

Intel® Management Engine

OEM Firmware Bring up Guide

Sept 2008

Rev 0.49

Intel Confidential



INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS OTHERWISE AGREED IN WRITING BY INTEL, THE INTEL PRODUCTS ARE NOT DESIGNED NOR INTENDED FOR ANY APPLICATION IN WHICH THE FAILURE OF THE INTEL PRODUCT COULD CREATE A SITUATION WHERE PERSONAL INJURY OR DEATH MAY OCCUR.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

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

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

This document contains information on products in the design phase of development.

All products, platforms, dates, and figures specified are preliminary based on current expectations, and are subject to change without notice. All dates specified are target dates, are provided for planning purposes only and are subject to change.

This document contains information on products in the design phase of development. Do not finalize a design with this information. Revised information will be published when the product is available. Verify with your local sales office that you have the latest datasheet before finalizing a design.

The Intel® Active Management Technology may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

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

Intel, Intel® Active Management Technology, Intel® Quiet System Technology and the Intel® logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

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

Copyright © 2005 - 2008, Intel Corporation. All rights reserved.



Contents

1 Introduction 8

1.1 Document scope 8

1.1.1 Important Assumptions 8

1.1.2 Known-good system configuration 8

2 Terminology 9

3 McCreary bring up overview 11

3.1 Flash image 11

3.2 Bring-up summary 12

4 Intel® QST Image Generation 15

4.1 Intel® QST INI files 15

4.2 Intel® QST Configuration Tool 16

4.2.1 Intel® QST Configuration DOS tool 16

4.2.2 Intel® QST Configuration Windows* tool 17

5 Final image generation 19

5.1 Image components 19

5.2 Flash image tool from VIP 20

5.3 Assemble the final image 20

5.3.1 Setup the flash image tool 20

5.4 Create the flash image 22

5.4.1 Descriptor region 22

5.4.2 ME region 30

5.4.3 QST Configuration 31

5.4.4 GbE region 31

5.4.5 BIOS region 32

5.4.6 Platform Data Region 32

5.4.7 ME Parameters 33

5.4.8 Save configuration 47

5.5 Building the flash image 48

5.6 Editing Image with FITC 49

Steps to edit the image 50

5.7 Programming the image onto the flash 54

5.8 Using a flash programmer 54

5.9 Using the flash programming tool 55

5.9.1 DOS environment 55

5.9.2 Load files onto the target platform 55

5.9.3 Query the flash devices 56

5.9.4 Program the image 56

5.9.5 Windows* environment 57

5.9.6 Load files onto the target platform 58

5.9.7 Query the flash devices 58

5.9.8 Program the image 59



6 BIOS Setup 60
6.1 Set Parameters 60
6.2 iTPM Parameters on BIOS 60

7 Intel® Management Engine BIOS Extension Setup 61
7.1 Intel® Management Engine BIOS Extension screen 61
7.1.1 Enabling Intel® AMT 61
7.1.2 Enabling ASF 2.0 64

8 Installing drivers 65
8.1 Driver identification 65

9 Basic Intel® AMT functional demonstration 67
9.1 Ping verification 67
9.2 Test Intel® AMT using the WebUI 69

10 Basic iTPM functional demonstration 70
10.1 Windows* XP Operating System Verification 70
10.2 Windows Vista* Operating System Verification 70
10.2.1 Method 1 70
10.2.2 Method 2 70

11 Intel® AMT Tools 71
11.1 Intel® AMT System Validation Tool – Local (AMTVTL) 71
11.1.1 AMTVTL from VIP 71
11.1.2 Executing AMTVTL 72
11.2 MEInfo tool 74
11.2.1 MEInfo DOS tool 74
11.2.2 MEInfo DOS tool from VIP 74
11.2.3 Executing MEInfo 74
11.2.4 MEInfo Windows tool 76
11.2.5 MEInfo Windows tool from VIP 76
11.2.6 Executing MEInfoWin 76
11.3 AMTVTR tool 78
11.3.1 AMTVTR tool from VIP 79
11.3.2 Executing AMTVTR 79
11.4 AMTVTR for testing System Defense features 81

