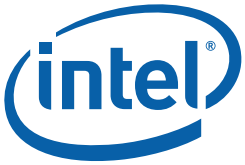


System Tools for Intel® 6 Series Chipset Family Intel® Management Engine Firmware 7.0 SKU's

User Guide

December 2010

Revision: 1.08

Intel Confidential

Contents

1 Introduction 11
1.1 Terminology 11
1.2 Reference Documents 17
2 Preface 18
2.1 Overview 18
2.2 CPT System Tools Changes 18
2.3 Image Editing Tools 19
2.4 Manufacturing Line Validation Tools 19
2.5 Intel® ME Setting Checker Tool 19
2.6 Operating System Support 20
2.7 Generic System Requirements 20
2.8 Error Return 21
2.9 Usage of the Double-Quote Character (") 21
2.10 PMX Driver Limitation 22
3 Flash Image Tool 23
3.1 System Requirements 23
3.2 Flash Image Details 23
3.2.1 Flash Space Allocation 24
3.3 Required Files 25
3.4 FITC Operation Mode 25
3.5 FITC Wizard Interface 25
3.5.1 FITC Wizard Interface Configuration 25
3.5.2 Screen Progression 26
3.6 FITC Advanced Mode 45
3.6.1 Configuration Files 45
3.6.2 Environment Variables 46
3.6.3 Build Settings 47
3.6.4 Modifying the Flash Descriptor Region 51
3.6.5 PCH Soft Straps 56
3.6.6 VSCC Table 56
3.6.7 Modifying the Intel® ME Region 57
3.6.8 Intel® ME FW Configuration 58
3.6.9 Modifying the GbE (LAN) Region 61
3.6.10 Modifying the PDR Region 62
3.6.11 Modifying the BIOS Region 63
3.6.12 Building a Flash Image 64
3.6.13 Change the Region Order on the SPI Device 65
3.6.14 Decomposing an Existing Flash Image 65
3.7 Command Line Interface 66
3.8 Example – Decomposing an Image and Extracting Parameters 67
3.9 More Examples of FITC CLI 68

4 Flash Programming Tool 69
4.1 System Requirements 69
4.2 Flash Image Details 69
4.3 Microsoft Windows Required Files 70
4.4 DOS Required Files 71
4.5 Programming the Flash Device 71
4.5.1 Stopping Intel® ME SPI Operations 71
4.6 Programming Fixed Offset Variables 72
4.7 Usage 72
4.8 Updating Hash Certificate through FOV 76
4.9 fparts.txt File 78
4.10 Examples 78
4.10.1 Example 1 – Flash SPI Flash Device with Binary File 79
4.10.2 Example 2 – Program a Specific Region 79
4.10.3 Example 3 – Program SPI Flash from a Specific Address 79
4.10.4 Example 4 – Dump Specific Region 79
4.10.5 Example 5 – Display SPI Information 80
4.10.6 Example 6 – Verify Image with Errors 80
4.10.7 Example 7 – Verify Image Successfully 81
4.10.8 Example 8 – Program FOV Parameter 81
4.10.9 Example 9 – Get ME settings 82
4.10.10 Example 10 – Compare ME settings 82

5 MEManuf and MEManufWin 84
5.1 Windows* PE Requirements 84
5.2 How to Use MEMANUF 84
5.3 Usage 85
5.4 MEMANUF –EOL Check 88
5.4.1 MEMANUF.cfg File 88
5.4.2 MEMANUF –EOL Variable Check 92
5.4.3 MEMANUF –EOL Config Check 92
5.4.4 Output/Result 92
5.5 Examples 93
5.5.1 Example 1 93
5.5.2 Example 2 93
5.5.3 Example 3 93
5.5.4 Example 4: Consumer Platform 98

6 MEInfo 99
6.1 Windows* PE Requirements 99
6.2 Usage 99
6.3 Examples 107
6.3.1 Example 1 107
6.3.2 Example 2 108
6.3.3 Local FWUpdate: EnabledExample 3 109

7 ME Firmware Update 110
7.1 Requirements 110
7.2 Windows* PE Requirements 111

7.3 Enabling and Disabling Intel® FWUpdate 111
7.4 Usage 111
7.5 Examples 113
7.5.1 Example 1 113
7.5.2 Example 2 113

8 Update Parameter Tool 114
8.1 Purpose of the Tool 114
8.2 Usage of the Tool 114
8.3 USB Utility 116
8.3.1 Syntax 116
8.4 Output 117
8.5 Intel® ME Parameters Intel® UpdParam can Change 119
8.6 Examples 119

Appendix A Fixed Offset Variables 121

Appendix B Tool Detail Error Codes 130

Appendix C Tool Option Dependency on BIOS/Intel® ME Status 145

Appendix D SKU Features 146

Figures

Figure 1: SPI Flash Image Regions 24
Figure 2: Serial Flash Configuration Screen 27
Figure 3: VSCC Configuration Screen 28
Figure 4: Image Source Files Screen 29
Figure 5: LAN Configuration Screen 30
Figure 6: Intel ME Application Permanent Disable screen for 5MB Intel® ME FW SKU 31
Figure 7: Intel® ME Kernel Configuration Screen 32
Figure 8: Manageability Application Screen 34
Figure 9: Intel® ME Networking Services Setup Screen 35
Figure 10: Intel® Anti Theft Technology Setup Screen 36
Figure 11: DMI/PCIe Configuration Screen 37
Figure 12: Thermal Reporting Screen 38
Figure 13: Boot Configuration Options Screen 39
Figure 14: Integrated Clock Configuration Screen - CPT 40
Figure 15: Single-Ended Clocks screen - CPT 41
Figure 16: Differential Clocks Screen - CPT 42
Figure 17: Production/Nonproduction Configuration Screen 43
Figure 18: Build Screen 44
Figure 19. Environment Variables Dialog 47
Figure 20. Build Settings Dialog 49
Figure 21: Selected an SKU Platform in FITC 51
Figure 22. Descriptor Region Length Parameter 52
Figure 23: Descriptor Region > Descriptor Map Parameters 52
Figure 24: Flash Components Dialog 53
Figure 25: Descriptor Region > Component Section Parameters 53
Figure 26: Descriptor Region > Master Access Section 55
Figure 27PCH Straps 56
Figure 28: Add VSCC Table Entry Dialog 57
Figure 29: Sample VSCC Table Entry 57
Figure 30: Intel® ME Section 58
Figure 31: Manageability Application Section 59
Figure 32: Power Packages Section 59
Figure 33: Features Supported Section 60
Figure 34: Setup and Configuration Section 61
Figure 35: GbE Region Options 61
Figure 36: PDR Region Options 62
Figure 37: BIOS Region Parameters 63
Figure 38: Region Order 65
Figure 39: Flash Image Regions 70
Figure 40: FPT Sample Input File 74
Figure 41: Raw Hash Values from Certificate File 77
Figure 42: Sample Hash.txt File 77
Figure 43: UPDParam Error Message for Incorrect Password 118
Figure 44: UPDParam Error Message for Failure to Update Parameter(s) 118

Tables

Table 1: OS Support for Tools 20
Table 2: Tools Summary 21
Table 3: Flash Image Regions – Description 24
Table 4: Build Settings Dialog Options 48
Table 5: Region Access Control Table 54
Table 6: CPU/BIOS Access 54
Table 7: Feature Default Settings by SKU 60
Table 8: FITC Command Line Options 66
Table 9: Flash Image Regions – Description 70
Table 10: Fixed Offset Variables Options 72
Table 11: Command Line Options for fpt.exe and fptw.exe 73
Table 12: Intel-Recommend Access Settings 76
Table 13: Options for DOS Version of the Tool 85
Table 14: Intel® MEMANUF Test Matrix 87
Table 15: MEMANUF - EOL Config Tests 92
Table 16: Intel® MEInfo Command Line Options 99
Table 17: List of Components for which Version Information is retrieved 101
Table 18: Image File Update Options 111
Table 19: Update Parameter Tool Options 114
Table 20: Required Reset for Updated Parameters 115
Table 21: USB Utility Options 116
Table 22: Fixed Offset Item Descriptions 121

Revision History

Revision Number	Description	Revision Date
0.5	Alpha	02/17/2010
0.6	Alpha for 5M updating FOV, updating OS support table	04/06/2010
0.7	updating FOV Adding what's new section Update FLOCKDN in Intel® MEINFO FOV override FITC setting RCR 1023314	04/12/2010
0.8	Re-write some of the English Update MEINFO table	05/13/2010
0.81	Update FOV table Fix FWupdate usage bug	06/01/2010
0.90	Update FITC screen shot and FWupdate error message	06/30/2010
1.00	FWupdate tool update Update the certificate changes	07/06/2010
1.01	Add explaination on fptw64 and –EOL check Remove RPAT	08/04/2010
1.02	update FITC build settings explaination	08/23/2010
1.03	Update MEMANUF, MEINFO examples	09/07/2010
1.04	Update Fwupdate command line options	09/16/2010
1.05	MEMANUF –no3G option Update the FOV changes for reset required after changes	10/08/2010
1.06	Update some FITC screen shots	10/27/2010
1.07	Update MEMANUF error message	11/02/2010

Revision Number	Description	Revision Date
1.08	Add limitation for FPT PSKfile option, Add note on FITC decomposing MEMANUF error message	12/02/2010

§

1 Introduction

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

1.1 Terminology

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

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

Acronym / Term	Definition
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 & Play feature. When installing a driver, this file provides the OS with the necessary information about driver filenames, driver components, and supported hardware.
Intel® AMT Firmware	The Intel® AMT Firmware running on the embedded processor
Intel® AT	Intel® Anti-Theft Technology
Intel® AT-p	Intel® Anti-Theft Protection (previously known as TDT)
Intel® ME	Intel® Management Engine. The embedded processor residing in the chipset GMCH.
Intel® MEBx	Intel® Management Engine BIOS Extensions
Intel® MEI	Intel® Management Engine Interface (renamed from HECI). The interface between the Intel® Management Engine and the Host system.
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® ME Setting Checker Tool
Intel® MEInfoWin	Windows version of Intel® MEINFO
Intel® MEManuf	Intel® MEManuf validates Intel® ME functionality on the manufacturing line
Intel® MEManufWin	Windows version of Intel® MEManuf
Intel® RPAT	Intel® Remote PC Assist Technology. Also known as Castle Peak. Intel® RPAT is a consumer PC manageability technology that helps connect PC support desks to a User's PC regardless of the state of the OS.
ISV	Independent Software Vendor
IT User	Information Technology User. Typically very technical and uses a management console to ensure multiple PCs on a network function.
JEDECID	Joint Electronic Device Engineering Councils ID. Standard Manufacturer's Identification Code that is assigned, maintained and updated by the JEDEC office
JTAG	Joint Test Action Group
KVM	Keyboard, Video, Mouse
LAN	Local Area Network
LED	Light Emitting Diode
LMS	Local Management Service. An SW application which runs on the host machine and provides a secured communication between the ISV agent and the Intel® Management Engine Firmware.
LPC	Low Pin Count Bus

Acronym / Term	Definition
M0	Intel® ME power state where all HW power planes are activated. Host power state is S0.
M1	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.**
M3	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.
M-Off	No power is applied to the management processor subsystem. Intel® ME is shut down.
MAC address	Media Access Control address
NM	Number of Masters
NVAR	Named Variable
NVM	Non-Volatile Memory
NVRAM	Non-Volatile Random Access Memory
OCKEN	Output Clock Enable
ODM	Original Device Manufacturer
OEM	Original Equipment Manufacturer
OEM ID	Original Equipment Manufacturer Identification
OOB	Out Of Band
OOB interface.	Out Of Band interface. An SOAP/XML interface over secure or non secure TCP protocol.
OS	Operating System
OS Hibernate	OS state where the OS state is saved on the hard drive.
OS not Functional	The Host OS is considered non-functional in Sx power state in any one of the following cases when the system is in S0 power state: • OS is hung • After PCI reset • OS watch dog expires • OS is not present
OVR	Override
PAVP	Protected Video and Audio Path
PC	Personal Computer
PCH	Platform Controller Hub
PCI	Peripheral Component Interconnect

Acronym / Term	Definition
PCIe*	Peripheral Component Interconnect Express
PDR	Platform Descriptor Region
PHY	Physical Layer
PID	Provisioning ID
PKI	Public Key Infrastructure
PM	Power Management
PRTC	Protected Real Time Clock
PSK	Pre-Shared Key
PSL	PCH Strap Length
QST	Intel® Quiet System Technology - Embedded hardware and FW solution that allows for algorithmic relationship between system cooling fans and temperature monitors so as to reduce noise without losing thermal efficiency
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 state 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

Acronym / Term	Definition
SPI Flash	Serial Peripheral Interface Flash
Standby	OS state where the OS state is saved in memory and resumed from the memory when the mouse/keyboard is clicked.
Sx	All S states which are different than S0
SW	Software
System States	Operating System power states such as S0, S1, S2, S3, S4, and S5.
TCP/IP	Transmission Control Protocol/Internet Protocol
TLS	Transport Layer Security
UI	User Interface
UIM	User Identifiable Mark
UMA	Unified Memory Access
Un-configured state	The state of the Intel® ME FW when it leaves the OEM factory. At this stage the Intel® ME FW is not functional and must be configured.
UNS	User Notification Services
UPDPARAM	Update Parameter Tool
USB	Universal Serial Bus
USBr	Universal Serial Bus Redirection
UUID	Universally Unique IDentifier
VE	Virtualization Engine
VLAN	Virtual Local Area Network
VSCC	Vendor Specific Component Capabilities
Windows* PE	Windows* Preinstallation Environment
WIP	Work in Progress
WLAN	Wireless Local Area Network
XML	Extensible Markup Language. Intel® AMT's XML-based protocol has 3 parts: • An envelope that defines a framework for describing what is in a message and how to process it • A set of encoding rules for expressing instances of application-defined data types • A convention for representing remote procedure calls and responses
ZTC	Zero Touch Configuration

1.2 Reference Documents

Document	Document No./Location
FW Bring Up Guide	Release kit
Firmware Variable Structures for Intel® Management Engine and Intel® Active Management Technology 7.0	ANACAPA document
PCH EDS	CDI
Cougar Point SPI Programming Guide	Release kit

§

2 Preface

2.1 Overview

This document covers the system tools used for creating, modifying, and writing binary image files, manufacturing testing, Intel® ME setting information gathering, and Intel® ME FW updating. The tools are located in **Kit directory\Tools\System tools**. For information about other tools, see the tool's user guides in the other directories in the FW release (e.g., Intel® AMT tools are located in **Kit directory\Tools\AMT tools)**.

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

- CPT platform – All tools in the CPT FW release kit are designed for CPT platforms only. These tools do not work properly on any other legacy platforms (Santa Rosa, Weybridge, Montevina, McCreary, and Capella/Piketon). Tools designed for other platforms also do not work properly on the CPT platform.
- Intel® vPro™ platform – All features listed in this document are available for Intel® vPro™ platforms with Intel® ME FW 7.0. There are some features that are specifically designed for the Intel® vPro™ platform and only work on it.
- Intel® ME Firmware 7.0 SKU – A common set of tools are provided for the following Intel® ME FW 7.0 SKUs: 1.5MB Intel® ME FW SKU and 5MB Intel® ME FW SKU. The following features are only available for 5MB Intel® ME FW SKUs and 1.5MB Intel® ME FW SKU users should generally ignore them:
  - Intel® AMT
  - Intel® ME BIOS Extension (Intel® MEBx)
  - The description of each tool command or option that is not available for 1.5MB Intel® ME FW SKU contains a note indicating this.

2.2 CPT System Tools Changes

Intel developed the following new system tools for CPT platforms:

- FITC now has a wizard that is fully integrated into FTC which helps create SPI image with both 1.5M and 5M ME firmware image. The previous UI is still available under an 'advanced' option.
- FPT supports the extraction and comparison of Intel® ME NVAR data using new command line flags.
- FPT supports the disabling of Intel® ME before burning a new Intel® ME image. This is supported automatically and does not require the use of special flags.
- FWUpdate supports FW downgrade options.

- FWupdate does not support FW update over the LMS interface
- Intel® MEManuf has the functionality for checking configuration settings at the end of the manufacturing process. These checks can be configured using an external file that is enabled or disabled, based upon platform requirements.
- MEMANUF –EOL check
- There are other minor changes across multiple tools. See each tool's documentation for more information.

2.3 Image Editing Tools

The following tools create and write flash images:

- FITC:
  - Combines the GbE, BIOS, PDR, and Intel® ME FW binaries into one image.
  - Configures softstraps and NVARs for Intel® ME settings that can be programmed by a flash programming device or the FPT.
- FPT:
  - Programs the flash memory of individual regions or the entire flash device.
  - Modifies some Intel® ME settings (FOV) 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 this feature.)

2.4 Manufacturing Line Validation Tools

The manufacturing line validation tools (Intel® MEMANUF) allow the Intel® ME and Intel® AMT functionality to be tested immediately after the PCH chipset is generated. These tools are designed to be able to run quickly. They can run on simple operating systems, such as MS-DOS 6.22, Windows* 98 DOS, FreeDOS, and DRMK DOS. The Windows versions are written to run on Windows* XP (SP1/2), Windows* Vista and Windows 7. These tools are mostly run on the manufacturing line to do manufacturing testing.

2.5 Intel® ME Setting Checker Tool

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

2.6 Operating System Support

Table 1: OS Support for Tools

	MS DOS V 6.22	Windows* 98 DOS	DRMK DOS Version 8.00	FreeDOS Version 1.1.32a	PC-DOS Version 7.01	PC-DOS Version 7.00C/V (US)	Windows* PE 32 (Based on Vista & XP)	Windows* PE 64 (Based on Vista & XP)	Windows* XP SP3 32 / latest service pack	Windows* XP SP3 64 / latest service pack	Windows* Vista 32 SP1 / Latest Sever ice pack	Windows* Vista 64 SP1 / Latest Service pack	Windows* 7 32	Windows* 7 64
Flash Image Tool	N	N	N	N	N	N	N	N	SV	SN	SN	SN	SV	SN
Flash Programming Tool	SV	SV	SV	SV	SV	N	SP	SN	SV	SN	SN	SN	SV	SN
Intel® MEManuf	SV	SV	SV	SV	SV	N	SP	SN	SV	SN	SN	SN	SV	SN
Intel® MEInfo	SV	SV	SV	SV	SV	N	SP	SN	SV	SP	SN	SP	SV	SP
Intel® ME Firmware Update Tool	SV	SV	SV	SV	SV	N		SP	SV	SP	SN	SP	SV	SP
Update Parameters tool	N	N	N	N	N	SV		N	N	N	N	N	N	N

NOTES:

1. N – Not supported by Intel
   SV – Supported and validated by Intel
   SN –Supported and Nost validated by Intel
   SP –Supported and Partially Validated by Intel
2. 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.
3. RCRs 1022532 and 1022814

2.7 Generic System Requirements

The installation of the following services is required by integration validation tools that run locally on the System Under Test with the Intel® Manageability Engine:

- Intel® MEI driver.
- Intel® AMT LMS – not applicable to 1.5MB Intel® ME FW SKU.

See the description of each tool for its exact requirements.

Table 2: Tools Summary

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

2.8 Error Return

Tools always return 0/1 for the error level (0 = success, 1= error). A detail error code is displayed on the screen and stored on an error.log file in the same directory as the tools. (See Appendix B for a list of these error codes.)

2.9 Usage of the Double-Quote Character (")

The command shell used to invoke the toosl in both DOS and Windows has a built-in CLI.

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

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

For example, you 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 you want these words to be input – input"string – the command line is: input\"string.

2.10 PMX Driver Limitation

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

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

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

Warning: Running the PMX driver with the Windows* 7 driver verifier turned on causes the OS to crash. Do not include PMX as part of the verifier driver list if you are running Windows* 7 with the driver verifier turned on.

§

3 Flash Image Tool

The Flash Image tool (**FITC.exe**) creates and configures a complete SPI image file for CPT platforms in the following way:

1. FITC creates and allows configuration of the Flash Descriptor Region, which contains configuration information for platform hardware and FW.
2. FITC assembles the following into a single SPI flash image:
   - Binary files of the following regions:
     - BIOS
     - Intel integrated LAN (GbE)
     - Intel® ME
     - Platform Descriptor Region
   - The Flash Descriptor Region created by FITC
3. You 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 you don't have to recreate a new image each time.

FITC supports a set of command line parameters that can be used to build an image from the CLI or from a makefile. When a previously stored configuration is used to define the image layout, you don't have to interact with the GUI.

FITC operates in one of two modes:

- Advanced – Same layout, look, and feel of FITC from previous generations
- Wizard – A step-by-step process limited to the essential features needed to create a full SPI image.

Note: FITC 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

FITC runs on Windows* XP, Windows* Vista, and Windows* 7. The tool does not have to run on an Intel® ME-enabled system.

3.2 Flash Image Details

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

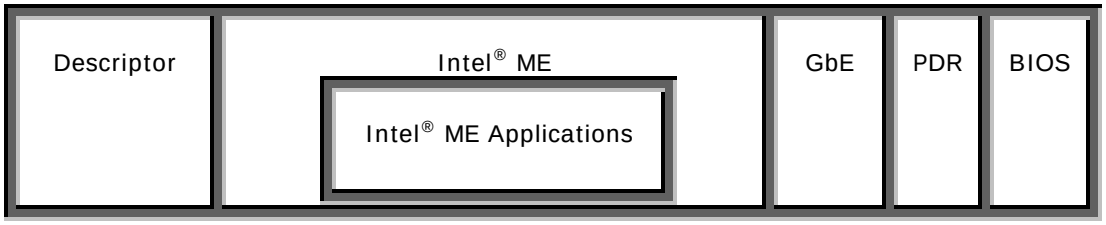


Figure 1: SPI Flash Image Regions

Table 3: Flash Image Regions – Description

Region	Description
Descriptor	This region contains information such as the space allocated for each region of the flash image, read-write permissions for each region, and a space which can be used for vendor-specific data. It takes up a fixed amount of space at the beginning of the flash memory. **Note:** This region MUST be locked before the serial flash device is shipped to end users. Please see 3.6.4.3 below for more information. Failure to lock the Descriptor Region leaves the Intel® ME device vulnerable to security attacks.
Intel® ME	This region contains code and configuration data for Intel® ME applications, such as Intel® AMT technology and Intel® AT-P. It takes up a variable amount of space at the end of the Descriptor.
GbE	This region contains code and configuration data for an Intel Integrated LAN (Gigabit Ethernet). It takes up a variable amount of space at the end of the Intel® ME region.
BIOS	This region contains code and configuration data for the entire computer.
PDR	This region lets system manufacturers describe custom features for the platform.

3.2.1 Flash Space Allocation

Space allocation for each region is determined as follows:

1. Each region can be assigned a fixed amount of space. If a region is not assigned a fixed amount of space, it occupies only as much space as it requires.
2. If there is still space left in the flash after allocating space to all of the regions, the Intel® ME region expands to fill the remaining space.
3. If there is leftover space and Intel® ME region is not implemented, the BIOS region expands to occupy the remaining space.
4. If there is leftover space and the BIOS region is not implemented, then the GbE region expands to occupy the remaining space.
5. If only the Descriptor region is implemented, it expands to occupy the entire flash.

3.3 Required Files

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

- fitctmpl.xml
- newfiletmpl.xml
- vsccommn.bin
- fitcwizardhelp.chm
- fitc.ini

FITC does not run correctly if any of the .xml and .bin files listed above are missing. FITC creates a blank **fitc.ini** file if there is no **fitc.ini** file in the folder.

3.4 FITC Operation Mode

FITC has two modes, Wizard mode and Advanced mode. The FITC Wizard uses several screens containing a selection of options and simple questions to guide you to build the image. After you select the options you want and answer the questions, the tool builds an image that is based on the information you provided.

To enter the Wizard: Press F9 or choose **Help > Wizard**.

To skip the Wizard: Press Cancel button on the Wizard's first screen. FITC then goes into FITC advanced mode, which is the legacy interface from previous generations of FITC.

3.5 FITC Wizard Interface

3.5.1 FITC Wizard Interface Configuration

You can find configuration information required by the FITC wizard in the following locations:

- General configuration information – see the FW Bring Up Guide from the appropriate Intel® ME FW kit.
- Detailed information on how to configure PCH Soft Straps and VSCC information – see the Cougar Point SPI programming guide.
- More detailed information on Intel® ME FW configuration parameters – see Appendix E.

3.5.2 Screen Progression

3.5.2.1 Progress Bar

A progress bar appears on every screen after the welcome screen. The progress bar shows how far you have progressed through the Wizard.

3.5.2.2 Serial Flash Configuration Screen

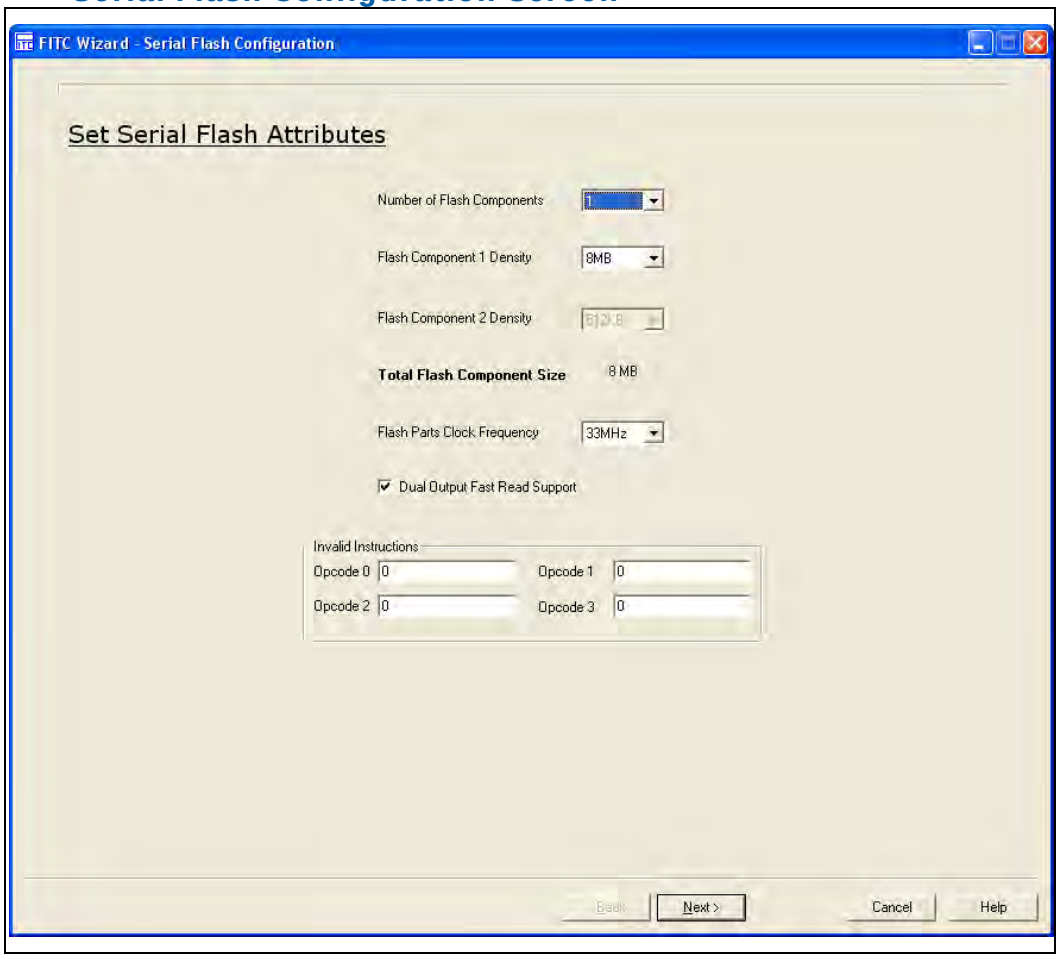


Figure 2: Serial Flash Configuration Screen

The Serial Flash Configuration screen is the first screen in the Wizard where you can configure settings which are specific to the SPI flash. You can configure the following settings:

- Number of flash components – the number of Flash components to be installed on the target machine. Valid values are 0, 1, 2. 0 causes only the ME region to be built.
- Flash component 1 Density – size of the first Flash component
- Flash component 2 Density – size of the second Flash component
- Flash Parts Clock Frequency – flash read frequency
- Opcodes – Each opcode is for an invalid instruction that the Flash Controller should protect against chip erase. An opcode that doesn't have an instruction to protect against chip erase should be set to zero.

