PCH Device ID             7A01
PCH Revision ID           A0
PCH SKU Type              Pre-Production ES

```

PCH Replacement State	Disabled		
PCH Replaceable Counter	0		
PCH Unlocked State	Disabled		
Transactional FW Information			
Original image type	Consumer		
Current sku type	Consumer		
Flash Information			
Storage Device Type	SPI		
SPI Flash ID 1	EF4019		
RPMC	Unsupported		
RPMC Bind Counter	0		
RPMC Bind Status	Pre-bind		
RPMC Rebind	Unsupported		
RPMC Replay Protection Max Rebind	1		
BIOS Read Access	0x000F		
BIOS Write Access	0x000A		
GBE Read Access	0x0009		
GBE Write Access	0x0008		
ME Read Access	0x000D		
ME Write Access	0x0004		
EC Read Access	0x0101		
EC Write Access	0x0100		
FW Capabilities	0x31009500		
Intel(R) Protected Audio Video Path	Present/Enabled		
Intel(R) Platform Trust Technology	Present/Enabled		
Persistent RTC and Memory	Present/Enabled		
End Of Manufacturing			
NVAR Configuration State	Locked		
EOM Settings	Lock(Flash,Config)		
EOM Flow	Not set		
HW Binding State	Disabled		
Flash Protection Mode	Protected		
FPF Committed	No		
Intel(R) Trusted Device Setup			
Intel(R) Protected Audio Video Path			
Attestation KeyBox	Not Provisioned		
PAVP State	Yes		
Security Version Numbers			
Trusted Computing Base SVN	1		
Firmware Version Control SVNs			
PMC	0	[minimum allowed: 0]	
CSE	1	[minimum allowed: 0]	
ROT KM	0	[minimum allowed: 0]	
IDLM	0	[minimum allowed: 0]	
CSME bootstrap	0	[minimum allowed: 0]	
SECURE BOOT BSMM	0	[minimum allowed: 0]	
OEM KM	0	[minimum allowed: 0]	
SECURE BOOT KM	0	[minimum allowed: 0]	
UCODE	0	[minimum allowed: 0]	
SECURE BOOT ACM	0	[minimum allowed: 0]	
HW Glitch Detection	0x00		
TRC Polarity	Falling Trans		
TRC Mode	Full-cycle polarity trans		
TRC State	Disabled		
Intel(R) Platform Trust Technology			
Intel(R) PTT initial power-up state	Enabled		
Intel(R) PTT State	Enabled		
SMx State	Enabled		
RSAltK Support	Disabled		
FW Supported FPFs	FPF	UEP	
	---	*In Use	
	---	---	
1st OEM Key Hash Revoked	Not set	Disabled	# Disabled=0, Enabled=1
1st OEM Key Hash size	Not set	Enabled	# Disabled=0, Enabled=1
1st OEM RSA Key size	Not set	Enabled	# Disabled=0, Enabled=1

[illegible]

FW Status Register5	0x80000133
FW Status Register6	0x00400308
Current FW State	Normal
Flash Partition Table	Valid
FW Memory State	CM0 with UMA
FW Initialization	Complete
BUP Loading state	Success
FW Error Code	No Error
FW Mode Of Operation	Normal
SPI Flash Log	Not Present
FW Loading Phase	HOSTCOMM Module
FW Loading Phase Status	UNKNOWN
ME File System Corrupted	No
RPMC status	OK
Platform Type	Desktop
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
OEM Chipset Init	Available
Factory Defaults Recovery Status	Enabled
Firmware Update OEM ID	00000000-0000-0000-0000-000000000000
Crypto HW Support	Enabled
Intel(R) ISH Power State	Enabled
OEM Tag	0x00
FW Update State	Enabled
TLS State	Enabled
CSME Measured Boot to TPM	Disabled
BIOS Recovery State	Disabled
Intel(R) ME Code Versions	
BIOS Version	RPLSFWI1.R00.1362.A51.2009020256
MEBx Version	15.0.0.0001
GbE Version	0.4
MEI Driver Version	2037.16.0.7231
FW Version	17.xx.xx.xxxx H Corporate
LMS Version	2037.16.0.7231
Wireless Hardware Version	2.1.77
Wireless Driver Version	21.115.0.2
IUPs Information	
PMC FW Version	160.2.0.1020
OEM FW Version	15.0.0.1237
ISHC FW Version	5.4.2.4521
LOCL FW Version	17.xx.xx.xxxx
WCOD FW Version	17.xx.xx.xxxx
NPHY FW Version	13.3.110.7017
SPHY FW Version	13.0.1.7022
PCHC FW Version	17.0.0.7013
PCH Information	
PCH Name	RPL
PCH Device ID	7A83
PCH Revision ID	A0
PCH SKU Type	Pre-Production ES
PCH Replacement State	Disabled

PCH Replaceable Counter	0
PCH Unlocked State	Enabled
Transactional FW Information	
Original image type	Corporate
Current sku type	Corporate
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	0x3DF6D507