12 Supplemental notes – CRB 82
12.1 BIOS Implementation 82
12.2 JUMPERS 82
12.2.1 Manufacturing Mode (Flash Override) jumper 82
12.2.2 Clear CMOS jumper 83
12.2.3 iTPM jumper 85
12.3 Testing SOL and IDE-R features 85
12.4 Final checklist 88



Figures

Figure 1. Snapshot of executing the Intel® QST Windows Configuration tool 18
Figure 2. Flash image tool – Configuration setup 21
Figure 3. ICH Strap 0 Configurations: 24
Figure 4. ICH Strap 1 Configurations: 25
Figure 5. ICH Strap 2 Configurations: 26
Figure 6. MCH Strap 0 Configurations: 27
Figure 7. VSCC Table Configuration Example 1:.....29
Figure 8. VSCC Table Configuration Example 2:.....29
Figure 9. ME Region 33
Figure 10. AMT Region.....38
Figure 11. iTPM Region 41
Figure 12. Setup and Configuration Region 43
Figure 13. Intel® Remote Connectivity Service Section 47
Figure 14. Manufacturing Mode jumper location: 83
Figure 15. CMOS Clear jumper location: 84
Figure 16. iTPM Enable jumper location: 85



Revision History

Revision Number	Description	Revision Date
0.1	Initial Release For Pre-Alpha	11/16/2007
0.2	Updated SPI flash reference	12/11/2007
0.3	Updated Bring-up guide with iTPM information	1/4/2008
0.31	Removed ICH Strap 0 SmBus parameter reference	1/31/2008
0.32	Adjusted Section 6.2 to properly reflect Eaglelake TPM BIOS options	2/1/2008
0.33	Updated FITC example images to match updated tool look and Danbury enabling section added	2/11/2008
0.34	Added information on enabling iTPM on the CRB Updated Intel® Quiet System Technology information to reflect changes to the .ini files	2/26/2008
0.35	Updated MAC address information to remove Dedicated ME MAC information not applicable for Eaglelake. Updated Danbury enabling information Updated FITc screen captures	3/11/2008
0.36	Updated FITc settings options to include additional ICH Soft Straps.	3/17/2008
0.37	Updated ICH Strap 0 information for the TCO Slave Select option	3/26/2008
0.38	Updated MCH Strap 0 information on the VE Enable Select option Updated VSCC programming section with screen capture examples	3/28/2008
0.39	Updated ICH Strap 1 configuration option for ICH VE Enable / Disable to properly enable Danbury	3/29/2008
0.40	Made changes to reflect firmware SKU naming	4/4/3008
0.41	Changed from Danbury to the official name Intel® Anti-Theft Technology Added Setup and Configuration section	4/23/2008
0.42	Section added for Intel® Remote Connectivity Service	5/7/2008
0.43	Updated image generation section with information on where to determine which firmware version should be used.	5/13/2008



Revision Number	Description	Revision Date
0.44	Updated the Intel® Remote Connectivity Service Removed Intel® Anti-Theft Technology sections Added information on new Auxiliary Well Select strap options.	5/27/2008
0.45	Added important notes that .NET 3.5 package is required Added note concerning ROM Bypass behavior	6/12/2008
0.46	Minor update to ROM Bypass note in MCH Configuration Figure 6	6/23/2008
0.47	Clarified AMTVTR and AMTVTL command line options and update System Defense usage for AMTVTR	7/8/2008
0.48	Minor changes to AMTVTR and AMTVTL command line options.	7/10/2008
0.49	Updated Terminology definitions Changed order of steps in Bring-up summary section to move Clear CMOS earlier in the flow. Added additional screen captures for Power Packages, Features and PET section under the Configuration Tab.	9/12/2008

§

1 Introduction

1.1 Document scope

This document describes the OEM-ODM bring up procedure for the McCreary/ICH10 platform with Intel® AMT 5.0/ASF 2.0/iTPM firmware. This document describes how to create an integrated image from the component images (BIOS, Gigabit Ethernet and Intel® ME images), program the integrated image onto the SPI flash devices. This document includes information on setting parameters required to enable Intel® AMT. This document also describes methods for testing the platform after Intel® AMT or iTPM is enabled to verify that Intel® AMT/iTPM is functioning.