3.5.2.3 VSCC Configuration Screen

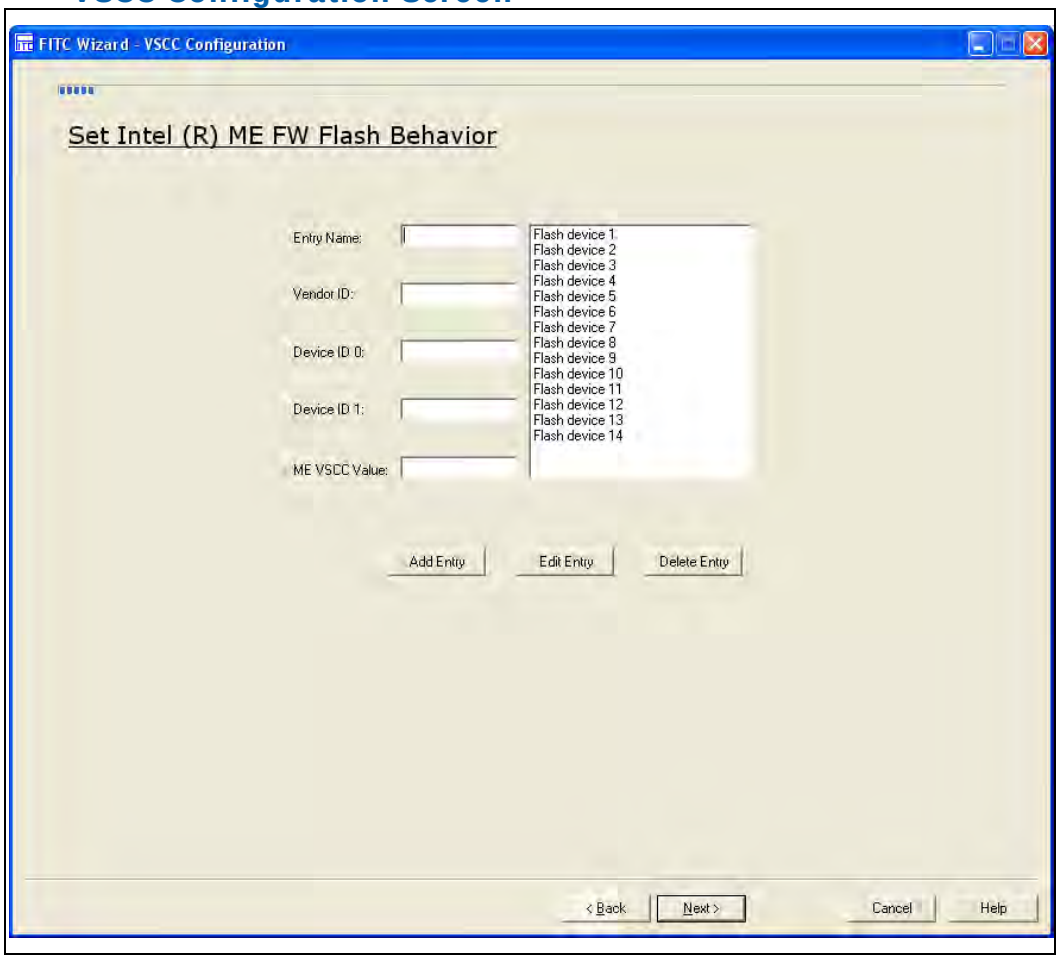


Figure 3: VSCC Configuration Screen

This screen lets you do the following to the VSCC table that will be in the Descriptor region in the final image:

- Add Entry – adds SCC entries to the table.
- Edit Entry – edits existing entries in the table.
- Remove Entry – removes an existing entry from the table.

The Wizard makes use of the **vsccommn.bin** file to validate user inputs when adding or editing a VSCC entry. The added or edited entry must be in the binary file.

The following information must be added for each entry:

- Entry name – name of the table entry
- Vendor ID – vendor-specific byte of the JEDEC ID

- Device ID 0 – first device-specific byte of the JEDEC ID
- Device ID 1 – second device-specific byte of the JEDEC ID
- ME VSCC Value – adds SPI support for Intel® ME FW

3.5.2.4 Image Source Files Screen

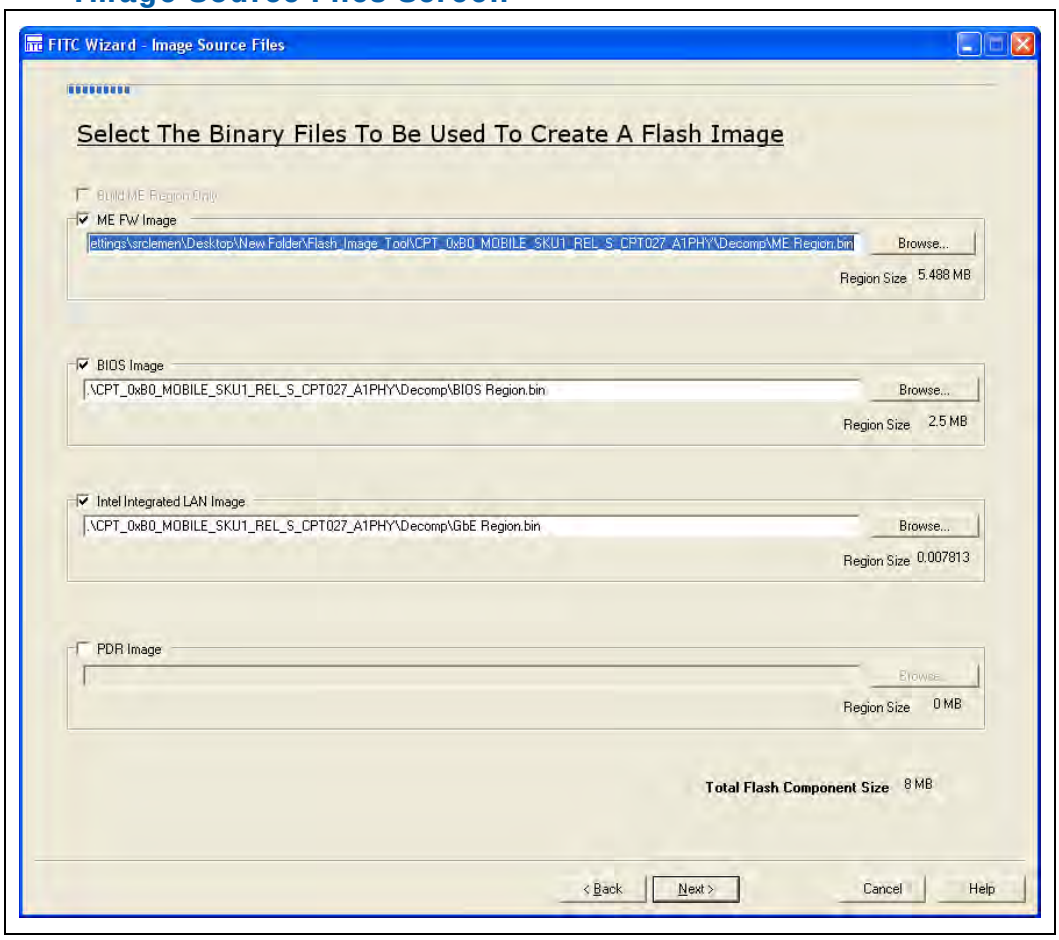


Figure 4: Image Source Files Screen

Note: The Intel Integrated LAN Image belongs to the GbE region.

The Image Source Files screen lets you select the following:

- The regions to be included in the Flash image.
- The binary image file for each of those regions.

To build a flash image from the Intel® ME region only: Select the **Build Intel® ME Region only** checkbox.

To include a region in the flash image: Select the region's image.

To exclude a region from the image: Deselect the region's image.

To select an image for a region: Click the **Browse** button and select the binary file for that image

3.5.2.5 LAN Configuration Screen

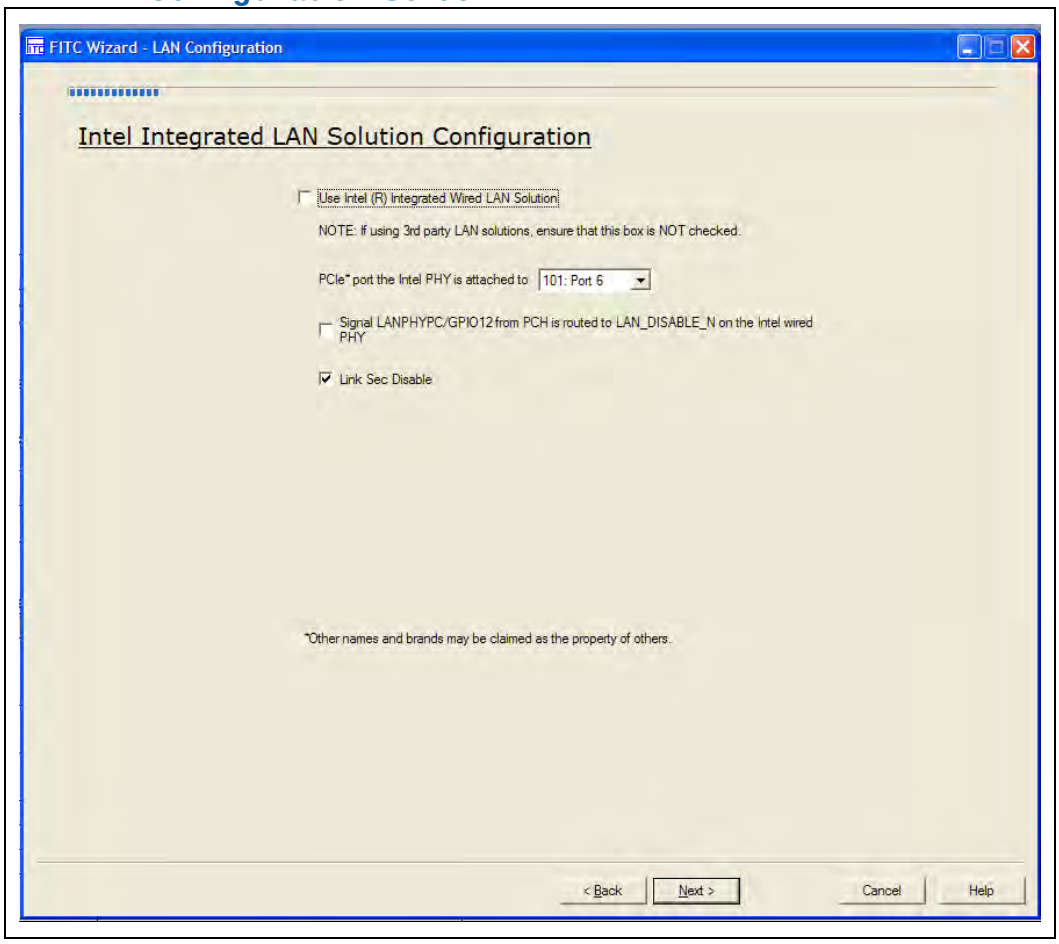


Figure 5: LAN Configuration Screen

Note: The LAN configuration screen only appears if you selected **Intel Integrated LAN Image** and a GbE binary file in the Image Source Files screen (see Figure 4).

The LAN configuration screen lets you configure the following parameters:

- Use Intel® integrated wired LAN solution – lets you use the Intel® integrated wired LAN.
- PCIe port the Intel PHY is attached to – sets the default value of the PRC.GBEPCIERPSEL register which is used to determine which PCIe port to use for MAC GBE/PHY over PCI express communication.

- Signal LANPHYPC/GPI012 from PCH is routed to LAN_Disable_N on the Intel wired PHY.
- Link Sec Disable – Disables LinkSec, a hop-by-hop network security solution that provides Layer 2 encryption and authenticity/integrity protection for packets traveling between the LinkSec-enabled nodes of the network.

3.5.2.6 Intel ME Application Permanent Disable Screen

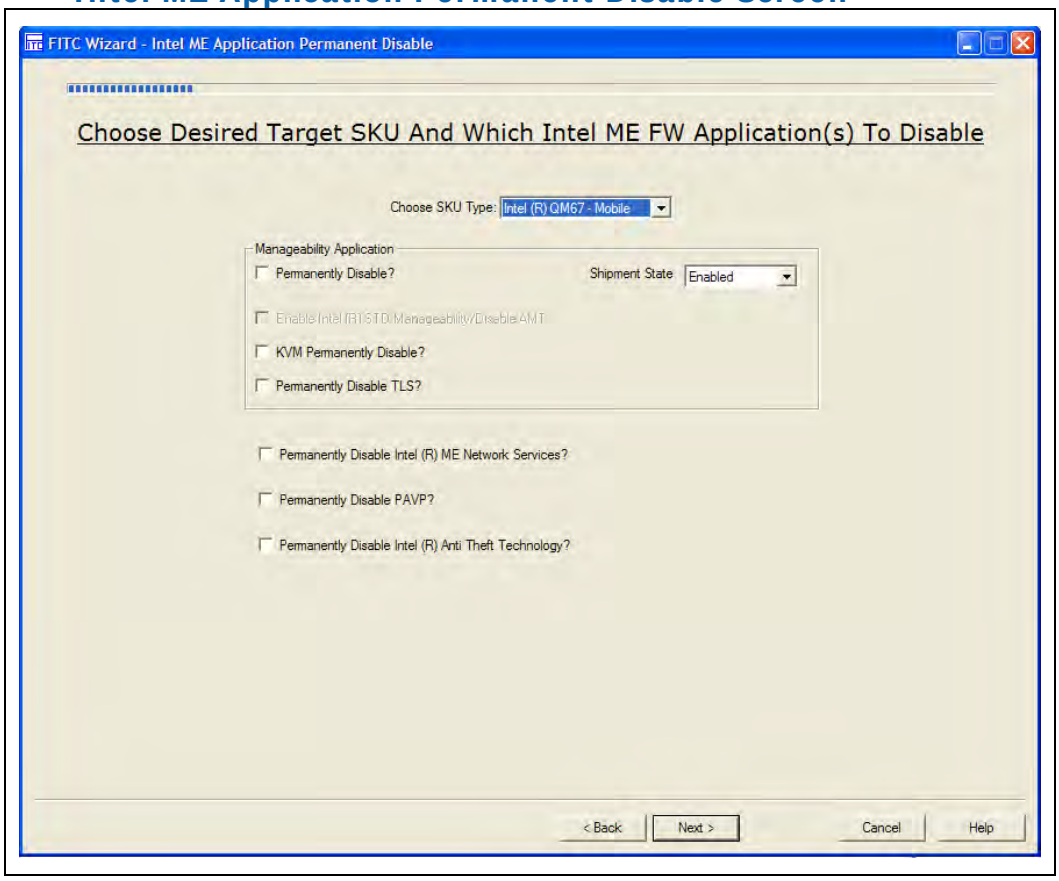


Figure 6: Intel ME Application Permanent Disable screen for 5MB Intel® ME FW SKU

Note: The screen's Manageability options are disabled for 1.5MB ME FW SKU – they are only enabled for 5MB ME FW SKU.

The Intel ME Application Permanent Disable screen lets you select an SKU Type for the FW image being built as well as the following feature-related configuration settings:

- Choose SKU type
- Manageability Application options:
  - Permanently disable – permanently disables Intel® AMT, Intel® Standard Manageability, and KVM
  - Shipment State

- Enable Intel® STD Manageability/Disable AMT – Only editable when Intel® Q67 is selected
- KVM Permanently Disable

- Permanently disable Intel® ME Network services
- Permanently disable TLS
- Permanently disable PAVP
- Permanently disable Intel® Anti Theft Technology

3.5.2.7 Intel® ME Kernel Configuration Parameters

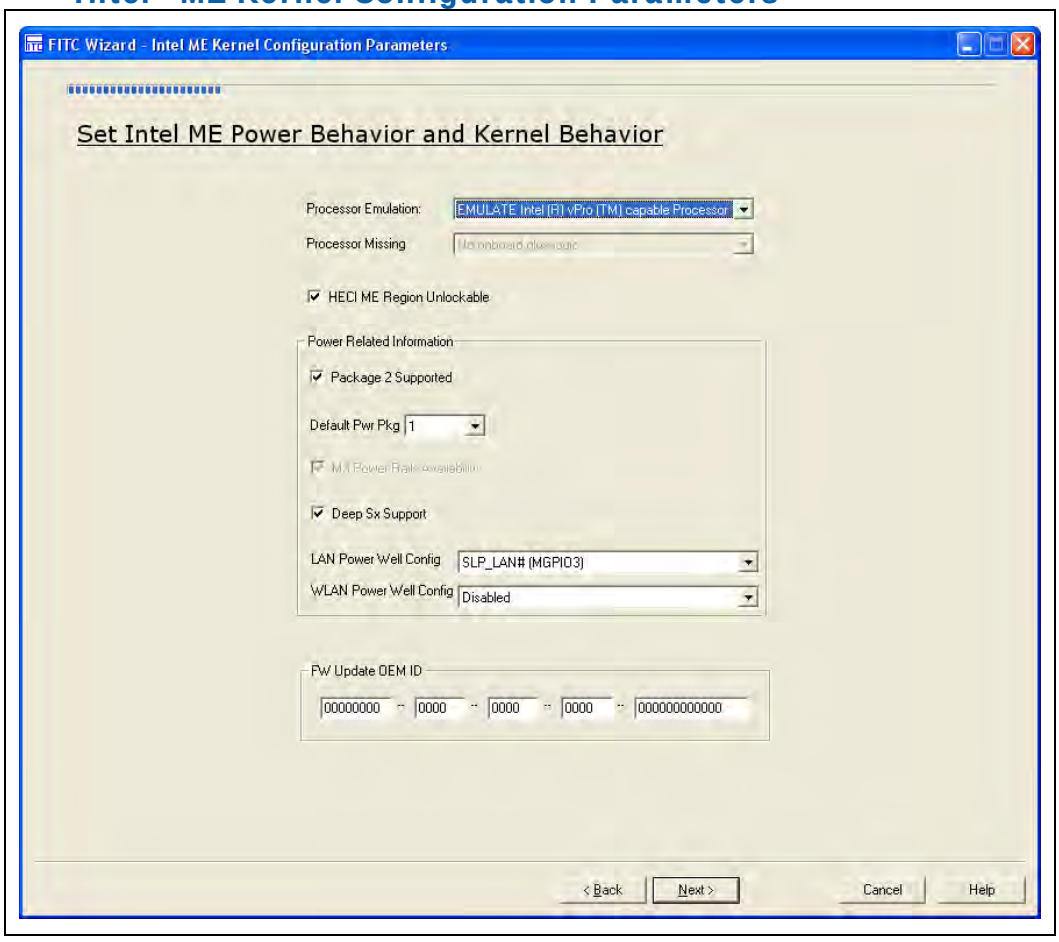


Figure 7: Intel® ME Kernel Configuration Screen

This screen contains the following options for configuring Intel® ME Kernel parameters:

- LAN Power Well Config – Options: Core Well, Sus Well, Me Well, SLP_LAN (MGPI03)
- WLAN Power Well Config – Options: Disabled, Sus Well, Me Well, WLAN Power controlled via SLP_M#|| SPDA
- M3 Power Rails Availability – Deselected=not available, Selected=available
- HECI ME Region Unlock – Selected=BIOS can write to the ME Region, Deselected=BIOS cannot write to the ME Region
- Deep Sx Support – Deep Sx is the Deep S4 and Deep S5 low power states
- Processor Missing – determines if there is glue logic present on the platform to detect a missing processor on desktop platforms
- FW Update OEM ID – Enter UUID or file containing the UUID to make sure that the platform can only be updated with an image coming from the OEM of the platform. If set to zero any input is valid (including none) when doing an FW update.
- Package 1 supported – Selected=power package available, Deselected=power package not available
- Package 2 supported – Selected=power package available, Deselected=power package not available
- Default Pwr Pkg – Select the default power package from the available packages.

3.5.2.8 Manageability Application Screen

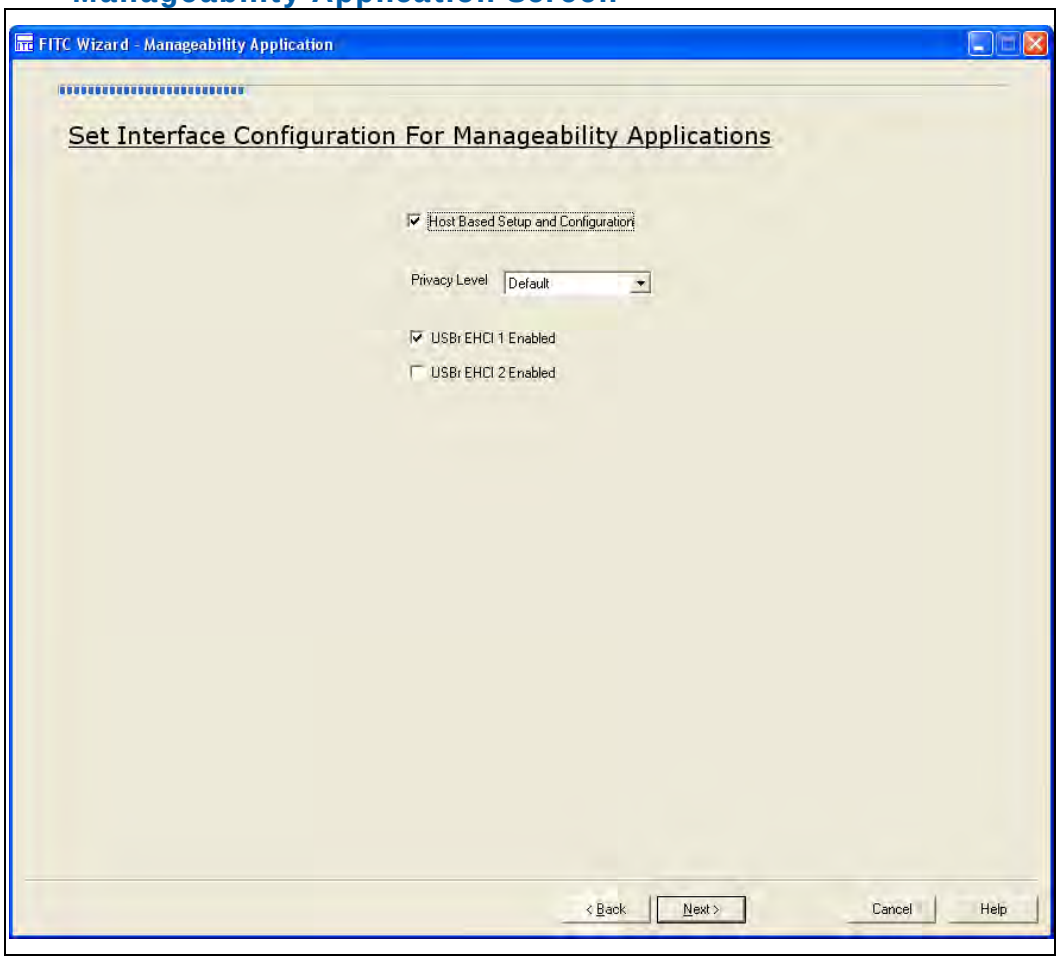


Figure 8: Manageability Application Screen

This screen configures Manageability Application parameters. Selecting an option enables it and deselecting an option disables it.

Note: This screen does not appear for 5MB ME FW SKUs if the Manageability Application **Permanently Disabled** option is selected on the Intel® ME Application Permanent Disable Screen. It also does not appear for 1.5MB ME FW SKUs.

3.5.2.9 Intel® ME Networking Services Setup Screen

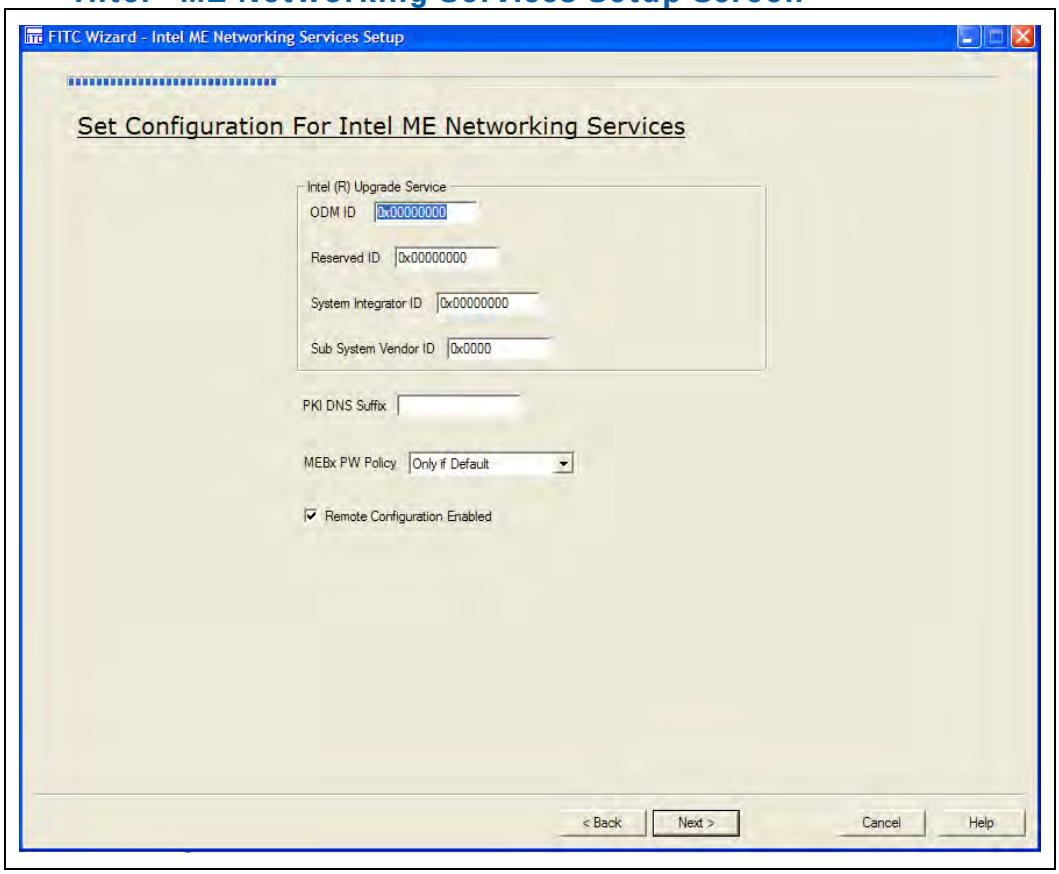


Figure 9: Intel® ME Networking Services Setup Screen

This screen contains the following options for configuring Networking Services parameters:

- Intel Upgrade Service Section:
  - ODM ID – ID generated by or registered with Intel® Upgrade web servers in order to identify the ODM/Board builder. First of three IDs stored in flash and accessible through Intel® MEI interface.
  - System Integrator ID – ID generated by or registered with Intel® Upgrade web servers in order to identify the System Integrator. Second of three IDs stored in flash and accessible through Intel® MEI interface.
  - Reserved ID – May be used as a reseller ID or other Intel® service IDs in the future
- PKI DNS Suffix – Sets PKI DNS Suffix in dotted string format
- Remote Configuration Enabled – Deselected=Disable, Selected=Enable
- MEBx PW Policy – Options:

- Over network password change is only allowed if it is the default password
- Over network password change is only allowed during setup and configuration
- Over network password change is always allowed

3.5.2.10 Intel® Anti Theft Technology Setup Screen

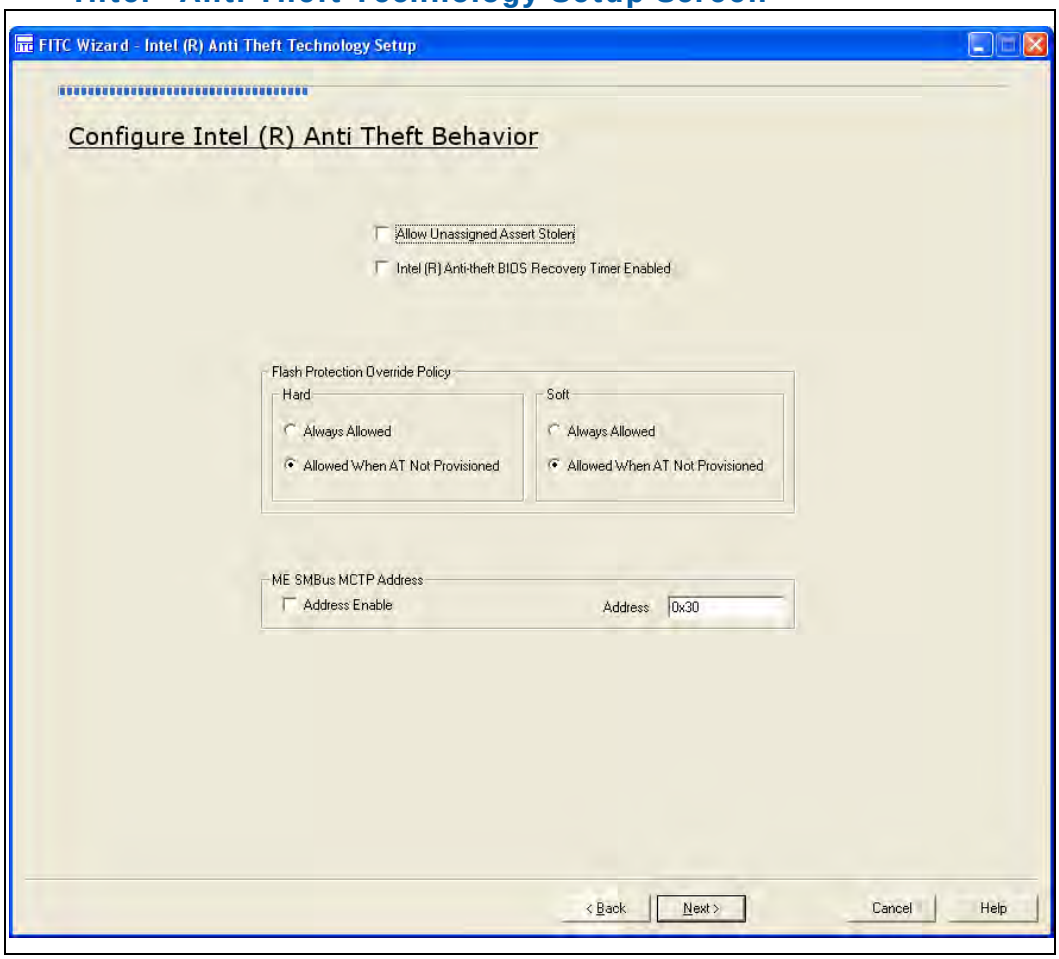


Figure 10: Intel® Anti Theft Technology Setup Screen

This screen contains the following options for configuring Anti Theft Technology parameters. Selecting an option enables it and deselecting an option disables it.