Intel(R) Active Management Technology	Present/Enabled
Intel(R) Protected Audio Video Path	Present/Enabled
Intel(R) Platform Trust Technology	Present/Enabled
Persistent RTC and Memory	Present/Enabled
End Of Manufacturing	
NVAR Configuration State	Unlocked
EOM Settings	Lock (Flash, Config)
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
IPv6 Enablement	Disabled
Configuration State	Not Started
Provisioning Mode	PKI
Auto-BIST State	Disabled
Wired AMT Link Status	Link Up
WLAN uCode Config ID Installed	0x2725
Wireless LAN in Firmware	Intel(R) Wi-Fi 6 AX210
WLAN uCode Config ID Required	0x2725
Wireless LAN Hardware	Intel(R) Wi-Fi 6 AX210
Localized Language	English
Wireless C-Link Status	Enabled
Intel(R) SMLink0 MCTP Address	0xFF
System UUID	5e24fe9c-c8d0-455d-a79f-54ea5fbd3d97
Wireless MAC Address	00-00-00-00-00-00
Wireless IPv4 Address	0.0.0.0
Intel(R) Manageability HW Status	Enabled
Discrete vPro NIC on-board State	Disabled
On Board Discrete vPro NIC SMBus address	0x00

vPRO TBT Dock State	Disabled		
On dock vPro NIC SMBus address	0x00		
Thunderbolt Port1 SMBus Address	0x20		
Thunderbolt Port2 SMBus Address	0x21		
Thunderbolt Port3 SMBus Address	0x22		
Thunderbolt Port4 SMBus Address	0x23		
AMT Global State	Enabled		
Redirection Privacy / Security Level	Default		
Intel(R) Trusted Device Setup			
Intel(R) Signing Policy	Seal Signing Required		
Intel(R) Reseal Timeout	0x06		
TDS Seal State	Disabled		
Intel(R) Trusted Device Setup Supported	Disabled		
Intel(R) Protected Audio Video Path			
Attestation KeyBox	Not Provisioned		
PAVP State	Yes		
Security Version Numbers			
Trusted Computing Base SVN	1		
Anti Rollback SVNs			
PMC	1	[minimum allowed: 1]	
CSE	1	[minimum allowed: 1]	
ROT KM	1	[minimum allowed: 1]	
IDLm	1	[minimum allowed: 1]	
OEM KM	1	[minimum allowed: N/A]	
Intel(R) Unique Platform ID			
UPID supported	Disabled		
Intel(R) Platform Trust Technology			
Intel(R) PTT initial power-up state	Enabled		
Intel(R) PTT State	Enabled		
SMx State	Enabled		
RSA1K Support	Enabled		
FW Supported FPFs	FPF	UEP	
		*In Use	
	---	---	
1st OEM Key Hash Revoked	Not set	Disabled	# Disabled=0, Enabled=1
1st OEM Key Hash size	Not set	Enabled	# Disabled=0, Enabled=1
1st OEM RSA Key size	Not set	Enabled	# Disabled=0, Enabled=1
2nd OEM Key Hash Revoked	Not set	Disabled	# Disabled=0, Enabled=1
2nd OEM Key Hash size	Not set	Enabled	# Disabled=0, Enabled=1
2nd OEM RSA Key size	Not set	Disabled	# Disabled=0, Enabled=1
BSMM Anti Rollback	Not set	Enabled	# Disabled=0, Enabled=1
DNX Anti Rollback	Not set	Enabled	# Disabled=0, Enabled=1
EOM Flow	Not set	Flexible	# Flexible=0, Full=1
Error Enforcement Policy 0	Not set	Disabled	# Disabled=0, Enabled=1
Error Enforcement Policy 1	Not set	Disabled	# Disabled=0, Enabled=1
Flash Descriptor Verification	Not set	Disabled	# Disabled=0, Enabled=1
IDLm Anti Rollback	Not set	Enabled	# Disabled=0, Enabled=1
Intel PTT Encryption Key	Not set	Not Revoked	# Not Revoked=0, Revoked=1
Intel(R) Manageability HW Fuse Status	Not set	Enabled	# Enabled=0, Disabled=1
Intel(R) PTT	Not set	Enabled	# Disabled=0, Enabled=1
OEM ID	Not set	0x00	
OEM KM Anti Rollback	Not set	Disabled	# Disabled=0, Enabled=1

[illegible]

6.4.3 Checking on Set-up and Configuration Process

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

Intel(R) ME INFO Version: 17.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) ME INFO Version: 17.x.x.xxxx
Copyright(C) 2005 - 2017, Intel Corporation. All rights reserved.

Local FWUpdate: Success - Value matches FW value.

Note: Starting Intel® CSME16.0, the MEBx is integrated within the BIOS reference code, therefore, MEInfo will return 0000s in the value field of the MEBx parameter in MEInfo. In future generations, the MEBx parameter will be removed from MEInfo's output.

§ §

7 *Intel® ME Firmware Update*