This document is written as a superset, meaning it contains all the components for bringing up the Eaglelake/ICH10 Manageability Engine firmware. Based on desired components/sku, use only the sections that apply.

1.1.1 Important Assumptions

- This document assumes the reader has already reviewed the Readme and Release Notes documents included in the kit distribution.
- This document assumes **Static IP mode only** for simplicity.
- This document assumes the Intel BTX form factor CRB is the system on which you are performing the bring-up process.
- This document assumes you have already performed a fresh operating system installation on the system.

NOTE: This document assumes you have downloaded and properly installed the required .NET version 3.5 provided as separate download from the actual Firmware / Tools kit releases posted on VIP prior to loading the MEI / LMS Driver stack.

1.1.2 Known-good system configuration

1. Two 512-Megabyte DDR2 800Mhz memory modules
2. Penryn 2.4 Ghz CPU, 1333Mhz Front Side Bus

§



2 Terminology

Acronym	Explanation
Intel® AMT	Intel® Active Management Technology (Intel® AMT)
Intel® QST	Intel® Quiet System Technology (Intel® QST)
ASF	Alerting Standard Format
BIOS	Basic Input Output System
CRB	Customer Reference Board
DHCP	Dynamic Host Configuration Protocol
FIT	flash image tool
FPT	Flash Programming Tool
FW	Firmware
G3	A system state of Mechanical OFF where all power is disconnected from the system
GbE	Gigabit Ethernet
GUI	Graphical User Interface
Intel® MEI	Intel® Management Engine Interface
HW	Hardware
ICH	I/O Controller Hub
IDER	Integrated Drive Electronics Redirection – the ability to redirect input and output of an IDE device to a remote console.
IP	Internet Protocol
iTPM	Integrated TPM—1.2 TPM Compliant
LAN	Local Area Network
LMS	Local Manageability Service
MAC	Media Access Control
MCH	Memory Controller Hub
Intel® ME	Intel® Management Engine
Intel® MEBX	Intel® Management Engine BIOS Extension (Intel® MEBX)
NIC	Network Interface Card
NVM	Non-Volatile Memory
OEM	Original Equipment Manufacturer
SOL	Serial Over LAN

Acronym	Explanation
SPI Flash	Serial Peripheral Interface Flash
Static IP	An Internet Protocol address set manually (not by DHCP)
SUT	System Under Test
SW	Software
S0	A system state where power is applied to all Hardware devices and system is running normally.
S1, S2, S3	A system state where the host CPU is not running. However power is connected to the memory system.
S4	A system state where the host CPU and memory is not active
S5	A system state where all power to the host system is off. However the power cord is still connected.
Sx	All S states that are not S0.
ATX	
BTX	
PDR	

§



3 McCreary bring up overview

The flash image consists of multiple regions, as shown in the figure below.

Region	Content
0	Flash Descriptor
1	BIOS
2	ME
3	GbE
4	Platform Data

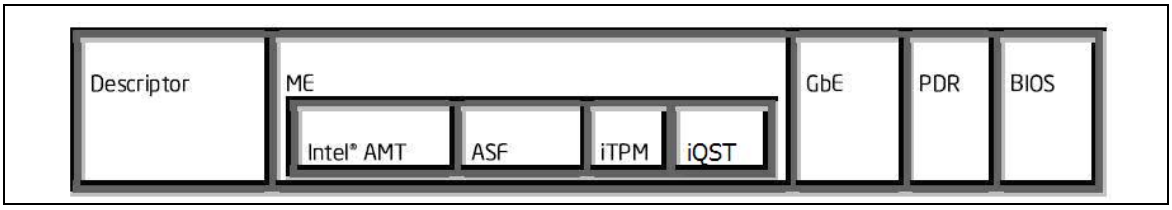
3.1 Flash image

The CRB ships with a total of two SPI flash devices installed. The image created below will span both devices as a single flash memory area. Bringing up a McCreary platform with the FW/SW kit involves the following steps:

1. Integrating the following images into a final, combined image:
   a. Flash Descriptor
   b. Intel® ME image
   c. BIOS image
   d. GbE image
   e. Platform Data Region
2. Programming this final image onto the SPI flash devices.
3. BIOS setup
4. Activating Intel® ME BIOS Extension (Intel® MEBx) Setup to configure the Management Engine and Intel® AMT.
5. Basic demonstration of Intel AMT/ASF 2.0/iTPM functionality.