- Allow unsigned Assert Stolen –
- Intel® Anti Theft BIOS Recovery Timer Enabled – Gives a stolen platform a 30 minute window to allow a FW/BIOS rehash before the system is powered down
- Sub System Vendor ID – ID that lets OEMs test boards using Manufacturing Test Permits

- ME SMBus MCTP Address section:
  - Address enable
  - Address – Address used by Intel® ME Anti-Theft Technology FW

3.5.2.11 DMI/PCIe Configuration Screen

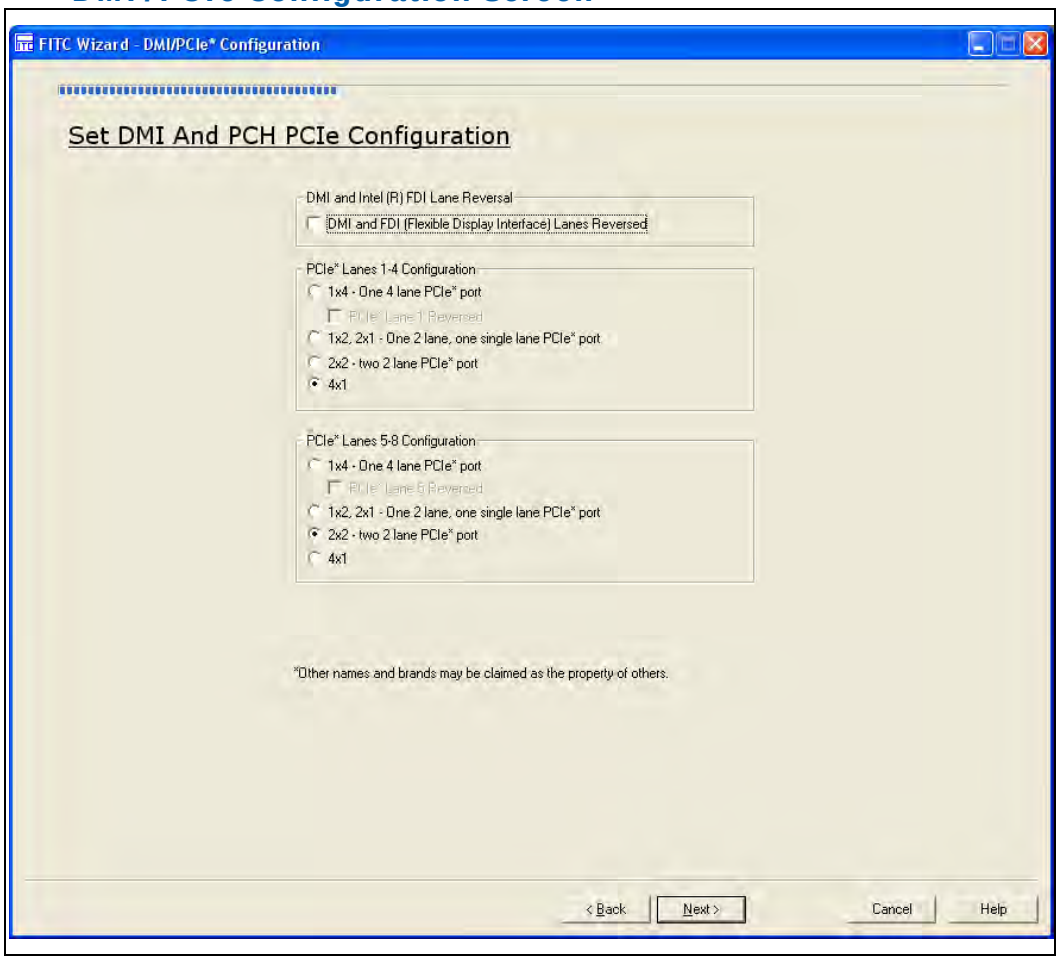


Figure 11: DMI/PCIe Configuration Screen

This screen contains the following options for configuring the PCIe and DMI parameters, which are reflected in the PCH Straps for the corresponding parameters:

- DMI and Intel FDI Lane Reversal section
- PCIe Lanes 1-4 configuration section – sets the default value of the PCIe Port Configuration 1 register covering PCIe ports 1-4
- PCIe Lanes 5-8 configuration section – sets the default value of the PCIe Port Configuration 2 register covering PCIe ports 5-8.

3.5.2.12 Thermal Reporting Screen

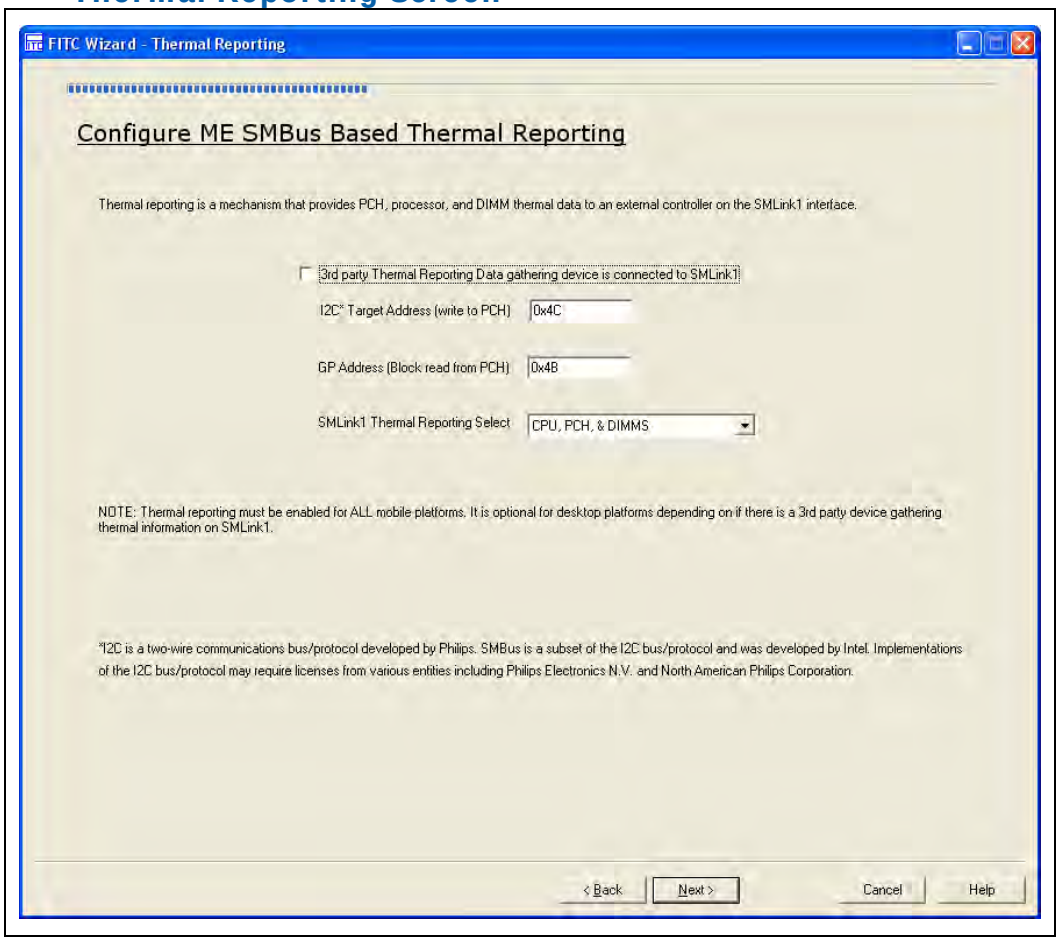


Figure 12: Thermal Reporting Screen

This screen contains the following options for configuring Thermal Reporting parameters:

- 3rd Party Thermal Reporting Data gathering device is connected to SMLink:
  - Deselected=GPI029 can only be used as SLP_LAN# for Intel integrated LAN solution
  - Selected=GPI029 is available for GPI0 configuration
- 12C Target Address
- GP address

3.5.2.13 Boot Configuration Options Screen

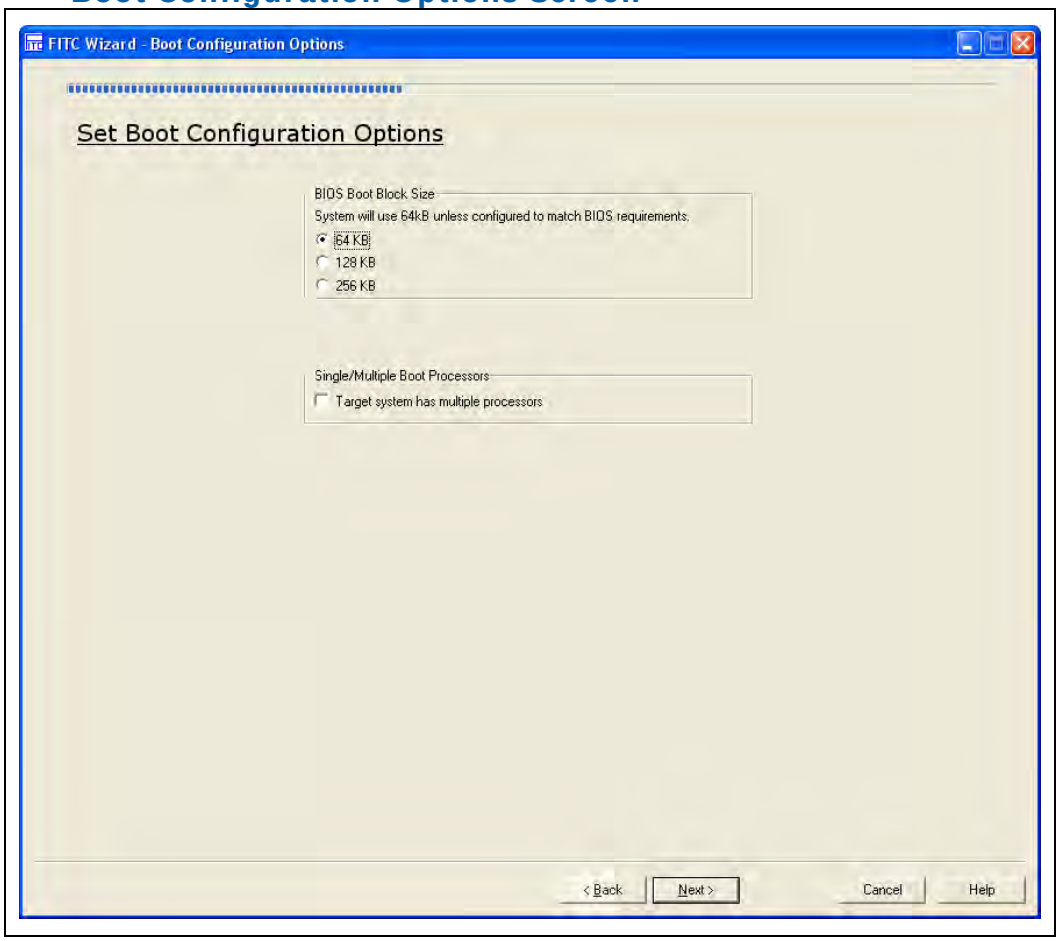


Figure 13: Boot Configuration Options Screen

This screen contains options for configuring the BIOS boot block size parameters and specifing whether or not the target system has multiple processors. Selecting an option enables it and deselecting an option disables it.

3.5.2.14 Integrated Clock Configuration Screen

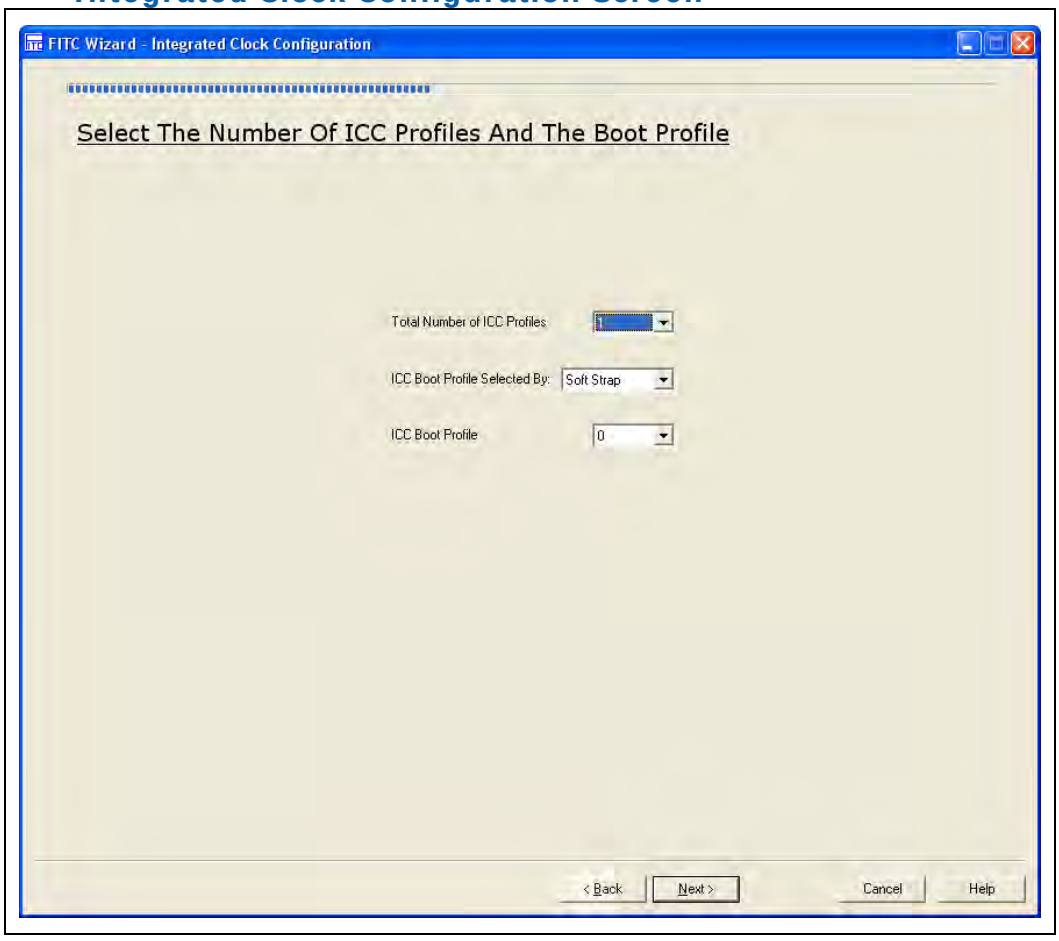


Figure 14: Integrated Clock Configuration Screen - CPT

This screen lets you select the following:

- The number of supported SKUs (1-7). There is an activated OEM (clock) record for each supported SKU.
- The clock record used by FITC.

3.5.2.14.1 Editing OEM Request Record Screens

To edit the clock record:

1. Click the **Edit** button on the Integrated Clock Configuration Screen; the Single-Ended Clocks screen appears.

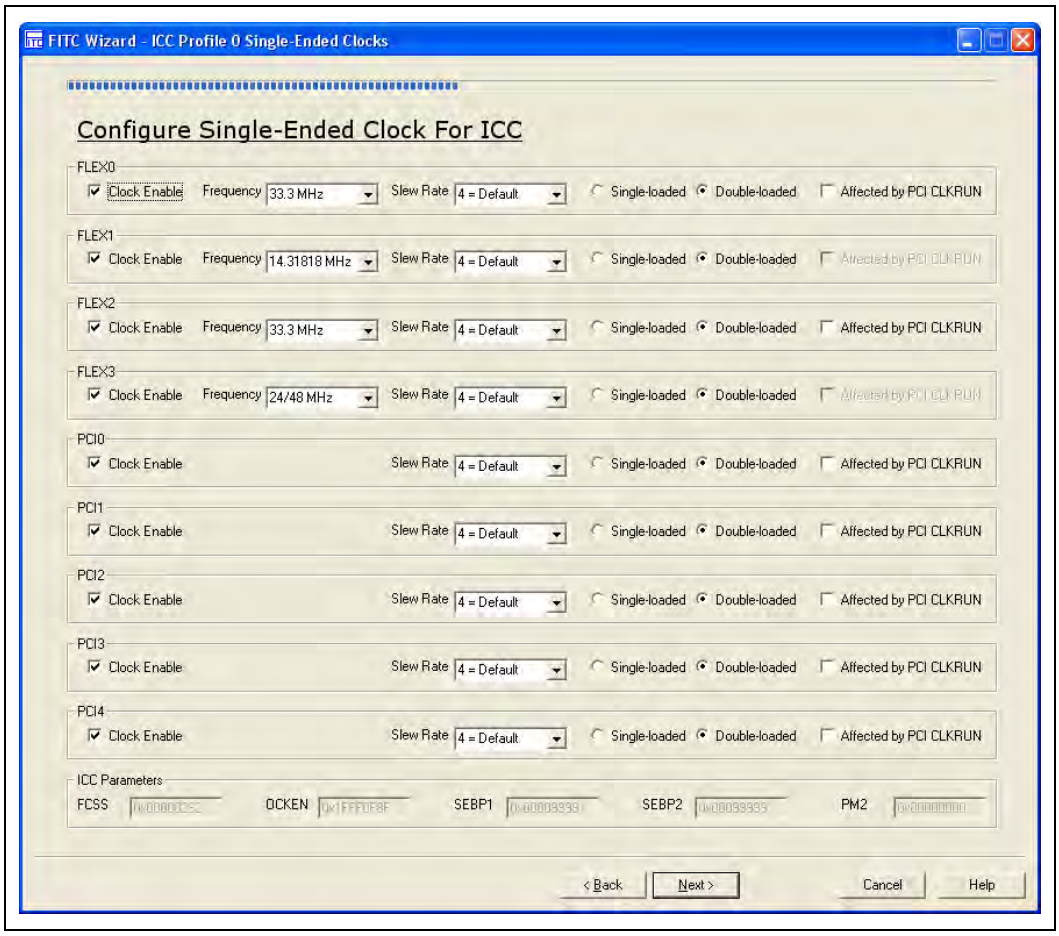


Figure 15: Single-Ended Clocks screen - CPT

2. Set the parameters on the Single-Ended Clocks screen and click **Next**; the Differential Clocks screen appears.

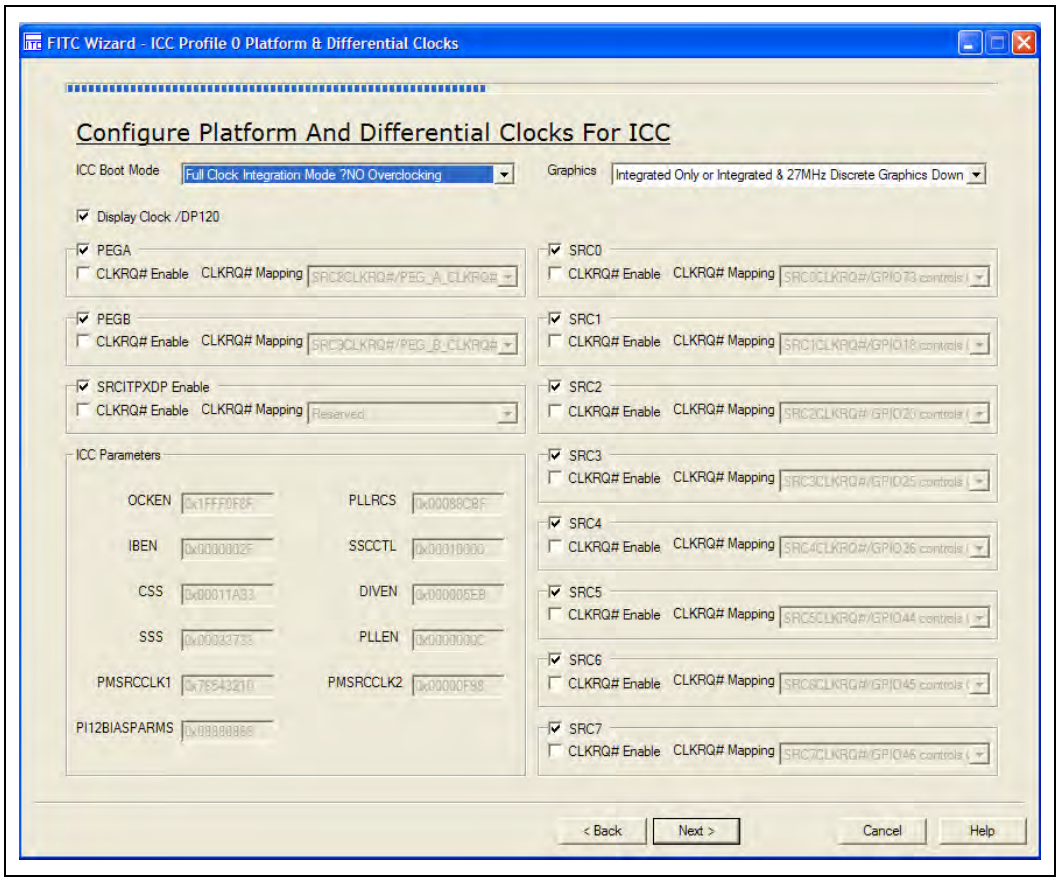


Figure 16: Differential Clocks Screen - CPT

3. Set the parameters on the Differential Clocks screen and click **OK**; the Integrated Clock Configuration Screen reappears with a **Status** of **Changed** for the edited record.

3.5.2.15 Production / Non-production Configuration Screen

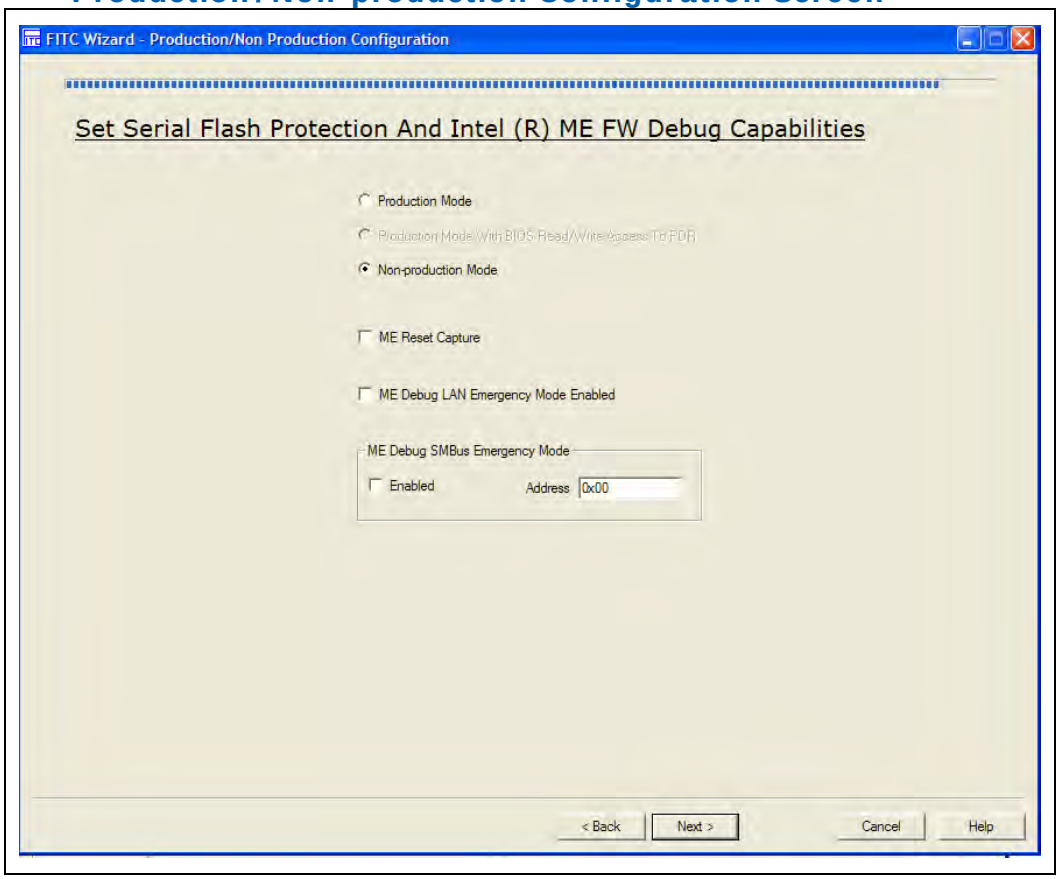


Figure 17: Production / Nonproduction Configuration Screen

This screen lets you specify whether the image being built is a production or non-production image.

3.5.2.16 Build Screen

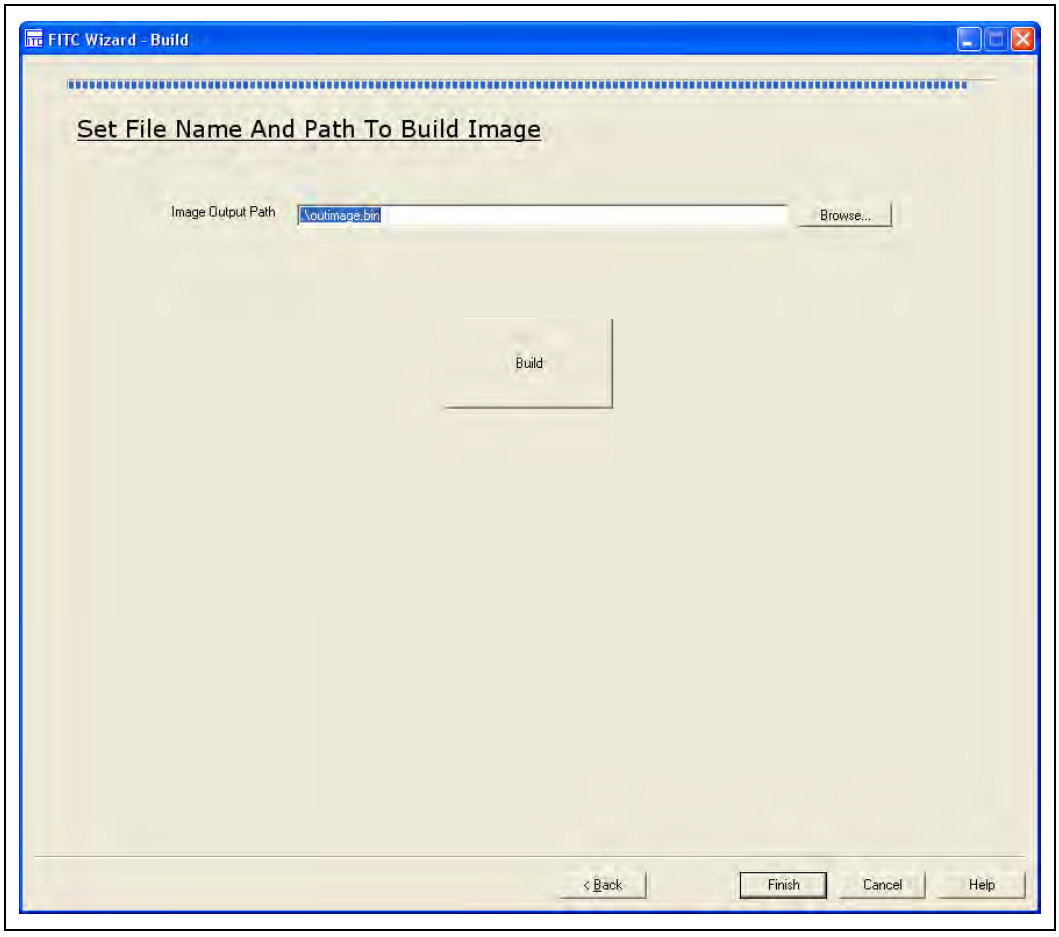


Figure 18: Build Screen

The build screen lets you define the path and name of the final image file and generate that image.

To define the path and name of the final image: Enter the image's name and path into the **Image Output Path** field.

To generate the final image in the output path and name defined in the Image Output Path field: Click **Build**.
(**Note**: FITC – Advanced mode also generates the final image in the defined output path and name when it builds the final image.)

To take the changes made in the Wizard to FITC – Advanced Mode: Click **Finish**.

- Note: pressing finish button without pressing build button will not build the image automatically

To switch back to FITC – Advanced Mode without making the changes selected in the Wizard: Click **Cancel**.

3.6 FITC Advanced Mode

See the following for further information:

- General configuration information – See the FW Bring Up Guide from the appropriate Intel® ME FW kit.
- Detailed information on how to configure PCH Soft Straps and VSCC information – See the Cougar Point SPI programming guide
- More detailed information on Intel® ME FW configuration parameters – See Appendix E.

3.6.1 Configuration Files

The flash image can be configured in many different ways, depending on the target hardware and the required FW options. FITC lets you 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.

3.6.1.1 Creating a New Configuration

FITC provides a default configuration file that you can use to build a new image. This default configuration file can be loaded by clicking **File** > **New**.

3.6.1.2 Opening an Existing Configuration

To open an existing configuration file:

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

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

3.6.1.3 Saving a Configuration

To save the current configuration in an XML file:

Choose **File > Save** or **File > Save As**; the **Save File** dialog appears if the configuration has not been given a name or if **File > Save As** was chosen.

4. Select the path and enter the file name for the configuration.
5. Click **Save**.

3.6.2 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. You can set the environment variables appropriate for your computer, or override the variables with command line options.

It is recommended that the environment variables be the first thing you set when working with a new configuration. This ensures that FITC 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 > Environment Variables**; a dialog appears displaying the current working directory on top, followed by the current values of all the environment variables:
   - $WorkingDir – the directory where the log file is kept and where the components of an image are stored when an image is decomposed.
   - $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 all intermediate files generated during the build.
   - $UserVar1-3 – used when the above variables are not populated.

