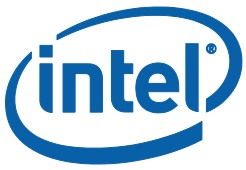


Ice Lake Platform System Tools - Intel® Management Engine Firmware 13.0

User Guide

September 2019

Revision 1.12

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, 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, 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.

Contents

Figures

Tables

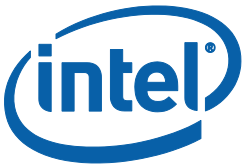
Revision History

Revision Number	Description	Date
0.6	• Initial Release.	August 2017
0.61	• Updated Firmware Update Errors list in Appendix B: B.2	January, 2018
0.7	• Added new section 3.4.18 under Intel FIT tool describing Power Delivery PD Controller Configuration • Updated Intel® MEInfo and Intel® MEManuf with RPMC implementation. • Updated section 6.3 – Added "*In Use* to indicate the currently used column for FPFs values in the Intel® MEInfo results examples. • Add details in Chapter 5 for –ALL command for Intel® MEManuf. • Added a note clarifying Privacy/Security Level Default Setting under "Appendix A – Intel® NVARs". • Updated Chapter 6 Intel® MEInfo list of components table. • Added a new section 3.4.15 for setting the PMC binary. • Added details across Intel® MEInfo, Intel® MEManuf, and Intel® FPT tools' sections related to Anti Rollback Mechanism. • Update section 3.4.5 Intel® FIT tool with the new ability to build a FWUpdate image without full image.	March, 2018
0.8	• Added new Chapter 8 "UEFI Sample Application Leveraging FWUpdate API Library" • Updated Appendix with Appendix B.3 "FWUpdate API Library Errors" • Updated Appendix A: Added the FPFs list • Updated Appendix A: Added OEM Secure Boot Policy bit mapping • Updated Appendix A: Updated NVARs naming	May 2018
0.81	• Fixed OEM Secure Boot Policy bit definition & length under Appendix A Intel® ME NVARs	June 2018
0.82	• Added a note with power cycle recommendation when flashing using Intel® FPT under Chapter 4: Flash Programming Tool • Fixed expected values for two FPFs under Appendix A Intel® ME NVARs • Updated Appendix A Intel® ME NVARs: Updated NVARs naming	June 2018
0.9	• Updated Chapter 8 "UEFI Sample Application Leveraging FWUpdate API Library" with new API implementations. • Marked old API implementations in Chapter 8 "UEFI Sample Application Leveraging FWUpdate API Library" as Deprecated.	September 2018

0.91	• Updated Appendix B with new system level error codes.
	• Added 2 new EOL tests in Chapter 5 "Intel® MEManuf"
	  • Boot Guard Status
	  • FW Status
	• Updated Chapter 8 "UEFI Sample Application Leveraging FWUpdate API Library" with new APIs and with RS mark on relevant Reduced Size APIs
	• Removed -ErrList command from Chapter 5 "Intel® MEManuf"
	• Update Appendix A – Intel® NVARs. eDP Port and LSPCON Port Config NVARs need only ME reset type.
		November 2018

	• Removed MEBx password protection requirement from Chapter 7 Intel® ME Firmware Update.	
0.92	• Removed System Integrator from MEManuf and Appendix A : Intel® ME NVARs	November 2018
1.0	• Updated Tools output to latest revision • Added details regarding the ErrorAction Field. • Updated Error Return sub-chapter. • Updated NVARs Descriptions. • Updated Appendix B with new system level error codes.	February 2019
1.01	• Removed "FWUpdLcl -generic" command from FWUpdate tool. • Updated Appendix B with new command line tools errors.	February 2019
1.02	• Updated supported -CLOSEMNF arguments in FPT tool. • Updated "Intel-Recommended Access Settings" table. • Updated Command Line Tools Errors with new errors.	May 2019
1.1	• Fixed shifting issue in the errors Index (Appendix B) • Fixed the Intel® FIT commands in Chapter 3 • Added new FWUpdate table displaying deprecated vs new functions in Chapter 8.2.1 • Updated MEManuf XML example in Chapter 5.4.2	June 2019
1.12	• Removed PMX chapter • Removed fparts.txt command • Removed -p -list -comparefpf -hashed commands • Split error list • Added new errors to error list • Updated MEINfo Consumer Intel® ME FW SKU • Updated MEManuf.xml config file • Updated server config FQDN length value	September 2019

§ §



1 Introduction

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

1.1.1 Terminology

Acronym/Term	Definition
3PDS	3rd Party Data Storage
AC	Alternating Current
Agent	Software that runs on a client PC with OS running
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
BBBS	BIOS Boot Block Size
BIN	Binary file
BIOS	Basic Input Output System
BIOS-FW	Basic Input Output System Firmware
BIST	Built In Self-Test
CCM	Client Control Mode (Host Based Setup and Configuration)
CLI	Command Line Interface
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
EC	Embedded Controller

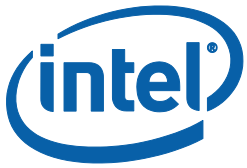
Acronym/Term	Definition
EEPROM	Electrically Erasable Programmable Read Only Memory
EFI	Extensible Firmware Interface
EHCI	Enhanced Host Controller Interface
EID	Endpoint ID
End User	The person who uses the computer (either Desktop or Mobile). In corporate, the user usually does not have administrator privileges. The end user may not be aware to the fact that the platform is managed by Intel® AMT.
EOP	End Of Post
FCIM	Full Clock Integrated Mode
FCSS	Flex Clock Source Select
FDI	Flexible Display Interface
FLOCKDN	Flash Configuration Lock-Down
FMBA	Flash Master Base Address
FOV	Fixed Offset Variable
FPSBA	Flash PCH Strap Base Address
FPT	Flash Programming Table
FQDN	Fully Qualified Domain Name
FRBA	Flash Region Base Address
FTP	Fault Tolerant Partition
Full Image	A full image starts with an FPT and contains FTP and NFTP partitions
Full Update	Updates all the regions
FW	Firmware
FWUpdate	Firmware Update
FWUpdateLib	Firmware Update Librart
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

Acronym/Term	Definition
HostIF	Host Interface
HTTP	Hyper Text Transfer Protocol
HW	Hardware
IBEN	Input Buffer Enable
IBV	Independent BIOS Vendor
ICC	Integrated Clock Configuration
ID	Identification
IDER	Integrated Drive Electronics Redirection
INF	An information file (.inf) used by Microsoft operating systems that support the Plug and Play feature. When installing a driver, this file provides the OS with the necessary information about driver filenames, driver components, and supported hardware.
Intel® DAL	Intel® Dynamic Application Loader (Intel® DAL)
Intel® FIT	Intel® Flash Image Tool
Intel® FPT	Intel® Flash Programming Tool
Intel® ME	Intel® Management Engine. The embedded processor residing in the chipset PCH.
Intel® MEBx	Intel® Management Engine BIOS Extensions
Intel® MEI driver	Intel® AMT host driver that runs on the host and interfaces between ISV Agent and the Intel® AMT HW.
Intel® MEINFO	Intel® Manageability Engine Information Tool to check whether ME is alive or not.
Intel® MEInfoWin	Windows® version of Intel® Manageability Engine Information Tool
Intel® MEManuf	Intel® Manageability Engine Manufacturing Tool validates Intel® ME functionality on the manufacturing line
Intel® MEManufWin	Windows® version of Intel® Manageability Engine Manufacturing Tool
ISV	Independent Software Vendor
IT User	Information Technology User. Typically very technical and uses a management console to ensure multiple PCs on a network function.
JEDECID	Joint Electronic Device Engineering Councils ID. Standard Manufacturer's Identification Code that is assigned, maintained and updated by the JEDEC office
JTAG	Joint Test Action Group
KVM	Keyboard, Video, Mouse
LAN	Local Area Network
LED	Light Emitting Diode
LOCL	Localization Language

Acronym/Term	Definition
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
NFTP	Non-Fault Tolerant Partition
NVM	Non-Volatile Memory
NVRAM	Non-Volatile Random Access Memory
OCKEN	Output Clock Enable
ODM	Original Device Manufacturer
OEM	Original Equipment Manufacturer
OEM ID	Original Equipment Manufacturer Identification
OOB	Out Of Band
OOB interface	Out Of Band interface. An SOAP/XML interface over secure or non-secure TCP protocol.
OS	Operating System
OS Hibernate	OS state where the OS state is saved on the hard drive.
OS not Functional	The Host OS is considered non-functional in Sx power state in any one of the following cases when the system is in S0 power state: OS is hung. After PCI reset. OS watch dog expires. OS is not present.
OVR	Override
PAVP	Protected Video and Audio Path
Partial Image	A partial image starts with either WVOD or LOCL partitions. No FPT, FTP, and NFTP in the file
Partial Update	Only updates regions that require an update such as WECOD or LOCL
PC	Personal Computer
PCH	Peripheral Controller Hub
PCI	Peripheral Component Interconnect
PCIe	Peripheral Component Interconnect Express
PDR	Platform Descriptor Region
PHY	Physical Layer

Acronym/Term	Definition
PID	Provisioning ID
PKI	Public Key Infrastructure
PM	Power Management
PRTC	Protected Real Time Clock
PSK	Pre-Shared Key
PSL	PCH Strap Length
RCFG	Remote Configuration
RCS	Remote Connectivity Service
RNG	Random Number Generator
ROM	Read Only Memory
RPAS	Remote Connectivity Service
RSA	A public key encryption method
RTC	Real Time Clock
S0	A system state where power is applied to all HW devices and the system is running normally.
S1, S2, S3	A system state where the host CPU is not running but power is connected to the memory system (memory is in self refresh).
S4	A system states where the host CPU and memory are not active.
S5	A system state where all power to the host system is off but the power cord is still connected.
SDK	Software Development Kit.
SEBP	Single Ended Buffer Parameters
SHA	Secure Hash Algorithm
SMB	Small Medium Business mode
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.

Acronym/Term	Definition
TCP/IP	Transmission Control Protocol/Internet Protocol.
TLS	Transport Layer Security
UEP	Unified Emulation Partition
UI	User Interface
UIM	User Identifiable Mark
UMA	Unified Memory Access
Un-configured state	The state of the Intel® ME FW when it leaves the OEM factory. At this stage the Intel® ME FW is not functional and must be configured.
UNS	User Notification Services
UPDPARAM	Update Parameter Tool
USB	Universal Serial Bus
USBr	Universal Serial Bus Redirection
UUID	Universally Unique Identifier
VLAN	Virtual Local Area Network
VSCC	Vendor Specific Component Capabilities
WCOD	Wireless Card Driver
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 13.0	CDI document
PCH EDS	CDI
Ice Lake SPI Programming Guide	Release kit
ISS Firmware Bring Up Guide	CDI

§ §

2 Preface

2.1 Overview

This document covers the system tools used for creating, modifying, and writing binary image files, manufacturing testing, Intel® ME setting information gathering, and Intel® ME FW updating. The tools are located in **Kit\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:

- Ice Lake PCH platform – All tools in the Ice Lake PCH FW release kit are designed for 9th Generation Intel® Core™ Processor and Ice 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 Ice Lake PCH platform.
- Intel® ME Firmware 13.0 SKU – A common set of tools are provided for the following Intel® ME FW 13.0 SKUs: Consumer Intel® ME FW SKU and Corporate Intel® ME FW SKU. The following features are only available for Corporate Intel® ME FW SKUs and Consumer Intel® ME FW SKU users should generally ignore them:

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

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

2.2 Image Editing Tools

The following tools create and write flash images:

- Intel® FIT:

  Combines the Descriptor, GbE, BIOS, PDR, ISH and Intel® ME FW binaries into one image.

  Configures soft straps and NVARs for Intel® ME settings and another for

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

- FPT:

  Programs the SPI flash memory of individual regions or the entire flash device.

  Modifies some Intel® ME settings (NVAR), FPFs after Intel® ME is flashed on the SPI part.

- FWUpdate – updates the Intel® ME FW code region on a flash device that has already been programmed with a complete SPI image.

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

2.3 Manufacturing Line Validation Tool

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

2.4 Intel® Management Engine Setting Checker Tool

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

2.5 Operating System Support

Table 2-1. OS Support for Tools

Intel® ME and Manufacturing Tools	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 Programing Tool	x	X		x	x
Intel® MEManuf 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® Flash Image Tool		x	x		x

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® ICC CCT Tool	x			x	

Notes:

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

2.6 Generic System Requirements

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

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

Refer the description of each tool for its exact requirements.

Table 2-2. Tools Summary

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

2.7 Error Return

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

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

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

2.8 Usage of Double-Quote Character (")

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

FPT.efi –f "" Wlan well power config"".

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

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

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

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

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

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

2.9 Control Handler Support

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

§ §

3 Intel® Flash Image Tool

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

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

   Binary files of the following regions:

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

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

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

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

3.1 System Requirements

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

3.2 Flash Image Details

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

Figure 3-1. SPI Flash Image Regions

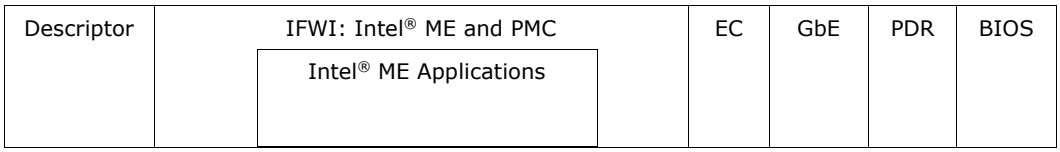


Table 3-1. Flash Image Regions – Description

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

3.2.1 Flash Space Allocation

Space allocation for each region is determined as follows:

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

3.3 Required Files

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

- vsccommn.bin
- .xml file

3.4 Intel® Flash Image Tool

Refer following for further information:

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

3.4.1 Configuration Files

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

3.4.2 Creating New Configuration

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

3.4.3 Opening Existing Configuration

To open an existing configuration file:

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

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

3.4.4 Saving Configuration

To save the current configuration in an XML file:

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

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

3.4.5 Environment Variables

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

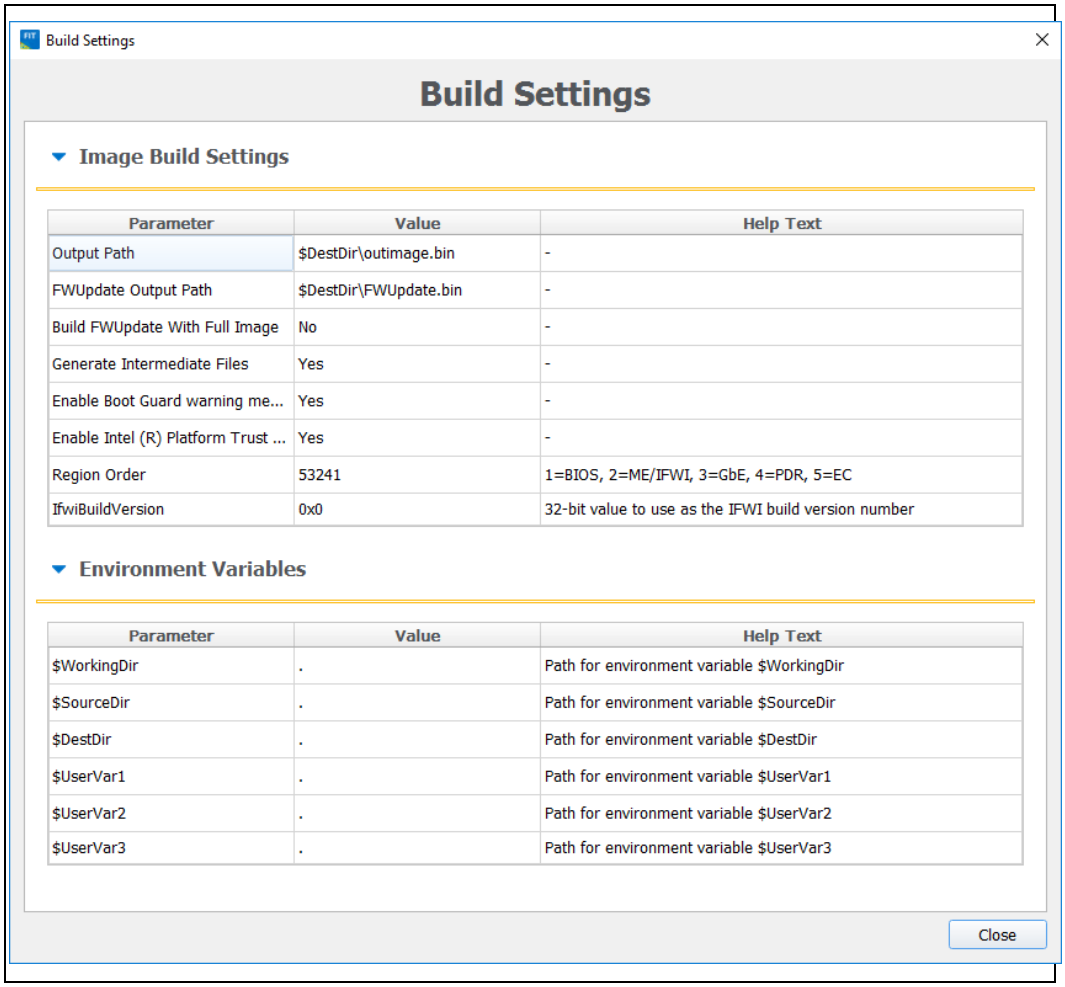
It is recommended that the environment variables be the first thing that the user sets when working with a new configuration. This ensures that FIT can properly substitute environment variables into paths to keep them relative. Doing this also speeds up configuration because many of the **Open File**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

To modify the environment variables:

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

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

Figure 3-2. Environment Variables Dialog



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

4. Click **OK**.

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

Note: Build Settings

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

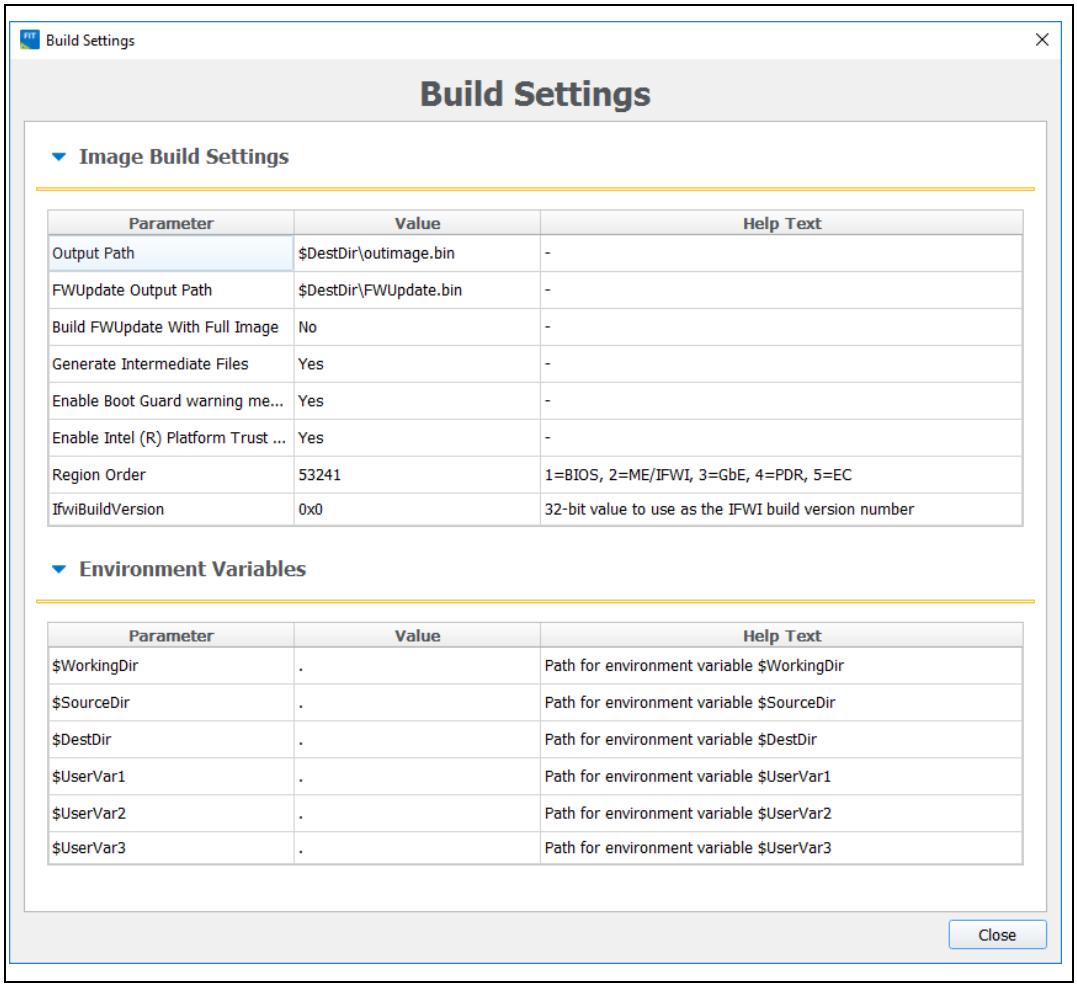
To modify the build setting:

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

Table 3-2. Build Settings Dialog Options

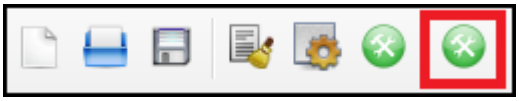
Option	Description
Output path	The path and filename where the final image should be saved after it is built. **NOTE:** Using the $DestDir environment variable makes the configuration more portable.
FWUpdate Output Path	The path and filename where the FWUpdate image should be saved after it is built.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

Figure 3-3. Build Settings Dialog



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

Figure 3-4. FWUpdate image build icon



3.4.6 Modifying the Flash Descriptor Region

The Flash Descriptor Region contains information about the flash image and the target hardware. This region contains the read/write values. It is important for this region to be configured correctly or the target computer may not function as expected. This region also needs to be configured correctly in order to ensure that the system is secure.