Each of these steps is addressed in detail in the following pages.

The figure below shows the components of the final firmware image which is built from the component images.



3.2 Bring-up summary

This section summarizes the complete bring-up procedure, which is detailed in the remainder of the document.

Step #	Description	Input	Output
1	Download kit from VIP	N/A	The downloaded zip file containing the following folders – NVM Image, System Tools, Drivers and based on the FW supported - ASF 2.0 Tools and Intel® AMT Tools.
2	Create the Flash Image using the flash image tool, which integrates component binary images (BIOS, GbE, ME) into a single flash image.	BIOS.rom, ME.bin and GbE.bin	outimage.bin
3	Clear CRB CMOS settings	N/A	N/A
4	Disable ME by removing memory from the Channel A DIMM 0 socket. **Note: If the platform only has Channel A DIMM 0 populated you will need to move the over to the Channel B DIMM 0 position in order to boot the platform.**	N/A	N/A
5	Program the Flash Image onto the SPI flash devices on the target platform using the Flash Programming Tool.	outimage.bin	Flash image programmed successfully on the target platform.
6	Remove AC power cord from CRB Power supply.	N/A	N/A
7	Modify the settings in the BIOS setup screen.	Enabling Steps: 1. Select the **"Chipset"** option on the top of the CRB BIOS menu using	Fields in the BIOS screen have been modified to support Intel® AMT/ASF/iTPM on the



Step #	Description	Input	Output
		the ←→ left / right arrow keys. 2. Select the "**ME Subsystem**" menu using the up / down ↑↓ arrow keys and hit enter. 3. Hit Enter on the "**Load ME Optimal**" option at the top of the screen and change the setting to "**Yes**" 4. Select the **"F9"** Function key to load **"BIOS Optimal Defaults"**. **Note: If the "Load ME Optimal" is not set to "Yes" ME functionality will not be enabled when using the "F9" BIOS Optimal Defaults option.**	target platform.
8	Modify the settings in the Intel® ME BIOS Extensions (Intel® MEBx) setup	Configure settings for Intel® ME and Intel® AMT/ASF	Intel® ME and Intel® AMT/ASF configured on the target platform.
9	Install the Chipset.INF file and then install the supported drivers.	Chipset.INF, LAN, LMS and Intel® ME Interface Setup files.	Chipset.INF, LAN, LMS and Intel® ME Interface drivers installed on the target platform.
10	Demonstrate the basic functionality of Intel® AMT.	Ping Verification and WebUI	The management console is able to connect to the Intel® AMT system and you can verify that Intel® AMT has been successfully enabled on the target platform.
11	Check the functionality of Intel® AMT locally using the AMTVTL tool	AMTVTL.exe + username and password (if Kerberos* is not used)	Checks the functionality of a local Intel® Active Management Technology (AMT) 5.0 device.
12	Verify Intel® ME features in a DOS environment using MEInfo DOS Tool.	MEInfo.EXE (DOS)	Verifies Intel® ME FW is alive. Returns version data about BIOS, Intel® ME, iTPM, Intel® AMT Firmware and components.
13	Verify Intel® ME features in a Windows* environment using MEInfo Windows Tool	MEInfoWin.EXE (Windows*)	Verifies Intel® ME FW is alive. Returns version data about BIOS, Intel® ME, iTPM, Intel® AMT Firmware and components.
14	Check the functionality of Intel® AMT remotely using	- Host IP address	Returns success if Intel® AMT has been enabled

Step #	Description	Input	Output
	AMTFeaturesRemote tool.	- AMT username and password (if Kerberos is not used)	successfully, otherwise returns a failure.
15	Check for the System Defense functionality using the AMTSD tool.	- Host IP address - AMT username and password (only if Kerberos is not used)	Successfully checks for System Defense functionality on Intel® AMT machine.
16	Verify that .NET 3.5 has been installed on the client platform prior to installation of the MEI / LMS Drivers. **Note: If .NET 3.5 is not installed prior to installing the MEI / LMS drivers you will receive an error message and the IMSS monitoring application will fail to operate properly.**	N/A	N/A
17	Verify MEI/LMS Drivers Installed	Review Device Manager to verify HECI device and LMS are present.	Review properties to confirm that the correct MEI and LMS versions are loaded in the OS.