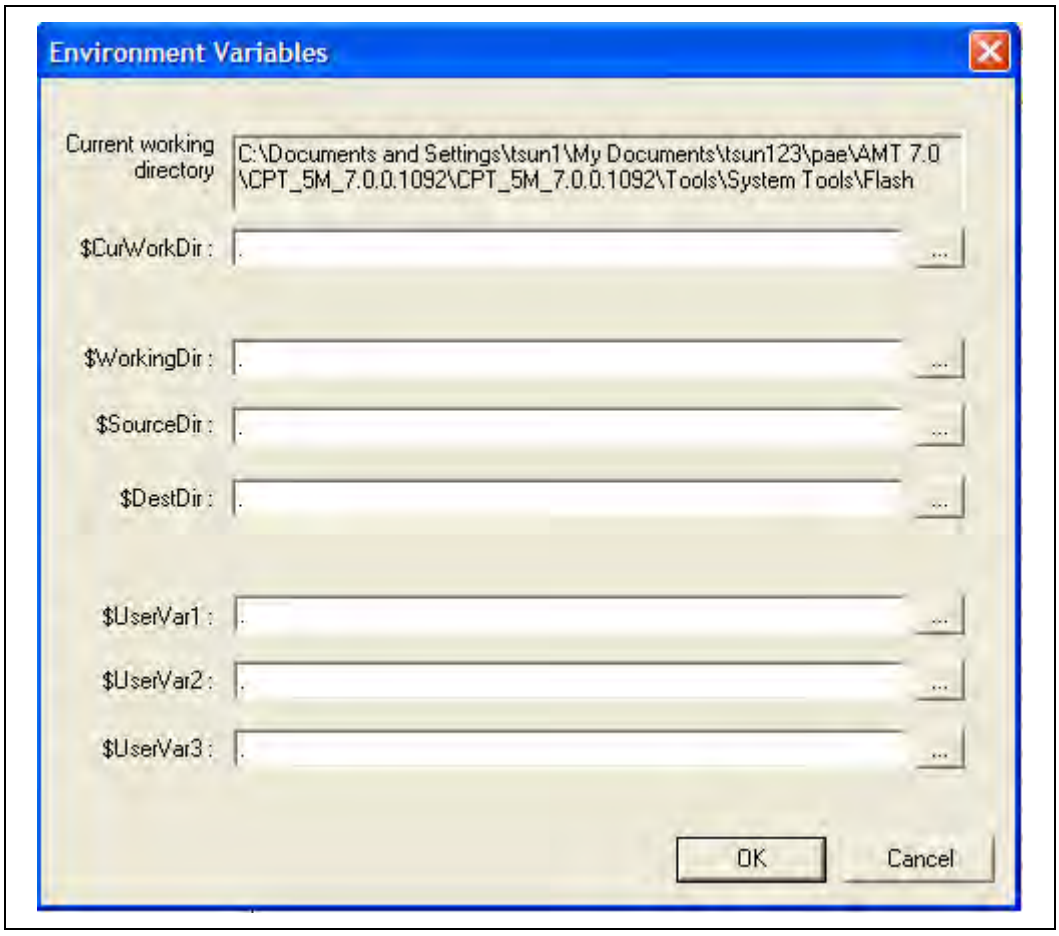


Figure 19. Environment Variables Dialog

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

Note: The environment variables are saved in the application's INI file, not the XML configuration file. This allows the configuration files to be portable across different computers and directory structures.

3.6.3 Build Settings

FITC lets you set several options that control how the image is built. The options that can be modified are described in Table 4.

To modify the build setting:

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

Table 4: Build Settings Dialog Options

Option	Description
Output path	The path and filename where the final image should be saved after it is built. (**Note**: Using the $DestDir environment variable makes the configuration more portable.)
Generate intermediate build files	Causes the application to generate separate (intermediate) binary files for each region, in addition to the final image file (see Figure 3). These files are located in the specified output folder's INT subfolder. These image files can be programmed individually with the FPT.
Build Compact Image	Creates the smallest flash image possible. (By default, the application uses the flash component sizes in the Descriptor to determine the image length.)
Do not set End of Manufacturing bit …	When descriptor permissions are set to production values, do not select the **Do not set End of Manufacturing bit** box unless not closing End of Manufacturing is explicitly desired. Intel strongly recommends that the Global Lock Bit/End of Manufacturing bit be set on all production platforms.
Flash Block/Sector Erase Size	All regions in the flash conform to the 4KB sector erase size. It is critical that this option is set correctly to ensure that the flash regions can be properly updated at runtime.
Assymetric Flash	Lets you specify a different sector erase size for the upper and lower flash block. Only 4KB erase is supported for Intel® ME FW. This option also lets you modify the flash partition boundary address.

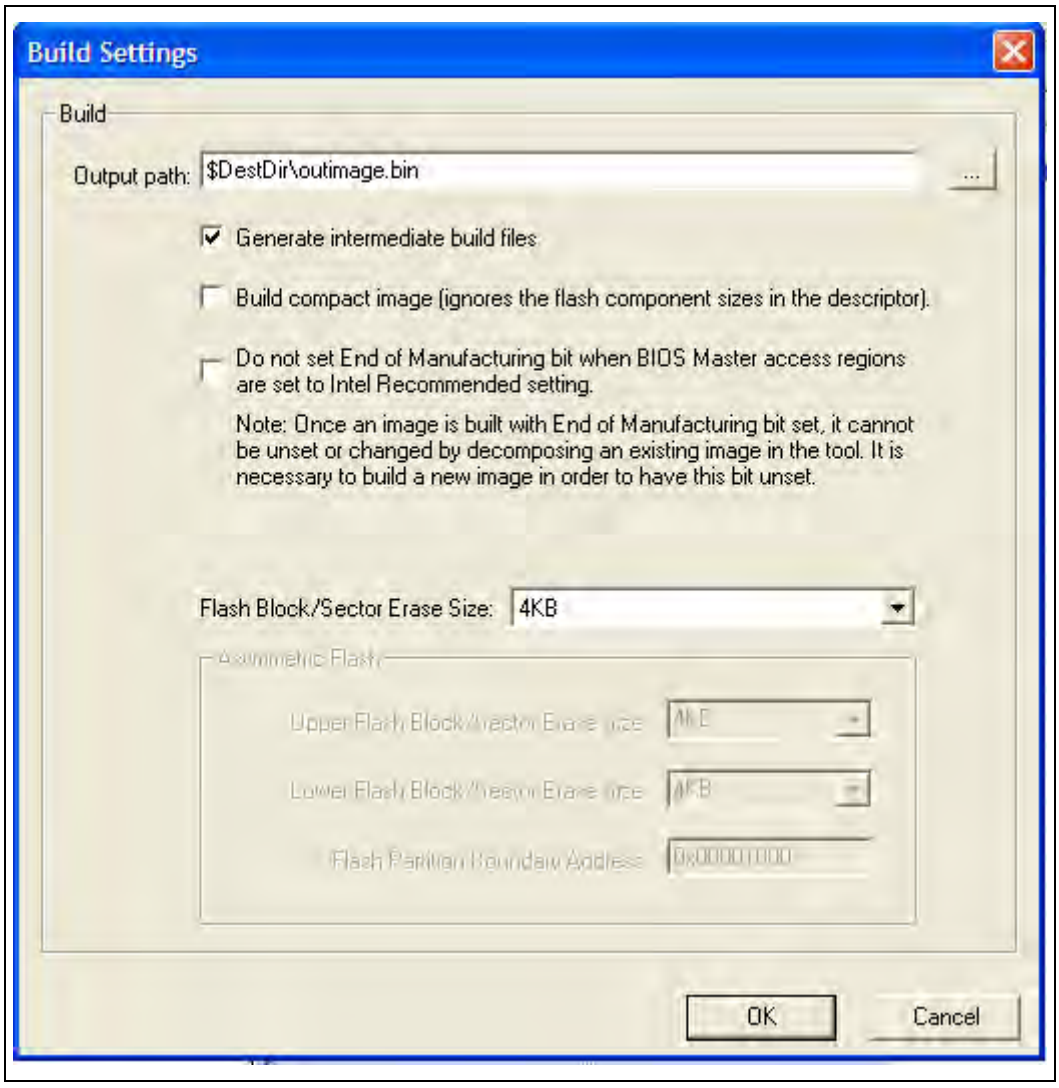


Figure 20. Build Settings Dialog

End of Manufacturing bit is simply a bit in the image. This is not an NVAR, or FOV . In previous generation, when creating an image, you could set the global valid bit automatically based on BIOS being set to production Master Access section, but to allow some customers not to set it, we show this checkbox. There is no bug here at the moment: this checkbox only does something if:

- ME manufacturing done bit is not set, BIOS is not set to production → FITc will not set ME manufacturing done bit – independent of this checkbox
- ME manufacturing done bit is not set, BIOS is set to production, checkbox is unchecked → FITc will set ME manufacturing done bit (new in IBX)
- ME manufacturing done bit is not set, BIOS is set to production, checkbox is checked → FITc will not set ME manufacturing done bit

- ME manufacturing done bit set → will stay set

A dumped image is never reflected in this checkbox – it does not show the actual value of ME manufacturing done bit. It shows what should be done in the next build. But if ME manufacturing done bit is set, this checkbox will never uncheck it.

3.6.3.1 Selecting the Platform SKU

The ability to select the Platform SKU lets you use "Full Featured Engineering samples" to test how the FW behaves like the production Intel® 6 Series Chipset, with the following reservations:

- Certain features only work with particular Chipset SKUs and FW kits (e.g., Intel® AMT only works with corporate SKUs with the 5MB Intel® ME FW kit).
- SKU Manager Selection has no effect on the Production PCH chipset

To select a Platform SKU:

1. Load the Intel® ME region (**Note**: Loading the Intel® ME region first ensures that the proper FW settings are loaded into FITC.
2. Select the appropriate platform type for your specific chipset from the SKU Manager drop-down list; the "Full Featured Engineering Samples" behaves as if it were the selected SKU PCH chipset.

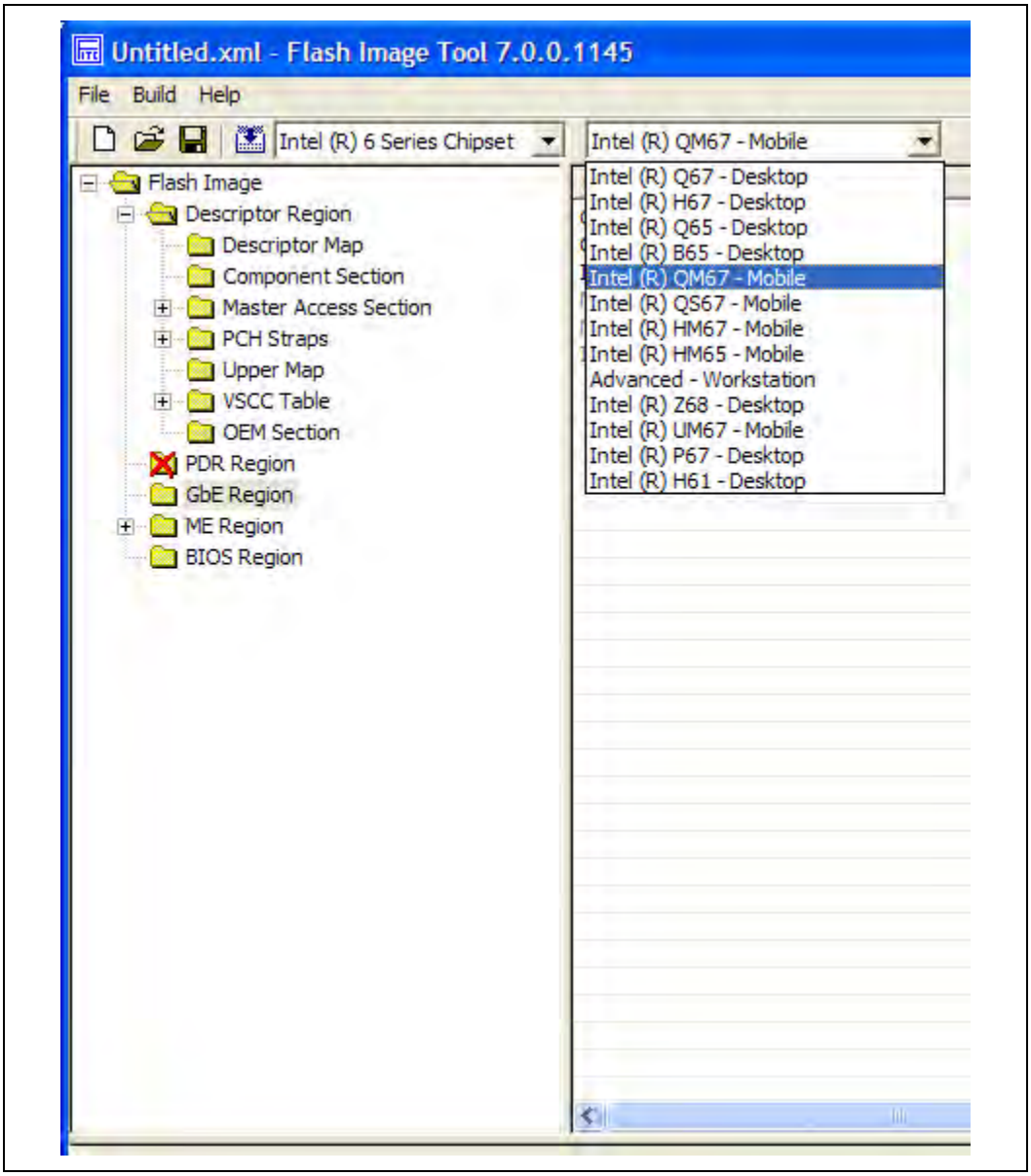


Figure 21: Selected an SKU Platform in FITC

3.6.4 Modifying the Flash Descriptor Region

The FDR 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.

3.6.4.1 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 **Descriptor Region** in the left pane; the **Descriptor Region Length** parameter appears in the right pane.
2. Double-click the **Descriptor Region Length** parameter; the **Descriptor Region Length** dialog appears.
3. Enter any non-zero value into the dialog to set the length of the region and click **OK**.

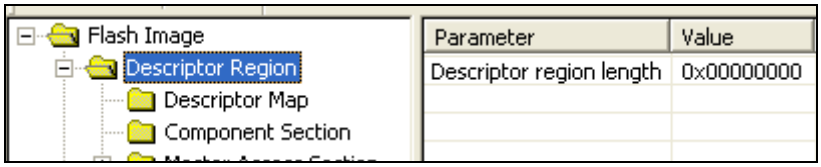


Figure 22. Descriptor Region Length Parameter

3.6.4.2 Setting the Number and Size of the Flash Components

To set the number of flash components:

1. Expand the **Descriptor Region** node of the tree in the left pane.
2. Select **Descriptor Map** (see Figure 23); all the parameters in the Descriptor Map section are listed in the right pane.

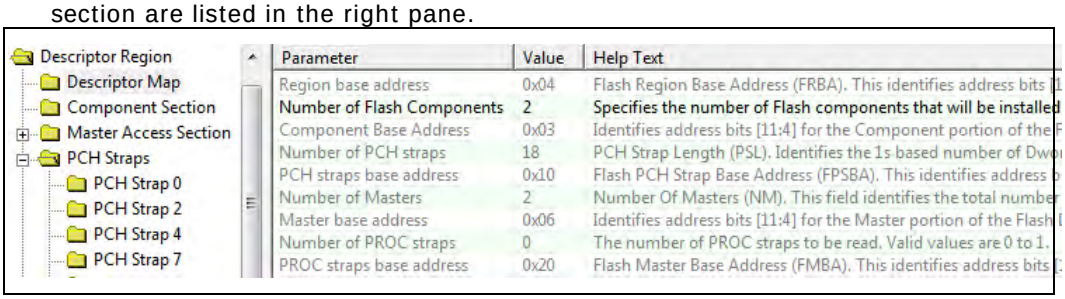


Figure 23: Descriptor Region > Descriptor Map Parameters

3. Double-click **Number of Flash Components** in the right pane (see Figure 24); the Flash Components dialog appears.
4. Enter the number of flash compents (valid values are 1 or 2).
5. Click **OK**; the parameter is updated.

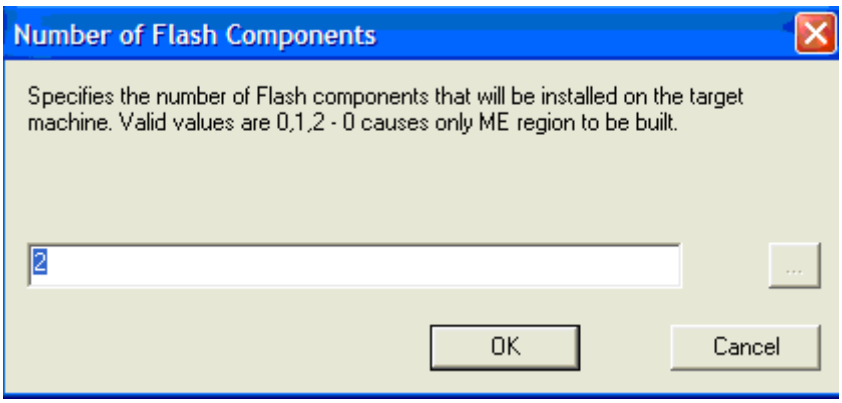
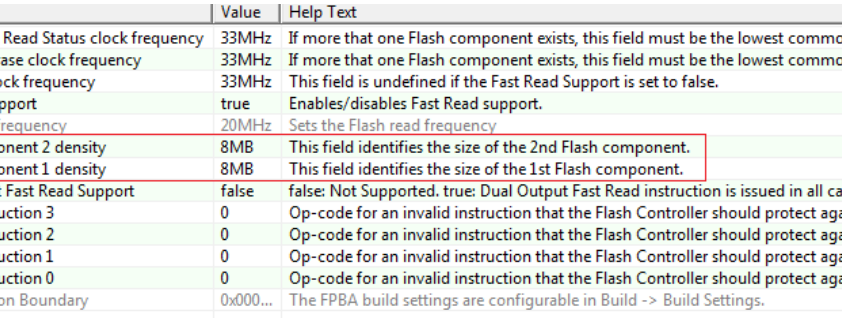


Figure 24: Flash Components Dialog

To set the size of each flash component:

1. Expand **Descriptor Region** node in the left pane and select **Component Section**; the Component Section parameters appear in the right pane. The **Flash component 1 density** and **Flash component 2 density** parameters specify the size of each flash component.
2. Double-click on one of these parameters; a dialog appears.
3. Select the correct component size from the dialog's drop-down list and click **OK**; that parameter is updated.
4. 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.



Parameter	Value	Help Text
Read ID and Read Status clock frequency	33MHz	If more that one Flash component exists, this field must be the lowest common frequency
Write and erase clock frequency	33MHz	If more that one Flash component exists, this field must be the lowest common frequency
Fast read clock frequency	33MHz	This field is undefined if the Fast Read Support is set to false.
Fast read support	true	Enables/disables Fast Read support.
Read clock frequency	20MHz	Sets the Flash read frequency
Flash component 2 density	8MB	This field identifies the size of the 2nd Flash component.
Flash component 1 density	8MB	This field identifies the size of the 1st Flash component.
Dual Output Fast Read Support	false	false: Not Supported. true: Dual Output Fast Read instruction is issued in all cases where th
Invalid Instruction 3	0	Op-code for an invalid instruction that the Flash Controller should protect against chip era
Invalid Instruction 2	0	Op-code for an invalid instruction that the Flash Controller should protect against chip era
Invalid Instruction 1	0	Op-code for an invalid instruction that the Flash Controller should protect against chip era
Invalid Instruction 0	0	Op-code for an invalid instruction that the Flash Controller should protect against chip era
Flash Partition Boundary	0x000...	The FPBA build settings are configurable in Build -> Build Settings.

Figure 25: Descriptor Region > Component Section Parameters

3.6.4.3 Region Access Control

Regions of the flash can be protected from read or write access by setting a protection parameter in the Descriptor Region. The Descriptor Region must be locked before Intel® ME devices are shipped. If the Descriptor Region is not locked, the Intel® ME

device is vulnerable to security attacks. The level of read/write access provided is at the discretion of the OEM/ODM. A cross-reference of access settings is shown below.

Table 5: Region Access Control Table

	Regions that can be accessed				
Region to Grant Access	**PDR**	**Intel® ME**	**GbE**	**BIOS**	**Descriptor**
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
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
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

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

Key:

0 – denied access

1 – allowed access

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

Table 6: CPU/BIOS Access

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

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

Example:

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

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

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

To set these access values in FITC:

1. Select **Descriptor Region > Master Access, Manageability Engine and GBE > CPU/BIOS** in the left pane; the access parameters are listed in the right pane (see Figure 26).
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 lock the SPI in the image creation phase: select the recommended setting for production (e.g., select 0x0D for Intel® ME read access and 0x0C for Intel® ME write access).

Note: If all Read/Write Master access settings for Intel® ME are set to production platform values, then the Intel® ME manufacturing mode done(Global Lock) bit is automatically set. If the Intel® ME manufacturing mode done(Global Lock) bit is set, the FOV mechanism is not available.

Parameter	Value	Help Text
PCI Bus ID	0	
PCI Device ID	0	
PCI Function ID	0	
Read Access	0xFF	0xFF = Debug/Manufacturing, 0x0D = Production. Each bit corresponds to Regions [7:0]. If the bit is set,
Write Access	0xFF	0xFF = Debug/Manufacturing, 0x0C = Production. Each bit corresponds to Regions [7:0]. If the bit is set,

Figure 26: Descriptor Region > Master Access Section

3.6.5 PCH Soft Straps

These sections contain configuration options for the PCH. The number of Soft Strap sections and their functionality differ based on the target PCH. Improper settings could lead to undesirable behavior from the target platform. (For more information on how to set them correctly, see the FW Bringup Guide or the PCH SPI programming guide, Appendix A.)

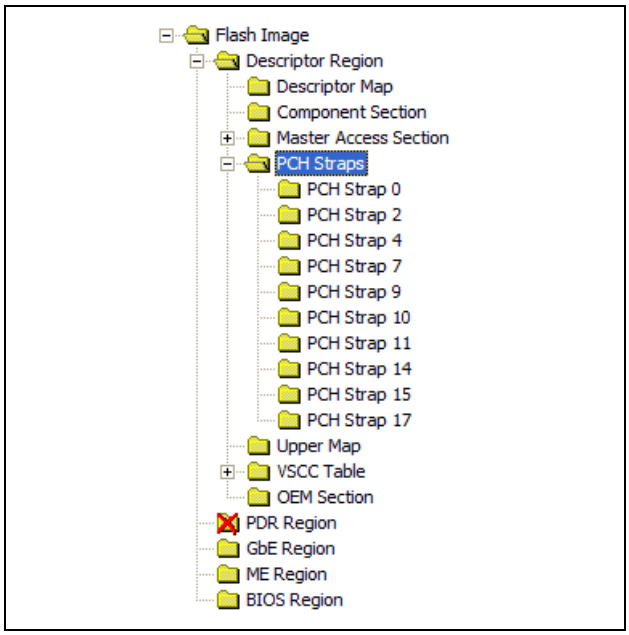


Figure 27PCH Straps

3.6.6 VSCC Table

This section is used to store information to setup flash access for Intel® ME. This does not have any effect on the usage of the FPT. **If the information in this section is incorrect, Intel® ME FW may not communicate with the flash device.** The information provided is dependent on the flash device used on the system. (For more information, see the PCH SPI programming guide, Section 6.4.)

3.6.6.1 Adding a New Table

To add a new table:

1. Right-click on **Descriptor Region > VSCC table**.
2. Choose **Add Table Entry** from the pop-up menu; the **Add Table Entry** dialog appears.

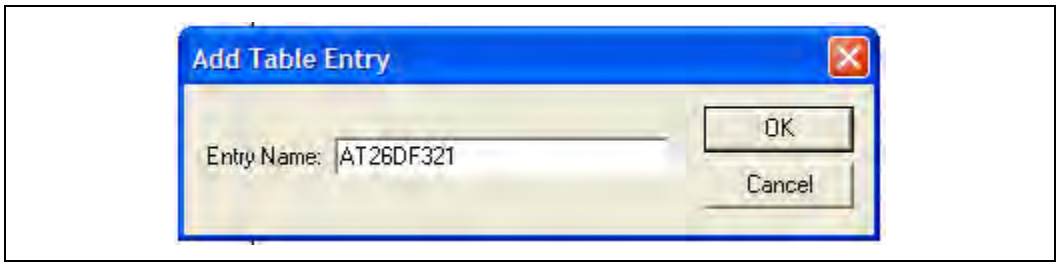
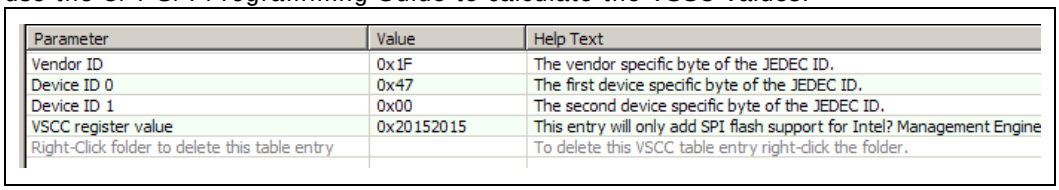


Figure 28: Add VSCC Table Entry Dialog

3. 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 FITC to prevent table entries that have the same name and no error message is displayed in such cases.)
4. Click **OK**; the new table is listed in the left pane under **VSCC Table** and you can enter into it the values for the flash device. (See Figure 29, which shows the parameters of a new VSCC table.)

Note: The values in the VSCC table can be found in the serial flash data sheet. You should use the CPT SPI Programming Guide to calculate the VSSC values.



Parameter	Value	Help Text
Vendor ID	0x1F	The vendor specific byte of the JEDEC ID.
Device ID 0	0x47	The first device specific byte of the JEDEC ID.
Device ID 1	0x00	The second device specific byte of the JEDEC ID.
VSCC register value	0x20152015	This entry will only add SPI flash support for Intel? Management Engine
Right-Click folder to delete this table entry		To delete this VSCC table entry right-click the folder.

Figure 29: Sample VSCC Table Entry

3.6.6.2 Removing an Existing VSCC Table

To remove an existing table:

1. Right-click on the name of the table in the left pane that you want to remove.
2. Choose Remove Table Entry; the table and all the information in it is removed.

3.6.7 Modifying the Intel® ME Region

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

3.6.7.1 Setting the Intel® ME Region Binary File

To select the Intel® ME region binary file:

1. Select the Intel® ME Region tree node.
2. Double-click on the **Binary file parameter** in the list; a dialog appears that lets you 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 iscopied into the Intel® ME Region.

Note: If you specified in the PCH Strap Section (0) that Intel® ME must boot from flash, the loaded FW must contain a ROM Bypass section. If the FW does not contain a ROM bypass section, a section becomes available in which you can enter the location of the ROM bypass file. Loading a exisiting ME FW image may not be able to show all the existing ME settings in the FITC ionterface. Some of the ME setting will display default value instead of the real value stored on the firmware image.

3.6.8 Intel® ME FW Configuration

Intel® ME FW parameters are visible and editable after a valid Intel® ME FW image has been loaded.

If any of the parameters do not have the Intel-recommended value, the offending row is highlighted yellow but no errors are reported. The highlighted yellow is designed to draw attention to these values to ensure these parameters are set correctly.

3.6.8.1 Intel® ME Section

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

The Intel® ME section lets you define the computer's manageability features. The parameter values can be found in the Help Text next to to the parameter value as shown in Figure 30.

Parameter	Value
FW Update OEM ID	00000000-0000-0000-0000-000000000000
LAN Power Well Config	3
WLAN Power Well Config	0x80
M3 Power Rails Availability	true
HECI ME Region Unlockable	true
Sub System Vendor ID	0x0000
PROC_MISSING	No onboard glue logic
Processor Emulation	No Emulation
OEM Tag	0x00000000
Hide FW Update Control	false
Debug Si Features	0x00000000
Prod Si Features	0x00000000

Figure 30: Intel® ME Section

3.6.8.2 Managability Application Section

Note: This section and its sub-sections are not applicable to 1.5MB Intel® ME FW SKU.

This section describes the Manageability Application parameters. (See the FW Bringup guide for general information and see Appendix E for more details.)

The Manageability section lets you 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).

Parameter	Value
Boot into BIOS Setup Capable	false
Pause during BIOS Boot Capable	false
BIOS Reflash Capable	false
USBr EHCI 1 Enabled	11b Enabled
USBr EHCI 2 Enabled	10b Disabled
Host Based Setup and Configuration	true
Privacy Level	Default
Idle Timeout - Manageability Engine	1

Figure 31: Manageability Application Section

3.6.8.3 Power Packages Section

The Power Packages section allows the OEM/ODM to specify which power packages are supported. (See the FW Bringup guide for general information and see Appendix E for more details.)

Parameter	Value
Power Pkg 2 Supported (Mobile: ON in S0, M...	true
Default Power Package	1

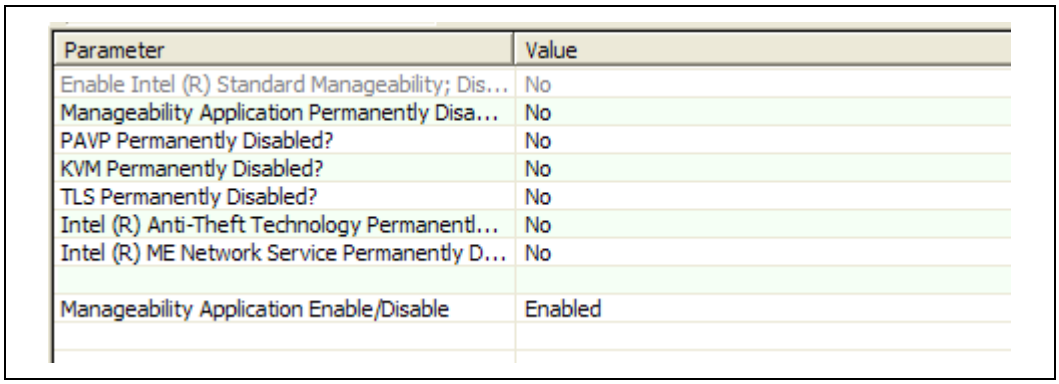
Figure 32: Power Packages Section

If the Power Package Supported value is set to false, that specific power package cannot be selected and is not visible to the end user.