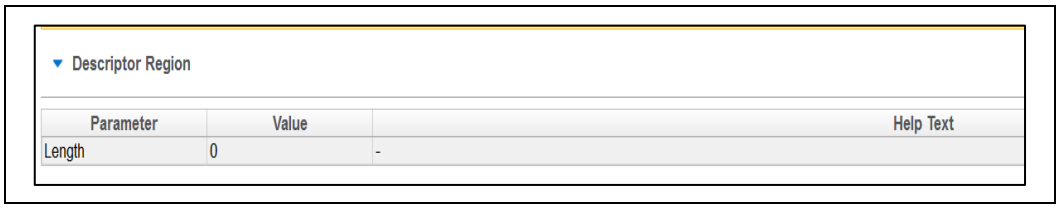
3.4.7 Descriptor Region Length

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

To set the value of the Descriptor Region Length parameter:

1. Select **Flash Layout** in the left pane; the **Length** parameter appears in the right pane.
2. Enter any non-zero value into the dialog to set the length of the region and click **OK**.

Figure 3-5. Descriptor Region Length Parameter

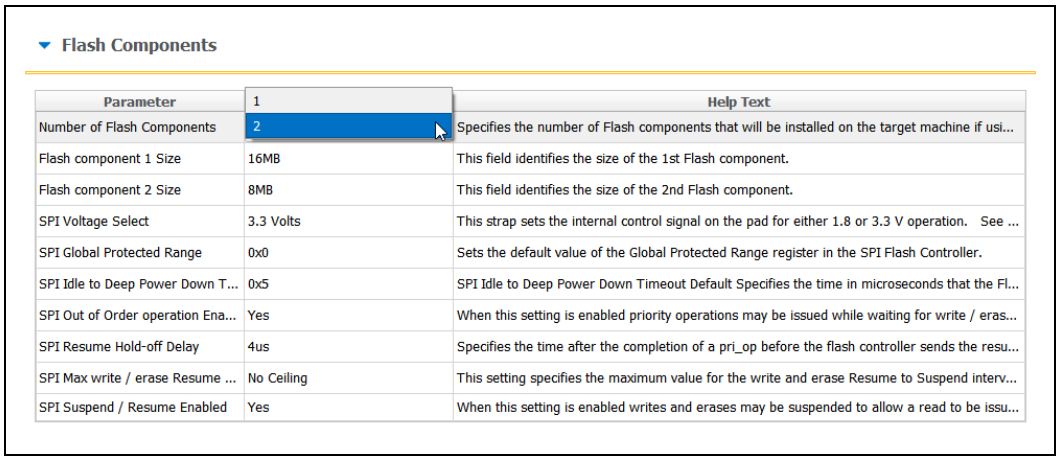


3.4.8 Setting the Number and Size of the Flash Components

To set the number of flash components:

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

Figure 3-6. Flash Settings > Flash Components



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

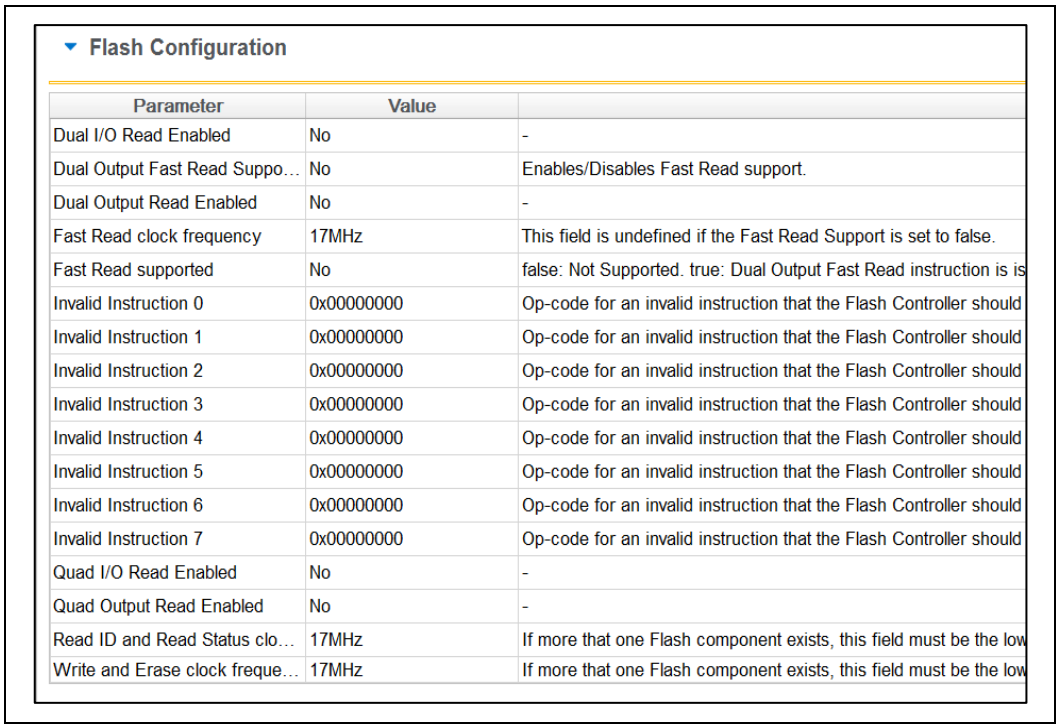
To set the size of each flash component:

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

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

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

Figure 3-7. Flash Settings → Flash Configuration



3.4.9 Region Access Control

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

Table 3-3. Region Access Control Table

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

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

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

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

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

Key:

0 – Denied access

1 – Allowed access

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

Table 3-4. CPU/BIOS Access

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

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

Example:

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

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

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

To set these access values in FIT:

1. Select **Flash Settings Tab → Host CPU/BIOS Master Access, Intel ME Master Access, Gbe Master Access and EC Master Access** in the right pane; the access parameters are listed in the right pane.
2. Double-click on each parameter and set its access value in one of the following ways:

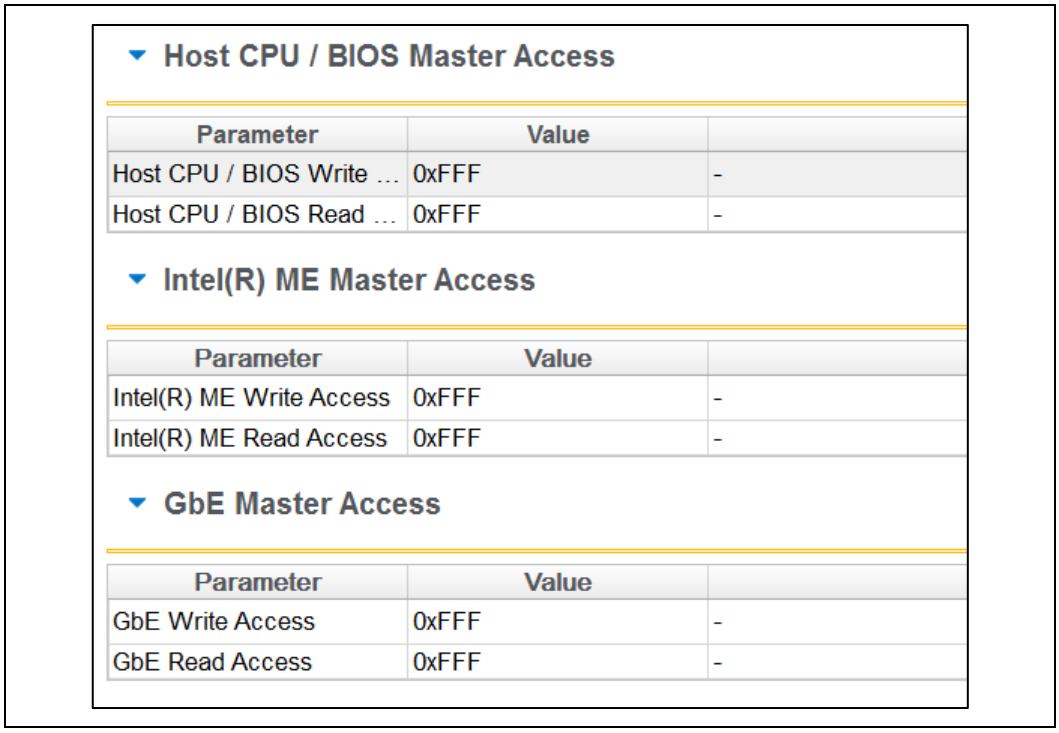
   To generate an image for debug purposes or to leave the SPI region open: select 0xFF for both read and write access in all three sections.

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

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

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

Figure 3-8. Descriptor Region → Master Access Section



3.4.10 VSCC Table

This section is used to store information to setup flash access for Intel® ME. This does not have any effect on the usage of the FPT. **If the information in this section is incorrect, Intel® ME FW may not communicate with the flash device.** The information provided is dependent on the flash device used on the system. (For more information, refer Ice Lake PCH-LP SPI Programming Guide, Section 6.4 and for Lewisburg C620 family platform, refer to LBG SPI Programming Guide, Section 4.4.)

VSCC Table can be accessed:

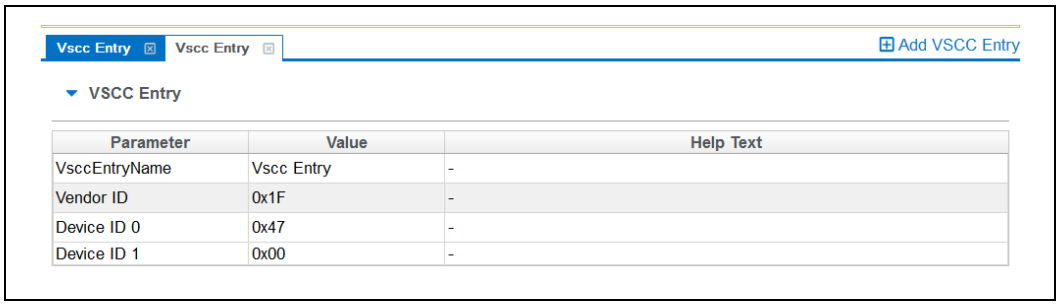
1. Select Flash Settings Tab on the left pan
2. Expand VSCC Entries on the right pan as shown in Figure3-9 below:

3.4.11 Adding New Table

To add a new table:

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

Figure3-9. Add VSCC Table Entry Dialog



2. Enter a name into the **Entry Name** field.

Note: To avoid confusion it is recommended that each table entry name be unique. There is no checking mechanism in FIT to prevent table entries that have the same name and no error message is displayed in such cases.

3. User can enter into the values for the flash device. (Figure3-9, which shows the parameters of a new VSCC table.)

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

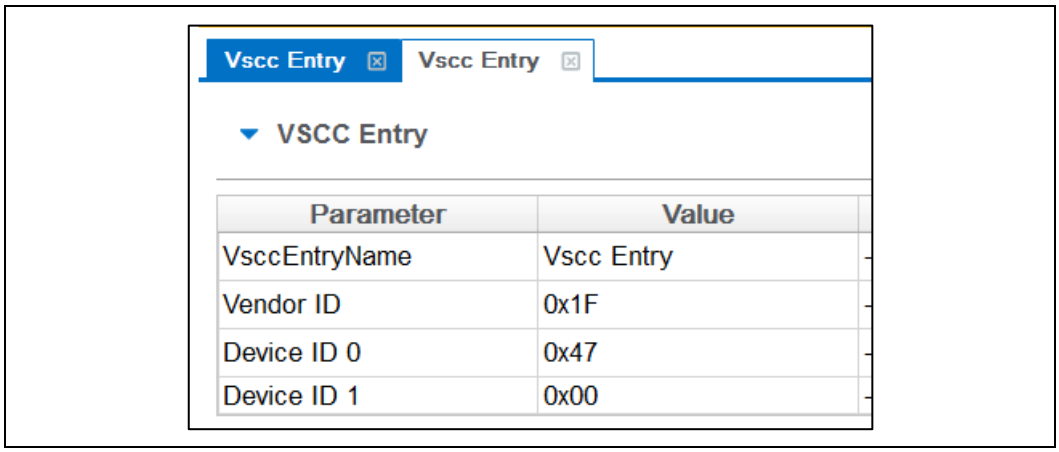
Note: If the descriptor region is being built manually the user will need to reference the VSCC table information for the parts being supported from the manufacturers' serial flash data sheet. The Ice Lake PCH-LP SPI Programming Guide should be used to calculate the VSSC values. For C620 family of workstation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

3.4.12 Removing Existing VSCC Table

To remove an existing table:

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

Figure 3-10. Deleting VSCC Table Entry Dialog



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

3.4.13 Modifying the Intel® Management Engine Region

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

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

3.4.14 Setting the Intel® Management Engine Region Binary File

To select the Intel® ME region binary file:

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

3.4.15 Setting the Intel® PMC Binary File

To select the Intel® PMC binary file:

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

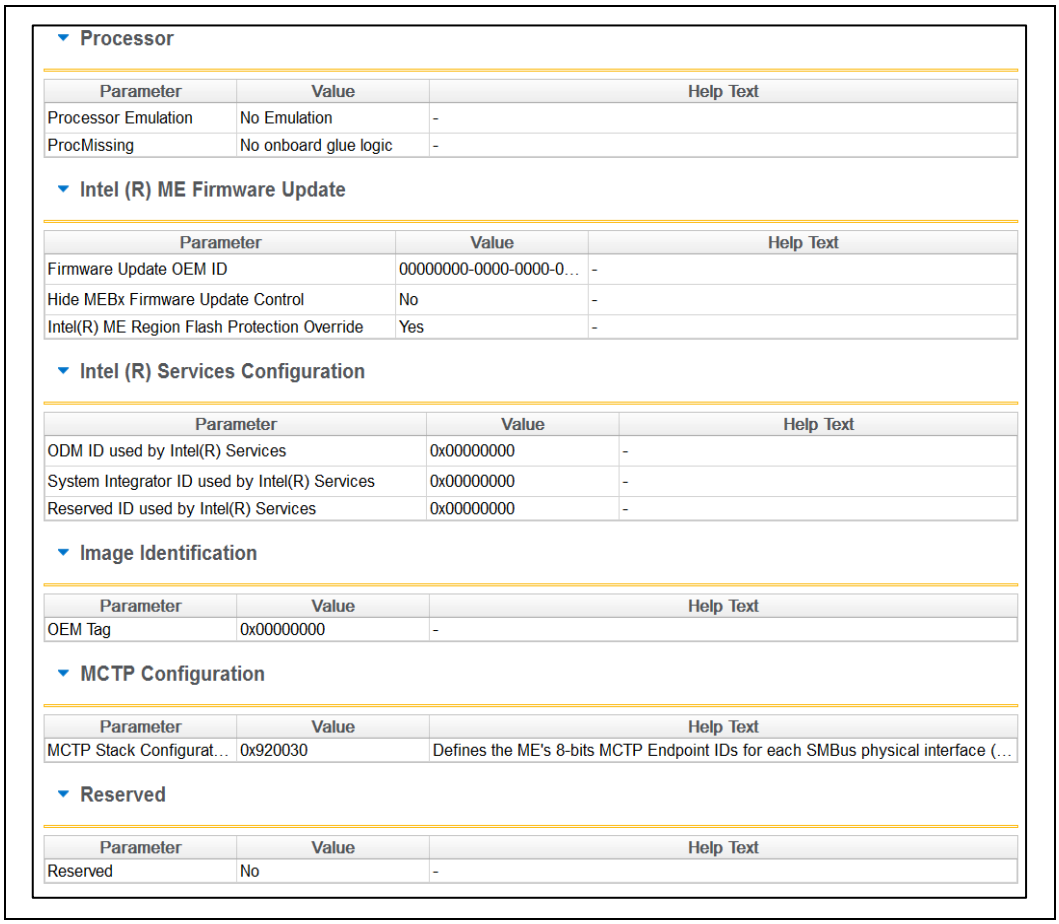
Note: Intel® FIT tool would return a build error in case wrong PMC binary is selected for stitching.

3.4.16 Intel® Management Engine Section

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

Click on Intel® ME Kernel Tab on the left pane to configure Intel® ME parameters. The parameter values can be found in the Help Text next to the parameter value as shown in Figure 3-11.

Figure 3-11. Intel® ME Kernel

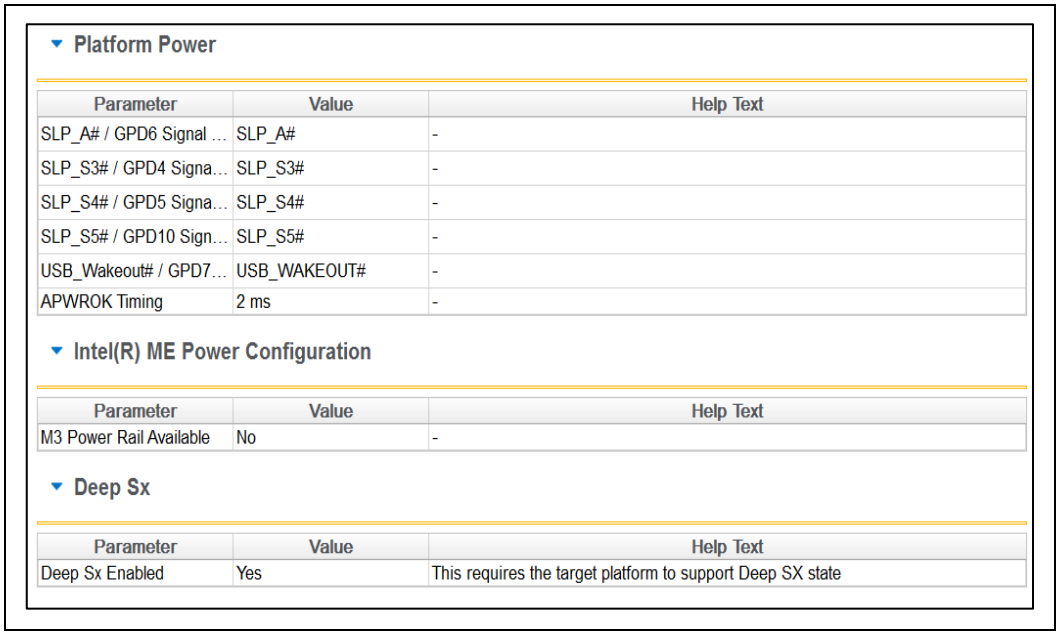


3.4.17 Power

This section describes the platform power configuration settings.