§



4 Intel® QST Image Generation

Note:

- This section is only applicable if Intel® QST is to be configured on the target platform. If not, please skip this section and continue with the next section.
- This section describes how to change the fan speed settings (Intel® QST settings), if default values are to be modified.
- INI files are provided in the kit with Intel-provided default values. If fan speed control values are to be modified, then change the default values provided in the INI files, before creating the Intel® QST image.

This section describes the creation of the Intel® QST binary image. This image will be integrated with the Intel ME firmware image along with the GbE and BIOS images using the Flash Image Tool to create the final flash image. The Intel® QST binary image is created using the Intel® QST Configuration tool. The Intel® QST Configuration tool takes in an INI (parameter initialization) file as input to create the Intel® QST binary image.

4.1 Intel® QST INI files

The INI files are used to create the Intel® QST binary image. The following table summarizes the location of the INI files:

INI File	Directory Location	Example	Use
QstCfgATXEL10D.ini	"Unzipped_folder"\Tools\ QST Tools\	C:\Digital_Office_5.0.0.xxxx\Tools\QST Tools\ QstCfgATXEL10D.ini	For ATX Corporate SKU Board
QstCfgBTXEL10D.ini	"Unzipped_folder"\Tools\ QST Tools\	C:\Digital_Office_5.0.0.xxxx\QST Tools\ QstCfgBTXEL10D.ini	For a BTX Corporate SKU Board
QstCfgATXEL10.ini	"Unzipped_folder"\Tools\ QST Tools\	C:\Digital_Office_5.0.0.xxxx\Tools\QST Tools\ QstCfgATXEL10.ini	For ATX Consumer SKU Board
QstCfgBTXEL10.ini	"Unzipped_folder"\Tools\ QST Tools\	C:\Digital_Office_5.0.0.xxxx\Tools\QST Tools\ QstCfgBTXEL10.ini	For a BTX Consumer SKU Board

4.2 Intel® QST Configuration Tool

The Intel® QST binary image can be created either using the DOS environment (as described in section 4.2.1) or using Windows* environment (as described in section 4.2.2).

4.2.1 Intel® QST Configuration DOS tool

The Intel® QST Configuration DOS Tool can be accessed under the following directory:

- "Unzipped_folder" Tools\QST Tools\

  (Example: C:\ C:\Digital_Office_5.0.0.xxxx\Tools\QST Tools\QstCfgD.exe)

Executing the following command at the DOS prompt generates the Intel® QST binary file:

- QstCfgD <INI File Pathname> [{-w|-o} <Binary File Pathname>]

  Where:

  INI File Pathname : Provides a pathname for the INI file

  -w [Binary File Pathname] : Specifies that the payload is to be written to a file. If the file already exists, the tool will ask for confirmation before overwriting it.

  -o [Binary File Pathname] : Specifies that the payload is to be written to a file. If the file already exists, it will be overwritten.

  (Example: QstCfgD.exe QstCfgBTXEL10D.ini -w QSTBinary.bin)

The Intel® QST binary image (Example: QSTBinary.bin) is created in the specified folder.

IMPORTANT:

- The Intel® QST Configuration DOS tool requires DOS4GW.exe in the same directory as the tool executable. Download DOS4GW.exe from the following URL: http://www.scene.org/file.php?file=%2Fresources%2Fdos4gw.exe&fileinfo.



Download the ‘dos4gw.exe’ file and save it in the same directory as ‘QstCfgD.exe’:

(Example: “Unzipped_folder”\Tools\QST Tools\)

Note:

- Choose the appropriate INI file for executing the command above depending on the type of the board (ATX or BTX).
- For more details on this tool, refer to the document – ‘Intel Quiet System Technology Configuration and Tuning Manual’. Please contact your Intel representative to get access to this document.