The selected Default Power Package must be supported. This is the value that is selected when the system is shipped. This value affects energy star compliance it is if not set correctly.

3.6.8.4 Features Supported

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



Parameter	Value
Enable Intel (R) Standard Manageability; Dis...	No
Manageability Application Permanently Disa...	No
PAVP Permanently Disabled?	No
KVM Permanently Disabled?	No
TLS Permanently Disabled?	No
Intel (R) Anti-Theft Technology Permanentl...	No
Intel (R) ME Network Service Permanently D...	No
Manageability Application Enable/Disable	Enabled

Figure 33: Features Supported Section

These options control the availability and visibility of FW features.

In cases where a specific feature is configurable in the Intel® MEBx, permanently disabling it through the **Features Supported** section hides/disables that feature in Intel® MEBx.

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: The Intel® Manageability Application setting combines several manageability technologies that are related to each other. This setting controls the following manageability technologies:

- Intel® Active Management Technology
- Intel® Standard Management
- Fast Call for Help
- Intel® KVM Remote Assistance Application

Setting **Intel® Manageability Application Permanently Disabled?** to "Yes" permanently disables all the features listed above without any way to enable them at a later time. The only way to re-enable these features is to completely re-burn the Intel® ME region with this setting set to "No". A FW update using ***FWUpdLcl.exe*** cannot re-enable features.

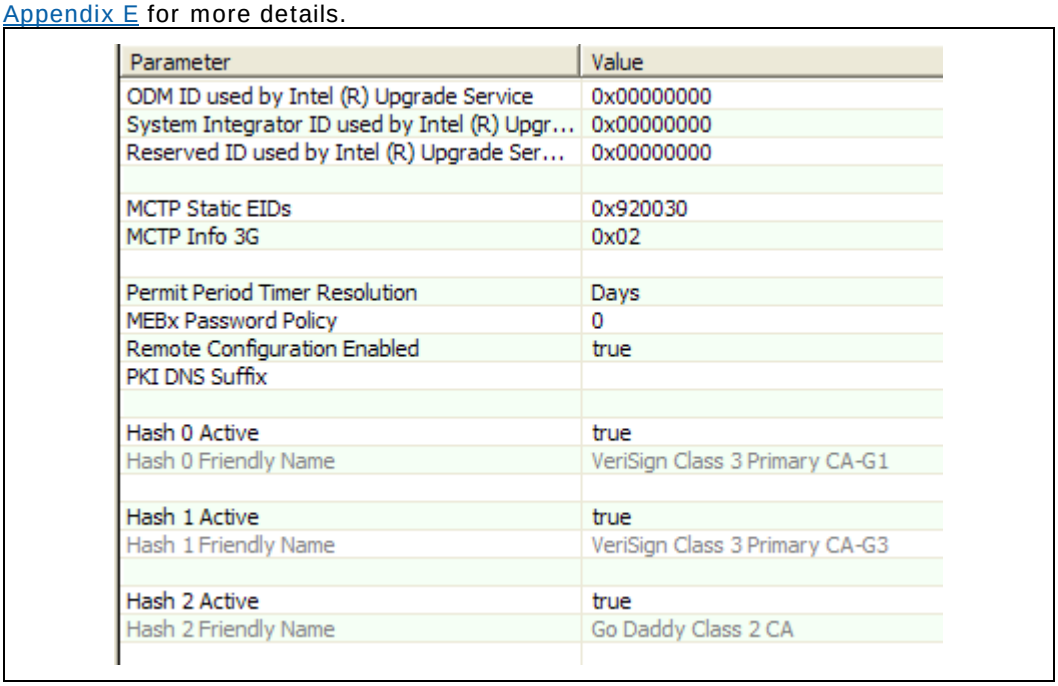
All parameters in this section are color-coded as per the key below.

The parameter can be changed.
The parameter is read only and cannot be changed.

Table 7: Feature Default Settings by SKU

3.6.8.5 Setup and Configuration Section

The Setup and Configuration section allows the end user to specify the configuration settings. These values determine the mode of the Intel® AMT device after the system has been configured. (See the FW Bringup guide for general information and see Appendix E for more details.

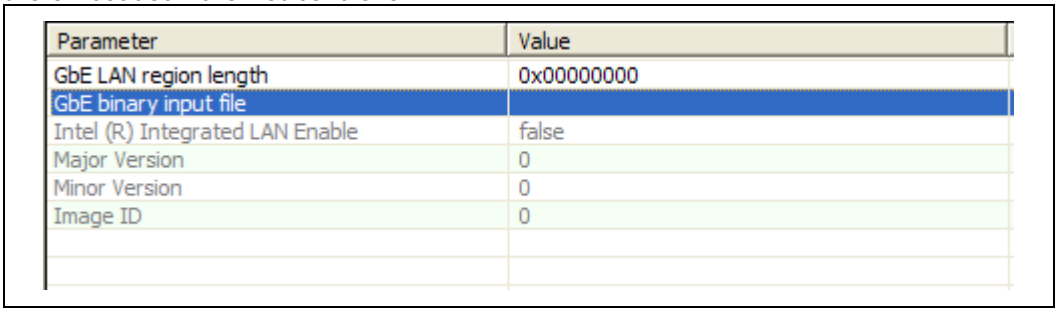


Parameter	Value
ODM ID used by Intel (R) Upgrade Service	0x00000000
System Integrator ID used by Intel (R) Upgr...	0x00000000
Reserved ID used by Intel (R) Upgrade Ser...	0x00000000
MCTP Static EIDs	0x920030
MCTP Info 3G	0x02
Permit Period Timer Resolution	Days
MEBx Password Policy	0
Remote Configuration Enabled	true
PKI DNS Suffix	
Hash 0 Active	true
Hash 0 Friendly Name	VeriSign Class 3 Primary CA-G1
Hash 1 Active	true
Hash 1 Friendly Name	VeriSign Class 3 Primary CA-G3
Hash 2 Active	true
Hash 2 Friendly Name	Go Daddy Class 2 CA

Figure 34: Setup and Configuration Section

3.6.9 Modifying the GbE (LAN) Region

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



Parameter	Value
GbE LAN region length	0x00000000
GbE binary input file	
Intel (R) Integrated LAN Enable	false
Major Version	0
Minor Version	0
Image ID	0

Figure 35: GbE Region Options

3.6.9.1 Setting the GbE Region Length Option

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

3.6.9.2 Setting the GbE Region Binary File

To select the GbE Region binary file:

1. Select **GbE Region** in the left pane; the GbE Region parameters are listed in the right pane.
2. Double-click on the **Binary input file** parameter; a dialog appears that lets you select the GbE file to use.
3. Select a file.
4. Click **OK** to update the parameter; when the flash image is built, the contents of this file is copied into the GbE Region.

3.6.9.3 Enabling / Disabling the GbE Region

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

To disable the GbE Region:

1. Right-click on **GbE Region** in the left pane.
2. Choose **Disable Region** from the pop-up menu; when the flash image is built it will not contain a GbE Region.

To enable the GbE Region:

1. Right-click on **GbE Region** in the left pane.
2. Choose **Enable Region** from the pop-up menu.

3.6.10 Modifying the PDR Region

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

Parameter	Value	Help Text
PDR region length	0x00000000	This is the size of the PDR region in bytes. Set this to zero and sp
Binary input file		This is the PDR image binary that will be copied into this region.

Figure 36: PDR Region Options

3.6.10.1 Setting the PDR Region Length Option

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

3.6.10.2 Setting the PDR Region Binary File

To select the PDR region binary file:

1. Select **PDR Region** in the left pane; the PDR Region parameters are listed in the right pane.
2. Double-click the **Binary input file** parameter; a dialog appears that lets you specify which PDR file to use.
3. Click **OK** to update the parameter; when the flash image is built, the contents of this file is copied into the BIOS region.

3.6.10.3 Enabling / Disabling the PDR Region

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

To disable the PDR Region:

1. Right-click on **PDR Region** in the left pane.
2. Choose **Disable Region** from the pop-up 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. Right-click on **PDR Region** in the left pane.
2. Choose **Enable Region** from the pop-up menu.

3.6.11 Modifying the BIOS Region

The BIOS Region contains the BIOS code run by the host processor. FITC always aligns this region with the end of the flash image. This is done so that if the flash descriptor becomes corrupt for any reason, the PCH defaults to legacy mode and looks for the reset at the end of the flash memory. 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.

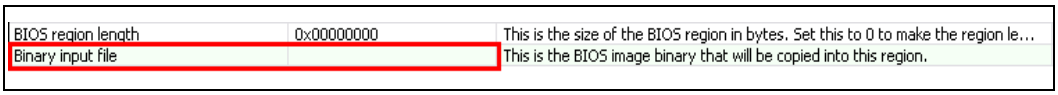


Figure 37: BIOS Region Parameters

3.6.11.1 Setting the BIOS Region Length Parameter

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

3.6.11.2 Setting the BIOS Region Binary File

To select the BIOS region binary file:

1. Select **BIOS Region** in the left pane; the BIOS Region parameters are listed in the right pane.
2. Double-click the **Binary input file** parameter; a dialog appears that lets you specify which BIOS file to use.
3. Click **OK** to update the parameter; when the flash image is built, the contents of this file are copied into the BIOS region.

3.6.11.3 Enabling/Disabling the BIOS Region

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

To disable the BIOS Region:

1. Right-click on **BIOS Region** in the left pane.
2. Choose **Disable Region** from the pop-up menu; when the flash image is built, there is no BIOS Region in it.

To enable the PDR Region:

1. Right-click on **BIOS Region** in the left pane.
2. Select **Enable Region** from the pop-up menu.

3.6.12 Building a Flash Image

The flash image can be built with the FITC 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.

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

- Binary file representing the image
- Text file detailing the various regions in the image
- Optional set of intermediate files (see Section 3.6.3).
- 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, you should select the single larger binary file when using FPT.

3.6.13 Change the Region Order on the SPI Device

The order and placement of the regions in the full SPI image created by FITC can be altered. The location of each region is determined by the order of the regions as they are displayed in left pane of the FITC window.

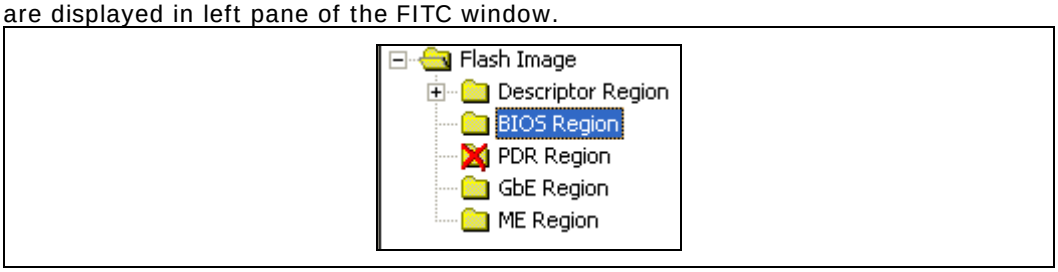


Figure 38: Region Order

Each region is added to the full SPI image in the order in which they appear in the list. The order of the regions in the full SPI image created from the regions listed in Figure 38 is as follows:

1. Descriptor Region
2. BIOS Region
3. GbE Region
4. ME Region.

This can be useful when programming a system with two SPI devices. It is possible to change the order of the regions by clicking and dragging the region to the required location. Figure 24 shows that the BIOS is placed on the first SPI device and the Intel® ME Region is placed on the second SPI device. The length of each region and the order determines if that region is on the first or second SPI device.

3.6.14 Decomposing an Existing Flash Image

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

To decompose an image:

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

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

3.7 Command Line Interface

FITC supports command line options.

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

The command line syntax for FITC is:

```
FITC [<XML_file>] [<BIN File>] [-?] [-H] [-B] [-O <file>] [-ROMBYPASS
<true|false>] [-ME <file>] [-GBE <file>] [-BIOS <file>]
[-PDR <file>] [-W <path>] [-S <path>] [-D <path>] [-U1 <value>] [-U2
<value>] [-U3 <value>] [-I <enable|disable>] [-FLASHCOUNT <1|2>] [-
FLASHSIZE1 <0|1|2|3|4|5>] [-FLASHSIZE2 <0|1|2|3|4|5>] [ -SKU <value>]
[PLATFORM <Value>]
```

Table 8: FITC Command Line Options

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

Option	Description
-D <path>	Overrides the destination directory environment variable $DestDir. It is recommended that you set these environmental variables before starting a project.
-U1 <value>	Overrides the $UserVar1 environment variable with the value specified. Can be any value required.
-U2 <value>	Overrides the $UserVar2 environment variable with the value specified. Can be any value required.
-U3 <value>	Overrides the $UserVar3 environment variable with the value specified. Can be any value required.
-FLASHCOUNT <0, 1 or 2>	Overrides the number of flash components in the Descriptor Region. If this value is zero, only the Intel® ME Region is built.
-FLASHSIZE1 <0, 1, 2, 3, 4 or 5>	Overrides the size of the first flash component with the size of the option selected as follows: • 0 = 512KB • 1 = 1MB • 2 = 2MB • 3 = 4MB • 4 = 8MB • 5 = 16MB.
-FLASHSIZE2 <0, 1, 2, 3, 4 or 5>	Overrides the size of the first flash component with the size of the option selected as follows: • 0 = 512KB • 1 = 1MB • 2 = 2MB • 3 = 4MB • 4 = 8MB • 5 = 16MB.
-Platform <value>	This option is used to change the platform you are building for. In CPT there is only one supported which is "6Series" (e.g., `/platform 6Series`).
-SKU <value>	This option is used to change the SKU you are building for. Use the words Q67, QM67, etc. as a reference to a SKU from the drop-down menu (e.g., `/sku Q67`).

3.8 Example – Decomposing an Image and Extracting Parameters

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

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

```
Fitc.exe output.bin /b
```

This command creates a folder named "output". The folder contains the individual regions (Descriptor, GBE, Intel® ME, BIOS) and the Map file (**<FILENAIntel® ME>.MAP**).

The xml file contains the current Intel® ME parameters.

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

3.9 More Examples of FITC 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:

```
Fitc.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:

```
Fitc.exe /b /me ".\..\..\Image Components\Firmware\PCH_REL_BYP_ME_UPD_PreProduction_0xB0.BIN" <file.bin or file.xml>
```

Take an existing image and put in a different Intel® ME region:

```
Fitc.exe /b /gbe ".\..\..\Image Components\GbE\82577_A2_CPT_A1_VER0PT21_MOBILE.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. You 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 fixed offset variables.

4.1 System Requirements

The DOS version of FPT (**fpt.exe**) runs on MS DOS 6.22, DRMKDOS, and FreeDOS.

The Windows version (**fptw.exe**) requires administrator privileges to run under Windows OS. You must use the **Run as Administrator** option to open the CLI in Windows* Vista 64/32 bit and Windows* 7 64/32 bit.

The Windows 64 bit version (fptw64.exe) is designed for running in ture 64 bit OS environment which does not have 32 bit compatible mode available for example WinPE 64.

FPT requires an operating system to run on. It is designed to deliver a custom image to a computer that is already able to boot and is not a means to get a blank system up and running. FPT must be run on the system with the flash memory that you are programming.

One possible workflow for using FPT is:

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

4.2 Flash Image Details

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

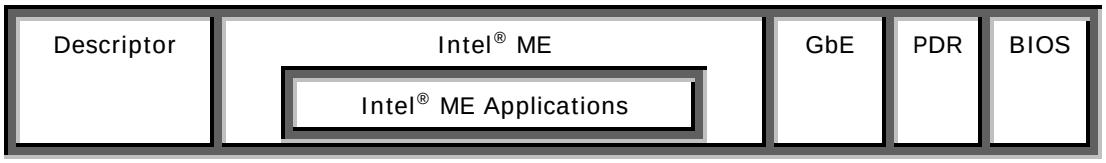


Figure 39: Flash Image Regions

Table 9: Flash Image Regions – Description

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

4.3 Microsoft Windows Required Files

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

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

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 *.inf` to load it into the running system each time Windows* PE reboots. Failure to do so causes errors for some features.

4.4 DOS Required Files

The DOS version of the FPT main executable is **fpt.exe**. The following files must be in the same directory as **fpt.exe**:

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

4.5 Programming the Flash Device

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

Note: Programming the flash device while Intel® ME is running may cause the flash device to become corrupted. Intel® ME SPI accessing should be stopped for any flash accessing before programming the full flash device.

This is **not** a requirement when writing to the fixed offset region.

4.5.1 Stopping Intel® ME SPI Operations

Intel® ME SPI Operations can be stopped in the following ways:

- Assert HDA_SDO (known as GPIO 33 or Flash descriptor override/Intel® ME manufacturing jumper) to high while powering on the system. This is not a valid method if the parameters are configured to ignore this jumper.
- Send the HMRFPO ENABLE Intel® MEI command to Intel® ME (for more information see the PCH Intel® ME BIOS writer's guide).

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

FPT will stop ME SPI accessing automatically if it try to write data into the ME region. Customers do not have to do extra step to stop ME if using FPT tool to update ME region.

4.6 Programming Fixed Offset Variables

FPT can program the fixed offset variables 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. The fixed offset variables can be continuously changed until the Intel® ME manufacturing mode done(***globallocked)*** bit is set to 0x01. The parameters can **NOT** be modified after this bit is set. To modify the default settings for the parameters, the entire flash device must be re-programmed.

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

Table 10: Fixed Offset Variables Options

Option	Description
fpt.exe –FOVs	Displays a list of the supported variables.
fpt.exe –EX -O <Text File>	Creates an empty text file that lets you update multiple fixed offset variables. The variables have the following format in the text file: [Parameter name] Enabled=0xff Value = In the created text file: • Variables that are NOT enabled (enabled=0xff) are not modified. • Only variables that ARE enabled (enabled=0x1) are modified.
fpt.exe –U –IN <Text file>	Updates the fixed offset variables with the values as they are entered in the text file.

See Appendix A for a description of all the Fixed Offset Variable parameters.

Note: If FOV is set before the first system boot, the Intel® ME setting takes the FOV changes and disregards the settings set in FITC.

4.7 Usage

Both the Windows and DOS versions of the FPT can run with command line options.

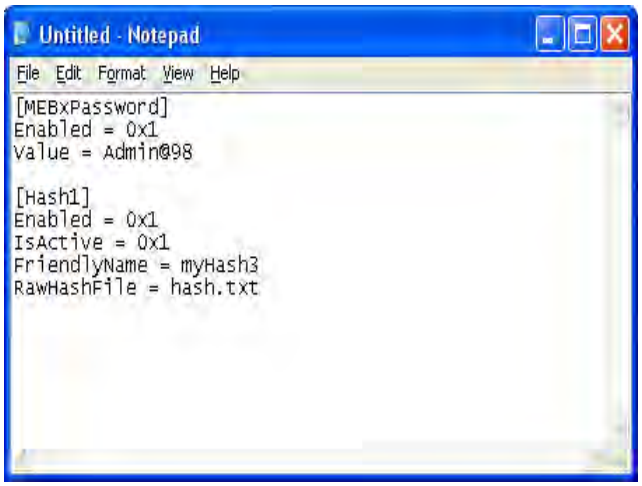
To view all of the supported commands: Run the application with the -? option.

The commands in both the DOS and Windows versions have the same syntax. The command line syntax for **fpt.exe** and **fptw.exe** is:

```
FPT.exe [-H|?] [-VER] [-EXP] [-VERBOSE] [-Y] [-P] [-LIST] [-I] [-F]
    [-ERASE] [-VERIFY] [-D] [-DESC] [-BIOS] [-ME] [-GBE] [-PDR] [-C] [-B]
    [-E] [-ADDRESS|A] [-LENGTH|L] [-FOVS] [-EX] [-U] [-O] [-IN] [-N] [-
    ID] [-V] [-LOCK] [-DUMPLOCK] [-PSKFILE] [-CLOSEMNF <No>] [-GRESET
    <NO>] [-SPIBAR] [-R] [-VARS] [-COMMIT] [-COMPARE] [-HASHED]
```

Table 11: Command Line Options for fpt.exe and fptw.exe

Option	Description
Help (-H, -?)	Displays the list of command line options supported by FPT tool.
-VER	Shows the version of the tools.
-EXP	Shows examples of how to use the tools.
-VERBOSE <file>	Displays the tool's debug information or stores it in a log file.
-Y:	Bypasses Prompt. FPT does not prompt user for input. This confirmation is automatically be answered with "y".
-P <file>:	Flash parts file. Specifies the alternate flash definition file which contains the flash parts description that FPT has to read. By default, FPT reads the flash parts definitions from **fparts.txt**.
-LIST:	Supported Flash Parts. Displays all supported flash parts. This option reads the contents of the flash parts definition file and displays the contents on the screen.
-I:	Info. Displays information about the image currently used in the flash.
-F <file>	Flash. Programs a binary file into an SPI flash. You must specify the binary file to be flashed. FPT reads the binary, erases the flash, 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 a image size.
-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 can not 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 you specify this flag.
-D <file> :	Dump. Reads the SPI flash and dumps the flash contents to a file or to the screen using the STDOUT option. The flash device must be written in 4KB sections. The total size of the flash device must also be in increments of 4KB.
-DESC:	Read/Write Descriptor region. Specifies that the Descriptor region is to be read, written, or verified. The start address is the beginning of the region.
-BIOS:	Read/Write BIOS region. Specifies that the BIOS region is to be read, written, or verified. Start address is the beginning of the region.
-Intel® ME:	Read/Write Intel® ME region. Specifies that the Intel® ME region is to be read, written, or verified. The start address is the beginning of the region.
-GBE:	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.
-C:	Chip erase. Erases the contents of SPI flash device(s). This function does NOT erase block by block.
-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.

Option	Description
-E:	Skip Erase. Does not erase blocks before writing. This option skips the erase operation before writing and must be used whenever you write to a blank flash device.
-A<value>, -ADDRESS <value>	Write/Read Address. Specifies the start address at which a read, verify, or write operation must be performed. You must 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. You must provide the length. This option is not used when providing a region since the region/file length determines this.
-FOVS:	Supported Fixed Offset Variables. Displays all supported FOVs supported by FPT. This option displays names and IDs of supported FOVs.
-EX:	Extract. Extracts the FOVs from flash. The names and values of FOVs are dumped into a text file. You must provide the text file to be used with the `-o <file>` parameter.
-U:	Update. Updates the FOVs in the flash. You can update the multiple FOVs by specifying their names and values in the parameter file. The parameter file must be in an INI file format (the same format generated by the –ex command). The `-in <file>` option is used to specify the input file.
-O <file>	Output File. The file used by FPT to output FOV information.
IN <file>	Input File. The file used by FPT for FOV input. This option flag must be followed by a text file (i.e., `fpt –u –in fov.txt`). The tool updates the FOVs contained in the text file with the values provided in the input file.  **Figure 40: FPT Sample Input File** User can also use FPT –ex –o <filename> to generate this file.
N <value>	Name. Specifies the name of the FOV that you want to update in the image file or flash. The name flag must be used with Value (-v).

Option	Description
-ID <value>	ID. The names of certain FOVs are quite lengthy. This option lets you to update the FOV by providing its unique identification number instead of its name. The ID for each FOV is specified in the configuration file.
-V <value>	Value. Specifies the value for the FOV variable. The name of variable is specified in the Name flag. The Value flag must follow the Name flag.
-LOCK:	Region Lock. Sets the SPI flash region access to the Intel recommended values (see Table 12)
-DUMPLOCK:	Dump Lock Settings. Displays the current lock settings on the screen. The lock settings are read from the descriptor region.
-PSKFILE <file>	PID/PPS/Password pair file. Specifies the input file that contains the one or more PID/PPS/Password key value pairs. This option is used to update the PID, PPS, and Password FOVs whose values are read from the input file. This option only support version 1 FiletypeHeader UUID
-CLOSEMNF <no> :	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. You 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. **Note**: Running `FPT-closemnf` also sets the default value for any unprovisioning process. Run `FPT –closemnf` first if you want 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.
-GRESET <NO> :	Global Reset. FPT performs a global reset. On mobile platforms this includes driving GPIO30 low. Mobile platforms require a SUS Well power-down acknowledge-driven low before the global reset occurs or the platform may not boot up from the reset. The "NO" afterwards disables the driving of GPIO30 for mobile SKUs.
-SPIBAR:	Display SPI BAR. FPT uses this option to display the SPI BAR.
-R <name>	NVAR Read . FPT uses this option to read a variable stored as a NVAR in the FW. 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.

Option	Description
-VARS:	Display Supported Variables. FPT uses this option to display all variables supported for the -R and -COMPARE commands.
-COMMIT:	Commit. FPT uses this option to commit FOVs 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.
-COMPARE <file>	NVAR Compare. FPT uses this option to compare a NVAR with the expected value filled in a text file. The compare entry should have the following format: "<name>" = <value> **Note**: <value> should have the form "xx ", where xx is a hexadecimal value. Each byte must be separated by a space and start with the least significant followed by the next significant byte.
-HASHED:	Hash Variable Output. FPT uses this option to distinguish whether the displayed output is hashed by the FW. For variables that can only be returned in hashed form (such as the Intel® MEBx password), this option has no effect – the data displayed is hashed regardless.

Table 12: Intel-Recommend Access Settings

	Intel® ME	GbE	BIOS
Read	0b 0000 1101 = 0x0d	0b 0000 1000 = 0x08	0b 0000 0011 = 0x0B
Write	0b 0000 1100 = 0x0c	0b 0000 1000 = 0x08	0b 0000 0010 = 0x0A

4.8 Updating Hash Certificate through FOV

Note: This section is not applicable for 1.5MB Intel® ME FW SKU.

There are 33 certificate hash values that can be stored in the Intel® ME region:

- 0-19, 23-32 are default certificates which are not deleted by the full un-provisioning process (caused by Intel® MEBX, RTC reset, or an application).
- Certificates 20-22 are not default certificates and are deleted after a full un-provisioning.
- Certificates 20-22 are configurable by FOV (with FPT or other flash programming methods) or FITC.
- Certificates 0-19 23-32 are not configurable by any tool.

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

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

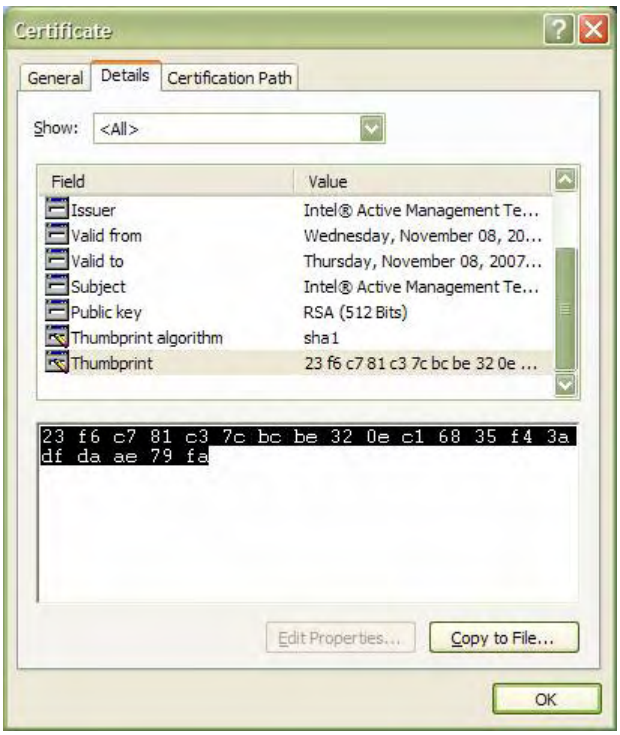


Figure 41: Raw Hash Values from Certificate File

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

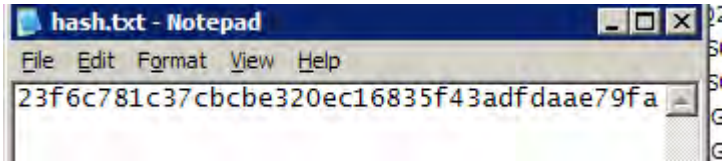


Figure 42: Sample Hash.txt File

4. Save the text file as **hash.txt** in the folder where **sampleparam.txt** is located.
5. Add the following to **sampleparam.txt** in order to update the FOV:

```
[ZTCEnable]
Enabled = 0x0
Value = 0x00
[Hash1]
Enabled = 0x1
IsActive = 0x1
FriendlyName = myHash3
RawHashFile = hash.txt
```

6. Flash Hash FOV with FPT's `-u –in` option (e.g., `fpt –u –in sampleparam.txt`).

Note: **Sampleparam.txt** is the file that is used to update multiple FOVs (`fpt.exe /ex /o sampleparam.txt`).

4.9 fparts.txt File

The **fparts.txt** file contains a list of all flash devices that are supported by FPT. The flash devices listed in this file must contain a 4KB erase block size. If the flash device is not listed, you receive the following error:

```
Flash Programming Tool.  Version X.X.X
Reading LPC BC register... 0x00000000
BIOS space write protection is enabled
Disabling BIOS space write protection
Reading LPC RCBA register... 0xFED1C001
SPI register base address... 0xFED1F020
Loading the flash definition file
Reading file "fparts.txt" into memory...
Initializing SPI utilities
Reading HSFSTS register... Flash Descriptor: Valid
        --- Flash Devices Found ---
>>> Error: There is no supported SPI flash device installed!
```

If the device is not located in **fparts.txt**, you are expected to provide information about the device, inserting the values into **fparts.txt** in same format as is used for the rest of the devices. Detailed information on how to derive the values in **fparts.txt** is found in the Intel® 6 Series Chipset SPI Programming Guide. The device must have a 4KB erase sector and the total size of the SPI Flash device must be a multiple of 4KB. The values are listed in columns in the following order:

- Display name
- Device ID (2 or 3 bytes)
- Device Size (in bits)
- Block Erase Size (in bytes - 256, 4K, 64K)
- Block Erase Command
- Write Granularity (1 or 64)
- Unused
- Chip Erase Command.