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

Figure 3-12. Power



▼ Platform Power

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

▼ Intel(R) ME Power Configuration

Parameter	Value	Help Text
M3 Power Rail Available	No	-

▼ Deep Sx

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

3.4.18 Power Delivery PD Controller Configuration

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

Figure 3-13. Power Delivery PD Controller Configuration

▼ Power Delivery PD Controller Configuration

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

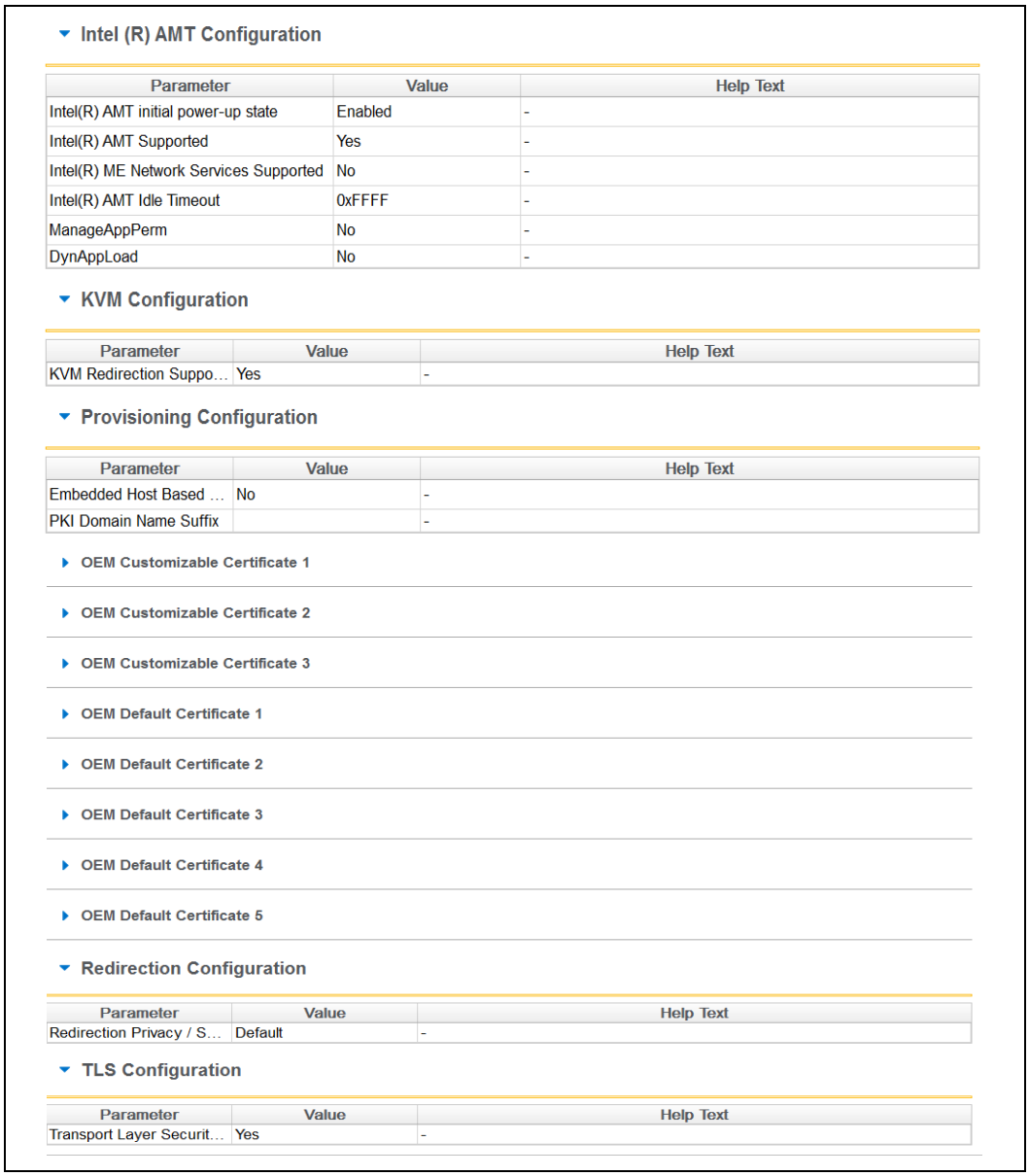
3.4.19 Manageability Application Section

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

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

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

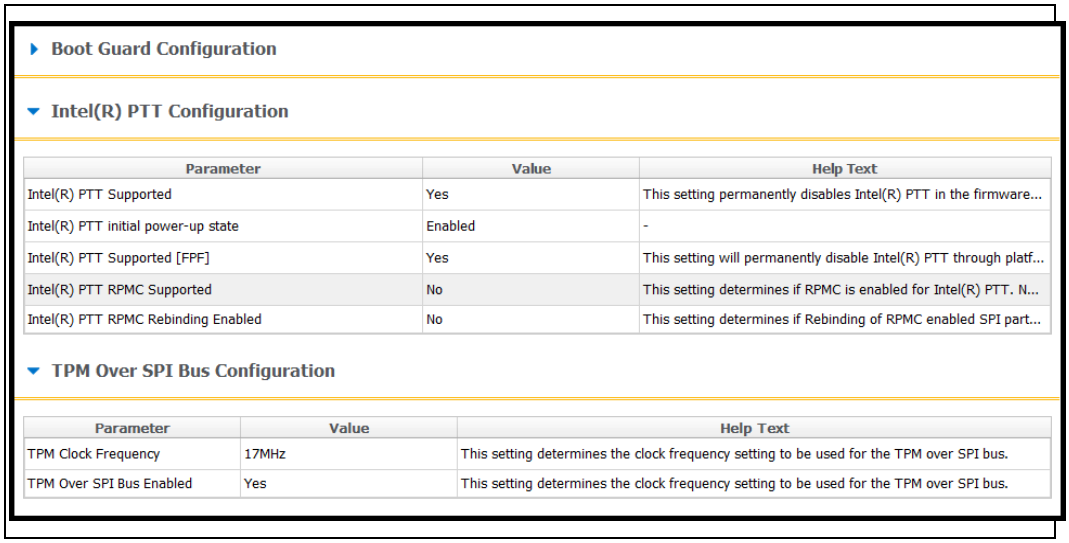
Figure 3-14. Manageability Application Section



3.4.20 Platform Protection

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

Figure 3-15. Platform Protection Section



These options control the availability and visibility of FW features.

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

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

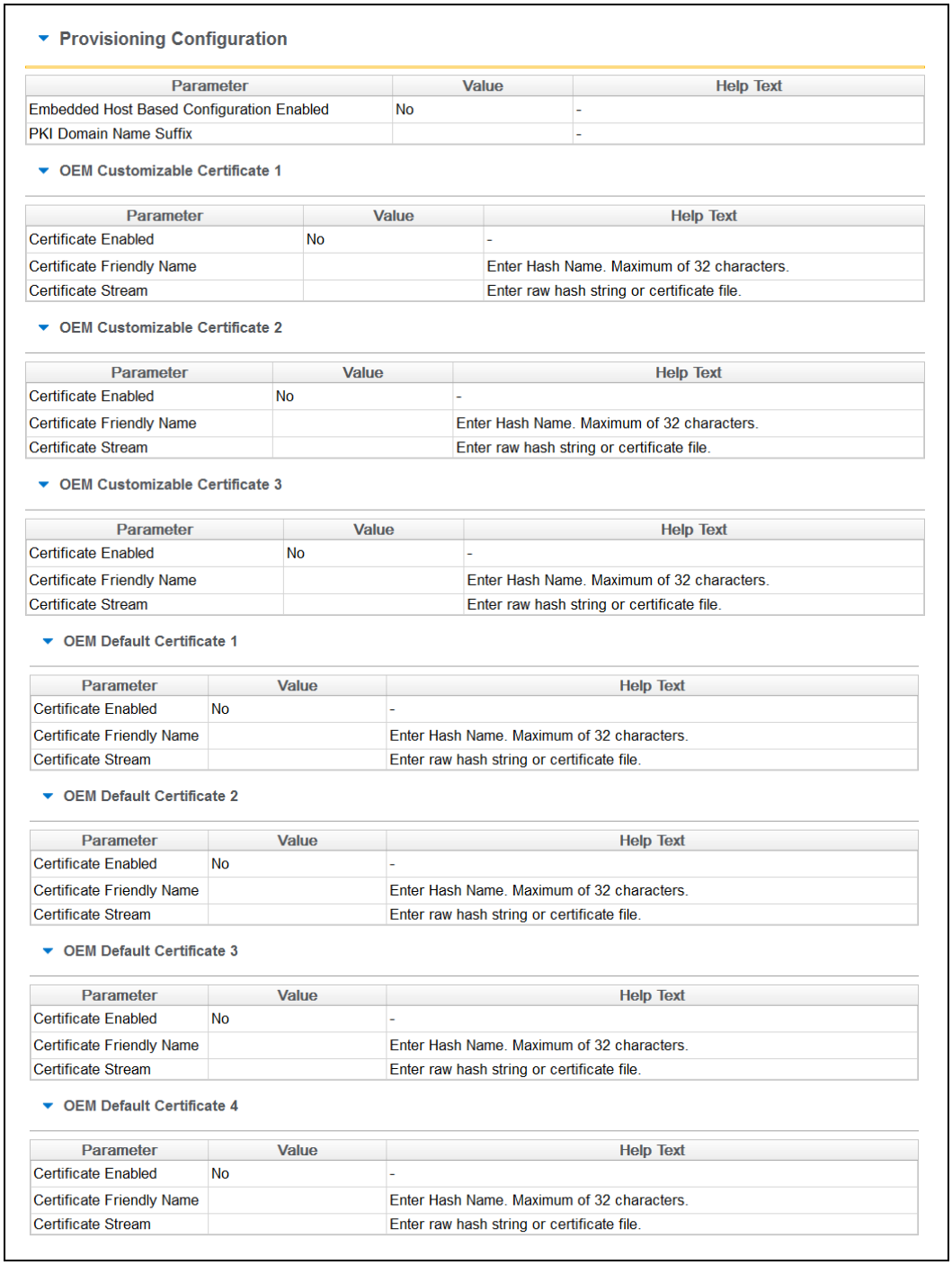
- Intel® Platform Trusted Technology
- Intel® Content Protection

3.4.21 Provisioning Section

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

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

Figure 3-15. Provisioning Configuration Section



Provisioning Configuration

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

OEM Customizable Certificate 1

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

OEM Customizable Certificate 2

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

OEM Customizable Certificate 3

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

OEM Default Certificate 1

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

OEM Default Certificate 2

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

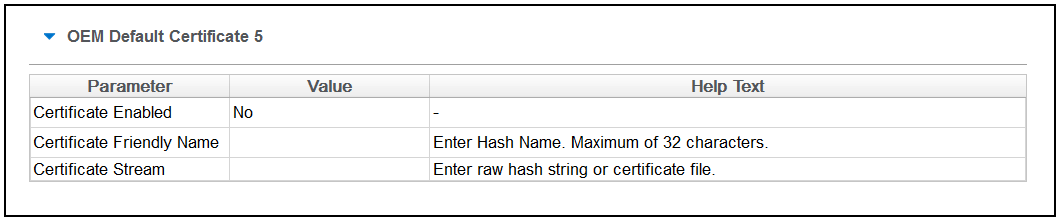
OEM Default Certificate 3

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

OEM Default Certificate 4

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

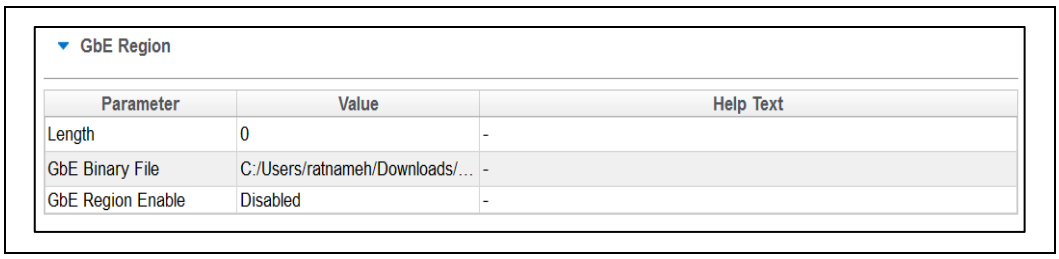
Figure 3-16. Provisioning Configuration Section (Cont..)



3.4.22 Gbe (LAN) Region Settings

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

Figure 3-17. GbE Region Options



3.4.23 Setting Gbe Region Length Option

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

3.4.24 Setting Gbe Region Binary File

To select the Gbe Region binary file:

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

3.4.25 Enabling/Disabling GbE Region

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

To disable the GbE Region:

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

To enable the GbE Region:

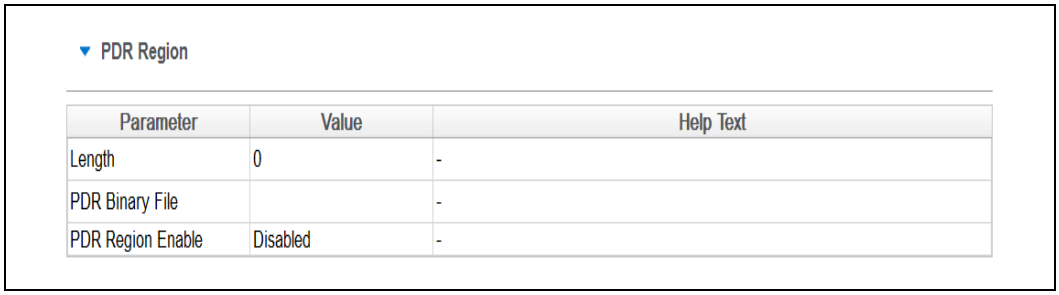
1. Click on Flash Layout tab on the left pane to load the binary file for Gbe region

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

3.4.26 Modifying PDR Region

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

Figure 3-18. PDR Region Options



3.4.27 Setting PDR Region Length Option

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

3.4.28 Setting PDR Region Binary File

To select the PDR region binary file:

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

3.4.29 Enabling/Disabling PDR Region

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

To disable the PDR Region:

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

Note: This region is disabled by default.

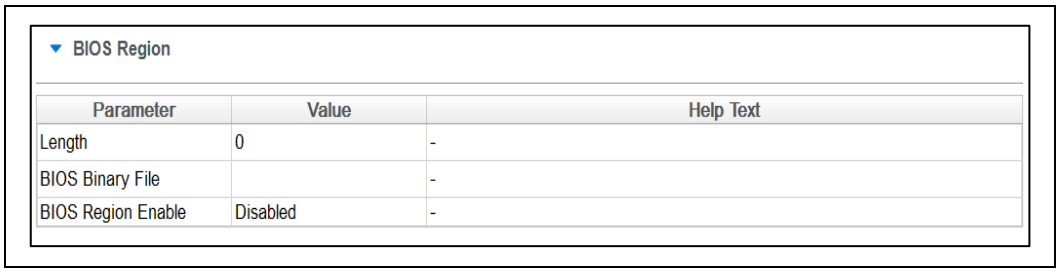
To enable the PDR Region:

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

3.4.30 Modifying BIOS Region

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

Figure 3-19. BIOS Region Parameters



3.4.31 Setting BIOS Region Length Parameter

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

3.4.32 Setting the BIOS Region Binary File

To select the BIOS region binary file:

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

3.4.33 Enabling/Disabling the BIOS Region

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

To disable the BIOS Region:

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

To enable the BIOS Region:

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

3.4.34 Building Flash Image

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

To build a flash image with the currently loaded configuration:

- Choose **Build > Build Image**.

  – OR –

- Specify an XML file with the /b option in the command line.

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

- Binary file representing the image
- Text file detailing the various regions in the image
- Optional set of intermediate files (refer Section Note:).
- Multiple binary files containing the image broken up according to the flash component sizes.

Note: These files are only created if two flash components are specified.)

The individual binary files can be used to manually program independent flash devices using a flash programmer. However, the user should select the single larger binary file when using FPT.

3.4.35 Decomposing Existing Flash Image

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

To decompose an image:

8. Chose **File → Open.**
9. Change the file type filter to the appropriate file type.
10. Select the required file and click **Open**; the image is automatically decomposed, the GUI is updated to reflect the new configuration, and a folder is created with each of the regions in a separate binary file.

Note: It is also possible to decompose an image by simply dragging and dropping the file into the main window. When decomposing an image, there are some NVARs will not be able to be decomposed by FIT. FIT will use Intel default value instead. User might want to check the log file to find out which NVARs were not parsed.

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

3.4.36 Command Line Interface

FIT supports command line options.

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

The command line syntax for FIT is:

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

Table 3-5. FIT Command Line Options

Option	Description
-exp	Displays example usage of this tool.
-H or -?	Displays the command line options.
-version|ver	Displays version of the tool.
-b	Automatically builds the flash image. The GUI does not appear if this flag is specified. This option causes the program to run in auto-build mode. If there is an error, a valid message is displayed and the image is not built. If a BIN file is included in the command line, this option decomposes it.
-bfwu	Build FWUpdate image.
-ofwu <filename>	Overrides the FWUpdate output file path.
-O <filename>	Path and filename where the image is saved. This command overrides the output file path in the XML file.
-f <filename>	Specifies input file. XML, full image binary, or ME only binary.
-me <file>	Overrides the binary source file for the Intel® ME 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.
-ec <file>	Overrides the binary source file for the EC region.
-gbe <file>	Overrides the binary source file for the GbE Region with the specified binary file.
-iunit <file>	Overrides the binary source file for the iUnit region.
-rombypass <true|false>	Overrides the ROM bypass setting in the XML file.
-sku <value>	Sets the SKU type to use.
-pmcp <file>	Overrides the binary source file for the PMCP region.
-ish <file>	Overrides the binary source file for the ISH region.

Option	Description
-sd_token <file>	Overrides the binary source file for the Secure Debug Token.
-iom <file>	Overrides the binary source file for the IOM region.
-nphy <file>	Overrides the binary source file for the North PHY region.
-tbt <file>	Overrides the binary source file for the TBT region.
-oem_km <file>	Overrides the binary source file for the OEM KM. override from cli enabled only in FWUpdate build
-W <path>	Overrides the working directory environment variable $WorkingDir. It is recommended that the user set these environmental variables first. (Suggested values can be found in the OEM Bringup Guide.)
-S <path>	Overrides the source file directory environment variable $SourceDir. It is recommended that the user set these environmental variables before starting a project.
-D <path>	Overrides the destination directory environment variable $DestDir. It is recommended that the user set these environmental variables before starting a project.
-U1 <value>	Overrides the $UserVar1 environment variable with the value specified. Can be any value required.
-U2 <value>	Overrides the $UserVar2 environment variable with the value specified. Can be any value required.
-U3 <value>	Overrides the $UserVar3 environment variable with the value specified. Can be any value required.
-I <enable | disable>	Enables or disables intermediate file generation.
-FLASHCOUNT <0, 1 or 2>	Overrides the number of flash components in the Descriptor Region. If this value is zero, only the Intel® ME Region is built.
-FLASHSIZE1 <0, 1, 2, 3, 4 or 5>	Overrides the size of the first flash component with the size of the option selected as follows: 0 = 512KB 1 = 1MB 2 = 2MB 3 = 4MB 4 = 8MB 5 = 16MB.
-FLASHSIZE2 <0, 1, 2, 3, 4 or 5>	Overrides the size of the first flash component with the size of the option selected as follows: 0 = 512KB 1 = 1MB 2 = 2MB 3 = 4MB 4 = 8MB 5 = 16MB.
-Save <file>	Saves the XML file.