FW Update 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.

FW Update 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® FW Update updates the entire Intel® ME code area. In addition, FW Update local can perform a partial update to update the different IUPs within the IFWI such as WCOD or LOCL partitions.

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

The image file that the tool uses for the update can be created using MFIT. Note that there is a separate layout for FWUpdate images in MFIT.

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

After a full FW Update, a host reset is needed to complete FW update. The user can also use the `-FORCERESET` option to do this automatically. However, in partial FW Update, a host reset is not required unless measurements are enabled and included

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 (FW Update).

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.

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® FW Update

The option to enable/disable FWUpdate supports three value which are:

1. FW_UPDATE_DISABLED: 0= Full Update disabled while Partial update is enabled
2. FW_UPDATE_ENABLED: 1= Both Full & Partial updates are enabled
3. FW_UPDATE_FULL_AND_PARTIAL_DISABLED: 3= Both Full & Partial updates are disabled

7.4 FW Update Flows

7.4.1 Full FW Update

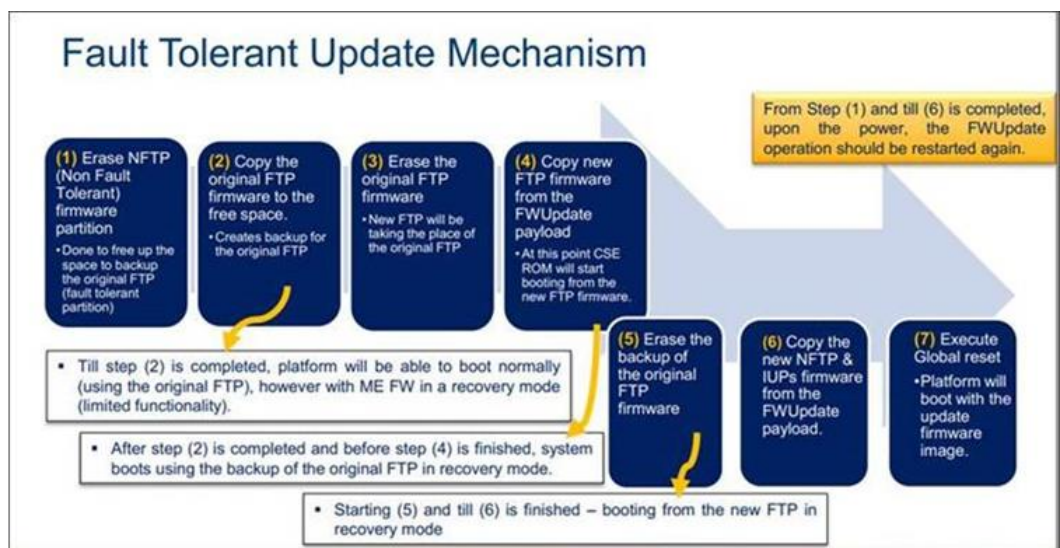
This allows the full update of Intel® ME Firmware. Starting CNL, if IUP's are present in the payload image along with Intel® ME Firmware, IUP's will also be updated along with Intel® ME as part of the Full FW Update.

Global Reset will be required to complete the FW Update operation.

PMC & PCHC Firmware Update: These FWs will be handled as part of the Full FW Update flow and cannot be updated on their own. PMC & PCHC FWs need to be stitched with Intel® ME Firmware using Intel® MFIT Tool and that image will be used as the payload to Full FW Update Flow for updating PMC & PCHC FWs.

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

The Update Flow explained: A full update flow is a fault tolerant one and consists of the following steps:



7.4.2 Partial FW Update

This allows the partial FW Update of IUP's (Independent Updatable Partitions) only e.g., WLAN micro-code, ISH Firmware, Localization, IUnit Loader etc.

7.5 Usage

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

Note: <File> is the image file of the FW to be updated. Is the same image file consumed by MFIT to stitch the final IFWI.

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 FW Update 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 FW Update 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>	This option is always used along with the -F option. The partition ID is requested using the "partid" option, which takes in the ID of the IUP 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 a detailed error including the missing partition will be returned by the tool. Also, 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 an FPT and contains FTP and NFTP partitions.
-FWVER	Display FW version in the flash
-FWVER<File>	Display FW version of the input update image file.
-H or -?	Displays the list of command line options supported by the Intel® FW Update tool.
-EXP	Shows examples about how to use the tools.
-VER	Shows the version of the tool.
-PARTVER<Partition ID>	Display flashed partition FW Version
-PARTVENDOR<Partition ID>	Display the Vendor ID of a specific partition
-INSTID <Instance ID>	Provide a specific instance ID of a partition to perform partial update

Option	Description
-SILENT	Update without any display or user prompts

7.6 Examples

7.6.1 Updates Intel® ME with Firmware Binary File

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