4.2.2 Intel® QST Configuration Windows* tool

The Intel® QST Configuration Windows* tool is located in the following directory:

- “Unzipped_folder”\Tools\QST Tools\

  (Example: C:\Digital_Office_5.0.0.xxxx\Tools\QST Tools\QstCfg.exe)

Executing the following command at the Windows command prompt generates the Intel® QST binary file:

- QstCfg <INI File Pathname> [{-w|-o} <Binary File Pathname>]

Where:

INI File Pathname : Provides a pathname for the INI file

-w [Binary File Pathname] : Specifies that the payload is to be written to

a file. If the file already exists, the tool will ask for confirmation before overwriting it.

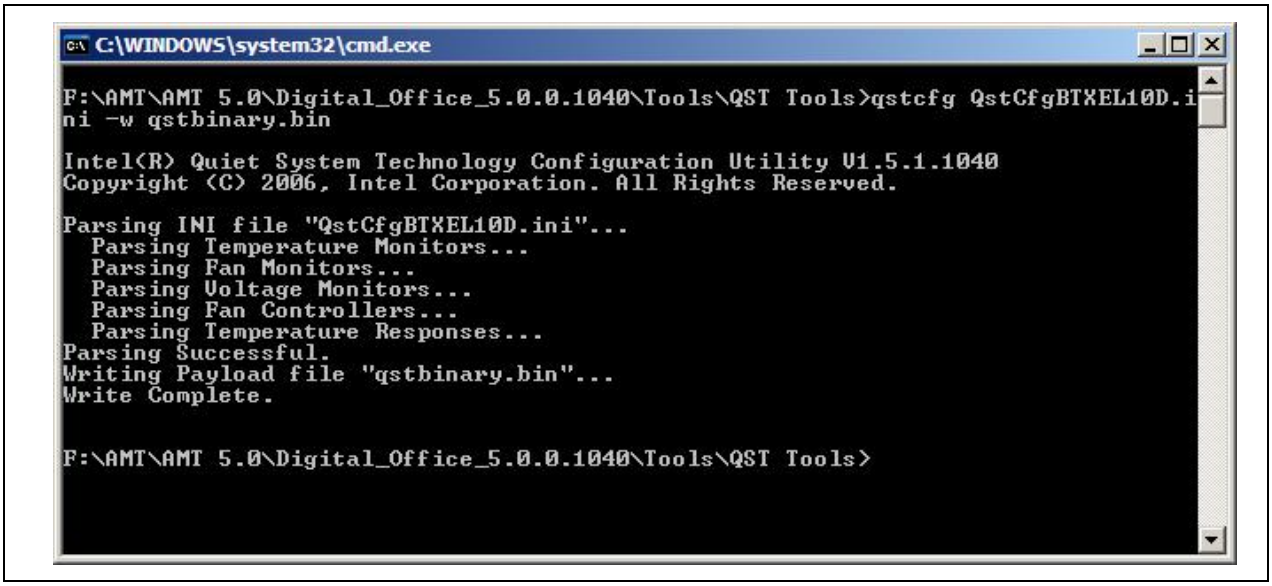
-o [Binary File Pathname] : Specifies that the payload is to be written to

a file. If the file already exists, it will be overwritten.

(Example: QstCfg.exe QstCfgBTXEL10D.ini -w QSTBinary.bin)

The Intel® QST binary image (Example: QSTBinary.bin) is created in the specified folder.

Figure 1. Snapshot of executing the Intel® QST Windows Configuration tool



Note:

- Choose the appropriate INI file for executing the Intel® QST Configuration Tool depending on the type of the CRB board (ATX or BTX).
- For more details on this tool, refer to the document – '*Intel® Quiet System Technology Configuration and Tuning Manual'*. Please contact your Intel representative for access to this document.
- The Intel® QST Configuration GUI Tool (QSTCT_GUI.exe), also included in this distribution, can be used to create the Intel® QST binary image in Windows* environment. See the document '*Intel® Quiet System Technology Configuration and Tuning Manual'* for a description of the GUI tool.
- Downloading of Microsoft VC runtime libraries is required for the Intel® QST Configuration GUI tool to work under Microsoft* Windows* 2000 Operating System. For more details, please refer to the document 'ReleaseNote.pdf' (Download from the same source as the kit is downloaded from)

§



5 Final image generation

The flash image tool is used to create the final flash image file that you will use to program the SPI flash devices on the CRB system board.