3.4.37 Example – Decomposing Image and Extracting Parameters

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

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

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

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

The xml file contains the current Intel® ME parameters.

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

3.4.38 More Examples of FIT CLI

Note: If using paths defined in the KIT, be sure to put "" around the path as the spaces cause issues.

Take an existing (dt_ori.bin) image and put in a new BIOS binary:

```
FIT.exe /b /bios "...\...\...\Image Components\BIOS\BIOS.ROM" <file.bin
or file.xml>
```

```
Take an existing image and put in a different Intel® ME region:
FIT.exe /b /me ".\...\...\Image
Components\Firmware\ME13.0_5M_PreProduction.BIN" <file.bin or file.xml>
```

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

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

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

§ §

4 Flash Programming Tool

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

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

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

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

4.1 System Requirements

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

One possible workflow for using FPT is:

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

4.2 Flash Image Details

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

4.3 Microsoft Windows® Required Files

The Microsoft Windows® version of the FPT executable is **fptw.exe**, (**FPTW64.exe** for 64-bit based Windows® OS). The following files must be in the same directory as **fptw.exe**:

- fptw.exe – the executable used to program the final image file into the flash.
- idrvdll.dll

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

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

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

4.4 EFI Required Files

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

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

4.5 Programming Flash Device

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

Note: After programing the flash device, whether fully or partially, it is recommended to perform a G3 power cycle to complete the flashing process.

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

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

4.6 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 SetFile. Attempting to execute Commit file when not necessary will result in firmware returning an error.

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

Note: Files output when using the Intel® FPT -CFGGEN command line option in EFI environments do not contain the Carriage Return code 0x0D (‘\r’) as part of 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 –u –n <nvar>	Overwrites a pending NVAR value update request with the file system's current.
fpt.exe –CVARS	Displays a list of the supported manufacturing configurable named variables (NVARs).
fpt.exe –cfggen	Creates a list of blank NVARs in a text file that lets the user update multiple line configurable NVARS. The variables have the following format in the text file: NVAR name = value which will be used by setfile.
fpt.exe –U –N <NVAR name> -v <value>	Accept for updating UEP values using setFileEx API
fpt.exe –U –IN <Text file>	Accepts cfggen file with values set and will use setFileEx 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:

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

Table 4-2. Command Line Options for fpt.efi, fpt.exe and fptw.exe

Option	Description
Help (-H, -?)	Displays the list of command line options supported by FPT tool.
-CLEAR	Overwrites a pending NVAR value update request with the file system's current.
-VER	Shows the version of the tools.
-EXP	Shows examples of how to use the tools.
-VERBOSE [<file>]	Displays the tool's debug information or stores it in a log file.
-Y	Bypasses Prompt. FPT does not prompt user for input. This confirmation will automatically be answered with "y".
-I	Info. Displays information about the image currently used in the flash.
-F <file> [NOVERIFY]	Flash. Programs a binary file into an SPI flash. The user needs to specify the binary file to be flashed. FPT reads the binary, and then programs the binary into the flash. After a successful flash, FPT verifies that the SPI flash matches the provided image. Without specify the length with –L option, FPT will use the total SPI size instead of an image size. The NOVERFY sub-option *must* follow the file name. This will allow flashing the SPI without verifying the programming was done correctly. The user will be prompted before proceeding unless '-y' is used.
-ERASE	Block Erase. Erases all the blocks in a flash. This option does not use the chip erase command but instead erases the SPI flash block by block. This option can be used with a specific region argument to erase that region. This option cannot be used with the –f, -b, -c, -d or –verify options.
-VERIFY <file>	Verify. Compares a binary to the SPI flash. The image file name has to be passed as a command line argument if this flag is specified.
-NOVERIFY	Suboption of –F, see –F for details.
-D <file>	Dump. Reads the SPI flash and dumps the flash contents to a file or to the screen using the STDOUT option. The flash device must be written in 4KB sections. The total size of the flash device must also be in increments of 4KB.

Option	Description
-DESC	Read/Write Descriptor region. Specifies that the Descriptor region is to be read, written, or verified. The start address is the beginning of the region.
-BIOS	Read/Write BIOS region. Specifies that the BIOS region is to be read, written, or verified. Start address is the beginning of the region.
-ME	Read/Write Intel® ME region. Specifies that the Intel® ME region is to be read, written, or verified. The start address is the beginning of the region.
-EC	Read/Write EC region. Specifies that the EC region is to be read, written, or verified. The start address is the beginning of the region.
-GBE	Read/Write GbE region. Specifies that the GbE region is to be read, written, or verified. The start address is the beginning of the region.
-PDR	Read/Write PDR region. Specifies that the PDR region is to be read, written, or verified. The start address is the beginning of the region.
-SAVEMAC	This is used to save the GbE MAC Address. It is appropriate only when GbE Firmware is being over written. It also saves the GbE SSID and SVID.
-SAVESXID	Saves the GbE SSID and SVID when GbE is being reflashed.
-B	Blank Check. Checks whether the SPI flash is erased. If the SPI flash is not empty, the application halts as soon as contents are detected. The tool reports the address at which data was found.
-E	Skip Erase. Does not erase blocks before writing. This option skips the erase operation before writing and should be used if the part being flashed is a blank SPI flash device.
-A<value>, -ADDRESS <value>	Write/Read Address. Specifies the start address at which a read, verify, or write operation must be performed. The user needs to provide an address. This option is not used when providing a region since the region dictates the start address.
-L <value>, -LENGTH <value>	Write/Read Length. Specifies the length of data to be read, written, or verified. The user needs to provide the length. This option is not used when providing a region since the region/file length determines this.
-CVARS	Lists all the current manufacturing line configurable variables.
-MASTERACCESSGEN	Generates a Manufacturing Line Configurable Master Access Input File.
-CFGGEN	NVAR Input file generation option. This creates a file which can be used to update the line configurable NVARS.
-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., fpt –u –in FPT.cfg). The tool updates the NVARs contained in the text file with the values provided in the input file. User can also use FPT –cfggen to generate this file.
-N <value>	Name. Specifies the name of the NVAR that the user wants to update in the image file or flash. The name flag must be used with Value (-v).
-V <value>	Value. Specifies the value for the NVAR variable. The name of variable is specified in the Name flag. The Value flag must follow the Name flag.
-CLOSEMNF [NO] [PDR] [EC] [BIOS] <file>	End of Manufacturing. This option is executed at the end of manufacturing phase. This option does the following: Sets the Intel® ME manufacturing mode done bit (Global Locked bit). Verifies that the Intel® ME manufacturing mode done bit (Global Locked) is set. Sets the master region access permission in the Descriptor region to its Intel-recommended value Verifies that flash regions are locked. If the image was properly set before running this option, FPT skips all of the above and reports PASS. If anything was changed, FPT automatically forces a global reset through the CF9GR mechanism. The user can use the no reset option to bypass the reset. If nothing was changed, based on the current setting, the tool reports PASS without any reset. The "NO" addition will prevent the system from doing a global reset following a successful update of the ME Manufacturing Mode Done, the Region Access permissions, or both. The PDR, BIOS, EC, or GBE addition will allow CPU\BIOS Read and Write access to the PDR region of flash. It is now supported to run -closemnf in master_access.xml Note: Running FPT-closemnf also sets the default value for any unprovisioning process. Run FPT –closemnf first if the user wants to test any unprovisioning related process. In order to allow FPT to perform a global reset, BIOS should not lock CF9GR when Intel® ME is in manufacturing mode. This step is highly recommended to the manufacturing process. Without doing proper end of manufacturing process would lead to ship platform with potential security/privacy risk. Important: Before using this option with Production MCP / FW verify that the values for the PTT and Anchor Cove are correct in your image. Once this setting is used it will permanently commit values into the Field Programmable Fuses and cannot be undone.
-GRESET	Global Reset. FPT performs a global reset.
-PAGE	Pauses the screen when a page of text has been reached. Hit any key to continue.

Option	Description
-SPIBAR	Display SPI BAR. FPT uses this option to display the SPI Base Address Register.
-R <name>	NVAR and FPFs Read. FPT uses this option to retrieve NVAR value for a specific NVAR file name. The value of the variable is displayed. By default, all non- secure variables are displayed in clear-text and secure NVAR will be displayed in HASH. The -hashed option can be used to display the hash of a value instead of the clear-text value.
-VARS	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	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.
-COMMITFPF <name>	Commits NVAR values to FPF via firmware and prevents further modification of FPFs.
-PROVHDCP <file><file>	Provision platform with the key and cert provided.
-READHDCP	Displays the HDCP Rx provisioning status.
-GETPID <file>	Retrieve the part id.
-REWRITE	Allows to rewrite the SPI with file data even if flash is identical.
-WRITETOKEN <file>	Write the token where the file name is the token name.
-ERASETOKEN	Delete the token.
-COMMITARBSVN	Commits ARB SVN to FPFs. This would commit the Anti Rollback SVN to the FPFs.

Note: After programing the flash device, whether fully or partially, it is recommended to perform a G3 power cycle to complete the flashing process.

Table 4-3. FPT–closemnf Behavior

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

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

Table 4-4. Intel-Recommend Access Settings

	ME	GBE	BIOS	EC
Read	0b 0000 0000 1101 = 0x00d	0b 0000 0000 1000 = 0x009	0b 0000 000† 000‡ 1011 = 0x0†‡F	0b 0000 0001 0000 00*1 = 0x0101 or 0x0103
Write	0b 0000 0000 1100 = 0x004	0b 0000 0000 1000 = 0x008	0b 000† 000‡ 1010 = 0x†‡A	0b 0000 0001 0000 0x100

Note:
1. ‡ = Value dependent on if PDR is implemented and if Host access is desired.
2. † = Optional BIOS access to the EC region.
3. * = Optional EC Read access to BIOS.

Notes:

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

2. Case **B** depends on platform design if optional BIOS access to the EC region, add EC parameter after -closemnf; BIOS Read = 0x10F, BIOS Write = 0x10A.

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

4.8 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-IO CONFIGURATION → SECURITY CONFIGURATION → BIOS LOCK : Disabled
3. BIOS MENU -> Intel Advanced Menu → PCH-IO CONFIGURATION -> Flash Protection Range: 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: xx.x.x.xxxx
Copyright (C) 2005 - 2019, Intel Corporation. All rights reserved.

Reading HSFSTS register... Flash Descriptor: Valid

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


Processing Flash memory block 950 from 2559.
- Erasing Flash Block [0x9B7000] - 100 percent complete.
- Programming Flash [0x09B7000] 2332KB of 2332KB - 100 percent
complete.
Processing Flash memory block 1550 from 2559.
- Erasing Flash Block [0xC0F000] - 100 percent complete.
- Programming Flash [0x0C0F000] 1916KB of 1916KB - 100 percent
complete.
Processing Flash memory block 1591 from 2559.
- Erasing Flash Block [0xC38000] - 100 percent complete.
- Programming Flash [0x0C38000]   160KB of   160KB - 100 percent
complete.
Processing Flash memory block 1748 from 2559.
- Erasing Flash Block [0xCD5000] - 100 percent complete.
- Programming Flash [0x0CD5000]   532KB of   532KB - 100 percent
complete.
Processing Flash memory block 1805 from 2559.
- Erasing Flash Block [0xD0E000] - 100 percent complete.
```

```
- Programming Flash [0x0D0E000]   188KB of   188KB - 100 percent
complete.
Processing Flash memory block 1816 from 2559.
- Erasing Flash Block [0xD19000] - 100 percent complete.
- Programming Flash [0x0D19000]    36KB of    36KB - 100 percent
complete.
Processing Flash memory block 1908 from 2559.
- Erasing Flash Block [0xD75000] - 100 percent complete.
- Programming Flash [0x0D75000]   344KB of   344KB - 100 percent
complete.
Processing Flash memory block 2042 from 2559.
- Erasing Flash Block [0xDFB000] - 100 percent complete.
- Programming Flash [0x0DFB000]   364KB of   364KB - 100 percent
complete.
Processing Flash memory block 2324 from 2559.
- Erasing Flash Block [0xF15000] - 100 percent complete.
- Programming Flash [0x0F15000]   596KB of   596KB - 100 percent
complete.
Processing Flash memory block 2540 from 2559.
- Erasing Flash Block [0xFED000] - 100 percent complete.
- Programming Flash [0x0FED000]    52KB of    52KB - 100 percent
complete.
Processing Flash memory block 2559 from 2559.
- Erasing Flash Block [0x1000000] - 100 percent complete.
- Programming Flash [0x1000000]    20KB of    20KB - 100 percent
complete.

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

FPT Operation Successful.
```

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

4.8.3 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: xx.x.x.xxxx

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

Reading HSFSTS register... Flash Descriptor: Valid

    --- Flash Devices Found ---

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


Warning: Not all of the file data will be written to flash because
          the file is longer than the flash area to be written to!
```

```
            File: "image.bin"
            File Length: 16777216
            Write Length: 2048.

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


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

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

Flash device was programmed. It is recommended to perform

G3 power cycle to complete the flashing process.

FPT Operation Successful.
```

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

4.8.4 Dump Full Image

```
fpt.exe –d imagedump.bin

EFI:
fpt.efi –d imagedump.bin

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

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

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

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

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

FPT Operation Successful
```

4.8.5 Dump Specific Region

```
fpt.exe –d descdump.bin -desc
```

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

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

Reading HSFSTS register... Flash Descriptor: Valid

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

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

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

FPT Operation Successful
```

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

4.8.6 Display SPI Information

```
fpt.exe -I
-----------------------------------------
Intel (R) Flash Programming Tool. Version:  XX.X.X.XXXX
Copyright (c) 2005 - 2019, Intel Corporation. All rights reserved.

Reading HSFSTS register... Flash Descriptor: Valid

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



        --- Flash Image Information --
        Signature: VALID
        Number of Flash Components: 1
                 Component 1 - 32768KB (262144Kb)
        Regions:
                 DESC      - Base: 0x00000000, Limit: 0x00000FFF
                 BIOS      - Base: 0x00600000, Limit: 0x00FFFFFF
                 CSME      - Base: 0x00083000, Limit: 0x005FFFFF
                 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: 16384KB, Total Installed SPI Memory: 32768KB

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 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.

4.8.7 Verify Image with Errors

```
fpt.exe -verify outimage.bin

EFI:
fpt.efi -verify outimage.bin

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

Reading HSFSTS register... Flash Descriptor: Valid

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


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

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

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

4.8.8 Verify Image Successfully

```
fpt.exe -verify outimage.bin

EFI:
fpt.efi -verify outimage.bin

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

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

4.8.9 Get Intel® ME settings

```
fpt.exe -r ''Privacy/SecurityLevel''
fpt.efi -r '''Privacy/SecurityLevel''~''
-------------------------------------------
Intel (R) Flash Programming Tool. Version:  x.x.x.xxxx
Copyright (c) 2007-2014, Intel Corporation. All rights reserved.
Platform: Intel(R) Qxx Express Chipset
Reading HSFSTS register... Flash Descriptor: Valid
--- Flash Devices Found ---
    W25Q64BV    ID: 0xEF4017    Size: 8192KB (65536Kb)
Variable: "Privacy/SecurityLevel"
Value: True / 01
Retrieve Operation: Successful
```

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

4.8.10 CVAR Configuration File Generation (-cfggen)

It creates an input file which can be used to update CVARs and FPFs. The file includes all the current CVAR or FPFs. 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.

Delayed Authentication Mode Config = 0x00

FWUpdLcl = 0x01

GPIO =
0x0104010006000101000000000104020007000101000000000104 0A000800010100000
00001040B000900010100000000001040C000A000101000000000 1040D000B0001010000
000001070E0000000100000000000 1070D000100010000000000

Integrated Sensor Hub Supported = 0x01

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 = 0x01

;   LSPCON Port Config NVAR value is not displayed because it is stored
encrypted.
LSPCON Port Config =

OEM Tag = 0x00000000

PAVP Supported = 0x01

TLS Supported = 0x01

Unconfigure On RTC = 0x00

;   eDP Port Config NVAR value is not displayed because it is stored
encrypted.
eDP Port Config =

Error Enforcement Policy 0 = 0x00

Error Enforcement Policy 1 = 0x00

Intel(R) PTT = 0x01

OEM ID = 0x0000

OEM Platform ID = 0x0000

;   OEM Public Key Hash NVAR value is not displayed because it is
stored encrypted.
OEM Public Key Hash =
```