```
EFI:
```

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

This command updates Intel® ME with FW.BIN file. If the firmware on current platform is newer than then version in FW.BIN file, tool will promote a warning to let user know there will be a firmware downgrade (rollback) 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, tool will return an error. User can use -allowsv to allow same version update.

7.6.2 Partial Firmware Update

This command will perform a partial update of the FW via Intel® MEI for any of the following partitions:

1. WCOD
2. LOCL
3. IOMP
4. TBTB
5. NPHY
6. ISHC
7. IUNP

If fault tolerant partial FW Update is supported, then in addition to the above IUPs, partial update can also be performed on the following partitions:

1. OEMP
2. SPHY

Example:

```
FWUpdLcl.exe -f FW.bin -partid <partition from the above list>
```

```
EFI:
```

```
FWUpdLcl.efi -f FW.bin -partid <partition from the above list>
```

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 *-FWUpdLcl.exe -exp partid* option

Intel (R) Firmware Update Utility Version: 16.0.0.7075

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]
```

7.6.4 Language Codes

The below table includes that applicable instance IDs for the LOCL partition.

Language	Language Code
English	0x01
French	0x02
German	0x03
Chinese Traditional	0x04
Japanese	0x05
Russian	0x06
Italian	0x07
Spanish	0x08
Brazilian Portuguese	0x09
Korean	0x0A
Chinese Simplified	0x0B
Arabic	0x0C
Czech	0x0D
Danish	0x0E
Greek	0x0F

Language	Language Code
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 FW Update API Library*

8.1 Getting Started - FW Update Full Library

8.1.1 Introduction

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

8.1.2 Environment

The provided FW Update Libraries, both the full and the RS, are compiled using EDKII.

8.1.3 Setup

OEMs will need to include the relevant "*.h" file in their program and links it to the relevant *.lib file. Both *.h and *.lib exist in the relevant FW Kit.

8.1.4 Files in the Kit

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

In FW Update (Full Size) folder, multiple OSs are supported; Taking Windows64 as example for the below table:

File Name	Description
errorlist.c & errorlist.h	Source and header files for the error generation.
FWUpdateLib.lib	FW Update FS Library. Static library.
FWUpdateSample.c	Source file including a sample code for customers who intend to incorporate the Full Size FW Update library. This is only relevant to Windows.
FWUpdLcl64.exe	Full FW Update tool.

In FW Update Reduced Size (RS) folder:

File Name	Description
errorlist.c & errorlist.h	Source and header files for the error generation.
fwupdatelib.h	Header file including all the functions that can be used by customers
FWUpdateEFILib.lib	FW Update RS Library compiled in EFI64 EDKII.
FWUpdLclApp.c	Source file including a sample code for customers who intend to incorporate the Reduced Size FW Update library. This sample code is in EFI EDKII.
FwUpdLcl.efi	Reduced Size FW Update tool compiled from the sample code in EFI64 EDKII. Compiled from file FWUpdLclApp.c.

8.2 Function Description

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

Note: Some function titles are marked with the initials RS. This is intended for functions that apply for the FW Update Reduced Size library.

Note: Some function titles are marked with the initials FS. This is intended for functions that apply for the FW Update Full Size library.

8.2.1 Full FW Update from Buffer (FS)(RS)

UINT32 FwuFullUpdateFromBuffer (UINT 8 *Buffer, UINT 32 BufferLength, _UUID *OemId, void *Func(UINT 32, UINT 32));

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

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

8.2.2 Partial FW Update from Buffer (FS)(RS)

UINT32 FwuPartialUpdateFromBuffer (UINT8 *Buffer, UINT32 BufferLength, UINT32 PartitionId, void *Func(UINT32, UINT32));

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

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

8.2.3 Checking update progress (FS) (RS)

UINT32 FwuCheckUpdateProgress (bool *InProgress, Out UINT32 *CurrentPercent, Out UINT32 FwUpdateStatus, Out UINT32 *NeedResetType);

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

Arguments	FwuCheckUpdateProgress
Returns	Success, otherwise, failure with error code. A success would return the following: InProgress – True if update is in progress. False if update is finished CurrentPercent – Current percent of the update if the update is in progress FwUpdateStatus – FW error code status of the update if it finished (success or error code). Caller allocated. NeedResetType – Calls out the needed reset type after the update has finished. <ul style="list-style-type: none">• 0 = No reset is required• 1 = Host reset is required• 2 = CSE reset is required• 3 = Global reset is required

8.2.4 Get FW Update ability (FS)(RS)

UINT32 FwuEnabledState (Out UINT16 *EnabledState);

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

Arguments	FwuEnabledState
-----------	------------------------

Returns	Success, otherwise, failure with error code. A success would return the following: EnabledState 0 = FW_UPDATE_DISABLED = 0. i.e., Full Disabled & Partial Enabled. 1 = FW_UPDATE_ENABLED = 1. i.e., Full Enabled & Partial Enabled. 3 = FW_UPDATE_FULL_AND_PARTIAL_DISABLED i.e., Full Disabled & Partial Disabled.
---------	---