4.10 Examples

The following examples illustrate the usage of the DOS version of the tool (**fpt.exe)**. The Windows version of the tool (**Fptw.exe**) behaves in the same manner apart from running in a Windows environment.

4.10.1 Example 1 – Flash SPI Flash Device with Binary File

```
C:\ fpt.exe –f spi.bin
```

This command writes the data in the **spi.bin** file into a whole SPI flash from address 0x0x

4.10.2 Example 2 – Program a Specific Region

```
fpt.exe –f –BIOS bios.rom
-----------------------------------------
Flash Programming Tool. Version X.X.X
Reading file "fparts.txt" into memory...
Initializing SPI utilities
Reading HSFSTS register... Flash Descriptor: Valid
    --- Flash Devices Found ---
    SST25VF016B                         ID:0xBF2541 Size: 2048KB
    (16384Kb)
Using software sequencing.
Reading LPC BC register... 0x00000001
Reading file "BIOS.ROM" into memory...
- Erasing Flash Block [0x101000]... - 100% complete.
- Programming Flash [0x100400]... - 100% complete.
Write Complete
```

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

4.10.3 Example 3 – Program SPI Flash from a Specific Address

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

This command loads 2KB of the binary file **image.bin** starting at address 0x0000. The starting address and the length must be a multiple of 4KB.

4.10.4 Example 4 – Dump Specific Region

```
fpt.exe –d -desc descdump.bin
-----------------------------------------
Flash Programming Tool. Version X.X.X
Reading file "fparts.txt" into memory...
Initializing SPI utilities
Reading HSFSTS register... Flash Descriptor: Valid
    --- Flash Devices Found ---
    SST25VF016B                         ID:0xBF2541   Size: 2048KB
    (16384Kb)
Using software sequencing.
- Reading Flash [0x000040]... 4KB of 4KB - 100% complete.
Writing flash contents to file "descdump.bin"...
Memory Dump Complete
```

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

4.10.5 Example 5 – Display SPI Information

```
fptw.exe –I
-----------------------------------------
Flash Programming Tool. Version X.X.X
Reading LPC BC register... 0x00000001
Reading LPC RCBA register... 0xFED1C001
SPI register base address... 0xFED1F020
Loading the flash definition file
Reading file "fparts.txt" into memory...
Initializing SPI utilities
Reading HSFSTS register... Flash Descriptor: Valid
   --- Flash Devices Found ---
   SST25VF016B                          ID:0xBF2541   Size: 2048KB
   (16384Kb)
Using software sequencing.
   --- Flash Image Information --
   Signature: VALID
   Number of Flash Components: 1
            Component 1 - 2048KB (16384Kb)
   Regions:
            Descriptor - Base: 0x000000, Limit: 0x000FFF
            BIOS       - Base: 0x100000, Limit: 0x1FFFFF
            ME         - Base: 0x001000, Limit: 0x0FDFFF
            GbE        - Base: 0x0FE000, Limit: 0x0FFFFF
   Master Region Access:
            CPU/BIOS - ID: 0x0000, Read: 0xFF, Write: 0xFF
            ME       - ID: 0x0000, Read: 0xFF, Write: 0xFF
            GbE      - ID: 0x0218, Read: 0xFF, Write: 0xFF
```

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

4.10.6 Example 6 – Verify Image with Errors

```
fpt.exe -verify outimage.bin
-----------------------------------------
Flash Programming Tool. Version X.X.X
Reading LPC BC register... 0x00000001
Reading LPC RCBA register... 0xFED1C001
SPI register base address... 0xFED1F020
Loading the flash definition file
Reading file "fparts.txt" into memory...
Initializing SPI utilities
Reading HSFSTS register... Flash Descriptor: Valid
   --- Flash Devices Found ---
   SST25VF016B                          ID:0xBF2541   Size: 2048KB
   (16384Kb)
   SST25VF016B                          ID:0xBF2541   Size: 2048KB
   (16384Kb)
Using software sequencing.
Reading file "outimage.bin" into memory...
RESULT: Data does not match!
```

```
0x00000000: 0x5A - 0x5A
0x00000001: 0xA5 - 0xA5
0x00000002: 0xF0 - 0xF0
0x00000003: 0x0F - 0x0F
0x00000004: 0x01 - 0x01
```

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, you are notified that the file is smaller than the binary image. This is due to extra padding that is added during the program process. The padding can be ignored when performing a comparison. The -y option proceeds with the comparison without warning.

4.10.7 Example 7 – Verify Image Successfully

```
fpt.exe -verify outimage.bin
-------------------------------------------
Flash Programming Tool. Version X.X.X
Reading file "fparts.txt" into memory...
Initializing SPI utilities
Reading HSFSTS register... Flash Descriptor: Valid
    --- Flash Devices Found ---
    SST25VF016B ID:0xBF2541 Size: 2048KB (16384Kb)
Using software sequencing.
Reading file "outimage.bin" into memory...
RESULT: Data does not match!
    [0x000000] Expected: 0x0B, Found: 0x5A
Total mismatches found in 64 byte block: 27
```

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

4.10.8 Example 8 – Program FOV Parameter

Note: This example is not applicable for 1.5MB Intel® ME FW SKU.

```
fpt.exe -u -n "AMTConfigMode" -v 0x03
-------------------------------------------
Flash Programming Tool. Version X.X.X
Reading file "fparts.txt" into memory...
Initializing SPI utilities
Reading HSFSTS register... Flash Descriptor: Valid
    --- Flash Devices Found ---
    SST25VF016B ID:0xBF2541 Size: 2048KB (16384Kb)

Updating software sequencing.
Reading region information from flash descriptor
Reading FOV configuration file "fptcfg.ini"
Updating variable [AMTConfigMode]..
```

This command updates the default configuration mode. In this example the Configuration mode was set to **Remote Connectivity Service.** This action is only supported if Remote Connectivity Service is supported on the system. FPT does not report dependency errors so you must be sure that the values selected are valid.

4.10.9 Example 9 – Get ME settings

FPT –r “Power Package 1” Read Power Package 1 variable and display the output in clear text on the screen without verbose information

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

4.10.10 Example 10 – Compare ME settings

FPT –verbose –compare vars.txt compare variables with suggested values in vars.txt, and report result on the screen. Vars.txt can have the following data with verbose information: You can use FPT –VARS to get the VAR list for the platform and get the value/format from FITC advanced mode.

```
"MEBxPassword" = 76 3C BE 3E B5 75 5F 6D 2D 5D 94 43 FD 79 A1 9D
54 D2 D5 9C 87 F8 FF 0E 6C 59 6F D2 17 37 13 5B
"OEMSkuRule" = EF DC EE 0F
"FeatureShipState" = EF FF EE 03
"OEM_TAG" = 78 56 34 12
"PID" = 8F DE B9 92 C3 88 03 71 12 A9 A7 3D FC 18 80 78 64 58 0A
E1 D9 E4 19 54 EF 6A 9F 33 F9 74 93 8C
"PPS" = 1A D3 16 1B A1 84 9A 7E 65 9E FB 67 1D 39 8E C0 06 92 81
67 4D 76 FB E4 09 1F 73 27 85 20 84 88
"USBrSettings" = 0B
"LAN Well Power Config" = SLP_LAN#(MGPIO3)
"WLAN Well Power Config" = Disabled
"Debug Si Features" = 00 00 00 00
"Prod Si Features" = 00 00 00 00
"M3 Power Rails Availability" = True
"HECI ME Region Unlockable" = True
"Sub System Vendor ID" = 00 00
"FW Update OEM ID" = 12345678-AABB-CCDD-EEFF-55AA11223344
"PROC_MISSING" = No onboard glue logic
"Power Package 1" = True
"Power Package 2" = True
"Default Power Package" = Power Package 2
"Enable Intel(R) Standard Manageability; Disable Intel(R) AMT" =
No
"Manageability Application Permanently Disabled?" = No
"PAVP Permanently Disabled?" = No
"KVM Permanently Disabled?" = No
"TLS Permanently Disabled?" = No
"Intel(R) Anti-Theft Technology Permanently Disabled?" = No
```

```
"Manageability Application Enable/Disable" = Enabled
"BIOS Reflash Capable" = False
"Boot into BIOS Setup Capable" = False
"Pause during BIOS Boot Capable" = False
"USBr EHCI 1 Enabled" = 11b Enabled
"USBr EHCI 2 Enabled" = 10b Disabled
"PrivacyLevel" = Default
"Host Based Setup and Configuration" = True
"Allow Unsigned Assert Stolen" = False
"Intel(R) Anti-Theft BIOS Recovery Timer" = Disabled
"MEBx Password Policy" = 00
"Hash 0 Active" = True
"Hash 0 Friendly Name" = VeriSign Class 3 Primary CA-G1
"Hash 0 Stream" = 74 2C 31 92 E6 07 E4 24 EB 45 49 54 2B E1 BB C5
3E 61 74 E2
"ODM ID used by Intel(R) Upgrade Service" = 00 00 00 00
```

§

5 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® MEManuf (Intel® MEMANUFWIN) requires administrator privileges to run under Windows OS. You must use the **Run as Administrator** option to open the CLI in Windows* Vista 64/32 bit and Windows* 7 64/32 bit.

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

5.1 Windows* PE Requirements

In order for the tools to work in a Windows* PE environment, you must manually load the driver by using the .inf file found in the Intel® MEI driver installation files. Once the .inf is located, you must use Windows* PE `cmd drvload *.inf` to load it into the running system each time Windows* PE reboots. Failure to do so causes the tools to report an error.

5.2 How to Use MEMANUF

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.

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.

MEMANUF tools report the result or cause a reboot. If there is a reboot, you should run `MEMANUF` again.

You can also run `MEMANFU –R` to retrieve/clean the test result from the previous MEMANUF test.

The Intel® AMT Sx (S4, S5) test must include a reset. It tests Intel® ME behavior when the system is in Sx state. Running this at least once in the manufacturing process is highly recommended to make sure HW components are working properly to support Intel® AMT working at Sx. (**Note**: This test is not applicable for 1.5MB Intel® ME FW SKU.)

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

5.3 Usage

The DOS version of the tool can be operated using the same syntax as the Windows version. The Windows version of the tool can be executed by:

```
MEMANUF [-S5/-S4/-S0][-NETOFF/-NETON][-R][-NOWLAN] [-NO3G] [-VERBOSE
<file>][-EXP][-VER] [-H][-?] [-EOL <Var|config>][-CFGGEN][-PAGE]
```

Table 13: Options for DOS Version of the Tool

Option	Description
No option	The tool runs some tests according to the FW SKU installed on the platform. If you have a platform that can enable Intel® AMT in the field, the tool requests the FW to run a complete BIST which includes a power reset (S4/S5) at the end of the test. This power reset is the only host-side power cycle that is triggered by Intel® ME. When the host resets, Intel® ME FW transitions from M0 to M3. It then automatically attempts to transition back from M3 to M0, bringing the host along back to S0. Once the host is booted back into OS, you must run the tool again in order to retrieve the test result. If you have a platform that can not enable Intel® AMT in the field (Intel® AMT is not present or it is permanently disabled), the tool requests that the FW run a complete BIST which doesn't involve any power transition at the end of the test. The test result is reported back right after the test is done and cleared. If the BIST test result isn't displayed after the BIST test is done, you must either run the tool again (with or without any BIST-related argument combinations) or you must use `-R` option to retrieve the result. The test result is cleared once it is displayed. Any attempt to explicitly retrieve the test result again after this causes a tool error. The tool is capable of remembering whether/what tests (including host-based tests) have been run from a previous invocation. Host-based tests are run for all cases (whether it's retrieving the test result or running the actual BIST). Currently there are two host-based tests: VSCC Table validation check and ICC data check. In any case, if the tool must run a reset at the end in the DOS environment; the tool runs the same test as `MEMANUF –S5`. In the Windows environment; the tool runs the same test as `MEMANUF –S4`. For AMT SKU, if there is a failure in the test, user will need to run MEMANUF again to retrieve the test result. This might be different than previous generation which in some case, tools will report error and stop the test.

Option	Description
-S0	The same as No option, except that there is no power reset/hibernation performed at the end of the BIST test including AMT SKU. The test result is reported back right after the test is done and cleared.
-R	Use this option to explicitly retrieve a test result. If there is no test result or it has been cleared, you will get a "No Intel® ME test result to retrieve" error. Once a test result is retrieved/displayed, it is cleared.
-EOL <Var|Config>	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`. (For more information, see VSCC test and ICC data check are performed for all options. 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. MEMANUF –EOL Check.)
-CFGGEN <filename>	Use this option along with a filename to generate a default configuration file. This file (with or without modification) can be used for the `-EOL` option. Rename it **MEManuf.cfg** before using it. It is highly recommended to use this option to generate a new **MEManuf.cfg** with an up-to-date variable names list before using the Intel® MEManuf End-Of-Line check feature.
–PAGE	When it takes more than one screen to display all the information, this option lets you pause the display and then press any key to continue on to the next screen.
-VER	Shows the version of the tools.
-VERBOSE <file>	Displays the debug information of the tool or stores it in a log file.
-EXP	Shows examples of how to use the tools.
-H or -?	Displays the help screen.
-S5	**Note**: This option is not applicable for 1.5MB Intel® ME FW SKU. The same as No option, except that it always requests FW to run a complete BIST that includes a power reset (S5) at the end of the test when AMT is present on the platform . Once the host is booted back into OS, you must run the tool again in order to retrieve the test result. **Note**: Both –S4 and –S5 require that the system run in AC mode.
-S4	**Note**: This option is not applicable for 1.5MB Intel® ME FW SKU. The same as the –S5 option, except that the system tries to hibernate (S4) instead of doing a power reset (S5). The hibernation is performed on the host side, and Intel® ME FW automatically transitions from M0 to M3 once the host is in S4 state. Once Intel® ME FW is in M3, it attempts to transition back from M3 to M0 bringing the host along back to S0. Once the host is booted back into OS, you must run the tool again in order to retrieve the test result.

Option	Description
-NETOFF	**Note**: This option is not applicable for 1.5MB Intel® ME FW SKU. This option re-enables the integrated GbE wired/wireless LAN interface so that network traffic can go in/out of it. If Intel® AMT is disabled, "Error 9257: Cannot run the command since Intel® AMT is not available" is returned.
-NETON	**Note**: This option is not applicable for 1.5MB 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.
-NO3G	This option will skip 3G test
-NOWLAN	**Note**: This option is not applicable for 1.5MB Intel® ME FW SKU. This option only applies to the AMT test so that you 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:** • -S5, -S4,-S0 can only be used on the platform which Intel® AMT is present and can be enabled in the field. • -S5/-S4 do not have any power policy dependency; even the current power policy does not support M3. Intel® MEMANUF is still able to test the Intel® ME M3 state.

Table 14: Intel® MEMANUF Test Matrix

Option	Intel® Vpro™ SKU	Consumer SKU	1.5MB Intel® ME FW SKU
No option	Run full BIST test (with Intel® ME-triggered reset under DOS, host-triggered hibernation under Windows)	Run runtime BIST test (with no reset)	Run runtime BIST test (with no reset)
-S4	Run full BIST test (with host-triggered hibernation)	Error: Invalid option -verbose: -S4 option is only available with Intel® AMT supported	Error: Invalid option -verbose: -S4 option is only available with Intel® AMT supported
-S5	Run full BIST test (with Intel® ME triggered reset)	Error: Invalid option -verbose: -S5 option is only available with Intel® AMT supported	Error: Invalid option -verbose: -S5 option is only available with Intel® AMT supported
-S0	Run runtime BIST test (with no reset)	Same as Intel® Vpro™ SKU	Same as Intel® Vpro™ SKU

Option	Intel® Vpro™ SKU	Consumer SKU	1.5MB Intel® ME FW SKU
-R	Retrieve the test result if there is one stored on the platform	Error: No result can be retrieved	Error: No result can be retrieved

Note: VSCC test and ICC data check are performed for all options.

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

5.4 MEMANUF –EOL Check

MEMANUF –EOL check is introduced in the CPT platform to give customers the ability to check Intel® ME-related configuration before shipping. There are two sets of tests that can be run: variable check and configuration check. Variable check is very similar as FPT –compare option. Please refer that section.

5.4.1 MEMANUF.cfg File

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

Here is an example of the **MEMANUF.cfg** file:

```
/ The end-of-line checks are broken into two categories. One is
// Variable Check, and the other is Configuration Check. If either
// of these check fails, by default MEManuf will report error and
// continue on to the next check. If a user doesn't wish to continue
// when an error is found, ErrAction field can be used. Please see
// the examples here for detailed explaination:
//
//    SubTestName="ME VSCC check", ErrAction="ErrorStop"
//
// If the above test fails, MEManuf will report error and stop. There
// are total of three different error actions user can choose from:
//
// ErrorContinue - report error and continue on to the next check
// ErrorStop - report error and stop any check after the current one
// WarnContinue - report warning and continue on to the next check
//
// To add comment or take out a specific test, leave // at the start
// of a line. This file is processed by MEManuf line by line as text
// file. Duplication of the same sub-tests are allowed, but MEManuf
// will always perform the last test to the first test from the file.

// All string comparisions given in this file are case insensitive
// compare. There might be multiple field name/value pairs in one
// entry, but each field needs to be specified in the following
// format where <field name> can be replaced by SubTestName, ReqVal
// or ErrAction, <field value> can be replaced by any string including
// dash and/or spaces surrounded by double quotation marks, or hex-
// decimal number(s) that not surrounded by double quotation marks.
```

```
// In case of numeric value, each value (without 0x prefix) needs to
// be specified in byte and deliminated by spaces if there are multiple
// bytes. No line Wrapping is supported:
//
//    <field name>="<field value>", such as ReqVal=" ", or
//    <field name>=<numeric value>, such as ReqVal=78, or
//    <field name>=<numeric value>, such as ReqVal=01 0A 0F FE 7B CD

////////////////////////////////////////////////////////////////////////
////
// Intel recommands default end-of-line checks includes the following
// list. If a user chooses to use his/her own version of MEManuf.cfg
// to skip or modify the error action of these checks as WarnContinue,
// MEManuf will report failure with warnings when these checks are
skipped,
// or have errors. It's suggested that a user should perform these
Intel(R)
// recommanded check on all type of SKUs.

SubTestName="EOP status check"
SubTestName="ME VSCC check"
SubTestName="BIOS VSCC check"
SubTestName="ME Manufacturing Mode status"
SubTestName="Flash Region Access Permissions"
SubTestName="Flash Descriptor Override Strap (GPIO33) check"
SubTestName="CF9GR lock check"
SubTestName="MAC address"
SubTestName="Wireless MAC address"
SubTestName="System UUID"

////////////////////////////////////////////////////////////////////////
////
// Please note that MAC address check will be skipped if Intel Gbe region
// is not present in SPI image. Wireless MAC address check will be
skipped
// if Intel wireless device is not found on the PCI bus. System UUID
check
// will be skipped if platform is not vPro platform.
//
// MAC address check, Wireless MAC address check and UUID check
// will be skipped if Intel(R) AMT is permanently disabled or not
present.
//
// MAC address and System UUID Checks can work with an optional ReqVal
field,
// which allows a user to specify his/her custom values to compare
against.
//
// For example, the test shown here checks the current wired LAN MAC
address
// against user provided value of 01-02-03-04-05-06:
//
//    SubTestName="MAC address", ReqVal="01-02-03-04-05-06"
//
// Here is the default values MEManuf uses if ReqVal field if obmitted:
//
// System UUID - all zeros and 0xff are considered as errors
// MAC address - all zeros and 0xff are considered as errors
// Wireless MAC address - all zeros and 0xff are considered as errors
//
// MAC address takes the format as XX-XX-XX-XX-XX-XX
// System UUID takes the format as XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX
////////////////////////////////////////////////////////////////////////
////
```

```
////////////////////////////////////////////////////////////////////////
///////
// The following Configuration Check requires a user to enter an expected
// value after ReqVal=, otherwise the lines without ReqVal field values
will
// be ignored.
//
// Please note that GBE version check will be skipped if Intel Gbe region
// is not present in SPI image.
//
// ME FW version is a string as <major ver>.<minor ver>.<hotfix
ver>.<build num>
// GBE version is a string as <major ver>.<minor ver>.<revision ver>
// BIOS version is string that vendor specific
////////////////////////////////////////////////////////////////////////
///////

// SubTestName="ME FW version", ReqVal=
// SubTestName="BIOS version", ReqVal=
// SubTestName="GBE version", ReqVal=

////////////////////////////////////////////////////////////////////
// Variable Check - user needs to put an expected value after ReqVal,
// otherwise the lines without ReqVal field values will be ignored
//
// There are variables that stored in encrypted format. When comparing
// with these variables, ReqVal can only specified as numeric values
// (in encrypted form) in byte order as mentioned aboved. ReqVal needs
// to be surrounded by double quotation marks if they are string input.
//
// To get a up-to-dated MEManuf.cfg with a complete variable names list,
// please run MEManuf -cfggen <filename>. Please note that variables
// that have # need to be replace by a number. Here defines the number:
//
// Note: The '#' for hash variables should be replaced with an entry
index.
//       The valid range is 0 to 22.
//
// !!! Please be sure to disable sending EOP or leave platform in ME
// !!! manufacturing mode to run this test, otherwise MEManuf will
// !!! report failure because this feature is only available in factory
// !!! mode environment.
////////////////////////////////////////////////////////////////////

// SubTestName="Allow Unsigned Assert Stolen", ReqVal=
// SubTestName="BIOS Reflash Capable", ReqVal=
// SubTestName="Boot into BIOS Setup Capable", ReqVal=
// SubTestName="Debug Si Features", ReqVal=
// SubTestName="Default Power Package", ReqVal=
// SubTestName="Enable Intel(R) Standard Manageability; Disable Intel(R)
AMT", ReqVal=
// SubTestName="FeatureShipState", ReqVal=
// SubTestName="Flash Protection Override Policy Hard", ReqVal=
// SubTestName="Flash Protection Override Policy Soft", ReqVal=
// SubTestName="FW Update OEM ID", ReqVal=
// SubTestName="Hash # Active", ReqVal=
// SubTestName="Hash # Friendly Name", ReqVal=
// SubTestName="HECI ME Region Unlockable", ReqVal=
// SubTestName="Host Based Setup and Configuration", ReqVal=
// SubTestName="Idle Timeout - Manageability Engine", ReqVal=
// SubTestName="Intel(R) Anti-Theft BIOS Recovery Timer", ReqVal=
// SubTestName="Intel(R) Anti-Theft Technology Permanently Disabled?",
ReqVal=
// SubTestName="KVM Permanently Disabled?", ReqVal=
```

```
// SubTestName="LAN Well Power Config", ReqVal=
// SubTestName="M3 Power Rails Availability", ReqVal=
// SubTestName="Manageability Application Enable/Disable", ReqVal=
// SubTestName="Manageability Application Permanently Disabled?", ReqVal=
// SubTestName="MCTP Info 3G", ReqVal=
// SubTestName="MCTP Static EIDs", ReqVal=
// SubTestName="MEBx Password Policy", ReqVal=
// SubTestName="MEBxPassword", ReqVal=
// SubTestName="ODM ID used by Intel (R) Upgrade Service", ReqVal=
// SubTestName="OEM Customizable Certificate 1", ReqVal=
// SubTestName="OEM Customizable Certificate 2", ReqVal=
// SubTestName="OEM Customizable Certificate 3", ReqVal=
// SubTestName="OEM Default Certificate", ReqVal=
// SubTestName="OEM_TAG", ReqVal=
// SubTestName="OEMSkuRule", ReqVal=
// SubTestName="PasswordFlag", ReqVal=
// SubTestName="Pause during BIOS Boot Capable", ReqVal=
// SubTestName="PAVP Permanently Disabled?", ReqVal=
// SubTestName="Permit Period Timer Resolution", ReqVal=
// SubTestName="PID", ReqVal=
// SubTestName="PKI DNS Suffix", ReqVal=
// SubTestName="Power Package 1", ReqVal=
// SubTestName="Power Package 2", ReqVal=
// SubTestName="PPS", ReqVal=
// SubTestName="PrivacyLevel", ReqVal=
// SubTestName="PROC_MISSING", ReqVal=
// SubTestName="Prod Si Features", ReqVal=
// SubTestName="Remote Configuration Enabled", ReqVal=
// SubTestName="Reserved ID used by Intel (R) Upgrade Service", ReqVal=
// SubTestName="Sub System Vendor ID", ReqVal=
// SubTestName="System Integrator ID used by Intel (R) Upgrade Service",
ReqVal=
// SubTestName="TLS Permanently Disabled?", ReqVal=
// SubTestName="USBr EHCI 1 Enabled", ReqVal=
// SubTestName="USBr EHCI 2 Enabled", ReqVal=
// SubTestName="USBrSettings", ReqVal=
```

// SubTestName="WLAN Well Power Config", ReqVal=

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

To select which test items to run: Create a line that begins with SubTestName="<specific sub test name>".

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

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

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

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

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

Note: The complete list of FOVs/NVARs to be included under "Variable Check" is still a WIP.

5.4.2 MEMANUF –EOL Variable Check

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

5.4.3 MEMANUF –EOL Config Check

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

Table 15: MEMANUF - EOL Config Tests

Test	Expected Configuration
EOP status check	Enabled
ME VSCC check	Set according to the Intel-recommended value
BIOS VSCC check	Set according to the Intel-recommended value
ME Manufacturing Mode status	Disabled
Flash Region Access Permissions	Set according to the Intel-recommended value
Flash Descriptor Override Strap (GPIO33)	Disabled
MAC address	None, all 0, or f
Wireless MAC address	None, all 0, or f
System UUID	None, all 0

Note:

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

5.4.4 Output / Result

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

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

- Fail - any customer-defined error occurred in the test.

5.5 Examples

5.5.1 Example 1

Note: Not applicable for 1.5MB Intel® ME FW SKU.

```
MEManufWin.exe –s4
```

This command runs the Intel® AMT test with a reboot if Intel® AMT is enabled on this platform. However, instead of a hard power cycle, Intel® MEManuf sends Windows into S4 hibernate mode and then brings the system back to the S0 state. Repeat this command to view the test results.

You must use `MEMANUFWIN.exe –S4 –R` or `MEMANUFWIN.exe` to retrieve the test result:

```
Intel(R) MEManuf Version: 7.0.0.1092
Copyright(C) 2005 - 2010, Intel Corporation. All rights reserved.


MEManuf Test Passed
```

5.5.2 Example 2

Note: Not applicable for 1.5MB Intel® ME FW SKU.

```
MEManufWin.exe –S5
```

Be sure to save your data before invoking this command on a Windows system because it immediately sends the system into an S5 state and any unsaved data may be lost. This command then immediately powers the computer back on if Intel® AMT is enabled on the platform.

To view the results, you can run MEMANUF again or `MEMANUF –S5 -R` option to retrieve the result.