```
CPU Debugging = 0x00

BSP Initialization = 0x00

Protect BIOS Environment = 0x01

Measured Boot = 0x01

Verified Boot = 0x01

Key Manifest ID = 0x01

Force Boot Guard ACM = 0x00

S3 Optimization = 0x00

Persistent PRTC Backup Power = 0x00

Txt Supported = 0x00
```

§ §

5 Intel® MEManuf and MEManufWin

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

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

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

5.1 Windows® PE Requirements

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

5.2 How to Use Intel® MEManuf

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

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

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

5.3 Usage

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

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

Table 5-1. Options for Tool

Option	Description
No option	There are differences depending on the firmware SKU type the system is running on: If BIST is disabled in the Intel® ME Boot: The first time running Intel® MEManuf, since there is no CM3 test result stored in SPI, the tool will request the FW to run a complete BIST which includes a power reset at the end of the test for the Hibernation for the Windows® version. This power reset is only host side power cycle that triggered by Intel® ME. When host resets, Intel® ME FW will transition from CM0 to CM3, and then attempt automatically transition back from CM3 to CM0 along bringing host back to S0. Once host is booted back into OS, user needs to run the tool again in order to run runtime BIST and retrieve the test result. If BIST is enabled in the Intel® ME Boot: If there is no CM3 test result, the tool will report error and request user to use –test to run a full BIST. If there is CM3 test result, the tool will execute the runtime BIST and report the result. If running on a Consumer SKU image, the tool will request the FW to run a complete BIST which does not involve any power transition at the end of the test. Test result will be reported back right after the test is done and cleared. If BIST test result is not displayed after BIST test is done, the tool needs to be run again (with or without any BIST related argument combinations) to retrieve the result, once test result is displayed, it will be cleared. Tool is capable of remembering whether/what tests (including host based tests) have been run from previous invocation. Host based tests will be run for all cases (whether it's retrieving test result or run the actual BIST). Currently there are two host based tests; they are VSCC Table validation check and ICC data check.
-EXP	Shows examples of how to use the tools.
-H	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
-NOWLAN	**Note:** This option is not applicable for Consumer Intel® ME FW SKU. This option only applies to the AMT test so that the user can skip the wireless LAN NIC test if there is no wireless LAN NIC attached to the hardware. When –nowlan switch is not used, Intel® MEManuf also checks for the HW presence of Intel WLAN card based on a pre-defined list. If Intel® MEManuf detects an Intel WLAN card present on the platform, Intel® MEManuf runs the WLAN BIST test and reports pass/fail accordingly. If Intel® MEManuf cannot find any known WLAN card, Intel® MEManuf skips the WLAN BIST test and does not report errors. With the –verbose option, it displays "No Intel wireless LAN card detected" Note: -S0 can only be used on the platform which Intel® AMT is present and can be enabled in the field.
-WLAN	Force wireless LAN test
-BLOCKNET	**Note**: This option is not applicable for Consumer Intel® ME FW SKU. This option blocks any network traffic that goes in/out of the integrated GbE wired/wireless LAN interface. If Intel® AMT is disabled, "Error 9257: Cannot run the command since Intel® AMT is not available" is returned.
-ALLOWNET	**Note**: This option is not applicable for Consumer Intel® ME FW SKU. This option allows any network traffic that goes in/out of the integrated GbE wired/wireless LAN interface. If Intel® AMT is disabled, "Error 9257: Cannot run the command since Intel® AMT is not available" is returned.
-BISTRESULT	Returns last BIST results.

Option	Description
-EOL <Var|Config> -F <filename>	This option runs several checks for the use of OEMs to ensure that all settings and configurations have been made according to Intel requirements before the system leaves the manufacturing process. The check can be configured by the customer to select which test items to run and their expected value (only applicable for Variable Values, FW Version, BIOS Version, and Gbe Version). The sub option config or var is optional. Using -EOL without a sub option is equivalent to the –EOL config. ICC data check is performed for all options. The Full BIST test for ME13.0 is a combination of M0_HW, Live_HW and M0_Config. The Runtime BIST is a combination of M0_HW and M0_Config. Intel® MEManuf Sx test will require system is capable to enter sleep state, keep pinging the platform with network package and keep the system up will make the test failed. Host based Tests ME/BIOS VSCC validation, Intel® MEManuf verifies that flash SPI ID on the system is described in VSCC table. If found, VSCC entry for relevant SPI part should match the known good values that pre-populated in the file. Intel® ME state check, Intel® MEManuf verifies Intel® ME is in normal state. This is done by checking the value of 4 fields (initialization state, mode of operation, current operation state, and error state) in FW status register1. If any of these fields indicates Intel® ME is in abnormal state, Intel® MEManuf will report error without running BIST test. ICC data check, Intel® MEManuf verifies that valid OEM ICC data is present and programmed accordingly. This is done by checking FW status register2 ICC bits (which are bit 1 and 2 equal to 3). Intel® MEManuf –EOL Check.) When –f flag is used along with a file name (<filename>), the tool will load the file as the configuration file, instead of using MEManuf.xml.
-NEXTREBOOT	Upon successful platform reboot CM3 Autotest will be performed. **Note:** This is a standalone command and will only work if CM3 Autotest has been enabled in the firmware image. CM3 Autotest will be executed on the next CMoff – CM0 transition (example: Cold Reset), Global Reset or G3. The option itself will not trigger any platform reboots.
-CFGGEN <filename>	Use this option along with a filename to generate a default configuration file. This file (with or without modification) can be used for the -EOL option. Rename it MEManuf.xml before using it. It is highly recommended to use this option to generate a new MEManuf.xml with an up-to-date variable names list before using the Intel® MEManuf End-Of-Line check feature.
-ALL	Use this option to generate all possible tests for configuration file. All BIST, EOLConfig, and EOLVAR types of tests will be included in the generated XML. **Note:** Intel recommended tests will be enabled regardless of *-all* parameter to meet corresponding dependencies
-VERBOSE <file>	Displays the debug information of the tool or stores it in a log file.

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

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

Table 5-2. Intel® MEManuf Test Matrix

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

Note: ICC data check is performed for all options.

Note: The Full BIST test for ME13.0 is a combination of M0_HW, Live_HW and M0_Config. The Runtime BIST is a combination of M0_HW and M0_Config.

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

5.3.1 Host based Tests

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

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

5.4 Intel® MEManuf –EOL Check

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

5.4.1 ErrorAction Field

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

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

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

5.4.2 MEManuf.xml File

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

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 -->
      <!-- Dependencies></Dependencies -->
      <!-- TestType>LIVE_HW</TestType -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
   </csmebist>
   <csmebist name="Policy Kernel - ME Password : Validate MEBx
password">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Verify password is
acceptable.</Description -->
      <!-- IntelRequired>True</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- TestType>M0_CONFIG</TestType -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
   </csmebist>
   <csmebist name="Policy Kernel - Boot Guard : Self Test">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Get test result from NVAR
SECURE_BOOT_SELF_TEST_RESULT.</Description -->
      <!-- IntelRequired>True</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- TestType>M0_HW</TestType -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
   </csmebist>
   <csmebist name="Policy Kernel - ME Configuration :
PROC_MISSING">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
```

```
      <!-- Description>Only on mobile. Test fails if rule is
not set to MEFWCAPS_NO_ONBOARD_GLUE_LOGIC.</Description -->
      <!-- IntelRequired>True</IntelRequired -->
      <!-- Dependencies>MOBILE</Dependencies -->
      <!-- TestType>M0_CONFIG</TestType -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
   </csmebist>
   <csmebist name="VDM - General : VDM engine">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test VDM.</Description -->
      <!-- IntelRequired>True</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- TestType>M0_HW</TestType -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
   </csmebist>
   <csmebist name="GFX - General : Sampling engine">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test KVM sampling engine..</Description
-->
      <!-- IntelRequired>True</IntelRequired -->
      <!-- Dependencies>IPV6_LAN_ADDR</Dependencies -->
      <!-- TestType>M0_HW</TestType -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
   </csmebist>
   <csmebist name="USBr - General : Storage">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test USBr Storage.</Description -->
      <!-- IntelRequired>True</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- TestType>M0_HW</TestType -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
   </csmebist>
   <csmebist name="USBr - General : KVM">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test USBr KVM.</Description -->
      <!-- IntelRequired>True</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- TestType>M0_HW</TestType -->
      <!-- End of uneditable fields -->
```

```
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
   </csmebist>
   <csmebist name="Common Services - LAN : Connectivity to NIC
in M3">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>LAN test runs only if AMT is not
permanently disabled and mDNSProxy is not
disabled.</Description -->
      <!-- IntelRequired>True</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- TestType>LIVE_HW</TestType -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
   </csmebist>
   <csmebist name="Common Services - LAN : Connectivity to NIC
in M0">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>LAN test runs only if AMT is not
permanently disabled and mDNSProxy is not
disabled.</Description -->
      <!-- IntelRequired>True</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- TestType>M0_HW</TestType -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
   </csmebist>
   <csmebist name="Common Services - EHBC State : EHBC and
Privacy Level states compatibility">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Check while both EHBC and privacy level
are available, (PrivLevel != Default) && (EHBCState ==
EHBC_STATE_ENABLE).</Description -->
      <!-- IntelRequired>True</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- TestType>M0_CONFIG</TestType -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
   </csmebist>
   <csmebist name="Common Services - EHBC State : Valid
Embedded Host Based Configuration (EHBC) state">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Check if EHBC state is
available.</Description -->
      <!-- IntelRequired>True</IntelRequired -->
```

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

```
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
   </csmebist>
   <csmebist name="AMT - Power : Valid LAN power well">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Run the tests verifying the internal
variables.</Description -->
      <!-- IntelRequired>True</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- TestType>M0_CONFIG</TestType -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
   </csmebist>
   <csmebist name="PAVP - General : Verify Edp and Lspcon
Configurations">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Check if LSPCON and 5K ports are
overlapped</Description -->
      <!-- IntelRequired>True</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- TestType>M0_HW</TestType -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
   </csmebist>
   <csmebist name="PAVP - General : Set Lspcon Port">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test the validility of the 5K port
configuration</Description -->
      <!-- IntelRequired>True</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- TestType>M0_HW</TestType -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
   </csmebist>
   <csmebist name="PAVP - General : Set Edp Port">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test the validility of the LSPCON port
configuration</Description -->
      <!-- IntelRequired>True</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- TestType>M0_HW</TestType -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
```

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

```
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Enabled/00/Disabled/01"
example="Enabled"> </RequiredValue>
   </eolconfig>
   <eolconfig name="Txt Supported">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Check fpf Txt Supported against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>1</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Disabled/00/Enabled/01"
example="Disabled"> </RequiredValue>
   </eolconfig>
   <eolconfig name="Secure boot KM SVN">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Check fpf Secure boot KM SVN against
expected value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>1</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Hex number with 0x prefix."
example="0x00"> </RequiredValue>
   </eolconfig>
   <eolconfig name="Secure boot KM Anti Rollback">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Check fpf Secure boot KM Anti Rollback
against expected value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>1</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Disabled/00/Enabled/01"
example="Disabled"> </RequiredValue>
   </eolconfig>
   <eolconfig name="Secure boot BSMM SVN">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
```

```
      <!-- Description>Check fpf Secure boot BSMM SVN against
expected value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>1</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Hex number with 0x prefix."
example="0x00"> </RequiredValue>
   </eolconfig>
   <eolconfig name="Secure boot ACM SVN">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Check fpf Secure boot ACM SVN against
expected value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>1</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Hex number with 0x prefix."
example="0x00"> </RequiredValue>
   </eolconfig>
   <eolconfig name="SPI Boot Source">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Check fpf SPI Boot Source against
expected value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>1</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Enabled/00/Disabled/01"
example="Enabled"> </RequiredValue>
   </eolconfig>
   <eolconfig name="SOC Config Lock State">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Check fpf SOC Config Lock State against
expected value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>1</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Disabled/00/Enabled/01"
example="Disabled"> </RequiredValue>
```

```
   </eolconfig>
   <eolconfig name="RPMB Migration Done">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Check fpf RPMB Migration Done against
expected value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>1</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Disabled/00/Enabled/01"
example="Disabled"> </RequiredValue>
   </eolconfig>
   <eolconfig name="ROT KM SVN">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Check fpf ROT KM SVN against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>1</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Hex number with 0x prefix."
example="0x00"> </RequiredValue>
   </eolconfig>
   <eolconfig name="ROT Anti Rollback">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Check fpf ROT Anti Rollback against
expected value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>1</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Disabled/00/Enabled/01"
example="Disabled"> </RequiredValue>
   </eolconfig>
   <eolconfig name="RBE Anti Rollback">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Check fpf RBE Anti Rollback against
expected value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>1</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
```

```
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Disabled/00/Enabled/01"
example="Disabled"> </RequiredValue>
   </eolconfig>
   <eolconfig name="Persistent PRTC Backup Power">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Check fpf Persistent PRTC Backup Power
against expected value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>1</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Enabled/00/Disabled/01"
example="Enabled"> </RequiredValue>
   </eolconfig>
   <eolconfig name="PMC SVN">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Check fpf PMC SVN against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>1</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Hex number with 0x prefix."
example="0x00"> </RequiredValue>
   </eolconfig>
   <eolconfig name="PMC Anti Rollback">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Check fpf PMC Anti Rollback against
expected value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>1</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Disabled/00/Enabled/01"
example="Disabled"> </RequiredValue>
   </eolconfig>
   <eolconfig name="PCH_COSIG">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Check fpf PCH_COSIG 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="OS SVN">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Check fpf OS 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="OS Anti Rollback">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Check fpf OS 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="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 Revocable Hash">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Check fpf OEM Revocable 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="Disabled/00/Enabled/01"
example="Disabled"> </RequiredValue>
   </eolconfig>
   <eolconfig name="OEM RSA Key 1 Size">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Check fpf OEM RSA Key 1 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="OEM Public Key Hash 1">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Check fpf OEM Public Key Hash 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="32 hex pairs with space between
pairs" example="04 AB F3 45 03 1D EF A2 B7 E8 98 79 10 45 AB DE
F2 35 49 A0 01 35 78 29 37 AB DE EF FA 10 EF 33">
</RequiredValue>
   </eolconfig>
   <eolconfig name="OEM Public Key Hash 0">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Check fpf OEM Public Key Hash 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="32 hex pairs with space between
pairs" example="04 AB F3 45 03 1D EF A2 B7 E8 98 79 10 45 AB DE
F2 35 49 A0 01 35 78 29 37 AB DE EF FA 10 EF 33">
</RequiredValue>
   </eolconfig>
   <eolconfig name="OEM Platform ID">
      <!-- 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 Hash 1 Valid">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Check fpf OEM Key Hash 1 Valid against
expected value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
```

```
      <!-- Level>1</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Disabled/00/Enabled/01"
example="Disabled"> </RequiredValue>
   </eolconfig>
   <eolconfig name="OEM KM SVN">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Check fpf OEM KM SVN against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>1</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Hex number with 0x prefix."
example="0x00"> </RequiredValue>
   </eolconfig>
   <eolconfig name="OEM KM Present">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Check fpf OEM KM Present against
expected value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- 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="NWLD SVN">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Check fpf NWLD 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="NWLD Anti Rollback">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Check fpf NWLD 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="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 Hardware Status
FPF">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Check fpf Intel(R) Manageability
Hardware Status FPF against expected value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>1</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Enabled/00/Disabled/01"
example="Enabled"> </RequiredValue>
   </eolconfig>
   <eolconfig name="IDLM SVN">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Check fpf IDLM SVN against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>1</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Hex number with 0x prefix."
example="0x00"> </RequiredValue>
   </eolconfig>
   <eolconfig name="IDLM Anti Rollback">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Check fpf IDLM Anti Rollback against
expected value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>1</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Disabled/00/Enabled/01"
example="Disabled"> </RequiredValue>
   </eolconfig>
   <eolconfig name="Glitch Detection Enabled">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Check fpf Glitch Detection Enabled
against expected value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>1</Level -->
      <!-- End of uneditable fields -->
```

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

```
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>1</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Disabled/00/Enabled/01"
example="Disabled"> </RequiredValue>
   </eolconfig>
   <eolconfig name="EMMC Boot Source">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Check fpf EMMC Boot Source against
expected value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>1</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Enabled/00/Disabled/01"
example="Enabled"> </RequiredValue>
   </eolconfig>
   <eolconfig name="DNX SVN">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Check fpf DNX SVN against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>1</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Hex number with 0x prefix."
example="0x00"> </RequiredValue>
   </eolconfig>
   <eolconfig name="DNX Anti Rollback">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Check fpf DNX Anti Rollback against
expected value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>1</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Disabled/00/Enabled/01"
example="Disabled"> </RequiredValue>
   </eolconfig>
   <eolconfig name="CPU KM SVN">
```

```
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Check fpf CPU 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="CPU FW Anti Rollback">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Check fpf CPU FW 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="CPU Co-signing">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Check fpf CPU Co-signing against
expected value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>1</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Disabled/00/Enabled/01"
example="Disabled"> </RequiredValue>
   </eolconfig>
   <eolconfig name="COMP EOM">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Check fpf COMP EOM 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="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="Attestation KeyBox test">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Check Attestation KeyBox data
validity</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>1</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
   </eolconfig>
   <eolconfig name="Manageability Hardware Support">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Check Manageability Hardware
Support</Description -->
      <!-- IntelRequired>True</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
   </eolconfig>
   <eolconfig name="Enforce RPMC Support">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Check if RPMC configuration is
enabled</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>SPI_DEP</Dependencies -->
      <!-- Level>1</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
```

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

```
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
   </eolconfig>
   <eolconfig name="Confirm ARB SVN value">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Confirms that the minimum ARB SVN saved
in the PCH fuses matches the ARB SVN of the FW
image</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>1</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
   </eolconfig>
   <eolconfig name="PCH Unlocked state">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Verifies that PCH is locked</Description
-->
      <!-- IntelRequired>True</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>1</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
   </eolconfig>
   <eolconfig name="HW Binding Disabled">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Verifies that HW binding is
disabled</Description -->
      <!-- IntelRequired>True</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>1</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
   </eolconfig>
   <eolconfig name="SOC Config Lock">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Check that SOC Config Lock FPF is
set.</Description -->
      <!-- IntelRequired>True</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>1</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
   </eolconfig>
```

```
   <eolconfig name="FPFs in UEP Committed">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Check that FPFs in UEP are committed to
Hardware.</Description -->
      <!-- IntelRequired>True</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>1</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
   </eolconfig>
   <eolconfig name="Validate Keybox Provisioning">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Check to see if Keybox is
provisioned</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
   </eolconfig>
   <eolconfig name="Firmware Update OEM ID">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Check Firmware Update OEM ID
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Hex" example="00000000-0000-0000-
0000-000000000000"> </RequiredValue>
   </eolconfig>
   <eolconfig name="GBE version">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Check Gbe Version against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>SPI_DEP</Dependencies -->
      <!-- Level>1</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="major_ver.minor_ver"
example="0.6"> </RequiredValue>
   </eolconfig>
```

```
   <eolconfig name="BIOS version">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Check BIOS Version against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>1</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Customer specific"
example="HSWLPTU1.86C.0117.R00.1303102001"> </RequiredValue>
   </eolconfig>
   <eolconfig name="ME FW version">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Check Firmware Version against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>1</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue
format="major_ver.minor_ver.hotfix_ver.build_num H|LP|ULT
Corporate|Consumer|Slim" example="12.0.0.1040 LP Consumer">
</RequiredValue>
   </eolconfig>
   <eolconfig name="System UUID">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Check System UUID against programmed
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>VPRO</Dependencies -->
      <!-- Level>1</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="See example" example="550e8400-
e29b-41d4-a716-446655440000"> </RequiredValue>
   </eolconfig>
   <eolconfig name="MAC address">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Check MAC address</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>VPRO|LAN|IPV4_LAN_HW</Dependencies -->
      <!-- Level>1</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
```

```
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="6 hex pairs separated by ':'"
example="00:01:12:A2:3B:45"> </RequiredValue>
   </eolconfig>
   <eolconfig name="Security Descriptor Override (SDO) check">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Check SDO pin</Description -->
      <!-- IntelRequired>True</IntelRequired -->
      <!-- Dependencies>SPI_DEP</Dependencies -->
      <!-- Level>1</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
   </eolconfig>
   <eolconfig name="RPMC Configuration">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Check RPMC configuration</Description --
>
      <!-- IntelRequired>True</IntelRequired -->
      <!-- Dependencies>SPI_DEP</Dependencies -->
      <!-- Level>1</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
   </eolconfig>
   <eolconfig name="EC Write Access Permissions">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Check EC write access</Description -->
      <!-- IntelRequired>True</IntelRequired -->
      <!-- Dependencies>SPI_DEP</Dependencies -->
      <!-- Level>2</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Hex number with 0x prefix."
example="0x0101. Value left empty will result in checking
against Intel recommended values."> </RequiredValue>
   </eolconfig>
   <eolconfig name="EC Read Access Permissions">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Check EC read access</Description -->
      <!-- IntelRequired>True</IntelRequired -->
      <!-- Dependencies>SPI_DEP</Dependencies -->
      <!-- Level>2</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
```