8.2.5 Set FW Update enabling state (FS)(RS)

UINT32 FwuSetEnabledState(IN UINT32 EnabledState);

Purpose: This function is only supported pre-EOP and defines the state of FWUpdate

Arguments	enabledState - FW Update enabling state. Caller allocated. 0 = FW_UPDATE_DISABLED = 0. i.e., Full Disabled & Partial Enabled. 1 = FW_UPDATE_ENABLED = 1. i.e., Full Enabled & Partial Enabled. 3 = FW_UPDATE_FULL_AND_PARTIAL_DISABLED i.e., Full Disabled & Partial Disabled.
Returns	Success, otherwise, failure with error code.

8.2.6 Retrieve OEM ID from Flash (FS)(RS)

UINT32 FwuOemId (Out _UUID *OemId);

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

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

8.2.7 Retrieve FW Type (FS)(RS)

UINT32 FwuFwType (In UINT32 *fwType);

Purpose: This function retrieves the FW type from flash.

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

	4 = FWU_FW_TYPE_CORPORATE
--	---------------------------

8.2.8 Retrieve FW Type from Buffer (FS)(RS)

UINT32 FwuFwTypeFromBuffer (IN UINT8 *buffer, IN UINT32 bufferLength, OUT UINT32 *fwType);

Purpose: This function retrieves the FW type from buffer.

Arguments	Buffer – Buffer of Update Image read from Update Image file. BufferLength –Length of the buffer in bytes.
Returns	Success, otherwise, failure with error code. A success would return the following: fwType 0 = FWU_FW_TYPE_INVALID 1 = FWU_FW_TYPE_RESERVED 2 = FWU_FW_TYPE_SLIM 3 = FWU_FW_TYPE_CONSUMER 4 = FWU_FW_TYPE_CORPORATE

8.2.9 Retrieve FW Type from image file (FS)

UINT32 FwuFwTypeFromFile(IN const char *fileName, OUT UINT32 *fwType);

Purpose: This function retrieves the FW type from an image file.

Arguments	fileName – File name of the update image
Returns	Success, otherwise, failure with error code. A success would return the following: fwType 0 = FWU_FW_TYPE_INVALID 1 = FWU_FW_TYPE_RESERVED 2 = FWU_FW_TYPE_SLIM 3 = FWU_FW_TYPE_CONSUMER 4 = FWU_FW_TYPE_CORPORATE

8.2.10 Retrieve PCH SKU from the flash image (FS)(RS)

UINT32 FwuPchSku(OUT UINT32 *pchSku);

Purpose: This function retrieves the PCH SKU.

Arguments	FwuPchSku
Returns	Success, otherwise, failure with error code. A success would return the following: 0 = FWU_PCH_SKU_INVALID

	1 = FWU_PCH_SKU_H 2 = FWU_PCH_SKU_LP
--	---

8.2.11 Retrieve PCH SKU from buffer (FS)(RS)

UINT32 FwuPchSkuFromBuffer(IN UINT8 *buffer, IN UINT32 bufferLength, OUT UINT32 *pchSku);

Purpose: This function retrieves the PCH SKU from buffer.

Arguments	Buffer – Buffer of Update Image read from Update Image file. BufferLength –Length of the buffer in bytes.
Returns	Success, otherwise, failure with error code. A success would return the following: 0 = FWU_PCH_SKU_INVALID 1 = FWU_PCH_SKU_H 2 = FWU_PCH_SKU_LP

8.2.12 Retrieve PCH SKU from image file (FS)

UINT32 FwuPchSkuFromFile(IN const char *fileName, OUT UINT32 *pchSku);

Purpose: This function retrieves the PCH SKU from an image file.

Arguments	fileName – File name of the update image
Returns	Success, otherwise, failure with error code. A success would return the following: 0 = FWU_PCH_SKU_INVALID 1 = FWU_PCH_SKU_H 2 = FWU_PCH_SKU_LP

8.2.13 Retrieves the product segment from the flash image (FS)(RS)

UINT32 FwuProductSegmentFromFlash(OUT UINT32 *productSegment);

Purpose: This function retrieves the product segment from the flash image.

Arguments	FwuProductSegmentFromFlash
Returns	Success, otherwise, failure with error code. A success would return the following: Product Segment: 1 = FWU_PRODUCT_SEGMENT_CLIENT 2 = FWU_PRODUCT_SEGMENT_HEDT 3 = FWU_PRODUCT_SEGMENT_WS 4 = FWU_PRODUCT_SEGMENT_CONVERGED_MOBILITY

	5 = FWU_PRODUCT_SEGMENT_IOT 6 = FWU_PRODUCT_SEGMENT_SMALL_CORE_ENTRY_LEVEL
--	---

8.2.14 Retrieves the product segment from buffer (FS)(RS)

UINT32 FwuProductSegmentFromBuffer(IN UINT8 *buffer, IN UINT32 bufferLength, OUT UINT32 *productSegment);

Purpose: This function retrieves the product segment from buffer.