5.5.3 Example 3

5.5.3.1 Example for 1.5MB Intel® ME FW SKU

```
MEMANUF –verbose

   Intel(R) MEManuf Version: 7.0.0.1061
   Copyright(C) 2005 - 2010, Intel Corporation. All rights reserved.

   Platform stepping value is 3

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

```
  CurrentState:                          Normal
  ManufacturingMode:                     Enabled
  FlashPartition:                        Valid
  OperationalState:                      M0 with UMA
  InitComplete:                          Complete
  BUPLoadState:                          Success
  ErrorCode:                             No Error
  ModeOfOperation:                       Normal
  ICC:                                   Valid OEM data, ICC
programmed

Get FWU info command...done

Get FWU version command...done

Get FWU feature state command...done

Get ME FWU platform type command...done

Get ME FWU feature capability command...done
Feature enablement is 0x1001C60
gFeatureAvailability value is 0x1
System is running on consumer/4M image, start Intel(R) ME Runtime
Test
OEM ICC data valid and programmed correctly

Request Intel(R) ME test result command...done
vsccommn.bin was created on 23:32:28 05/05/2010 GMT
SPI Flash ID #1 ME VSCC value is 0x2005
SPI Flash ID #1 (ID: 0xEF4017) ME VSCC value checked
SPI Flash ID #1 BIOS VSCC value is 0x2005
SPI Flash ID #1 (ID: 0xEF4017) BIOS VSCC value checked
SPI Flash ID #2 ME VSCC value is 0x2005
SPI Flash ID #2 (ID: 0xEF4017) ME VSCC value checked
SPI Flash ID #2 BIOS VSCC value is 0x2005
SPI Flash ID #2 (ID: 0xEF4017) BIOS VSCC value checked
FPBA value is 0x0
No Intel Wireless device was found

Request Intel(R) ME Runtime BIST test command...done

Get Intel(R) ME test data command...done
Total of 22 Intel(R) ME test result retrieved
Micro Kernel - Blob Manager: Set - Passed
Micro Kernel - Blob Manager: Get - Passed
Micro Kernel - Blob Manager: Remove - Passed
Policy Kernel - SMBus: Read byte - Passed
Policy Kernel - ME Password: Valid MEBx password - Passed
Policy Kernel - Power Package: Package 1 supported - Passed
```

```
Policy Kernel - Power Package: Default package supported - Passed
Policy Kernel - ME Configuration: Wlan Power Well - Passed
Policy Kernel - ME Configuration: CPU Missing Logic - Passed
Policy Kernel - ME Configuration: M3 Power Rails Available - Passed
Policy Kernel - Embedded Controller: Get power source - Passed
Common Services - General: Low power idle timeout - Passed
Common Services - Provisioning: Valid MEBX password change policy -
Passed
Common Services - Provisioning: Zero-Touch configuration enabled -
Passed
Common Services - Provisioning: Client Config mode is valid - Passed
Common Services - General: Vlan not enabled on mobile - Passed
Common Services - Provisioning: Both PID and PPS are set - Passed
Common Services - Provisioning: MEBX password set when PID and PPS
set - Passed
Common Services - Wireless LAN: Connectivity to NIC - Skipped
AMT - Privacy Level: Valid Privacy Level settings - Passed
Policy Kernel - Power Package: Live Heap Test - Passed

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

MEManuf Test Passed
```

5.5.3.2 Example for 5MB Intel® ME FW SKU

```
MEMANUF –verbose
```

Intel(R) MEManuf Version: 7.0.0.1092

Copyright(C) 2005 - 2010, Intel Corporation. All rights reserved.

Platform stepping value is 3

FW Status Register1: 0x1E000255

FW Status Register2: 0x62000006

CurrentState:	Normal
ManufacturingMode:	Enabled
FlashPartition:	Valid
OperationalState:	M0 with UMA
InitComplete:	Complete

BUPLoadState: Success

ErrorCode: No Error

ModeOfOperation: Normal

ICC: Valid OEM data, ICC programmed

Get FWU info command...done

Get FWU version command...done

Get FWU feature state command...done

Get ME FWU platform type command...done

Get ME FWU feature capability command...done

Feature enablement is 0xDE65C61

gFeatureAvailability value is 0x1

OEM ICC data valid and programmed correctly

Request Intel(R) ME test result command...done

vsccommn.bin was created on 22:29:43 06/28/2010 GMT

SPI Flash ID #1 ME VSCC value is 0x2005

SPI Flash ID #1 (ID: 0xEF4017) ME VSCC value checked

SPI Flash ID #1 BIOS VSCC value is 0x2005

SPI Flash ID #1 (ID: 0xEF4017) BIOS VSCC value checked

SPI Flash ID #2 ME VSCC value is 0x2005

SPI Flash ID #2 (ID: 0xEF4017) ME VSCC value checked

SPI Flash ID #2 BIOS VSCC value is 0x2005

SPI Flash ID #2 (ID: 0xEF4017) BIOS VSCC value checked

FPBA value is 0x0

No Intel Wireless device was found

Request Intel(R) ME Runtime BIST test command...done

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

Total of 30 Intel(R) ME test result retrieved

Micro Kernel - Blob Manager: Set - Passed

Micro Kernel - Blob Manager: Get - Passed

Micro Kernel - Blob Manager: Remove - Passed

Policy Kernel - SMBus: Read byte - Passed

Policy Kernel - ME Password: Valid MEBx password - Passed

Policy Kernel - Power Package: Package 1 supported - Passed

Policy Kernel - Power Package: Default package supported - Passed

Policy Kernel - ME Configuration: Wlan Power Well - Passed

Policy Kernel - ME Configuration: CPU Missing Logic - Passed

Policy Kernel - ME Configuration: M3 Power Rails Available - Passed

Policy Kernel - Embedded Controller: Get power source - Passed

Common Services - General: Low power idle timeout - Passed

Common Services - Provisioning: Valid MEBX password change policy - Passed

Common Services - Provisioning: Zero-Touch configuration enabled - Passed

Common Services - Provisioning: Client Config mode is valid - Passed

Common Services - General: Vlan not enabled on mobile - Passed

Common Services - Provisioning: Both PID and PPS are set - Passed

Common Services - Provisioning: MEBX password set when PID and PPS set - Passed

AMT - Privacy Level: Valid Privacy Level settings - Passed

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

AMT - Power: WLAN enabled on mobile - Failed

AMT - Power: Power-package 2 supported - Passed

AMT - KVM: USBr is enabled when KVM is enabled - Passed

AMT - EC: Basic connectivity - Passed

AMT - KVM: Compare engine - Passed

AMT - KVM: Compression engine - Passed

AMT - KVM: Sampling engine - Passed

AMT - KVM: VDM engine - Passed

AMT - USBr: Hardware - Passed

Policy Kernel - Power Package: Live Heap Test - Passed

Error 9314: Intel(R) ME test result reports error(s)

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

Error 9296: MEManuf Test Failed (9314)

5.5.4 Example 4: Consumer Platform

`MEManuf.exe`

This command executex tests on a consumer platform. The system does not power cycle at the end of this test.

6 MEInfo

MEInfoWin and Intel® MEInfo provide a simple test to check whether the ME FW is alive or not. Both tools perform the same test, query the Intel® ME FW – including Intel® AMT – and retrieve data. Table 16 contains a list of the data that each tool returns.

The Windows version of MEInfo (MEInfoWin) requires administrator privileges to run under Windows OS. You must use the **Run as Administrator** option to open the CLI in Windows* Vista 64/32 bit and Windows* 7 64/32 bit.

6.1 Windows* PE Requirements

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

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
MEInfo.exe [-feat <name> -value <value>]
MEInfo.exe [-feat <name>]
MEINFO.exe -FWSTS
MEINFO.exe [-H]
MEINFO.exe [-?]
MEINFO.exe –verbose <filename>
MEINFO.exe [-VER]
MEINFO.exe [-EXP]
MEINFO.exe [-BLIST]
MEINFO.exe [-PAGE]
```

Table 16: Intel® MEInfo Command Line Options

Option	Description
-feat < name> -value <value>	Compares the value of the given feature name with the value in the command line. If the feature name or value is more than one word, the entire name or value must be enclosed in quotation marks. If the values are identical, a message indicating success appears. If the values are not identical, the actual value of the feature is returned. Only one feature may be requested in a command line.

Option	Description
-feat <name>	Retrieves the current value for the specified feature. If the feature name is more than one word, the entire feature name must be enclosed in quotation marks. The feature name entered must be the same as the feature name displayed by Intel® MEInfo. Intel® MEInfo can retrieve all of the information detailed below. However, depending on the SKU selected, some information may not appear.

–FWSTS	Decodes the Intel® ME FW status register value field and breaks it down into the following bit definitions for easy readability:
	`FW Status Register: 0x00000245`
	`FW Status Register1: 0x60000000`
	`CurrentState: Normal`
	`ManufacturingMode: Disabled`
	`FlashPartition: Valid`
	`OperationalState: M0 with UMA`
	`InitComplete: Complete`
	`BUPLoadState: Success`
	`ErrorCode: No Error`
	`ModeOfOperation: Normal`
	`Phase: HOSTCOMM Module`

-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`.
–BLIST	Displays the Black List information of the FW currently running on the platform. This list shows you which versions are NOT eligible for an update on the platform. The black list information displayed here is the same information that the FWUpdate tool exposed.
-H or -?:	Displays the list of command line options supported by the Intel® MEInfo tool.
-VER	Shows the version of the tools.
- PAGE	When it takes more than one screen to display all the information, this option lets you pause the display and then press any key to continue on to the next screen.
-VERBOSE <file>	Displays the tool's debug information or stores it in a log file.
-EXP	Shows examples about how to use the tools.
No option:	**ol is invoked without parameters, it reports information for all components listed in** Table 17 below for full SKU FW.

Table 17: List of Components for which Version Information is retrieved

Feature Name	Feature Data Source (ME Kernel/AMT/ SW/Other)	Consumer SKU	Corporate SKU	Specific Feature Dependency	Field Value
Tools Version	SW (MEInfo)	X	X	N/A	A version string
PCH Version	ME Kernel	X	X	N/A	A version string
FW Version	ME Kernel	X	X	N/A	A version string
BIOS Version	ME Kernel	X	X	MEBx needs to be present. Not available on 4M Sku	A version string
GbE Version	Other (Directly reading from SPI)	X	X	GbE Region to be present in the image	A version string
MEBx Version	ME Kernel	X	X	MEBx needs to be present. Not available on 4M Sku	A version string
VendorID	ME Kernel	X	X	N/A	A number (in Hex)
Wireless Driver/Hardware Version*	Other (Reading Windows registry entries)	X	X	Only when wireless HW is present, and wireless windows driver is installed	A version string
Link Status	AMT	X	X	AMT CEM (a.k.a Common Service) is used. Not available on 4M Sku	Link up/down

Feature Name	Feature Data Source (ME Kernel/AMT/ SW/Other)	Consumer SKU	Corporate SKU	Specific Feature Dependency	Field Value
FW Capabilities	ME Kernel	X	X	N/A	Combination of feature name list breakdown (with Decimal bits value) *This is a display of the Feature State for the ME. Is enabled / disabled on the system. Each bit in the value represents a feature state. ME feaurtes including Full manageability, standard manageability, Anti-theft technology etc.
Cryptography Support	ME Kernel	X	X	N/A	Enabled/Disabled
BIOS and GbE Config Lock	Other (Directly reading from SPI)	X	X	N/A	Enabled/Disabled/Unknown If shown as enabled, both FLOCKDN for BIOS and Gbe are set. If shown as disabled, either/all FLOCKDN for BIOS and Gbe are not set.
Host Read Access to ME	Other (Directly reading from SPI)	X	X	N/A	Enabled/Disabled/Unknown

Feature Name	Feature Data Source (ME Kernel/AMT/ SW/Other)	Consumer SKU	Corporate SKU	Specific Feature Dependency	Field Value
Host Write Access to ME	Other (Directly reading from SPI)	X	X	N/A	Enabled/Disabled/Unknown
Last ME Reset Reason	ME Kernel	X	X	N/A	Power up/Firmware reset/Global system reset/Unknown
Intel® AMT State	ME Kernel	N/A	X	Both Full Manageability and Manageability Application has to be PRESENT (Capable)	Enabled/Disabled
Intel® Standard Manageability State	ME Kernel	N/A	X	Full Manageability should not be PRESENT (Capable), but Manageability Application has to be PRESENT	Enabled/Disabled
VE Enablement Status	ME Kernel (VE Enablement in Softstrap 10 [bit3] and 14 [bit 8])	X	X	N/A	Enabled/Disabled/Invalid Enabled when both Softstraps indicate VE is enabled Disabled when both Softstraps indicate VE is disabled Invalid when either one of the Softstraps indicates VE is enabled
BIOS Boot State	ME Kernel	X	X	N/A	Pre Boot/In Boot/Post Boot

Feature Name	Feature Data Source (ME Kernel/AMT/ SW/Other)	Consumer SKU	Corporate SKU	Specific Feature Dependency	Field Value
System UUID	AMT	N/A	X	AMT CEM (a.k.a. Common Service) is used. Not available on 4M Sku	UUID of the system
OEM Id	ME Kernel	X	X	Only if fw image supports OEM Id	UUID for OEM to check during FW Update
Configuration State	AMT	N/A	X	AMT CEM (a.k.a. Common Service) is used. Not available on 1.5M Sku	Not started/In process/Completed/Unknown
Provisioning Mode	AMT	N/A	X	AMT CEM (a.k.a. Common Service) is used. Not available on 1.5M Sku	PKI/PSK/Unknown
MAC Address	AMT	X	X	AMT CEM (a.k.a. Common Service) is used only when wired Hw is present. Not available on 1.5M Sku	A MAC address (in Hex separated by “=”)
Wireless MAC Address	AMT	X	X	AMT CEM (a.k.a. Common Service) is used only when wireless HW is present. Not available on 1.5M Sku	A MAC address (in Hex separated by “=”)
IPv4 Address (Wired and Wireless)	AMT	X	X	AMT CEM (a.k.a. Common Service) is used only when wired/wireless Hw is present. Not available on 1.5M Sku	IPv4 IP address (in decimal separated by “.”)

Feature Name	Feature Data Source (ME Kernel/AMT/ SW/Other)	Consumer SKU	Corporate SKU	Specific Feature Dependency	Field Value
IPv6 Address (Wired and Wireless)	AMT	N/A	X	AMT CEM (a.k.a. Common Service) is used only when wired/wireless Hw is present. Not available on 1.5M Sku	All IPv6 IP addresses
IPv6 enabled (Wired and Wireless)	AMT	N/A	X	AMT CEM (a.k.a. Common Service) is used only when wired/wireless Hw is present. Not available on 1.5M Sku	Enabled/Disabled
Local FWUpdate	ME Kernel	X	X	N/A	Enabled/Disabled/Password Protected
MEI Driver version*	Other (Reading Windows registry entries)	X	X	Only when Windows MEI driver is installed	A version string
LMS version*	Other (Reading Windows registry entries)	X	X	Only when Windows LMS driver is installed	A version string
UNS version*	Other (Reading Windows registry entries)	X	X	Only when Windows UNS driver is installed	A version string
SPI Flash ID	Other (Directly reading from SPI)	X	X	Only when there are flash parts HW installed	A JEDEC ID number (in Hex)
ME/BIOS VSCC register values	Other (Directly reading from SPI)	X	X	Only when there are flash parts HW installed	A 32bit VSCC number (in Hex)

Feature Name	Feature Data Source (ME Kernel/AMT/ SW/Other)	Consumer SKU	Corporate SKU	Specific Feature Dependency	Field Value
Capability Licensing Service	ME Kernel	X	X	Not available on 4M Sku. Not shown unless Fw feature capability supports it	Enabled/Disabled
Capability Licensing Service Status	ME Kernel	X	X	Not available on 4M Sku. Not shown unless FW feature capability supports it. This feature is only shown if there is a Level III PCH devices, or the feature is enabled	Permit info not available/Upgraded/Not Upgraded/Not Upgradable
Level III Manageability Upgrade	ME Kernel (ICLS)	X	X	B65 with non-Celeron CPU	Upgraded/Upgrade Capable/Not Upgradable
CPU Upgrade State	ME Kernel (ICLS)	N/A	H65, H67, H61, HM65, HM67	Not available on 4M SKU. Not shown unless Fw feature capability supports it	Upgraded/Upgrade Capable/Not Upgradable
Privacy Level	AMT	X	X	Not available on 4M SKU. Only shown when AMT is enabled	Default/Enhanced/Extreme/Unknown
OEM Tag	ME Kernel	X	X	N/A	A 32bit Hexadeimal number
FWSTS	ME Kernel	X	X	N/A	Two 32bit Hexadecimal numbers and their bit definition breakdown

6.3 Examples

6.3.1 Example 1

This is a simple test that indicates whether the FW is alive. If the FW is alive, the test returns device-specific parameters. The output is from the Windows version. The DOS version does not display the UNS version, Intel® Management Engine Interface, or LMS version numbers.

6.3.1.1 Example for 1.5MB Intel® ME FW SKU

```
MEINFOWIN.exe

Intel(R) MEInfo Version: 7.0.0.1061
Copyright(C) 2005 - 2010, Intel Corporation. All rights reserved.

GBE Region does not exist.
Intel(R) ME code versions:

BIOS Version:                           ASNBCPT1.86C.0028.B00.1006171024
MEBx Version:                           0.0.0.0
Gbe Version:                            Unknown
VendorID:                               8086
PCH Version:                            600003
FW Version:                             7.0.0.1061

FW Capabilities:                        16784480

    Intel(R) Anti-Theft Technology - PRESENT/ENABLED
    Intel(R) Capability Licensing Service - PRESENT/ENABLED
    Protect Audio Video Path - PRESENT/ENABLED

Level III Manageability Upgrade State:  Not Upgradable
CPU Upgrade State:                      Upgrade Capable
Cryptography Support:                   Disabled
Last ME reset reason:                   Global system reset
Local FWUpdate:                         Enabled
BIOS and GbE Config Lock:               Disabled
Host Read Access to ME:                 Enabled
Host Write Access to ME:                Enabled
SPI Flash ID #1:                        EF4017
SPI Flash ID VSCC #1:                   20052005
SPI Flash ID #2:                        EF4017
SPI Flash ID VSCC #2:                   20052005
SPI Flash BIOS VSCC:                    20050000
VE Enablement Status:                   Invalid
BIOS boot State:                        Post Boot
OEM Id:                                 00000000-0000-0000-0000-
000000000000
Link Status:                            Link down
MAC Address:                            00-00-00-00-00-00
IPv4 Address:                           0.0.0.0
Capability Licensing Service:           Enabled
Capability Licensing Service Status:    Permit info not available
```

6.3.1.2 OEM Tag: 0x00000000Example for 5MB Intel® ME FW SKU

```
MEINFOWIN.exe

Intel(R) MEInfo Version: 7.0.0.1092
Copyright(C) 2005 - 2010, Intel Corporation. All rights reserved.

Intel(R) ME code versions:

BIOS Version:                           ASNBCPT1.86C.0036.B00.1008051344
MEBx Version:                           7.0.0.43
Gbe Version:                            7.16.0
VendorID:                               8086
PCH Version:                            600003
FW Version:                             7.0.0.1092

FW Capabilities:                        233200741

    Intel(R) Active Management Technology - PRESENT/DISABLED
    Intel(R) Anti-Theft Technology - PRESENT/ENABLED
    Intel(R) Capability Licensing Service - PRESENT/ENABLED
    Protect Audio Video Path - PRESENT/ENABLED

Intel(R) AMT State:                     Disabled
CPU Upgrade State:                      Upgrade Capable
Cryptography Support:                   Enabled
Last ME reset reason:                   Power up
Local FWUpdate:                         Enabled
BIOS and GbE Config Lock:               Disabled
Host Read Access to ME:                 Enabled
Host Write Access to ME:                Enabled
SPI Flash ID #1:                        EF4017
SPI Flash ID VSCC #1:                   20052005
SPI Flash ID #2:                        EF4017
SPI Flash ID VSCC #2:                   20052005
SPI Flash BIOS VSCC:                    20050000
VE Enablement Status:                   Invalid
BIOS boot State:                        Post Boot
OEM Id:                                 00000000-0000-0000-0000-
000000000000
Link Status:                            Link down
System UUID:                            00000000-0000-0000-0000-
000000000000
MAC Address:                            88-88-88-88-87-88
IPv4 Address:                           0.0.0.0
IPv6 Enablement:                        Disabled
Configuration state:                    Not started
Provisioning Mode:                      PKI
Capability Licensing Service:           Enabled
Capability Licensing Service Status:    Permit info not available
OEM Tag:                                0x00000000
```

6.3.2 Example 2

This example retrieves the current value of the Flash version:

```
C:\ MEInfo.exe -feat "Local FWUpdate"
Intel(R) MEInfo Version: 7.0.0.1092
Copyright(C) 2005 - 2010, Intel Corporation. All rights reserved.
```

6.3.3 Local FWUpdate: EnabledExample 3

This example checks whether the computer has completed the setup and configuration process. If the parameter name or the value has a space, the value or name should have be preceded and followed by a quotation mark.

```
C:\ MEInfo.exe -feat "Setup and Configuration" –value "Not Completed"
Intel(R) MEInfo Version: 7.0.0.1092
Copyright(C) 2005 - 2010, Intel Corporation. All rights reserved.
```

Local FWUpdate: Success - Value matches FW value.

§

7 ME Firmware Update

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).

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 only updates the FW code portion that Intel provides on the OEM website. Intel® FWUpdate updates the entire Intel® ME code area.

The image file that the tool uses for the update is not the image file used to create the complete SPI FW image file. A sample FW image file for updating, **Base_Corporate_Intel® ME_UPD.BIN**, is located in the kit's Image Components\ME folder. It only contains the ME code region and it can not be used to generate a whole SPI image with the FITC tool.

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. You can use the –FORCERESET option to do this automatically.

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.

If Intel® Anti-theft technology is enabled, a system restart must occur to complete the FW update process.

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

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

7.2 Windows* PE Requirements

In order for tools to work under Windows* PE environment, you must manually load a driver by using the .inf file in the Intel® MEI driver installation files. Once you locate the .inf file, you must 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 MEBx (or BIOS depending on customer implementation), there is an option to enable/disable local firmware update.

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

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

For more details please refer to MEBx user guide.

7.4 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 FITC to program the entire flash memory.

To differentiate between the image files used for updating and those used for programming the entire flash memory, files used for Intel® FWUpdate include the string UPD in their file names.

```
FWUpdLcl.exe [-VERBOSE <file>] [-Y] [-ALLOWSV] [-FORCERESET] [-
OEMID <UUID>][-PASS <PASSWORD>] [-GENERIC] -F <file>
FWUpdLcl.exe –HALTRCFG
FWUpdLcl.exe [-H|?] [-VER] [-EXP] [-FWVER]
```

Note: Image File is the image file of the FW to be updated. It is not the same image file used by FITC.

Table 18: 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

Option	Description
	automatically take "y" as the input.
-F <FILE>	File. Specifies the FWUpdate image file to be used for performing an update.
-BLIST	Display Black List. Displays the Black List information of the FW currently running on the platform. This list shows you which versions are NOT eligible for an update on the platform.
-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 you 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).
-HALTRCFG	Halt Remote Configuration. The tool halts remote configuration. **Note**: This is NOT an option used with updating the FW image.
-PASS <PASSWORD>	This is used to specify the MeBx password to perform the update. A valid password is required to perform the update especially when FW Update setting in Mebx is set to “password protected”.
-generic	MEI. Specifies that the tool performs the update over the MEI interface. MEI is used even if the FW supports a network-based update. **Note**: This option is only supported in the Windows version of the tool.
-FWVER	Display FW version
-H or -?:	Displays the list of command line options supported by the Intel® MEInfo tool.
-EXP	Shows examples about how to use the tools.
-VER	Shows the version of the tools.

7.5 Examples

7.5.1 Example 1

```
FWUpdLcl.exe –f CPT_5M__UPD_Production.BIN
```

This command updates ME with CPT_5M_UPD_Production.BIN file. If the firmware on current platform is newer than then version in CPT_5M_UPD_Prodcution.BIN file, tools will promote a warning to let user know there will be a firmware downgard (rollback) happen 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 CPT_5M_UPD_Production, tools will return with an error. User can use –allowsv to allow same version update.

7.5.2 Example 2

```
FWUpdLcl.exe -haltRCFG
```

Calling the `-haltRCFG` option halts all remote configuration traffic and prevents remote configuration. `-haltRCFG` can NOT be used as a command line argument while performing FWUpdate.

8 Update Parameter Tool

Note: This section is not applicable for 1.5MB Intel® ME FW SKU.

8.1 Purpose of the Tool

UPdParam is used to change certain Intel® ME FW parameters (both Intel® AMT and Kernel) even after the Intel® ME manufacturing mode done bit (global locked bit) is set and the Descriptor region is locked. This tool only works on DOS when BIOS does not send an EOP message.

8.2 Usage of the Tool

```
UpdParam.exe [-?] [-h] [-f] [-v] [-r] [-u] [-ver] [-s] [-c] [-exp] [-
verbose <file>]
```

Table 19: Update Parameter Tool Options

Option	Description
-H|?	Displays help screen
-F <filename>	Inputs USB file name
-V <MeBXCurrPwd>	Overrides Intel® MEBx Admin password
-R	Global reset
-U	Unprovisioning (use this option with -f<fname>)
-S	Saves updated parameters as factory defaults on FW image. This feature was implemented in CPT to save the updated parameter as the factory default. This saves the settings even after CMOS is cleared. **Note**: All the other Intel® ME settings – except Intel® MEBx password change – should be saved after you send the –s command.
-C	Commit Option (used with -f<filename>). The use of the commit option is the same as in FPT. Based on which parameter gets updated, the tool performs either Intel® ME reset, Global reset, or none. Reset gets performed at the very end (after all the parameters are updated). Global reset is easy to verify as you can see the system rebooting. To verify whether or not the Intel® ME reset was performed successfully: Run Meinfo –fwsts.
-EXP	Displays sample usage of this tool.

Option	Description
-VERBOSE <FILE>	Displays the tool's debug information.

Table 20: Required Reset for Updated Parameters

Parameter	Required Reset
Current MEBx password	No
New MEBx password	No
Manageability Feature selection (Enable AMT)	Global Reset
FW Local update	No
FW update qualifier	No
Power package	No
PID	Intel® ME Reset
PPS	Intel® ME Reset
PKIDNSSuffix	No
ConfiServerFQDN	No
ZeroTouchSetupEnabled	Intel® ME Reset
PreInstalledCertEnabled	Intel® ME Reset
UserDefinedCertEnabled	Intel® ME Reset
UserDefinedCertAdd	Intel® ME Reset
SolIderConfig	No
HostName	No
DomainName	No
DHCP	No
Idle Timeout	Intel® ME Reset
Provisioning Server Address	No
Provisioning server port	No
StaticIPv4Parameters	No
KVM State (Enable/Disable)	Intel® ME Reset
KVM Remote IT	Intel® ME Reset
KVM User	Intel® ME Reset

Note: This table might get updated in future.

8.3 USB Utility

Intel® UPDParam uses as an input a binary file that is created with a USB Utility (**USBfile.exe**).

8.3.1 Syntax

The following parameters can be set in **USBfile.exe** to generate the binary file.

```
USBfile -create <usb output file name> <current MEBx password>
   <new MEBx password> [-v 1|2|2.1] [-amt] [-v1file <version 1 outfile>]
   [-dns <DNS suffix>] [-fqdn <prov server fqdn>] [-ztc 0|1]
   [-dhcp 0|1] [-fwu 0|1] [-pm 0|1] [-fwuq 0|1|2] [-pp <16 byte GUID>]
   [-pspo <port number>] [-psadd <ip addr>] [-ito <4 byte of idle time
   out>] [-gen <num of records>] [-xml <xml file name>] [-pid <pid> -pps
   <pps>] [-hash <cert file name> <friendly name>] [-redir <n>] [-s4p
   <StacticIPv4Params>] [-hostname <hostname>] [-domname <domain name>]
   [-passPolicyFlag <0|1|2>]
```

Table 21: USB Utility Options

Option	Description
-v 1|2|2.1	Setup file version; 2.1 by default
-v1file <version 1 outfile>	Creates a version 1 setup file
-amt	Sets the manageability selection value to AMT
-dns <DNS suffix>	Sets the PKI DNS suffix name (up to length 255)
-fqdn <prov server fqdn>	String, maximum length 255
-ztc 0|1	Disables/enables PKI Configuration
-dhcp 0|1	Disables/enables DHCP
-fwu 0|1	Disables/enables FW local update
-pm 0|1	Enterprise/SMB provisioning mode
-fwuq 0|1|2	Always/Never/Restricted FW Update Qualifier
-pp <GUID>	Sets the power package. GUID should be in network order.
-pspo <port number>	Provision server port number
-psadd <ip addr>	IP address for provision server (e.g., 123.222.222.121)
-ito <4 byte of idle time out>	4 char of idle time out
-gen <n>	Number of records to create
-xml <xml file name>	Configuration xml fil
-pid <pid> -pps <pps>	PSK pair. This is ignored if -gen was chosen

Option	Description
-hash <certificate file name> <friendly name>	Computes and adds the hash of the given root certificate file. Up to three certficate hashes may be specified.
-redir <n>:	An integer that is calculated as follows: • bit 0 : 1 (Enable) or 0 (Disable) - SOL feature • bit 1 : 1 (Enable) or 0 (Disable) - IDER feature • bit 2 : 1 (Enable) or 0 (Disable) - Username/password authentication type of the SOL/IDER in the Intel® ME FW
-s4p <localHost:SubnetMask:GatewayAddr:DNSaddr:SecondaryDNSaddr>	E.g., 10.0.0.1:255.255.255.0:10.0.0.2:10.0.0.3:10.0.0.4 **Note**: The DHCP flag should be disabled.
-hostname <hostname>	ASCII representation of host name. Maximum length 63.
-domname <domain name>	Domain name. Maximum length 255
-vlan <0|1-VlanTag(1-4096)>	VlanStatus enable/disable, e.g., 0-4011
-passPolicyFlag <0|1|2>	Default/block in post/always open

For more details on how to use **USBfile.exe**, use the help command in the USB file utility. Once you have set all the parameters that you need to change (along with the current Intel MEBx password) **USBfile.exe** creates a binary file.

For example, you could enter the command `Usbkey.exe –create test.bin Admin Admin@98` (supposing the System current Intel® MEBx password is Admin). When you run **USBfile.exe**, this command creates a binary file named **test.bin** that sets the new password for Intel MEBx to Admin@98.

Once the binary file is created it is used by the UpdateParam tool as an input.

To use the binary file created by USBfile.exe:

- The binary file must contain the current Intel MEBx password.
- This tool (UpdateParam) must be in either pre-boot or in-boot mode in order to run:
  - Pre boot – you just flashed the image but haven't changed the password yet
  - In-boot – you have changed the password and entered Intel® MEBx
- BIOS does not send an EOP to ME

8.4 Output

If the binary file contains the right Intel® MEBx password, it proceeds to make the appropriate changes to the settings. It either returns a Success/Fail status for each of the parameters that are in the binary file or the tool returns an error code and error message and exits.

```
-------------------------------------------
Intel(R) UpdParam Version:          6.0.0.9290
Copyright (c) 2007-2009, Intel Corporation. All rights reserved.

-------------------------------------------
Chipset: Ibexpeak.
Validating Password... Failed.

Error 3037: The CurrentMEBx password is invalid.
```

Figure 43: UPDParam Error Message for Incorrect Password

Once the password validation is successfully completed, Intel® UPDParam changes the rest of the parameters as listed in the .bin file. If there is a failure changing/updating any of the parameters, Intel® UPDParam returns the error code and error message associated with the failure.

```
-------------------------------------------
Intel(R) UpdParam Version:          6.0.0.9290
Copyright (c) 2007-2009, Intel Corporation. All rights reserved.

-------------------------------------------
Chipset: Ibexpeak.

Validating Password... Success.
Updating Local Firmware Update Qualifier... Success.
Updating PID/PPS... Success.
Note: No change in ZTC status required.  Same as input.
Updating PID/PPS... Success.
Updating PKI DNS Suffix... Failed..
Error 3: Command is not permitted in current operating mode
Updating Config Server FQDN... Failed..
Error 1: AMT device internal error
Updating SOL/IDER Configuration... Success.
Setting FW update Parameter... Failed.
Setting Host Name... Success.
Setting Domain Name... Success.
Setting Idle Timeout... Success.
Setting Provisioning Mode... Success.
Setting ProServer Port Parameter... Success.
Setting IPv4 Parameters... Success.
Changing Password... Success.
```

Figure 44: UPDParam Error Message for Failure to Update Parameter(s)

Note: Error messages are displayed in red and warning messages are displayed in yellow.