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

```
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Hex number with 0x prefix."
example="0x0101. Value left empty will result in checking
against Intel recommended values."> </RequiredValue>
   </eolconfig>
   <eolconfig name="ME Write Access Permissions">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Check ME write access</Description -->
      <!-- IntelRequired>True</IntelRequired -->
      <!-- Dependencies>SPI_DEP</Dependencies -->
      <!-- Level>2</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Hex number with 0x prefix."
example="0x0101. Value left empty will result in checking
against Intel recommended values."> </RequiredValue>
   </eolconfig>
   <eolconfig name="ME Read Access Permissions">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Check ME read access</Description -->
      <!-- IntelRequired>True</IntelRequired -->
      <!-- Dependencies>SPI_DEP</Dependencies -->
      <!-- Level>2</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Hex number with 0x prefix."
example="0x0101. Value left empty will result in checking
against Intel recommended values."> </RequiredValue>
   </eolconfig>
   <eolconfig name="ME Manufacturing Mode status">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Check End of Manufacturing Mode against
Intel recommended value</Description -->
      <!-- IntelRequired>True</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>1</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
   </eolconfig>
   <eolconfig name="EOP status check">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Check that EOP was
sent/recieved</Description -->
      <!-- IntelRequired>True</IntelRequired -->
```

```
      <!-- Dependencies></Dependencies -->
      <!-- Level>1</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Enabled</State>
      <ErrAction>ErrorContinue</ErrAction>
   </eolconfig>
   <!-- END OF EOL CONFIG TESTS -->
   <!-- EOL VAR TESTS -->
   <eolvar name="vPro TBT I2C Re-timer 4 address">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Hex number with 0x prefix."
example="0x00"> </RequiredValue>
   </eolvar>
   <eolvar name="vPro TBT I2C Re-timer 3 address">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Hex number with 0x prefix."
example="0x00"> </RequiredValue>
   </eolvar>
   <eolvar name="vPro TBT I2C Re-timer 2 address">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Hex number with 0x prefix."
example="0x00"> </RequiredValue>
   </eolvar>
   <eolvar name="vPro TBT I2C Re-timer 1 address">
```

```
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Hex number with 0x prefix."
example="0x00"> </RequiredValue>
   </eolvar>
   <eolvar name="eDP Port Config">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="32 hex pairs with space between
pairs" example="04 AB F3 45 03 1D EF A2 B7 E8 98 79 10 45 AB DE
F2 35 49 A0 01 35 78 29 37 AB DE EF FA 10 EF 33">
</RequiredValue>
   </eolvar>
   <eolvar name="WLAN Power Well">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue
format="Disabled/80/CoreWell/81/SusWell/82/MEWell/83/WLAN Sleep
via SLP_WLAN#/86" example="Disabled"> </RequiredValue>
   </eolvar>
   <eolvar name="Unconfigure On RTC">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
```

```
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Enabled/00/Disabled/01"
example="Enabled"> </RequiredValue>
   </eolvar>
   <eolvar name="Trusted Device Setup Supported">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Disabled/00/Enabled/01"
example="Disabled"> </RequiredValue>
   </eolvar>
   <eolvar name="TLS Supported">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="No/00/Yes/01" example="No">
</RequiredValue>
   </eolvar>
   <eolvar name="StorageState">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Disabled/00/Enabled/01"
example="Disabled"> </RequiredValue>
   </eolvar>
   <eolvar name="Signing Policy">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
```

```
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="PMF and Seal signing/00/PMF
Signing Only/01/No Signing/02" example="PMF and Seal signing">
</RequiredValue>
   </eolvar>
   <eolvar name="Seal State">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Disabled/00/Enabled And
Intact/01/Enabled And Broken/02" example="Disabled">
</RequiredValue>
   </eolvar>
   <eolvar name="SOL">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Disabled/00/Enabled/01"
example="Disabled"> </RequiredValue>
   </eolvar>
   <eolvar name="Reseal Timeout">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Hex number with 0x prefix."
example="0x00"> </RequiredValue>
```

```
   </eolvar>
   <eolvar name="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>AMT_MNG|CORP</Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Default/01/Enhanced/02/Extreme/03"
example="Default"> </RequiredValue>
   </eolvar>
   <eolvar name="Redirection">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Hex number with 0x prefix."
example="0x00000000"> </RequiredValue>
   </eolvar>
   <eolvar name="RCFG/ZTC">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Disabled/00/Enabled/01"
example="Disabled"> </RequiredValue>
   </eolvar>
   <eolvar name="Processor Emulation">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
```

```
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="No
Emulation/00/vPro/01/Core/02/Celerno/03/Pentium/04/Xeon/05/Xeon
Manageability Capable/06" example="No Emulation">
</RequiredValue>
   </eolvar>
   <eolvar name="PROC_MISSING">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>MOBILE</Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="No onboard glue logic/ff"
example="No onboard glue logic"> </RequiredValue>
   </eolvar>
   <eolvar name="PKI Domain Name Suffix">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="String" example="Any">
</RequiredValue>
   </eolvar>
   <eolvar name="PAVP Supported">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="No/00/Yes/01" example="No">
</RequiredValue>
   </eolvar>
   <eolvar name="Opt-in Policy">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
```

```
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Hex number with 0x prefix."
example="0x00"> </RequiredValue>
   </eolvar>
   <eolvar name="On dock vPro NIC SMBus address">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Hex number with 0x prefix."
example="0x00"> </RequiredValue>
   </eolvar>
   <eolvar name="On Board Discrete vPro NIC SMBus address">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Hex number with 0x prefix."
example="0x00"> </RequiredValue>
   </eolvar>
   <eolvar name="OEMSkuRule">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Hex number with 0x prefix."
example="0x00000000"> </RequiredValue>
   </eolvar>
   <eolvar name="OEM Tag">
```

```
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Hex number with 0x prefix."
example="0x00000000"> </RequiredValue>
   </eolvar>
   <eolvar name="OEM Default Cert Stream">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Hex number with 0x prefix."
example="0x00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000
0000000000000"> </RequiredValue>
   </eolvar>
   <eolvar name="OEM Default Cert Friendly Name">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="String" example="Any">
</RequiredValue>
   </eolvar>
   <eolvar name="OEM Default Cert Active">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
```

```
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue
format="False/00/NotActive/00/True/01/Active/01"
example="False"> </RequiredValue>
   </eolvar>
   <eolvar name="OEM Default Cert 5 Stream">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Hex number with 0x prefix."
example="0x00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000
0000000000000"> </RequiredValue>
   </eolvar>
   <eolvar name="OEM Default Cert 5 Friendly Name">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="String" example="Any">
</RequiredValue>
   </eolvar>
   <eolvar name="OEM Default Cert 5 Active">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue
format="False/00/NotActive/00/True/01/Active/01"
example="False"> </RequiredValue>
   </eolvar>
   <eolvar name="OEM Default Cert 4 Stream">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
```

```
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Hex number with 0x prefix."
example="0x00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000
0000000000000"> </RequiredValue>
   </eolvar>
   <eolvar name="OEM Default Cert 4 Friendly Name">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="String" example="Any">
</RequiredValue>
   </eolvar>
   <eolvar name="OEM Default Cert 4 Active">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue
format="False/00/NotActive/00/True/01/Active/01"
example="False"> </RequiredValue>
   </eolvar>
   <eolvar name="OEM Default Cert 3 Stream">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
```

```
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Hex number with 0x prefix."
example="0x0000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000
0000000000000"> </RequiredValue>
   </eolvar>
   <eolvar name="OEM Default Cert 3 Friendly Name">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="String" example="Any">
</RequiredValue>
   </eolvar>
   <eolvar name="OEM Default Cert 3 Active">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue
format="False/00/NotActive/00/True/01/Active/01"
example="False"> </RequiredValue>
   </eolvar>
   <eolvar name="OEM Default Cert 2 Stream">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Hex number with 0x prefix."
example="0x0000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000
0000000000000"> </RequiredValue>
   </eolvar>
   <eolvar name="OEM Default Cert 2 Friendly Name">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
```

```
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="String" example="Any">
</RequiredValue>
   </eolvar>
   <eolvar name="OEM Default Cert 2 Active">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue
format="False/00/NotActive/00/True/01/Active/01"
example="False"> </RequiredValue>
   </eolvar>
   <eolvar name="OEM Custom Cert 3 Stream">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Hex number with 0x prefix."
example="0x00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000
0000000000000"> </RequiredValue>
   </eolvar>
   <eolvar name="OEM Custom Cert 3 Friendly Name">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
```

```
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="String" example="Any">
</RequiredValue>
   </eolvar>
   <eolvar name="OEM Custom Cert 3 Active">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue
format="False/00/NotActive/00/True/01/Active/01"
example="False"> </RequiredValue>
   </eolvar>
   <eolvar name="OEM Custom Cert 2 Stream">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Hex number with 0x prefix."
example="0x00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000
0000000000000"> </RequiredValue>
   </eolvar>
   <eolvar name="OEM Custom Cert 2 Friendly Name">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="String" example="Any">
</RequiredValue>
   </eolvar>
   <eolvar name="OEM Custom Cert 2 Active">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
```

```
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue
format="False/00/NotActive/00/True/01/Active/01"
example="False"> </RequiredValue>
   </eolvar>
   <eolvar name="OEM Custom Cert 1 Stream">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Hex number with 0x prefix."
example="0x00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000
0000000000000"> </RequiredValue>
   </eolvar>
   <eolvar name="OEM Custom Cert 1 Friendly Name">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="String" example="Any">
</RequiredValue>
   </eolvar>
   <eolvar name="OEM Custom Cert 1 Active">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
```

```
      <RequiredValue
format="False/00/NotActive/00/True/01/Active/01"
example="False"> </RequiredValue>
   </eolvar>
   <eolvar name="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 -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Disabled/00/Enabled/01"
example="Disabled"> </RequiredValue>
   </eolvar>
   <eolvar name="Manageability App Supported">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="No/00/Yes/01" example="No">
</RequiredValue>
   </eolvar>
   <eolvar name="MEBxPassword">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="32 hex pairs with space between
pairs" example="04 AB F3 45 03 1D EF A2 B7 E8 98 79 10 45 AB DE
F2 35 49 A0 01 35 78 29 37 AB DE EF FA 10 EF 33">
</RequiredValue>
   </eolvar>
   <eolvar name="MCTP Device Ports">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
```

```
      <!-- Dependencies></Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Hex number with 0x prefix."
example="0x00000000"> </RequiredValue>
   </eolvar>
   <eolvar name="LSPCON Port Config">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="32 hex pairs with space between
pairs" example="04 AB F3 45 03 1D EF A2 B7 E8 98 79 10 45 AB DE
F2 35 49 A0 01 35 78 29 37 AB DE EF FA 10 EF 33">
</RequiredValue>
   </eolvar>
   <eolvar name="LAN Power Well">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue
format="CoreWell/00/SusWell/01/MEWell/02/SLP_LAN#(MGPIO3)/03"
example="CoreWell"> </RequiredValue>
   </eolvar>
   <eolvar name="KVM Redirection Supported">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="No/00/Yes/01" example="No">
</RequiredValue>
```

```
   </eolvar>
   <eolvar name="KVM">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Disabled/00/Enabled/01"
example="Disabled"> </RequiredValue>
   </eolvar>
   <eolvar name="Intel(R) Precise Touch Technology Supported">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="No/00/Yes/01" example="No">
</RequiredValue>
   </eolvar>
   <eolvar name="Intel(R) PTT initial power-up state">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Disabled/00/Enabled/01"
example="Disabled"> </RequiredValue>
   </eolvar>
   <eolvar name="Intel(R) PTT Supported">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
```

```
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Disabled/00/Enabled/01"
example="Disabled"> </RequiredValue>
   </eolvar>
   <eolvar name="Intel(R) Manageability Hardware Status NVAR">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Enabled/00/Disabled/01"
example="Enabled"> </RequiredValue>
   </eolvar>
   <eolvar name="Intel(R) ME Region Flash Protection Override">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="False/00/True/01" example="False">
</RequiredValue>
   </eolvar>
   <eolvar name="Intel(R) ME Network Services Supported">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Yes/00/No/01" example="Yes">
</RequiredValue>
   </eolvar>
   <eolvar name="Intel(R) ME CLINK Signal">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
```

```
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Disabled/00/Enabled/01"
example="Disabled"> </RequiredValue>
   </eolvar>
   <eolvar name="Intel(R) AMT WD Auto Reset">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="No/00/Yes/01" example="No">
</RequiredValue>
   </eolvar>
   <eolvar name="Intel(R) AMT Supported">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="No/00/Yes/01" example="No">
</RequiredValue>
   </eolvar>
   <eolvar name="Intel(R) AMT Idle Timeout">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Hex number with 0x prefix."
example="0x0000"> </RequiredValue>
   </eolvar>
   <eolvar name="Integrated Sensor Hub Supported">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
```

```
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Disabled/00/Enabled/01"
example="Disabled"> </RequiredValue>
   </eolvar>
   <eolvar name="Host Name">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="String" example="Any">
</RequiredValue>
   </eolvar>
   <eolvar name="Firmware Update OEM ID">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Hex" example="00000000-0000-0000-
0000-000000000000"> </RequiredValue>
   </eolvar>
   <eolvar name="Firmware KVM Screen Blanking">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="No/00/Yes/01" example="No">
</RequiredValue>
```

```
   </eolvar>
   <eolvar name="Firmware Dynamic Application Loader
Supported">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="No/00/Yes/01" example="No">
</RequiredValue>
   </eolvar>
   <eolvar name="FeatureShipState">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Hex number with 0x prefix."
example="0x00000000"> </RequiredValue>
   </eolvar>
   <eolvar name="FWUpdLcl">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Disabled/00/Enabled/01/Full and
Partial disabled/03" example="Disabled"> </RequiredValue>
   </eolvar>
   <eolvar name="End of Manufacturing Enable">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
```

```
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="No/00/Yes/01" example="No">
</RequiredValue>
   </eolvar>
   <eolvar name="Embedded Host Based Config">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Disabled/00/Enabled/01"
example="Disabled"> </RequiredValue>
   </eolvar>
   <eolvar name="EOM Configuration">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Lock Descriptor and OEM
Configs/00/Lock Descriptor and OEM Configs via First
Boot/01/Lock OEM Configs Only/02/Lock OEM Configs Only via
First Boot/03/Lock Descriptor Only/04/Lock Descriptor Only via
First Boot/05/Do not lock Descriptor and OEM Configs/06/Do not
lock Descriptor and OEM Configs via First Boot/07"
example="Lock Descriptor and OEM Configs"> </RequiredValue>
   </eolvar>
   <eolvar name="Domain Name">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="String" example="Any">
</RequiredValue>
   </eolvar>
```

```
   <eolvar name="Delayed Authentication Mode Config">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Disabled/00/Enabled/01"
example="Disabled"> </RequiredValue>
   </eolvar>
   <eolvar name="Debug Override Production Silicon">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Hex number with 0x prefix."
example="0x00000000"> </RequiredValue>
   </eolvar>
   <eolvar name="Debug Override Pre-Production Silicon">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Hex number with 0x prefix."
example="0x00000000"> </RequiredValue>
   </eolvar>
   <eolvar name="Config Server IPv6/IPv4 Port">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
```

```
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Hex number with 0x prefix."
example="0x0000"> </RequiredValue>
   </eolvar>
   <eolvar name="Config Server IPv6/IPv4 Address">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="String" example="Any">
</RequiredValue>
   </eolvar>
   <eolvar name="Config Server FQDN">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies>CORP</Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="String" example="Any">
</RequiredValue>
   </eolvar>
   <eolvar name="CSME Measured Boot to TPM">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>3</Level -->
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Disabled/00/Enabled/01"
example="Disabled"> </RequiredValue>
   </eolvar>
   <eolvar name="Automatic Built in Self Test">
      <!-- The commented fields below CANNOT be edited. Any
edits will be ignored by the tool -->
      <!-- Description>Test variable against expected
value</Description -->
      <!-- IntelRequired>False</IntelRequired -->
      <!-- Dependencies></Dependencies -->
      <!-- Level>3</Level -->
```

```
      <!-- End of uneditable fields -->
      <!-- Please edit the fields below ONLY with the State or
ErrAction -->
      <State>Disabled</State>
      <ErrAction>ErrorContinue</ErrAction>
      <RequiredValue format="Disabled/00/Enabled/01"
example="Disabled"> </RequiredValue>
   </eolvar>
   <!-- END OF EOL VAR TESTS -->
</memanuf_config>
```

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

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

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

```
<eolconfig name="GBE version">
        <!-- The commented fields bellow CANNOT be edited. Any edits will be
ignored by the tool -->
        <!-- Description>Check Gbe Version against expected value</Description -->
        <!-- IntelRequired>False</IntelRequired -->
        <!-- Dependencies>LAN</Dependencies -->
        <!-- End of uneditable fields -->
        <!-- edit the fields below ONLY with the State or ErrAction -->
        <State>Enabled</State>
        <ErrAction>ErrorContinue</ErrAction>
        <RequiredValue format="major_ver.minor_ver" example="0.6">
</RequiredValue>
    </eolconfig>
```

5.4.3 MEManuf –EOL Variable Check

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

Table 5-3. MEManuf - EOL Config Tests

Test	Expected Configuration
EOP status check	Enabled.
RPMC Configuration	Complicated check.
ME FW Version	User expected version.
BIOS Version	User expected version.
GBE Version	User expected version.
Firmware Update OEM ID	User expected value (Format: "`00000000-0000-0000-0000-000000000000`").
Touch – Vendor ID	Expected vendor ID.
FPFs in UEP Committed	All UEPs committed and equal to the FPFs' values.
SOC Config Lock	1=Enabled.
HW Binding Disabled	True.
PCH Unlocked State	Unlocked.
Confirm ARB SVN Value	Image SVN = Minimum ARB SVN.
Checking NVM for fatal Flash Logs	No errors in log.
FW Status	Complicated check – FWStatus is ok.
Boot Guard Status	Complicated check – BootGuard files in FWStatus are ok.
PCHC FW Version	User expected version.
ME Manufacturing Mode status	Disabled.
ME Read Access Permissions	User value (0x001 for example).
ME Write Access Permissions	User value (0x001 for example).
GBE Read Access Permissions	User value (0x001 for example).
GBE Write Access Permissions	User value (0x001 for example).
BIOS Read Access Permissions	User value (0x001 for example).
BIOS Write Access Permissions	User value (0x001 for example).
EC Read Access Permissions	User value (0x001 for example).
EC Write Access Permissions	User value (0x001 for example).
Security Descriptor Override (SDO) check	Disabled.

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

5.4.5 Output/Result

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

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

5.5 Examples

5.5.1 Example 1

5.5.1.1 Example for Consumer Intel® ME FW SKU

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

LPC Device Id: xxxx.
Platform: Icelake Platform

Windows OS Version : 10.0

FW Status Register1: 0x94000255
FW Status Register2: 0x60000506
FW Status Register3: 0x00000020
FW Status Register4: 0x00004000
FW Status Register5: 0x00001F03
FW Status Register6: 0x04400308

    CurrentState:                           Normal
    ManufacturingMode:                      Enabled
    FlashPartition:                         Valid
    OperationalState:                       CM0 with UMA
    InitComplete:                           Complete
    BUPLoadState:                           Success
    ErrorCode:                              No Error
    ModeOfOperation:                        Normal
    SPI Flash Log:                          Not Present
    Phase:                                  HOSTCOMM Module
    ME File System Corrupted:               No
    PhaseStatus:                            UNKNOWN
    FPF and ME Config Status:               Not committed
    RPMC status:                            OK

FW Capabilities value is 0x31309650
Feature enablement is 0x31309650
Platform type is 0x42000341
Feature enablement is 0x31309650
ME initialization state valid
ME operation mode valid
Current operation state valid
ME error state valid
MFS is not corrupted
PCH SKU Emulation is correct

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

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

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


Policy Kernel - Boot Guard : Self Test - Passed

VDM - General : VDM engine - Passed

PAVP - General : Set Edp Port - Passed

Touch - General : Reset Panel - Failed

Error 255: Sensor could not be found. Either no sensor is connected, the sensor
has not yet initialized,
or the system is improperly configured.

Touch - General : Generate Test Packets - Failed

Error 255: Sensor could not be found. Either no sensor is connected, the sensor
has not yet initialized,
or the system is improperly configured.

Touch - General : Panel Detect - Failed

Error 255: Sensor could not be found. Either no sensor is connected, the sensor
has not yet initialized,
or the system is improperly configured.