Arguments	Buffer – Buffer of Update Image read from Update Image file. BufferLength –Length of the buffer in bytes.
Returns	Success, otherwise, failure with error code. A success would return the following: Product Segment: 1 = FWU_PRODUCT_SEGMENT_CLIENT 2 = FWU_PRODUCT_SEGMENT_HEDT 3 = FWU_PRODUCT_SEGMENT_WS 4 = FWU_PRODUCT_SEGMENT_CONVERGED_MOBILITY 5 = FWU_PRODUCT_SEGMENT_IOT 6 = FWU_PRODUCT_SEGMENT_SMALL_CORE_ENTRY_LEVEL

8.2.15 Retrieves the product segment from image file (FS)

UINT32 FwuProductSegmentFromFile(IN const char *fileName, OUT UINT32 *productSegment);

Purpose: This function retrieves the product segment from an image file.

Arguments	fileName – File name of the update image
Returns	Success, otherwise, failure with error code. A success would return the following: Product Segment: 1 = FWU_PRODUCT_SEGMENT_CLIENT 2 = FWU_PRODUCT_SEGMENT_HEDT 3 = FWU_PRODUCT_SEGMENT_WS 4 = FWU_PRODUCT_SEGMENT_CONVERGED_MOBILITY 5 = FWU_PRODUCT_SEGMENT_IOT 6 = FWU_PRODUCT_SEGMENT_SMALL_CORE_ENTRY_LEVEL

8.2.16 Checking compatibility with update image from buffer (FS)(RS)

UINT32 FwuCheckCompatibilityFromBuffer(IN UINT8 *buffer, IN UINT32 bufferLength);

Purpose: This function checks if the Update Image from the buffer, is compatible to the Flash Image, by comparing TCB SVN, ARB SVN, VCN of all the updatable partitions in the Flash Image and in the Update Image.
It is not possible to update to a lower value.

Note: Update may not be possible for other reasons, that are not checked by this function.

Arguments	Buffer – Buffer of Update Image read from Update Image file. BufferLength –Length of the buffer in bytes.
Returns	Success if compatible, otherwise, failure with error code.

8.2.17 Checking compatibility with update image from image file (FS)

UINT32 FwuCheckCompatibilityFromFile(IN const char *fileName);

Purpose: This function checks if the Update Image from an image file, is compatible to the Flash Image, by comparing TCB SVN, ARB SVN, VCN of all the updatable partitions in the Flash Image and in the Update Image.
It is not possible to update to a lower value.

Note: Update may not be possible for other reasons, that are not checked by this function.

Arguments	fileName – File name of the update image
Returns	Success if compatible, otherwise, failure with error code.

8.2.18 Get version of specific partition from flash image (FS)(RS)

UINT32 FwuPartitionVersionFromFlash(UINT32 PartitionId, UINT16 *Major, UINT16 *Minor, UINT16 *Hotfix, UINT16 *Build);

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

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

8.2.19 Get version of specific partition from buffer (FS)(RS)

UINT32 FwuPartitionVersionFromBuffer (UINT8 *Buffer, UINT32 BufferLength, UINT32 PartitionId, UINT16 *Major, UINT16*Minor, UINT16 *Hotfix, UINT16 *Build);

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

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

8.2.20 Get vendor ID for a specific partition (FS)(RS)

```
UINT32 FwuPartitionVendorIdFromFlash (UINT32 PartitionId, Out UINT32 VendorId);
```

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

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

8.2.21 Performing a full FW Update (FS)

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

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

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

8.2.22 Performing a partial FW Update (FS)

```
UINT32 FwuPartialUpdateFromFile (const char *fileName, UINT32 PartitionId, void(*func)( UINT32, UINT32));
```

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

Arguments	fileName – File name referring to the update image to be provided PartitionId – ID of the partition to update. Please refer to our list of IUPs to learn about partially updateable partitions
-----------	---

	func – A callback function that reports the progress of sending the buffer to FW.
Returns	Success, otherwise, failure with error code.

8.2.23 Retrieving partition version from image file (FS)

UINT32 FwuPartitionVersionFromFile(const char *fileName, UINT32 partitionId, Out UINT16 *major, Out UINT16 *minor, Out UINT16 *hotfix, Out UINT16 *build);

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

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

8.2.24 Retrieving instance of a partition (FS)

UINT32 FwuPartitionInstances(UINT32 partitionId, Out UINT32 *currentInstanceId, Out UINT32 *expectedInstanceId);

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

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

8.2.25 Performing a partial FW Update with Instance ID from buffer (FS)

UINT32 FwuPartialUpdateWithInstanceIdFromBuffer(UINT8 *buffer, UINT32 bufferLength, UINT32 PartitionId, UINT32 instanceId, void (*func)(UINT32, UINT32));

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

Arguments	Buffer – Buffer of the update image read from the update image file BufferLength – Length of the buffer in bytes PartitionId – ID of the partition to update, only partially updateable partitions apply
-----------	---

	InstanceId – Instance ID of the partition to update func – A callback function that reports the progress of sending the buffer to FW.
Returns	Success, otherwise, failure with error code.