Since Intel® UpdParam uses Intel® MEI to communicate with different components of the Intel® ME it also returns the Intel® MEI status.

A log file is also created that contains details about all the steps run. The log file can be found in the same folder as the executable.

8.5 Intel® ME Parameters Intel® UpdParam can Change

- Current MEBx password
- New MEBx password
- Manageability Feature selection (Enable AMT)
- FW Local update
- Power package
- PID
- PPS
- PKIDNSSuffix
- ConfiServerFQDN
- ZeroTouchSetupEnabled
- PreInstalledCertEnabled
- UserDefinedCertEnabled
- UserDefinedCertAdd
- SolIderConfig
- HostName
- DomainName
- DHCP
- Idle Timeout
- Provisioning Server Address
- Provisioning server port
- StaticIPv4Parameters
- KVM

8.6 Examples

`UpdParam –f <filename>`
Inputs the binary file and updates the parameters.

`UpdParam –f <filename> -v <CurrentMebxPwd>`
Inputs a binary file containing the MEBX current password entered at the command prompt.

`UpdParam –f <filename> -v <CurrentMebxPwd> -u`
Inputs a binary file containing the following:

- MEBX current password entered at the command prompt.
- An option to do partial unprovisioning.

`Updparam –r`
Performs a global reset.

`Updparam –h`
Displays the help screen.

Appendix A Fixed Offset Variables

This appendix only covers fixed offset variables that are directly available to FPT and FPTW. A complete list of fixed offset variables can be found in the *Firmware Variable Structures for Intel® Management Engine* (Document number 24571). All of the fixed offset variables have an ID and a name. The `-fov` 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 22: Fixed Offset Item Descriptions

Fixed Offset Name	FPT ID	Fixed Offset ID	Description	Data Length (in Bytes)	Expected Value	Secure	Reset Type
Non-Application Specific Fixed Offset Item Descriptions							
MEBx Password	1	0x0003	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	**No**	**ME**

<table>
<tr><th>Fixed Offset Name</th><th>FPT ID</th><th>Fixed Offset ID</th><th>Description</th><th>Data Length (in Bytes)</th><th>Expected Value</th><th>Secure</th><th>Reset Type</th></tr>
<tr><td>Default Power Package</td><td>3</td><td>0x0005</td><td>Default Power Package (Desktop):

Pkg1 - ON in S0
Pkg2 - ON in S0, ME Wake in S3, S4-5

Default Power Package (Mobile):

Pkg1 - ON in S0
Pkg2 - ON in S0, ME Wake in S3, S4-5 (AC-Only)</td><td>1</td><td>Package 1: 0x01
Package 2: 0x02</td><td>No</td><td>ME</td></tr>
<tr><td>AMT MEBx Password Rule Flag</td><td>6</td><td>0x0009</td><td>Controls Password Manager ability to allow AMT to set MEBx password from remote.</td><td>1</td><td>0 = Do not allow the MEBx password to be set from remote
1 = Allow the MEBx password to be set from remote</td><td>No</td><td>ME</td></tr>
<tr><td>OEM Permanent Disable</td><td>7</td><td>0x000A</td><td>UINT32 (little endian) value. This controls what features are permanently disabled by OEM.
Notes:
User must set all non-reserved bits to the value they want. There is NO ability to change features one at a time. This FOV sets OEM Permanent Disable for ALL features.
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 6 Series Chipset.
Examples:
• Intel® Q67 with Intel® AMT, KVM and PAVP 1.5 and TLS enabled: Bits: 0,2, 12, 18,21 set to ‘1’ (0x241005)</td><td>4</td><td>Feature Capable: 1
Feature Permanently disabled: 0
<table><tr><th>Bit</th><th>Description</th><th></th><th>Notes</th></tr><tr><td>31:22</td><td>Reserved</td><td></td><td></td></tr><tr><td>21</td><td>TLS</td><td></td><td></td></tr><tr><td>20:19</td><td>Reserved</td><td></td><td></td></tr><tr><td>18</td><td>KVM</td><td></td><td>2</td></tr><tr><td>17</td><td>Reserved</td><td></td><td></td></tr><tr><td>16</td><td>HAP</td><td></td><td></td></tr><tr><td>15:13</td><td>Reserved</td><td></td><td></td></tr><tr><td>12</td><td>PAVP</td><td></td><td></td></tr><tr><td>11:6</td><td>Reserved</td><td></td><td></td></tr><tr><td>5</td><td>Intel® AT</td><td></td><td></td></tr></table></td><td>No</td><td>GR</td></tr>
</table>

Fixed Offset Name	FPT ID	Fixed Offset ID	Description	Data Length (in Bytes)	Expected Value	Secure	Reset Type
			• Intel® QM67 with disabling Intel® AMT, PAVP 1.5 enabled: Bits: 12 set to '1' (0x1000) • Intel® HM67 with PAVP 1.5 and KVM and TLS disabled: Bits: 12, 18, 21 (0x241000)		<table><tr><td>4:3</td><td>Reserved</td><td></td><td></td></tr><tr><td>2</td><td>Manageability and Security Application</td><td></td><td>1</td></tr><tr><td>1</td><td>Reserved</td><td></td><td></td></tr><tr><td>0</td><td>Manageability Full</td><td></td><td>1</td></tr></table> 1. For corporate SKUs (Intel® Q67, Intel® QM67, Intel® QS67) 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'. Reserved bits should be set to '0'.		

<table>
<tr><th>Fixed Offset Name</th><th>FPT ID</th><th>Fixed Offset ID</th><th>Description</th><th>Data Length (in Bytes)</th><th>Expected Value</th><th>Secure</th><th>Reset Type</th></tr>
<tr>
<td>Feature Shipment Time State</td>
<td>8</td>
<td>0x000B</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.

Notes:

User must set all non-reserved bits to the value they want. There is NO ability to change features one at a time.

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 6 Series Chipset.

Examples:
• Intel® Q67 with Manageability Application, ship enabled: Bit: 2 set to ‘1’ (0x4)
• Intel® QM67 with disabling Manageability Application, Bit: 2 none set to ‘0’ (0x4)</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>31:3</td><td>Reserved</td><td></td></tr><tr><td>2</td><td>Manageability Full</td><td></td></tr><tr><td>1:0</td><td>Reserved</td><td></td></tr></table>
All other bits are reserved. Reserved bits should be set to 0.</td>
<td>No</td>
<td>GR</td>
</tr>
</table>

Fixed Offset Name	FPT ID	Fixed Offset ID	Description	Data Length (in Bytes)	Expected Value	Secure	Reset Type
SetWLANPower Well	35	0x000E	Sets which power will the board uses for WLAN cards	4	**0x80** = Disabled **0x82** = Sus Well **0x83** = ME Well The following option will be available if **Deep Sx Enable** is set to '**false**' on Mobile platforms. **0x84** = WLAN Power Controlled via SLP_M# || SPDA The following option will be available if **Deep Sx Enable** is set to '**true**' on Mobile platforms. **0x85 (default)** = WLAN Power Controlled via SLP_M# || SLP_ME_CSW_DEV #	**No**	**ME**
Idle Timeout – ME		0x2008	UINT16 representing the time in minutes for the Idle Timeout	2	Value **0x0000 < n <:0xFFFF**	**No**	**ME**
OEM_TAG	34	0x000F	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).	**No**	**ME**
Intel® AMT Related Fixed Offset Item Descriptions							
PID	9	0x2001	A 64 bit quantity made up of ASCII codes of some combination of 8 characters – capital alphabets (A–Z), and numbers (0–9). **Must be set along with PPS**.	8	Please see the PSK algorithm section on how to generate a valid PID.	**No**	**ME**
PPS	10	0x2002	A 256 bit quantity made up of ASCII codes of some combination of 32 characters – capital alphabets (A–Z), and numbers (0–9). **Must be set along with PID**.	32	Please see the PSK algorithm section on how to generate a valid PPS.	**No**	**ME**
Remote Configuration Enabled	13	0x2009	Remote Configuration Enable setting	1	Enabled:**0x01**	**No**	**ME**

Fixed Offset Name	FPT ID	Fixed Offset ID	Description	Data Length (in Bytes)	Expected Value	Secure	Reset Type
Customized Certificate Hash Entry 1	14	0x200B	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	**No**	**ME**
Customized Certificate Hash Entry 2	15	0x200C	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	**No**	**ME**
Customized Certificate Hash Entry 3	16	0x200D	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	**No**	**ME**
MEBx Password Change Policy	17	0x200E	The policy that controls MEBx password change over the network interface. Policy 0 – change allowed only if the password is still default Policy 1 – change allowed only during Setup and Configuration Policy 2 – change always allowed	1	Policy 0: **0x00** Policy 1: **0x01** Policy 2: **0x02**	**No**	**ME**
USBr Settings	24	0x2017	USBr feature settings	1	**b11** – Enabled **b10** - Disabled Bit mask: Bits 7:0 Bit 0..1 - EHCI 1 enabled (EHCI1Enabled) Bit 2..3 - EHCI 2 enabled (EHCI2Enabled) Bit 4..7 - reserved **At least one of the EHCIs should be enabled. This is not required but recommended.**	**No**	**GR**

Fixed Offset Name	FPT ID	Fixed Offset ID	Description	Data Length (in Bytes)	Expected Value	Secure	Reset Type
Host Based Setup and Configuration	32	0x2018	Enables configuring Host Based Setup and Configuration in Intel ® AMT	4	Disabled (Admin Control Mode only): **0x00** Enabled (Admin and Client Control Mode): **0x01**	**No**	**ME**

Fixed Offset Name	FPT ID	Fixed Offset ID	Description	Data Length (in Bytes)	Expected Value	Secure	Reset Type
Privacy Level	33	0x2019	Redirection (KVM, SOL, IDE-r) privacy level settings.	1	Default **0x01** Enhanced **0x02** Extreme **0x03** Default: SOL enabled = true IDER enabled = true KVM enabled = true Opt-in can be disabled= true KVM opt-in configurable remotely = true Enhanced: SOL enabled = true IDER enabled = true KVM enabled = true Opt-in can be disabled= false KVM opt-in configurable remotely = true Extreme SOL enabled = false IDER enabled = false KVM enabled = false Opt-in can be disabled= false KVM opt-in configurable remotely = N/A	**No**	**ME**

Fixed Offset Name	FPT ID	Fixed Offset ID	Description	Data Length (in Bytes)	Expected Value	Secure	Reset Type
Intel® AT-p Related FOV Item Descriptions							
AT FW Flash Protection Override Policy Hard GPIO33	27	0x6001	Indicates whether Hard-GPIO-33 is allowed, and under what conditions.	1	Always Allowed: **0x01** Allowed when AT NOT provisioned: **0x02**	**No**	**ME**
AT FW Flash Protection Override Policy Soft GPIO33	28	0x6002	Indicates whether Soft-GPIO-33 is allowed, and under what conditions.	1	Always Allowed: **0x01** Allowed when AT NOT provisioned: **0x02**	**No**	**ME**

Appendix B Tool Detail Error Codes

B.1 Common Error Code for all Tools

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

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

Error Code	Error Message	Response
101	No SPI flash device could be identified. Please verify if Fparts.txt has support for this part	Verify **Fparts.txt** contains device supported.
102	Failed to read the device ID from the SPI flash part	Verify **Fparts.txt** has correct values
103	There are no supported SPI flash devices installed. Check connectivity and orientation of SPI flash device	Verify **Fparts.txt** has correct values. Check SPI Device
104	The two SPI flash devices do not have compatible command sets	Verify both SPI devices on the system are compatible
105	An error occurred while writing to the write status register of the SPI flash device. This program will not be able to modify the SPI flash	Check SPI Device
8196	HECI message receive buffer memory allocation failed	
8193	Intel® ME Interface: Cannot locate Intel® ME device driver	
8199	Could not issue %s command message Where %s can be the following: Get FWU Version Get FWU Info Get FWU Feature State Block LAN Unblock LAN Intel® ME Kernel Test Intel® AMT Extended Test Intel® AMT Partial Test Intel® AMT Full Test Intel® AMT Graceful Test Intel® AMT Test Result Intel® ME Kernel Test Result Block Intel® AMT Full Test Get Intel® AMT Test Counter	Contact Intel

Error Code	Error Message	Response
8203	Unexpected result in %s command response Where %s can be the following: Get FWU Version Get FWU Info Get FWU Feature State Block LAN Unblock LAN Intel® ME Kernel Test Intel® AMT Extended Test Intel® AMT Partial Test Intel® AMT Full Test Intel® AMT Graceful Test Intel® AMT Test Result Intel® ME Kernel Test Result Block Intel® AMT Full Test Get Intel® AMT Test Counter	Contact Intel
8204	Intel® ME Interface: Unsupported message type	
8213	Requesting HECI receive buffer size is too small	

B.2 Firmware Update Errors

Error Code	Error Message	Explanation	Suggestion
0	Success		
1	An internal error to the AMT device has occurred	haltrcfg related	
2	AMT Status is not ready	haltrcfg related	
3	Invalid AMT Mode	haltrcfg related	
4	An internal error to the AMT device has occurred	haltrcfg related	
8199	Intel® ME Interface: ME Device not ready for data transmission		
8703	PLEASE REBOOT YOUR SYSTEM. Firmware update cannot be initiated without a reboot.	You can try to update firmware twice without a reboot	Reboot the system
8704	Firmware update operation not initiated due to a SKU mismatch		
8705	Firmware update not initiated due to version mismatch		
8706	Firmware update not initiated due to integrity failure or invalid FW image		
8707	Firmware update failed due to an internal error		

Error Code	Error Message	Explanation	Suggestion
8707	Firmware update failed due to an internal error. Firmware returns SAL notification error. Please try after Intel® ME-reset or re-flashing the Intel® ME image.		
8707	Firmware update failed due to an internal error. Firmware returns Audit policy error. Please try after Intel® ME-reset or re-flashing the Intel® ME image		
8707	Firmware update failed due to an internal error. Firmware failed to create fault tolerant partition. Please try after Intel® ME-reset or re-flashing the Intel® ME image		
8708	Firmware Update operation not initiated because a firmware update is already in progress		
8710	Firmware update tool failed due to insufficient memory		
8710	Firmware update failed due to insufficient memory		
8712	Firmware update failed due to authentication failure		
8713	Firmware update not initiated due to an invalid FW image		
8713	Firmware update not initiated due to an invalid FW image header		
8714	Firmware update not initiated due to file <file> open or read failure		
8714	Firmware update not initiated due to file open or read failure		
8715	Firmware update tool failed to connect iAMT through LMS, due to a HTTP operation failure		
8715	Firmware update tool failed to connect iAMT through LMS, due to a HTTP operation failure. Please verify the inputs (host, user, password, certificate, work mode, etc.).		
8716	Invalid usage		
8716	Invalid usage, -allowsv switch required to update the same version firmware		
8717	Firmware update not initiated due to invalid hostname specified		
8718	Update operation timed-out; cannot determine if the operation succeeded		

Error Code	Error Message	Explanation	Suggestion
8719	Firmware update cannot be initiated because Local Firmware update is disabled		
8720	Firmware update cannot be initiated because Secure Firmware update is disabled		
8722	Cannot receive the current version from the firmware after update		
8723	No Firmware update is happening		
8724	Update finished but version mismatch after the update		
8725	Failed to receive last update status from the firmware		
8727	Firmware update tool failed to get the firmware parameters		
8728	Firmware update iAMT communication failed, Failed to find certificate <certName> in certificate store		
8728	Firmware update iAMT communication failed, Failed to set HTTP certificate options <lastError>: <errMsg>		
8728	Firmware update iAMT communication failed, Failed to find certificate names		
8728	Firmware update iAMT communication failed, Failed to open system certificate store <lastError>: <errMsg>		
8728	Firmware update iAMT communication failed, HTTP request failed: Certificate rejected		
8734	Firmware update iAMT communication failed, WSMAN not supported		
8740	Unsupported Operating System		
8741	Firmware updated failed		
8743	Unknown or Unsupported Platform		
8744	OEM ID verification failed		
8745	Invalid UUID provided with the OEM ID switch		
8745	Firmware update cannot be initiated because the OEM ID provided is incorrect		
8746	Firmware update not initiated due to invalid image length		
8747	Firmware update not initiated due to an unavailable global buffer		

Error Code	Error Message	Explanation	Suggestion
8748	Firmware update not initiated due to invalid firmware parameters		
8749	Invalid version specified		
8758	Image blackListed		
8759	Internal Error		
8761	Firmware write file failure		
8762	Firmware read faile failure		
8763	Internal error		
8764	The image provided is not supported by the platform		
8766	Password did not match		
8767	Password exceeded maximum retry		
8768	Password not provided when required		
8769	Polling for FW Update failed		

B.3 Intel® MEManuf Errors

Error Codes	Error Messages
9248	Intel® ME internal communication error (BIST)
9249	Intel® ME internal communication error (AMT)
9251	Failed to create verbose log file %s Where %s is the log file name user specified
9256	Communication error between application and Intel® ME module (FW Update client)
9257	Internal error (Could not determine FW features information)
9261	Hibernation isn't supported by the OS, Intel® ME test cannot run
9267	Failed to establish communication with SPI flash interface
9268	Failed to load vsccommn.bin
9269	Zero flash device found for VSCC check
9270	Failed to load driver (PCI access for Windows)

Error Codes	Error Messages
	Tool needs to run with an administrator privilege account.
9271	Flash ID 0x%06X Intel® ME VSCC mismatch Programmed value of 0x%X doesn't match the recommended value of 0x%X See PCH SPI programming Guide for more details
9272	Flash ID 0x%06X Intel® ME VSCC value didn't find recommended value
9273	Intel® VE is disabled by PCH Softstrap
9276	Failed to read FW Status Register value 0x%X
9277	Intel® VE internal error
9278	Cannot locate hardware platform identification. This program cannot be run on the current platform. Unknown or unsupported hardware platform. or A %s hardware platform is detected. This program cannot be run on the current platform. Unknown or unsupported hardware platform. Where %s is the official name of the hardware platform
9279	SPI flash Intel® ME region is not locked
9280	Intel® Gbe/Intel® ME has read or write access to BIOS region
9281	SPI flash descriptor region is not locked
9282	BIOS has granted Intel® Gbe and/or Intel® ME access to its region
9283	Region access permissions don't match Intel recommended values
9284	Read firmware flash master region permission failure
9292	The SKU does not have any test assigned to be run -S4 Intel® AMT test only runs under Windows
9296	Intel® MEManuf Test Failed Use <VERBOSE> option for more details
9297	Intel® NAND needs to be enabled to perform the test
9299	Single flash part found, Flash Partition Boundary Address must be zero
9300	Flash Partition Boundary Address should be in between flash parts
9301	The two flash parts on this platform require different BIOS VSCC values
9302	Intel® NAND module test failed (feature not enabled)
9303	Memory allocation failed for checking variable "<Variable Name>"
9304	Variable "<Variable Name>" mismatch, actual value is - <Variable Value>

Error Codes	Error Messages
9305	Intel® ME firmware version mismatch, actual value is - <Version String> Intel® Gbe version mismatch, actual value is - <Version String> BIOS version mismatch, actual value is - <Version String>
9306	System UUID mismatch, actual value is - <UUID> System UUID mismatch, feature is not supported
9307	Intel® Wired/Wireless LAN MAC address mismatch, feature is not supported Intel® Wired/Wireless LAN MAC address mismatch, actual value is - <MAC Address>
9308	Flash Descriptor Override Strap is enabled
9309	End-Of-Post message is not sent
9310	Unable to determine Intel® ME Manufacturing Mode status Intel® ME is still in Manufacturing Mode
9311	Intel® ME test not started, error 0x%X returned
9312	Intel® ME test didn't finish within 30 seconds
9313	No Intel® ME test result to retrieve
9314	Intel® ME test result reports error(s)
9315	Intel® ME test is currently running, try later again
9316	Intel(R) ME can not run FULL Bist Possible Causes (1) power package 2 not supported, (2) This is mobile system with DC power
9317	No valid OEM ICC data programmed

B.4 Intel® MEInfo Errors

Error Code	Error Messages
9450	Communication error between application and Intel® AMT module (Manageability client)
9451	Communication error between application and Intel® AMT module (PTHI client)
9452	Communication error between application and Intel® ME module (iCLS client)
9455	Failed to read FW Status Register value 0x%X
9457	Failed to create verbose log file %s: Where %s is the log file name user specified
9458	Communication error between application and Intel® ME module (FW Update client)

Error Code	Error Messages
9459	Internal error (Could not determine FW features information)
9460	Cannot locate hardware platform identification This program cannot be run on the current platform. Unknown or unsupported hardware platform Or A %s hardware platform is detected This program cannot be run on the current platform. Unknown or unsupported hardware platform Where %s is the official name of the hardware platform
9467	Cannot use zero as SPI Flash ID index number
9468	Couldn't find a matching SPI Flash ID
9469	Access to SPI Flash device(s) failed
9470	Failed to load driver (PCI access for Windows) Tool needs to run with an administrator privilege account.
9471	Invalid feature name XXXXX: Where XXXXX is the feature name
9472	XXXXX feature was not available: Where XXXXX is the feature name
9473	XXXXX actual value is – YYYYY: Where XXXXX is the feature name Where YYYY is the feature value

B.5 FPT Errors

Error Code	Error	Response
1	Memory allocation error occurred	Make sure there is enough memory in the system
200	Invalid parameter value specified by the user. Use -? Option to see help.	Check the command line arguments supported by using the "-?"
201	FPT.exe cannot be run on the current platform. Please contact your vendor.	Contact your vendor.
202	Confirmation is not received from the user who performed the operation.	User input required
203	Flash is not blank. Data <data> found at address <address>.	Attempt to erase the device again
204	Data verify mismatch found at address <address>.	Reprogram the device

Error Code	Error	Response
205	Failure. Unexpected error occurred	File a sighting
206		PDR region exists
240	Access was denied while opening the file <file>	Check the permissions for the file
241	Access was denied while creating the file <file>	Check the permissions for the file
242	An unknown error occurred while opening the file <file>	Verify the file is not corrupt
243	An unknown error occurred while creating <file>	Verify the file is not corrupt
244	<name> is not a valid file name.	Check the filename
245	<file> file not found	Check file location
246	Failed to read the entire file into memory. File: <file>	Check system memory. Verify the file is not corrupt
247	Failed to write the entire flash contents to file	Check system memory
248	<file> file already exists	Delete the file that already exists
249	The file is longer than the flash area to write	Check file size
250	The file is smaller than the flash area to write	Check file size
251	Length of image file extends past the flash area	Check file size
252	Image file <file> not found	Check filename
253	<file> file does not exist	Check filename
254	Not able to open the file <file>	Check filename
255	Error occurred while reading the file <file>.	Check filename
256	Error occurred while writing to the file <file>	Check filename
280	Failed to disable write protection for the BIOS space!	Verify BIOS does not have write protection enabled
281	The Enable bit in the LPC RCBA register is not set. The value of this register cannot be used as the SPI BIOS base address	
282	Failed to get information about the installed flash devices	Check descriptor region access settings
283	Unable to write data to flash. Address <address>.	Check descriptor region access settings
284	Failed to load driver (PCI access for Windows). Tool needs to run with an administrator privilege account.	

Error Code	Error	Response
320	General Read failure	Attempt the command again. If symptom persists file a sighting
321	The address <address> is outside the boundaries of flash area	Check address
360	Invalid Block Erase Size value in <file>.	Check **fparts.txt** or its equivalent file
361	Invalid Write Granularity value in <file>	Check **fparts.txt** or its equivalent file
362	Invalid Enable Write Status Register Command value in <file>	Check fparts.txt or its equivalent file
363	Invalid Chip Erase Timeout value in <file>	Check fparts.txt or its equivalent file
400	Flash descriptor does not have correct signature	Verify file is not corrupt
401	An error occurred reading the flash mapping data	Check SPI device
402	An error occurred while reading the flash components data	Check SPI device
403	An error occurred while reading the flash region base/limit data	Check SPI device
404	An error occurred while reading the flash master access data	Check SPI device
405	An error occurred while reading the flash descriptor signature	Check SPI device
406	System booted in Non-Descriptor mode, but the flash appears to contain a valid signature	Check SPI device
407	User-provided Chip Erase Timeout has been reached. If the timeout value was set incorrectly the chip erase may still occur.	Check fparts.txt or its equivalent file
440	Invalid Fixed Offset variable name	Check Variable name
441	Invalid Fixed Offset variable Id	Check Variable ID
442	Param file <file> is already opened	Close parameter file
444	Invalid name or Id of FOV	Check variable name or ID
445	Invalid length of FOV value. Check FOV configuration file for correct length	Check length of FOV parameter in parmeter file
446	Password does not match the criteria	Password does not meet strong password requirements
447	Error occurred while reading FOV configuration file	
448	Invalid hash certificate file	Check hash certificate file

Error Code	Error	Response
449	Valid PID/PPS/Password records are not found in	Check PID/PPS/Password records and ensure that all 3 values exist
450	Invalid ME manufacturing mode done(Global locked) value entered	ME manufacturing mode done(Global locked) value is incorrect. Value should be 0x01 when modifying FOV parameters is no longer desired
451	Unable to get master base address from the descriptor	Check file integrity
452	Verification of End Of Manufacturing settings failed	Attempt command again. If problem persists, file a sighting
453	End Of Manufacturing Operation failure - Verification failure on ME manufacturing mode done (Global Locked)settings	Verify Intel® ME manufacturing mode done bit (global locked bit) has not been previously set
454	End Of Manufacturing Operation failure - Verification failure on Intel® ME Manuf counter	Verify Intel® MEManuf counter is valid
455	End Of Manufacturing Operation failure - Verification failure on Descriptor Lock settings	Verify Descriptor region is present and not corrupt
456	Invalid hexadecimal value entered for the FOV	Check value for FOV supplied
457	Parsing of file <file> failed	
480	The setup file header has an illegal UUID	UUID must be valid before Intel® ME is turned on
481	The setup file version is unsupported	Check setup file integrity
482	A record encountered that does not contain an entry with the Current Intel® MEBx password	Current Intel® MEBX password must be supplied
483	The given buffer length is invalid	Check buffer length value
484	The record chunk count cannot contain all of the setup file record data	Setup file number exceeded
485	The setup file header indicates that there are no valid records	Setup file has no valid records. Check setup file integrity
486	The given buffer is invalid	Check buffer value
487	A record entry with an invalid Module ID was encountered	Check record values. Check Setup file integrity
488	A record was encountered with an invalid record number	Check record values. Check Setup file integrity

Error Code	Error	Response
489	The setup file header contains an invalid module ID list	Check record values. Check Setup file integrity
490	The setup file header contains an invalid byte count	Check record values. Check Setup file integrity
491	The setup file record ID is not RECORD_IDENTIFIER_DATA_RECORD	Check record values. Check Setup file integrity
492	The list of data record entries is invalid	Check record values. Check Setup file integrity
493	The Current MEBx password is invalid	The Intel® MEBX password does not meet strong password requirements. See the Intel® MEBx Users Guide for information about strong password requirements.
494	The New MEBx password is invalid	The Intel® MEBX password does not meet strong password requirements. See the Intel® MEBx Users Guide for information about strong password requirements.
495	The PID is invalid	Check to see if value is valid. Check file integrity
496	The PPS is invalid	Check to see if value is valid. Check file integrity
497	The PID checksum failed	Check to see if value is valid. Check file integrity
498	The PPS checksum failed	Check to see if value is valid. Check file integrity
499	The data record is missing a CurrentIntel® MEBx password entry	Missing value is required
500	The data record is missing a NewIntel® MEBx password entry	Missing value is required
501	The data record is missing a PID entry	Missing value is required
502	The data record is missing a PPS entry	Missing value is required
503	The file <file> has an invalid entry	

B.6 UPDPARAM Errors:

Note: This section is not applicable to 1.5MB FW SKU.

Error Codes	Description
0	Success
3001	Invalid arguments specified
3002	Invalid Parameter value
3003	Error occurred while opening image file
3004	Parsing of image file failed
3005	HECI communication failed
3006	File does not exist
3007	Operating system is not supported
3008	AMT Internal error occurred
3009	User-defined certificate hash table is full
3010	Unable to start HECI
3011	Invalid input file name
3012	Chipset is not supported by the tool
3013	PID value is NULL
3014	PPS value is NULL
3015	Configuration Server FQDN value is NULL
3016	PKI DNS Suffix value is NULL
3017	Host Name value is NULL
3018	Domain Name value is NULL
3054	Unable to create Log file
3055	System failed to retrieve current firmware feature state
3056	Unable to save updated parameter as factory defaults on FW image
3057	Unable to complete FOV commit option

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

Tools' Options	Intel® ME manufacturing mode donebit		End of post		CF9GR locking	
	1	0	Yes	No	Yes	No
FPT -Greset	Not related	Not related	Not related	N/A Not related	Fail	Work
FPT –R	Depends on End of post status	Work	Depends on Intel® ME manufacturing mode donebit status	Work	Not related	Not related
Intel® MEMANUF –EOL config	Depends on End of post status	Work	Depends on Intel® ME manufacturing mode donebit status	Work	Not related	Not related
All options for UpdPARAM	Not related	Not related	Fail	Work	Not related	Not related

Appendix D SKU Features

TBD