Policy Kernel - ME Configuration : PROC_MISSING - Passed

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

Error 81: MEManuf Operation Failed.
```

6 Intel® MEInfo

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

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

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

6.1 Windows® PE Requirements

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

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

6.2 Usage

The executable can be invoked by:

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

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

Table 6-1. Intel® MEInfo Command Line Options

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

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

–FWSTS	`Intel ® MEInfo Version: xx.x.x.xxxx`
	`Copyright © 2005 – 2019, Intel Corporation. All rights reserved.`
	
	`FW Status Register1: 0x94000255`
	`FW Status Register2: 0x60000506`
	`FW Status Register3: 0x00000020`
	`FW Status Register4: 0x00004000`
	`FW Status Register5: 0x00001F03`
	`FW Status Register6: 0x04400308`
	
	`CurrentState Normal`
	`ManufacturingMode Enabled`
	`FlashPartition Valid`
	`OperationalState CM0 with UMA`
	`InitComplete Complete`
	`BUPLoadState Success`
	`ErrorCode No Error`
	`ModeOfOperation Normal`
	`SPI Flash Log Not Present`
	`Phase HOSTCOMM Module`
	`ME File System Corrupted No`
	`PhaseStatus UNKNOWN`
	`FPF and ME Config Status Not committed`
	`RPMC status OK`

-VERBOSE <filename>	Turns on additional information about the operation for debugging purposes. This option has to be used together with the above mentioned option(s). Failure to do so generates the error: "Error 9254: Invalid command line option". This option works with no option and `–feat`.
-H	Displays the list of command line options supported by the Intel® MEINFO tool.
-VER	Shows the version of the tools.

Option	Description
- PAGE	When it takes more than one screen to display all the information, this option lets the user pause the display and then press any key to continue on to the next screen.
-EXP	Shows examples about how to use the tools.
No option:	If the tool is invoked without parameters, it reports information for all components listed in Table 6-2 below for full SKU FW.

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

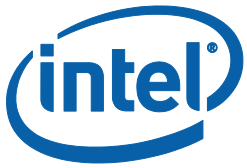
Feature Name	Feature Data Source (Intel® ME Kernel/ Intel® AMT/SW/ Other)	Consumer SKU	Specific Feature Dependency	Field Value
Tools Version	SW (Intel® MEInfo)	X	N/A	Version string Example: 13.x.y.ZZZZ; where x=minor, y = HF/MR, ZZZZ = Build Number.
BIOS Version	Intel® ME Kernel	X	MEBx needs to be present. Not available on Corporate Sku	Version string
GbE Version	Other (Directly reading from SPI)	X	GbE Region to be present in the image	A version string
PMC FW Version	Intel® ME Kernel	X	PMC Region to be present in the image	Version string
Descriptor Version	Intel® ME Kernel	X	Descriptor Region to be present in the image	Version string
IOMP FW Version	Intel® ME Kernel	X	IO Manageability Engine binary version	Version string
MGPP FW Version	Intel® ME Kernel	X	MG PHY Binary version	Version string
TBTP FW Version	Intel® ME Kernel	X	Thunderbolt™ Binary	Version string
VendorID	Intel® ME Kernel	X	N/A	A number (in Hex)
FW Version	Intel® ME Kernel	X	N/A	Version string 13.x.y.ZZZZ A B; where x=minor, y = HF/MR, ZZZZ = Build Number, A=LP/H, B=SKU type [Consumer/Corporate].

Feature Name	Feature Data Source (Intel® ME Kernel/ Intel® AMT/SW/ Other)	Consumer SKU	Specific Feature Dependency	Field Value
LMS version*	Other (Reading Windows® registry entries)	X	Only when Windows® LMS driver is installed	A version string
Intel® MEI Driver version*	Other (Reading Windows® registry entries	X	Only when Windows® Intel® MEI driver is installed	A version string
PCH Information	Intel® ME Kernel	X	N/A	Display of PCH Information including: Version Device ID Step Data SKU Type Replacement Counter Replacement State Unlock State
FW Capabilities	Intel® ME Kernel	X	N/A	Combination of feature name list breakdown (with a Hexadecimal value) *This is a display of the Feature State for the Intel® ME. Is enabled / disabled on the system. Each bit in the value represents a feature state. Intel® ME features including Full manageability, standard manageability, Anti-theft technology etc.
FW Type	Intel® ME Kernel	X	N/A	Pre-Production/Production
TLS State	Intel® ME Kernel	X	N/A	Enabled/Disabled
Last Intel® ME Reset Reason	Intel® ME Kernel	X	N/A	Power up/ Firmware reset/ Global system reset/ Unknown
FWUpdLcl	Intel® ME Kernel	X	N/A	Enabled/Disabled
TCSS FW partial update	Intel® ME Kernel	X	FIT TCSS enable feature set to 'Enabled'	Enabled/Disabled
BIOS and GbE Config Lock	Other (Directly reading from SPI)	X	N/A	Enabled/Disabled/ Unknown

Feature Name	Feature Data Source (Intel® ME Kernel/ Intel® AMT/SW/ Other)	Consumer SKU	Specific Feature Dependency	Field Value
				If shown as enabled, both FLOCKDN for BIOS and Gbe are set. If shown as disabled, either/all FLOCKDN for BIOS and Gbe are not set.
Host Read Access to Intel® ME	Other (Directly reading from SPI)	X	N/A	Enabled/Disabled/ Unknown
Host Write Access to Intel® ME	Other (Directly reading from SPI)	X	N/A	Enabled/Disabled/ Unknown
SPI Flash ID	Other (Directly reading from SPI)	X	Only when there are flash parts HW installed	A JEDEC ID number (in Hex)
ME/BIOS VSCC register values	Other (Directly reading from SPI)	X	Only when there are flash parts HW installed	A 32bit VSCC number (in Hex)
BIOS Boot State	Intel® ME Kernel	X	N/A	Pre Boot/ In Boot/ Post Boot
OEM Id	Intel® ME Kernel	X	Only if fw image supports OEM Id	UUID for OEM to check during FW Update
Capability Licensing Service State	Intel® ME Kernel	X	Not available on Corporate Sku. Not shown unless Fw feature capability supports it	Enabled/Disabled
OEM Tag	Intel® ME Kernel	X	N/A	A 32bit Hexadecimal number
M3 Autotest	Intel® ME Kernel	X	FIT CM3 Autotest Enabled set to 'true'	Enabled/Disabled
C-Link Status	Intel® ME Kernel	X	Intel® Wireless LAN	Enabled/Disabled
RPMC Replay Protection	Intel® ME Kernel	X	FIT PTT RPMC Supported feature set to 'Yes'	Supported/Not Supported
RPMC Replay Protection Bind Counter	Intel® ME Kernel	X	N/A	Counter indicating the number that SPI flash has been rebound

Feature Name	Feature Data Source (Intel® ME Kernel/ Intel® AMT/SW/ Other)	Consumer SKU	Specific Feature Dependency	Field Value
RPMC Replay Protection Bind Status	Intel® ME Kernel	X	N/A	Pre-Bind/Bound
RPMC Protection Rebind	Intel® ME Kernel	X	FIT PTT RPMC Rebinding Enabled feature set to 'Yes'	Supported/Not Supported
RPMC Replay Protection Max Rebind	Intel® ME Kernel	X	N/A	Counter indicating the maximum number of rebinds
Storage Device Type	Intel® ME Kernel	X	N/A	SPI/UFS
FWSTS	Intel® ME Kernel	X	N/A	Two 32bit Hexadecimal numbers and their bit definition breakdown
EPID Group ID	Intel® ME Kernel	x	N/A	HEX Value
Keybox	Intel® ME Kernel	x	N/A	Provisioned/ Not provisioned
Intel® PTT State	Intel® ME Kernel	X	FIT PTT is set to 'Enable'	Yes/No
Intel® PTT initial power-up state	Intel® ME Kernel	X	N/A	Enabled/Disabled
PAVP State	Intel® ME Kernel	X	N/A	Yes/No
ISH initial power state	Intel® ME Kernel	X	N/A	Yes/No
End of Manufacturing Enable	Intel® ME Kernel	X	Running closemanuf on platform	Yes/No
OEM Public Key Hash FPF	Intel® ME Kernel	X	BIOS	Yes / No
OEM Public Key Hash ME	Intel® ME Kernel	X	BIOS	SHA-256bit Hash entry
Minimum Allowed Anti Rollback SVN	Intel® ME Kernel	X	BIOS	Counter indicating the minimum allowed ARB SVN value
Image Anti Rollback SVN	Intel® ME Kernel	X	BIOS	Counter indicating the ARB SVN existing in the FW Image

Feature Name	Feature Data Source (Intel® ME Kernel/ Intel® AMT/SW/ Other)	Consumer SKU	Specific Feature Dependency	Field Value
Trusted Computing Base SVN	Intel® ME Kernel	X	BIOS	Counter indicating TCB SVN
Crypto HW Support	Intel® ME Kernel	X	BIOS	Enabled/Disabled
BSMM SVN FPF	Intel® ME Kernel	X	BIOS	Hash of Public Key to verify Boot Policy Manifest
Protect BIOS Environment	Intel® ME Kernel	X	BIOS	Yes / No
CPU Debugging	Intel® ME Kernel	X	BIOS	Enabled / Disabled
BSP Initialization	Intel® ME Kernel	X	BIOS	Enabled / Disabled
Measured Boot	Intel® ME Kernel	X	BIOS	Yes / No
Verified Boot	Intel® ME Kernel	All	BIOS	Yes / No
Key Manifest ID	Intel® ME Kernel	All	BIOS	Hash of Public Key to verify Boot Policy Manifest
Enforcement Policy	Intel® ME Kernel	All	BIOS	Unrestricted / Remediation / Restricted
iTouch	SW (Intel® MEInfo)	All	iTouch	iTouch information includes: • Device ID • HW Revision ID • FW Revision ID • Frame Size • Feedback Size • Sensor Mode • Maximum Number of Touch Point • SPI Frequency • SPI I/O Mode



6.3 Examples

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

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

6.3.1 Consumer Intel® ME FW SKU

```
Intel (R) MEInfo Version: 14.0.YY.XXXX
Copyright (C) 2005 - 2019, Intel Corporation. All rights reserved.

Intel(R) ME code versions:

BIOS Version                                CMLSFWR1.R00.1344.D00.1908230552
MEBx Version                                14.0.0.XXXX
GbE Version                                 0.4
Descriptor Version                          1.0
Vendor ID                                   8086
FW Version                                  14.0.YY.XXXX LP Consumer
LMS Version                                 1927.14.0.XXXX
MEI Driver Version                          1931.14.0.XXXX

PMC FW Version                              140.1.1.XXXX
OEM FW Version                              14.0.10.XXXX
ISHC FW Version                             5.0.0.XXXX
PCHC FW Version                             14.0.0.XXXX

PCH Information
    PCH Version                             0
    PCH Device ID                           284
    PCH Step Data                           A0
    PCH SKU Type                            Pre-Production ES
    PCH Replacement Counter                 0
    PCH Replacement State                   Disabled
    PCH Unlocked State                      Enabled

FW Capabilities                             0x31309640

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


Capability Licensing Service State          Enabled
Crypto HW Support                           Enabled
End of Manufacturing Enable                 No
FWUpdLcl                                    Enabled
Firmware Update OEM ID                      00000000-0000-0000-0000-000000000000
Integrated Sensor Hub Initial Power State   Enabled
Intel(R) PTT State                          Enabled
Intel(R) PTT initial power-up state         Enabled
OEM Tag                                     0x00
PAVP State                                  Yes
Post Manufacturing NVAR Config              Yes
```

```
TLS State                                       Enabled

FW Type                                         Pre-Production
Last ME reset reason                            Unknown
BIOS Config Lock                                Enabled
Host Read Access to ME                          Enabled
Host Write Access to ME                         Enabled
Host Read Access to EC                          Enabled
Host Write Access to EC                         Enabled
SPI Flash ID 1                                  EF4019
SPI Flash ID 2                                  Not Available
BIOS boot State                                 Post Boot
M3 Autotest                                     Disabled
EPID Group ID                                   0x4DC
Keybox                                          Not Provisioned
RPMC Replay Protection                          Unsupported
RPMC Replay Protection Bind Counter             0
RPMC Replay Protection Bind Status              Pre-bind
RPMC Replay Protection Rebind                   Unsupported
RPMC Replay Protection Max Rebind               1
Storage Device Type                             SPI
Minimum Allowed Anti Rollback SVN               1
Image Anti Rollback SVN                         2
Trusted Computing Base SVN                      1
Re-key needed                                   False
HW Binding                                      Enabled


                                                FPF          UEP          ME FW
                                                *In Use
                                                ---          ---          -----
ACM SVN                                         0x00         0x00         0x00
BSMM SVN                                        0x00         0x00         0x00
EK Revoke State                                 Not Revoked  Not Revoked  Not Revoked
Error Enforcement Policy 0                      Disabled     Disabled     Disabled
Error Enforcement Policy 1                      Disabled     Disabled     Disabled
Intel(R) PTT                                    Enabled      Enabled      Enabled
KM SVN                                          0x00         0x00         0x00
OEM ID                                          0x00         0x00         0x00
OEM KM Present                                  Enabled      Enabled      Enabled
OEM Platform ID                                 0x00         0x00         0x00
OEM Secure Boot Policy                          0x78         0x78         0x78
CPU Debugging                                   Enabled      Enabled      Enabled
BSP Initialization                              Enabled      Enabled      Enabled
Protect BIOS Environment                        Enabled      Enabled      Enabled
Measured Boot                                   Enabled      Enabled      Enabled
Verified Boot                                   Enabled      Enabled      Enabled
Key Manifest ID                                 0x01         0x01         0x01
Force Boot Guard ACM                            Disabled     Disabled     Disabled
PTT Lockout Override Counter                    0x00         0x00         0x00
Persistent PRTC Backup Power                    Enabled      Enabled      Enabled
RPMC Rebinding                                  Disabled     Disabled     Disabled
RPMC Support                                    Disabled     Disabled     Disabled
SOC Config Lock State                           Enabled      Disabled     Enabled
SPI Boot Source                                 Enabled      Enabled      Enabled
Txt Supported                                   Disabled     Disabled     Disabled

OEM Public Key Hash FPF
4D19B4F23FF9170C2C46B3D76BF05919A7FA8B6B113DF53C86C0E8003C23A8DC
```

```
OEM Public Key Hash UEP
4D19B4F23FF9170C2C46B3D76BF05919A7FA8B6B113DF53C86C0E8003C23A8DC
OEM Public Key Hash ME FW
4D19B4F23FF9170C2C46B3D76BF05919A7FA8B6B113DF53C86C0E8003C23A8DC
```

6.3.2 Retrieve Current Value of Flash Version

```
C:\ MEINFO.exe -feat "BIOS boot state"
Intel(R) MEINFO Version: 13.0.0.xxxx
Copyright(C) 2005 - 2017, Intel Corporation. All rights reserved.

BIOS boot State: Post Boot

> MEINFO.efi -feat `^"BIOS boot state"^'
Intel(R) MEINFO Version: 13.0.0.xxxx
Copyright(C) 2005 - 2017, Intel Corporation. All rights reserved.
```

BIOS boot State: Post Boot

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

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

Intel(R) MEINFO Version: 13.0.0.xxxx
Copyright(C) 2005 - 2017, Intel Corporation. All rights reserved.

Local FWUpdate: Success - Value matches FW value.


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

Intel(R) MEINFO Version: 13.0.0.xxxx
Copyright(C) 2005 - 2017, Intel Corporation. All rights reserved.

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

§ §

7 Intel® ME Firmware Update

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

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

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

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

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

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

7.1 Requirements

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

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

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

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

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

7.2 Windows® PE Requirements

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

7.3 Enabling and Disabling Intel® FWUpdate

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

This option supports two values, enabled and disabled.

Disabled – does not allow FW to be updated

Enabled – allows FW to be updated

For more details, refer Intel® MEBx user guide.

7.4 FWUpdate Flows

7.4.1 Full FWUpdate

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

Global Reset will be required to complete the FWUpdate operation.

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

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

7.4.2 Partial FWUpdate

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

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

7.5 Usage

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

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

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

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

Table 7-1. Image File Update Options

Option	Description
-VERBOSE [<FILE>]	Verbose. Enables additional information about the tool's operation to be displayed for debugging purposes.
-Y	Ignore warning. If the warning asks for input "Y/N", this flag makes the tool automatically take "y" as the input.
-F <FILE>	File. Specifies the FWUpdate image file to be used for performing an update.
-SAVE <file>	Restore Point. Retrieves an update image from the FW based on the currently running FW. The update image is saved to the user-specified file.
-ALLOWSV	Allow Same Version. Allows the version of the input FW (based on the file input) to be the same as the version of the FW currently on the platform. Without this option, an attempt to perform an update on the same version will not proceed.
-FORCERESET	Force Reset. The tool automatically reboots the system after the update process with FW is complete. The system reboot is necessary for the new FW to take effect. An attempt to update the FW without this option will end with a message telling the user to reset the platform for the changes to take effect.
-OEMID <UUID>	OEM ID. The tool uses the specified OEM ID during the transaction of the new FW image with the Manageability Engine. The purpose of the OEM ID is for manufacturers to have an identifier for their system. Using any other OEM ID value other than what is on the FW running on the target platform results in a failure of the FWUpdate process. The full image (including all necessary flash partitions) flashed to the system can be configured with the Flash Image Tool to specify the OEM ID (this tool specifies a default of zeros for the OEM ID.) If this command line option is not used, the default OEM ID used for the update is zeros. The OEM ID is configured in the existing FW image running on the platform. The OEM ID value is specified in the UUID format (8-4-4-4-12).
–PARTID	This option is always used along with the –F option. The partition ID is requested using the "partid" option. If the requested partition is expected by the Firmware the tool will search for the expected partition in the image provided, extract it and send it to the FW to perform the update. If the expected partition is not found in the image or if the requested partition is not expected by the firmware an error will be returned to the user.
-FWVER	Display FW version
-H or -?	Displays the list of command line options supported by the Intel® MEINFO tool. **Note:** Use -H for help when running in the EFI Shell.
-EXP	Shows examples about how to use the tools.
-VER	Shows the version of the tools.
-PARTVER	Display flashed FW partition with its FW Version
-SILENT	Runs FWUpdate in Silent
-PARTVENDOR	Vendor ID of the partition

7.6 Examples

7.6.1 Updates Intel® ME with Firmware Binary File

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

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

FWUpdLcl.exe –f FW.BIN

EFI:
FWUpdLcl.efi –f FW.BIN
```

7.6.2 Partial Firmware Update

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

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

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

Non-Verbose Mode

```
…\FWUpdLcl.exe –f FW.bin –partid WCOD

Intel (R) Firmware Update Utility version 13.0.0.xxxx
Copyright (C) 2007-2018, Intel Corporation.  All rights reserved.

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

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

Verbose Mode

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

Intel (R) Firmware Update Utility version 13.0.0.xxxx
Copyright (C) 2007-2018, Intel Corporation.  All rights reserved.

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

Firmware last update status = Firmware update success
```

```
Firmware last update reset type = 2

FW Update is completed successfully.
```