8.2.26 Performing a partial FW Update with Instance ID from file (FS)

```
UINT32 FwuPartialUpdateWithInstanceIdFromFile( const char *fileName, UINT32
partitionId, UINT32instanceId, void(*func)( UINT32, UINT32));
```

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

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

8.2.27 Creating a restore point image into buffer (FS)(RS)

```
UINT32 FwuSaveRestorePointToBuffer(OUT UINT8 **buffer, OUT UINT32
*bufferLength);
```

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

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

8.2.28 Creating a restore point image into file (FS)

```
UINT32 FwuSaveRestorePointToFile( const char *fileName);
```

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

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

8.2.29 Checking power source (FS)

UINT32 FwuPowerSource(OUT UINT32 *powerSource);

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

Arguments	<i>FwuPowerSource</i>
Returns	Success, otherwise, failure with error code. A success would return the following: <i>powerSource</i> = power source would show one of the below values <ul style="list-style-type: none">• 0 = Unknown• 1 = AC• 2 = DC

8.2.30 Set ISH configuration file (RS Only)

UINT32 FwuSetIshConfig (UINT8 *Buffer, UINT32 BufferLength);

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

Arguments	<i>Buffer</i> – Buffer of IUP <i>BufferLength</i> – Length of the buffer in bytes
Returns	Success, otherwise, failure with error code

8.2.31 Get PDT version and VDV version (RS Only)

UINT32 FwuGetIshPdtVersion (Unit8 *PdtVersion, UINT8 *VdvVersion);

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

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

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

The Intel® Manifest Extension Utility (MEU) inputs a firmware binary created by a 3rd party and outputs an independent-updateable partition (IUP) that is compressed and signed. After completing this process the signed binary can be added to the SPI flash image using the Intel® MFIT 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 *RPL 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
Non-Application Specific Fixed Offset Item Descriptions					
MEBxPassword	<p>Overrides the MEBx default password. It must be at least eight characters and not more than 32 characters in length. All characters must meet the following:</p> <p>ASCII(32) <= char <= ASCII(126)</p> <p>Cannot contain these characters: , : "</p> <p>Must contain for complexity:</p> <ol style="list-style-type: none"> 1. At least one Digit character (0 - 9) 2. At least one 7-bit ASCII non alpha-numeric character above 0x20 (Example: ! \$;) 3. Both lower-case and upper case Latin. 4. Underscore and space are valid characters but are not used in determination of complexity. 	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>NOTES: 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><tr><th>Bit</th><th>Description</th><th>Note</th></tr><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></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	0x80 = Disabled 0x81 = CoreWell 0x82 = Primary Well 0x83 = ME Well 0x86 = WLAN Sleep via SLP_WLAN#	Global	No
OEM Tag	A human readable 32-bit number to describe the flash image represented by value.	32	Readable 32 bit hex value identifying the image. Can be empty (Null).	Global	No
OEM FW Version	64-bit number for OEMs to store digested FW version if needed	64	OEM choice	ME	Yes

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	ME	No
FW Update State	Enabled Firmware Update Capability	1	0 = Disabled 1 = Enabled 3 = Full and Partial disabled	Global	Yes
TCSS FW Partial Update	Enable partial update of TCSS components	1	0 = Disabled 1 = Enabled	ME	No
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	Global	No

Fixed Offset Name	Description	Data Length (in Bytes)	Expected Value	Reset Type	Manufacturing Post EOM/Pre EOP
Attestation keyBox	A key hash for the attestation of the keyBox	16384	File	ME	No
CSME Measured Boot to TPM	Enables/Disables Measured boot	1	0 = Disabled 1 = Enabled	Global	No
Discrete vPro NIC on-board State	Enables/Disables the discrete vPro NIC	1	0 = Disabled 1 = Enabled	ME	No
EOM Settings	Flexible EOM 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	ME	No
On Board Discrete vPro NIC SMBus address	Address of the SMBus on which discrete Dock would operate	8	Address	ME	No
On dock vPro NIC SMBus address	Address of the SMBus on which vPro Dock would operate	8	Address	ME	No
TDS Reseal Timeout	Defines the Timeout value for resealing the TDS capability	8	Timeout Value	ME	No
TDS Signing Policy	Defines the signing policy for TDS; whether to seal sign or sign only the PMF or no signing at all	8	{0, 1, 2}, // PossibleValuesIDs {"Seal Signing Requested", "PMF Signing Only", "No Signing"}, // possibleValuesStrings	ME	No
Trusted Device Setup Supported	Enables/Disables the TDS support on the platform	1	0 = Disabled 1 = Enabled	ME	No
vPRO TBT Dock State	Enables/Disabled the Thunderbolt vPro Dock capability	1	0 = Disabled 1 = Enabled	ME	No