7.6.3 Display Supported Commands

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

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

```
…\FWUpdLcl.exe -exp partid

Intel (R) Firmware Update Utility version 13.0.0.xxxx
Copyright (C) 2007-2017, Intel Corporation.  All rights reserved.


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

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

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


EFI:
> FWUpdLcl.efi -exp partid

Intel (R) Firmware Update Utility version 13.0.0.xxxx
Copyright (C) 2007-2017, Intel Corporation.  All rights reserved.


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

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


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

7.6.4 Language Codes

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

Language	Language Code
English	0x01
French	0x02

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

8 UEFI Sample Application Leveraging FWUpdate API Library

8.1 Getting Started - FWUpdate 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 FWUpdate 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 FWUpdate (Full Size) and FWUpdate RS (reduced size) folders released within the relevant FW Kit. Users will find the following files:

In FWUpdate (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.
fwudef.h	Header file including FWUpdate definitions.
fwupdatelib.h	Header file including all the functions that can be used by customers.
FWUpdateLib.lib	Static library with dynamic links to import DLLs.

Fwupdatelibdeprecated.h	Old deprecated FWUpdate header file. Functions within this file will be deprecated in future projects.
FWUpdateSample.c	Source file including a sample code for customers who intend to incorporate the Full Size FWUpdate library. This is only relevant to Windows.
FWUpdLcl64.exe	Full FWUpdate tool.

In FWUpdate Reduced Size (RS) folder:

errorlist.c & errorlist.h	Source and header files for the error generation.
fwudef.h	Header file including FWUpdate definitions.
fwupdatelib.h	Header file including all the functions that can be used by customers
FWUpdateEFILib.lib	FWUpdate RS Library compiled in EFI64 EDKII.
Fwupdatelibdeprecated.h	Old deprecated RS FWUpdate header file. Functions within this file will be deprecated in future projects.
FWUpdLclApp.c	Source file including a sample code for customers who intend to incorporate the Reduced Size FWUpdate library. This sample code is in EFI EDKII.
FWUpdLclAppDeprecated.c	Source file including a deprecated sample code for customers who intend to incorporate the Reduced Size FWUpdate library. This sample code is in EFI EDKII. It uses deprecated functions from fwupdatelibdeprecated.h.
FwUpdLcl.efi	Reduced Size FWUpdate 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 as *deprecated*, this is intended for functions that have new replacement functions and will be deprecated in future projects.

Note: Some function titles are marked with the initials *RS*. This is intended for functions that apply for the FWUpdate RS library.

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

8.2.1 FWUpdate deprecated functions vs new functions

The below table displays the summary of the deprecation of old functions and the replacement (if applicable) with the new functions.

[FS] – Functions in Full Size library only.

[RS] – Functions in Reduced Size library only.

All other functions are in both Reduced Size library and full size library.

Old Function (Deprecated)	**New function**	**Description**
FwUpdateFullBuffer [FS] FwUpdateFull	FwuFullUpdateFromBuffer [FS] FwuFullUpdateFromFile	Full Update from a buffer. [FS] From file name.
FwUpdatePartialBuffer [FS] FwUpdatePartial	FwuPartialUpdateFromBuffer [FS] FwuPartialUpdateFromFile	Partial Update of IUP from a buffer. [FS] From file name.
FWUpdate_QueryStatus_Get_Response	FwuCheckUpdateProgress	Check for FW Update progress and completion.
GetInterfaces	FwuEnabledState	Return FW Update enabling state: enabled, disabled, password protected.
[RS] GetFwFlashVersion (no parameter) [FS] GetFwVersion (file name) (if file name null, return flash version) GetPartVersion	FwuPartitionVersionFromFlash FwuPartitionVersionFromBuffer [FS] FwuPartitionVersionFromFile	Old function: Return CSE version from flash or update image. New function: use partition FTPR version to get CSE version. Get version of a specific partition.
[RS] HeciPdt	[RS] FwuSetIshConfig	Update ISH configuration file.
VerifyOemId	FwuOemId	Get OEM ID from flash.

[FS] GetOemId		Removed function VerifyOemId.
GetPartVendorID	FwuPartitionVendorIdFromFlash FwuPartitionVendorIdFromBuffer	Get vendor ID of a specific partition.
GetLastStatus	x	Return the last Update status. Removed function, since it is returned in FwuUpdateCheckProgress().
GetLastUpdateResetType	x	Return the needed reset type after Update. Removed function
IsUpdateStatusPending	x	Removed function
CheckPolicyBuffer [FS]CheckPolicy	x	Check if downgrade / same version/ upgrade. Removed function
GetExtendedIpuPartitionAttributes	x	Removed function
GetIpuPartitionAttributes	x	Removed function
CheckVersion	x	Removed function. Defined in h file, but not implemented.
StartUpdate	x	Removed function. Defined in h file, but not implemented.
EndUpdate	x	Removed function. Defined in h file, but not implemented.

[RS] FwUpdateCheckPowerSource	x	Removed function. Defined in h file, but not implemented.
[FS] FwUpdateCheckPowerSource	[FS] FwuPowerSource	Check FWSTS register 1 [28:29] – power source
[FS] GetPchSKU	FwuPchSku	Return LP or H.
[FS] GetFwType	FwuFwType	Return slim / consumer / corporate.
[FS] SaveRestorePoint	FwuSaveRestorePointToBuffer [FS] FwuSaveRestorePointToFile	Save the current image from the flash.
[FS] IsRestorePointImage	x	Removed function
[FS] GetFwUpdateInfoStatus	x	Return FWSTS register bits: FID, IUP needed, FwuInProgress.
[FS] FwUpdateRestore [FS] FwUpdateRestoreBuffer	x	Removed function. (use regular Full Update function instead)
[FS] FwUpdatePartialWithInstanceId [FS] FwUpdatePartialWithInstanceId Buffer	[FS] FwuPartialUpdateWithInstanceIdFromFile [FS] FwuPartialUpdateWithInstanceIdFromBuffer	Partial Update to a specific instance id.

8.2.2 Full FWUpdate 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 FWUpdate using buffer as the base for the FWUpdate.

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

8.2.3 Partial FWUpdate from Buffer (*FS*)(*RS*)

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

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

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

8.2.4 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.

- 0 = No reset is required
- 1 = Hot reset is required
- 2 = CSE reset is required
- 3 = Global reset is required

8.2.5 Get FWUpdate ability (*FS*)(*RS*)

UINT32 FwuEnabledState (Out UINT16 *EnabledState);

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

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

8.2.6 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 (OUT UINT32 *fwType);

Purpose: This function retrieves the FW type from flash.

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

2 = FWU_FW_TYPE_SLIM
3 = FWU_FW_TYPE_CONSUMER
4 = FWU_FW_TYPE_CORPORATE

8.2.8 Retrieve PCH SKU (*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

8.2.9 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)

8.2.10 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)

8.2.11 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

8.2.12 Performing a full FWUpdate (*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.

8.2.13 Performing a partial FWUpdate (*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.

8.2.14 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 partition ID from a given update image file.

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

8.2.15 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: ***CurentInstanceId*** – Current instance ID ***ExpectedInstanceId*** – Expected instance ID

8.2.16 Performing a partial FWUpdate 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 FWUpdate with the provided instance ID from a buffer

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

8.2.17 Performing a partial FWUpdate with Instance ID from file (*FS*)

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

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

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

8.2.18 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

8.2.19 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.

8.2.20 Checking power source (*FS*)

UINT32 FwuPowerSource(OUT UINT32 *powerSource);

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

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

8.2.21 Set ISH configuration file (*RS* Only)

UINT32 FwuSetIshConfig (UINT8 *Buffer, UINT32 BufferLength);

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

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

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

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

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

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

8.2.23 Get Interfaces (Deprecated) (*FS*)(*RS*)

unsigned int GetInterfaces(unsigned short *interfaces);

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

Arguments	***Interfaces*** - whether the Local FW Update is disabled (0) or enabled (1) or password protected (2)

Returns	Gets the Interfaces from HECI 0 = Success Non-zero value = Failure

8.2.24 Get Last Status (Deprecated) (*FS*)(*RS*)

unsigned int GetLastStatus(unsigned int *lastStatus);

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

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

8.2.25 Get Last Update Reset Type (Deprecated) (*FS*)(*RS*)

unsigned int GetLastUpdateResetType(unsigned int *lastResetType);

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

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

8.2.26 Check Policy (Deprecated) (*FS*)

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

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

Arguments	***Image File*** - Binary Image file ***AllowSV*** - Allow Same Version flag (Set to 1 to execute same version flow)

	Update Type - Update Type Output. Can be DOWNGRADE_SUCCESS = 0, DOWNGRADE_FAILURE = 1, SAMEVERSION_SUCCESS = 2, SAMEVERSION_FAILURE = 3, UPGRADE_SUCCESS = 4, UPGRADE_PROMPT = 5, ***Ver***- FW Version (Major, Minor, Hotfix, Build)
Returns	0 = Success Non-zero value = Failure

8.2.27 Check Policy Buffer (Deprecated) (*FS*)(*RS*)

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

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

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

8.2.28 Verify OEM Id (Deprecated) (*FS*)(*RS*)

bool VerifyOemId(_UUID id);

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

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

8.2.29 Get Ipu Partition Attributes (Deprecated) (*FS*)(*RS*)

unsigned int GetIpuPartitionAttributes(FWU_GET_IPU_PT_ATTRB_MSG_REPLY *FwuGetIpuAttrbMsgInfo);

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

Arguments	Out parameter: FWU_GET_IPU_PT_ATTRB_MSG_REPLY – is a data structure with IPU related information
Returns	0 = Success Non-zero value = Failure

8.2.30 Get FW Update Info Status (Deprecated) (*FS*)

unsigned int GetFwUpdateInfoStatus(FWU_INFO_FLAGS *StatusFlags);

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

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

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

8.2.31 FW Update Query Status Get Response (Deprecated) (*FS*)(*RS*)

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

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

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

Returns	***LastResetType –*** indicates Reset type required for the fwupdate process just completed 0 = Success Non-zero value = Failure

8.2.32 FW Update Full – Using Buffer (Deprecated) (*FS*)

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

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

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

8.2.33 FW Update Partial Buffer (Deprecated) (*FS*)(*RS*)

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

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

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

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

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

Arguments	***Buffer*** - Buffer ***Buffer Length*** – Length of buffer
Returns	***Partition ID*** - denotes the partition ID, which could be WLAN (wcod) or language (locl). ***WOCD ID*** = 0x244f4357 and ***LOCL ID*** = 0x4C434F4C ***Flags***: not used. ***IpuUpdatedInfo*** – not used. 0 = Success Non-zero value = Failure

8.2.34 PDT Data (Sensor Calibration Data) Update (Deprecated) (*RS Only*)

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

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

Arguments	***Buffer*** - Buffer ***Buffer Length*** – Length of buffer
Returns	0 = Success Non-zero value = Failure

8.2.35 Retrieving Firmware Version (Deprecated) (*FS*)

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

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

9 Intel® Manifest Extension Utility (Intel® MEU)

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

The Intel® MEU tool completes the following steps:

- Creates an Independent Updatable Partition (IUP) by adding manifest and meta-data information to the firmware.
- Calls an external LZMA tool for compression of the firmware binary
- Calls the OpenSSL tool as the signing infrastructure tool to sign the partition.

The Intel® Manifest Extension Utility (MEU) tool can also be used to create OEM Key Manifest and OEM Unlock Token.

9.1 Usage

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

§ §

Appendix A: Intel® ME NVARs

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

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

Table A-1. NVARs Descriptions

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

<table>
<tr><th>Fixed Offset Name</th><th>Description</th><th>Data Length (in Bytes)</th><th>Expected Value</th><th>Reset Type</th><th>Mfg. Post EOM/Pre EOP</th></tr>
<tr>
<td>OEMSkuRule</td>
<td>UINT32 (little endian) value. This controls what features are permanently disabled by OEM.
Notes:
There are reserved bits that the must not be changed for proper platform operation. The user should only modify the bit(s) for the feature(s) they wish to change. There is NO ability to change features one at a time. This NVAR sets OEM Permanent Disable for ALL features. In addition prior updating or changing any of available settings it is highly recommended that the user first retrieves the current OEM Sku Rule and toggling only the desired bits, and then resave them.
This will not enable functionality that is not capable of working in the target hardware SKU. Please see the respective Firmware Bring-up Guide for a list of what features are capable with what firmware bundle and Hardware SKU of Intel 9 Series Chipset.</td>
<td>4</td>
<td>Feature Capable: 1
Feature Permanently disabled: 0
<table>
<tr><th>Bit</th><th>Description</th><th>Notes</th></tr>
<tr><td>31</td><td>Reserved</td><td></td></tr>
<tr><td>30</td><td>Reserved</td><td></td></tr>
<tr><td>29</td><td>PTT</td><td></td></tr>
<tr><td>29:22</td><td>Reserved</td><td></td></tr>
<tr><td>21</td><td>TLS</td><td></td></tr>
<tr><td>20</td><td>DAL</td><td></td></tr>
<tr><td>19</td><td>Reserved</td><td></td></tr>
<tr><td>18</td><td>KVM</td><td>2</td></tr>
<tr><td>17</td><td>Reserved</td><td></td></tr>
<tr><td>16</td><td>ME Network Disable</td><td></td></tr>
<tr><td>15:13</td><td>Reserved</td><td></td></tr>
<tr><td>12</td><td>PAVP</td><td></td></tr>
<tr><td>11</td><td>Reserved</td><td></td></tr>
<tr><td>10</td><td>ISH</td><td></td></tr>
<tr><td>9:6</td><td>Reserved</td><td></td></tr>
<tr><td>5</td><td>Reserved</td><td></td></tr>
<tr><td>4:3</td><td>Reserved</td><td></td></tr>
<tr><td>2</td><td>Manageability and Security Application</td><td>1</td></tr>
<tr><td>1</td><td>Reserved</td><td></td></tr>
<tr><td>0</td><td>Manageability Full</td><td>1</td></tr>
</table>
1. For corporate SKUs bits 0 and 2 need to be both set to ‘1’ to allow for Intel® AMT to work.
2. KVM (bit 18) should only be set to ‘1’ when Manageability Application (bit 2) is set to ‘1’. If using a Corporate SKU, then Manageability Full (bit 0) must also be set to ‘1’.</td>
<td>Global</td>
<td>No</td>
</tr>
</table>

<table>
<tr>
<td>FeatureShipState</td>
<td>UINT32 (little endian) value. This controls what features are enabled or disabled. These features may be enabled /disabled by mechanisms such as MEBx or provisioning. This setting is only relevant for features NOT permanently disabled by the OEM Permanent Disable.

This will not enable functionality that is not capable of working in the target hardware SKU. Please see the respective Firmware Bring-up Guide for a list of what features are capable with what firmware bundle and Hardware SKU of Intel 8 Series Chipset.

Notes:

There are reserved bits that the must not be changed for proper platform operation. The user should only modify the bit(s) for the feature(s) they wish to change. There is NO ability to change features one at a time. This NVAR sets OEM Permanent Disable for ALL features. In addition prior updating or changing any of available settings it is highly recommended that the user first retrieves the current Feature Shipment Time State and toggling only the desired bits, and then resave them.</td>
<td>4</td>
<td>Feature Enabled: 1

Feature Disabled: 0

<table>
<tr><th>Bit</th><th>Description</th><th>Notes</th></tr>
<tr><td>31:30</td><td>Reserved</td><td></td></tr>
<tr><td>29</td><td>PTT</td><td></td></tr>
<tr><td>28:11</td><td>Reserved</td><td></td></tr>
<tr><td>10</td><td>ISH</td><td></td></tr>
<tr><td>3:9</td><td>Reserved</td><td></td></tr>
<tr><td>2</td><td>Manageability Full</td><td></td></tr>
<tr><td>1:0</td><td>Reserved</td><td></td></tr>
</table>

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

Fixed Offset Name	Description	Data Length (in Bytes)	Expected Value	Reset Type	Mfg. Post EOM/Pre EOP
WLAN Power Well	Sets which power well the board uses for WLAN cards	4	**0x80** = Disabled **0x81** = CoreWell **0x82** = Primary Well **0x83** = ME Well **0x86** = WLAN Sleep via SLP_WLAN#	**Global**	**No**
OEM Tag	A human readable 32-bit number to describe the flash image represented by value	4	Readable 32 bit hex value identifying the image. Can be empty (Null).	**Global**	**No**

GPIO	GPIO	60	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 0-19 GPP_E 0-23 GPP_F 0-19 GPP_G 0-7 GPP_H 0-23 GPD 0-11 For H platforms: GPP_A 0-23 GPP_B 0-23 GPP_C 0-23 GPP_D 0-15 GPP_E 0-12 GPP_F 0-23 GPP_G 0-15 GPP_H 0-23 GPP_I 0-14 GPP_J 0-9 GPP_K 0-11 GPD 0-11 Example read of GPIO: Variable: "gpio" Value: 0x0000 : 00 00 00 00 04 00 00 00 06 00 00 00 01 00 00 00 0x0010 : 00 00 00 00 01 00 00 00 04 00 00 00 0C 00 00 00 0x0020 : 01 00 00 00 00 00 00 00 08 00 00 00 01 00 00 00 0x0030 : 0F 00 00 00 01 00 00 00 00 00 00 00 Note: the only locations that can be modified are underlined above. The format for updating the GPIO is as follows... Gpio = 0x00000000030000001100000001000000000000001000000020000001700000001000000000000000800000003000000130000000100000000000000	**ME**	**No**

Fixed Offset Name	Description	Data Length (in Bytes)	Expected Value	Reset Type	Mfg. Post EOM/Pre EOP
			RST = GPP_D_17 IRQ = GPP_C_23 DFU = GPP_D_19		
FWUpdLcl	Enabled Firmware Update Capability	1	0 = disabled 1 = enabled	**Global**	**Yes**
eDP Port Config	EDP Port Configuration. Up to two ports can be enabled 0x00 - 0x01 – A 0x02 - B 0x04 - C 0x08 - D 0x10 – E 0x20 - F	1	0x0:None/None 0x1:A/None 0x2:B/None 0x3:A/B 0x4:C/None 0x5:C/A 0x6:C/B 0x8:D/None 0x9:D/A 0xa:D/B 0xC:D/C 0x10:E/None 0x11:E/A 0x12:E/B 0x14:E/C 0x18:E/D 0x20:F/None 0x21:F/A 0x22:F/B 0x24:F/C 0x28:F/D 0x30:F/E	**Global**	**No**
LSPCON Port Config	LSPCON Port Configuration.	1		**Global**	**No**
Delayed Authentication Mode Config	Enables Delayed Authentication Mode on the platform	1	0 = Disabled 1 = Enabled	**ME**	**Yes**
Unconfigure On RTC	Enables platform to unconfigure on RTC removal	1	0 = Disabled 1 = Enabled	**ME**	**Yes**
AMT Related NVARs					
OEM Custom Cert 1	Cert Hash Data. See Certificate Hash Entry Structure definition **Note: If the platform is un-configured the Certificate Hash will be deleted.**	55 => n >= 83	Valid Certificate Hash Entry (SHA1, SHA256 or SHA384)	**ME**	**Yes**

Fixed Offset Name	Description	Data Length (in Bytes)	Expected Value	Reset Type	Mfg. Post EOM/Pre EOP
OEM Custom Cert 2	Cert Hash Data. See Certificate Hash Entry Structure definition **Note: If the platform is un-configured the Certificate Hash will be deleted.**	55 => n >= 83	Valid Certificate Hash Entry (SHA1, SHA256 or SHA384)	**ME**	**Yes**
OEM Custom Cert 3	Cert Hash Data. See Certificate Hash Entry Structure definition **Note: If the platform is un-configured the Certificate Hash will be deleted.**	55 => n >= 83	Valid Certificate Hash Entry (SHA1, SHA256 or SHA384)	**ME**	**Yes**

Fixed Offset Name	Description	Data Length (in Bytes)	Expected Value	Reset Type	Mfg. Post EOM/Pre EOP
Redirection Privacy / Security Level	Redirection (KVM, SOL, Storage Redirection) privacy level and configuration (RCFG, CCM) settings.	1	Default **0x01** Enhanced **0x02** Extreme **0x03** Default: SOL enabled = true Storage Redirection enabled = true KVM enabled = true Opt-in can be disabled= true KVM opt-in configurable remotely = true RCFG and CCM = true Enhanced: SOL enabled = true Storage Redirection enabled = true KVM enabled = true Opt-in can be disabled= false Opt-in configurable remotely = true RCFG and CCM = true Extreme SOL enabled = false Storage Redirection enabled = false KVM enabled = false Opt-in can be disabled= false KVM opt-in configurable remotely = N/A RCFG and CCM = false	**ME**	**No**
Embedded Host Based Config	Embedded Host Based Configuration State.	1	0 = Disabled 1 = Enabled	**ME**	**No**
Firmware KVM Screen Blanking	Screen Blanking Enabled	1	0 = Disabled 1 = Enabled	**ME**	**No**
PKI Domain Name Suffix	PKI DNS Suffix. Null terminated string	32	PKI DNS Suffix in dotted string format	**ME**	**Yes**
Config Server FQDN	Configuration Server FQDN (Fully Qualified Domain Name)	255	Example: "intelFVE.com"	**ME**	**Yes**

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

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

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

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