Fixed Offset Name	Description	Data Length (in Bytes)	Expected Value	Reset Type	Manufacturing Post EOM/Pre EOP
Thunderbolt Port1 SMBus Address	Address to TBT port1 SMBus	8	Address	ME	No
Thunderbolt Port2 SMBus Address	Address to TBT port2 SMBus	8	Address	ME	No
Thunderbolt Port3 SMBus Address	Address to TBT port3 SMBus	8	Address	ME	No
Thunderbolt Port4 SMBus Address	Address to TBT port4 SMBus	8	Address	ME	No
AMT Related NVARs					
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)	ME	Yes
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)	ME	Yes
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

Fixed Offset Name	Description	Data Length (in Bytes)	Expected Value	Reset Type	Manufacturing Post EOM/Pre EOP
Redirection Privacy / Security Level	Redirection (KVM, SOL, Storage Redirection) privacy level and configuration (RCFG, CCM) settings.	1	Default 0x01 Enhanced 0x02 Extreme 0x03 Default: SOL enabled = true Storage Redirection enabled = true KVM enabled = true Opt-in can be disabled = true KVM opt-in configurable remotely = true RCFG and CCM = true Enhanced: SOL enabled = true Storage Redirection enabled = true KVM enabled = true Opt-in can be disabled = false Opt-in configurable remotely = true RCFG and CCM = true Extreme SOL enabled = false Storage Redirection enabled = false KVM enabled = false Opt-in can be disabled = false KVM opt-in configurable remotely = N/A RCFG and CCM = false	ME	No
Embedded Host Based Config	Embedded Host Based Configuration State.	1	0 = Disabled 1 = Enabled	ME	No
Firmware KVM Screen Blanking	Screen Blanking Enabled	1	0 = No 1 = Yes	ME	No
PKI Domain Name Suffix	PKI DNS Suffix. Null terminated string	223	PKI DNS Suffix in dotted string format	ME	Yes
Config Server FQDN	Configuration Server Fully Qualified Domain Name (FQDN)	256	Example: "intelFVE.com"	ME	Yes

Fixed Offset Name	Description	Data Length (in Bytes)	Expected Value	Reset Type	Manufacturing Post EOM/Pre EOP
RCFG/ZTC	R Configuration	1	0 = Disabled 1 = Enabled This is update only NVAR. Tool won't be able to read expected value.	ME	Yes
*Redirection	This is a bit-field indicating the enable/disable status of Storage Redirection, SOL, and KVM features in Intel® AMT. bit[0]: 1 – Storage Redirection enabled, 0 – disabled bit[1]: 1 – SOL enabled, 0 – disabled bit[2]: 1 – KVM enabled, 0 – disabled	4	Range: 0-7 Example: Value of 4 (100b) indicates that KVM is enabled. Value of 3 (011b) indicates that Storage Redirection, and SOL are enabled. Value of 7 (111b) indicates that Storage Redirection, SOL, and KVM are enabled. This is update only NVAR. Tool won't be able to read expected value	ME	Yes
*Opt-in Policy	Change User Opt-in (lower nibble). NONE = 0, KVM = 1, ALL = F Disable Opt-In Configurable from Remote IT (upper nibble). 0 - Opt-in is NOT Configurable from Remote IT 1 - Opt-in is Configurable from Remote IT	1	0x00 0x10 0x01 0x11 0x0F 0x1F Examples: In addition to the following, the values may not be configured remotely: Value of 0x00 indicates User Consent is not required. Value of 0x01 indicates User Consent is required for KVM only. Value of 0x0F indicates User Consent is required for (ALL). In addition to the following, the values may be configured remotely: Value of 0x10 indicates User Consent is not required. Value of 0x11 indicates User Consent is required for KVM only. Value of 0x1F indicates User Consent is required for (ALL).	ME	Yes

Fixed Offset Name	Description	Data Length (in Bytes)	Expected Value	Reset Type	Manufacturing Post EOM/Pre EOP
Host Name	Set Host Name Only	64	RaptorLake	ME	Yes
Domain Name	Set Domain Name Only	192	myserver.intel.com amr.corp.intel.com www.intel.com mymail.somecollege.edu	ME	Yes
Config Server IPv6/IPv4 Address	Set Provisioning Server (IPv4/IPv6) Address	60	Example of IPV4: 192.168.1.200 255.255.255.0	ME	Yes
Config Server IPv6/IPv4 Port	Set Provisioning Server (IPv4/IPv6) Port	2	Within Range: 0 – 0xFFFF	ME	Yes
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
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
Field Programming Fuses					
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

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

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

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