The purpose of this tool is to create and configure a flash image for the McCreary/ICH10 platform. The following binary files are provided as input to this tool:

- BIOS (.rom)
- Intel® ME firmware (.bin)
- GbE (.bin)
- QSTBinary.bin

The tool usage is explained in detail in the following sections.

5.1 Image components

The various image components can be downloaded from VIP:

(https://platformsw.intel.com/)

The kit will contain one component archive as shown in the table below.

Archive name	Intel® AMTFirmware	ASF2.0 Firmware
Digital_Office_5.0.0.xxxx.zip	✓	✓

NOTE: The "xxxx" shown herein throughout in this text will actually be a 4-digit number corresponding to the build number. Example: Digital_Office_5.0.0.1002.zip

Download the required archive and expand it to a folder using an archive utility, such as WinZIP. (Example: E:\)

- Example: If the 'Digital_Office_5.0.0.xxxx' archive has been downloaded, then the contents will be downloaded into the following directory structure - \Digital_Office_5.0.0.xxxx
- Refer to the remainder of this document to create the proper build directory structure.

Under the root of the above directory hierarchy, the actual composite binary image files you will use to build the final image (BIOS, ME, GbE) are located under the **NVM Image** directory. (Example: E:\Digital_Office_5.0.0.xxxx\NVM Image) When you configure the flash image tool to build your final image, you will set an environment variable called $SourceDir to point to this directory for the source images. All environment variables required by the flash image tool are covered in detail in the sections below.

Note: Be sure to check in both the 'Message of the Week' and kit 'Release Notes' for further information on which Firmware should be used for the specific GMCH silicon revision to ensure proper ME operation.

5.2 Flash image tool from VIP

The flash image tool can be accessed from the following directory:

- "Unzipped_folder"\Tools\System Tools\Flash Image Tool\

(Example: E:\Digital_Office_5.0.0.xxxx\Tools\System Tools\Flash Image Tool\fitc.exe)

The 'Flash Image Tool' folder contains the following files:

a. fitc.exe
b. fitc.ini
c. fitctmpl.xml
d. newfiletmpl.xml

Note:

- For details on each of the files listed above, please refer to the document: 'Intel® Flash Image Tool.pdf'.
- The flash image tool main executable is 'fitc.exe'. It is required that the following files be in the same directory as fitc.exe:
  - fitctmpl.xml
  - newfiletmpl.xml
- The flash image tool will NOT execute properly if the above files are missing.
- This tool runs in Microsoft Windows* XP only.

5.3 Assemble the final image

5.3.1 Setup the flash image tool

Follow the steps given below to set up the flash image tool.

1. Go to the directory containing 'fitc.exe'.
2. Double click on 'fitc.exe'.
3. Read through the license agreement, accept it, and then click 'OK'.
4. The flash image tool opens a new configuration file entitled:

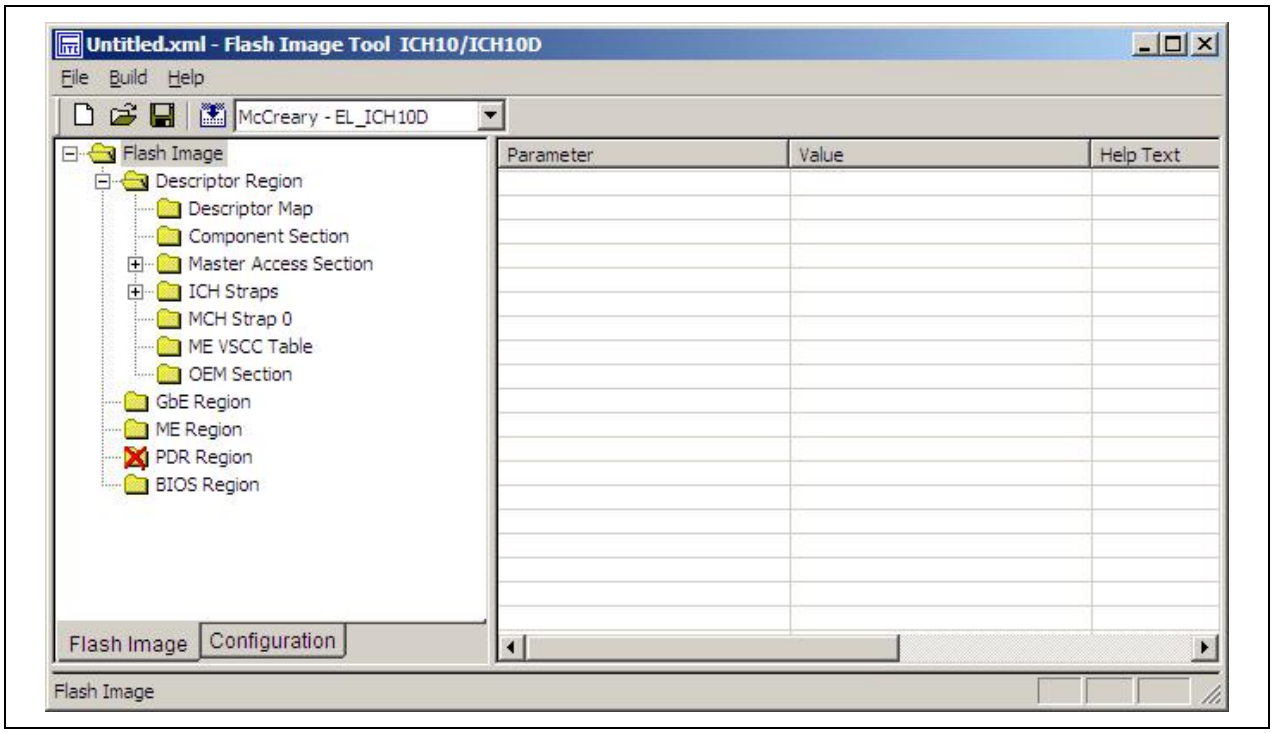


'Untitled.xml – Flash Image Tool'.

5. Click on the 'Build' menu.
6. Select 'Environment Variables…'
7. the Environment Variables window opens. Set the following values:
   a. Set '$WorkingDir:' to: .\Working
   b. Set '$SourceDir:' to: .\..\..\..\NVM Image (this corresponds to the directory path containing the NVM images.)
   c. Set '$DestDir:' to: .\Output
   d. For '$UserVar1:', '$UserVar2:' and '$UserVar3:' do not change the default values. (Retain the values as '.')
   e. Click 'OK', and the 'Environment Variables' window will close.
8. Click on the 'Build' menu at the top of the current window.
9. Select 'Build Settings…'
10. In the new window opened, entitled 'Build Settings':
    a. Set the 'Output path:' to include the final flash image name.

       (Example: $DestDir\outimage.bin) Note: You may opt to name your output file so it includes the build number, i.e. "$DestDir\1002_Outimage.bin".
    b. Leave 'Generate intermediate build files' as checked.
    c. Leave 'Build compact image (ignores the flash component sizes in the descriptor)' as unchecked.
    d. Set 'Flash Block/Sector Erase Size:' to the sector size of the target flash device. (Example: 4KB for 16 Mbit SST device with ID SST25VF016B)
    e. Click 'OK', and the 'Build Settings' window will close.

Figure 2. Flash image tool – Configuration setup





5.4 Create the flash image

This section details the steps required to configure the flash image tool and to set the various required parameters, before the final flash image is created from its component images, i.e. BIOS image, GbE image and the Intel® ME image. The following regions are configured using the flash image tool:

5.4.1 Descriptor region

1. Expand the 'Descriptor Region' node in the left pane of the main window.
2. Click on 'Descriptor Map' node under the 'Descriptor Region' node.
   a. All of the parameters of the 'Descriptor Map' section will appear in the list in the right pane.
   b. Double click on 'Number of Flash Components' parameter.
   c. A dialog box will open, entitled 'Number of Flash Components'.
   d. Set this value to the number of flash components installed on the target board. Valid values are 1 or 2. Note: the McCreary CRB comes with 2 flash devices installed, so you would select 2 for this setting. Other platforms may have only one flash device.
   e. Click 'OK' and the dialog box closes.
3. In the left pane, open the 'Descriptor Region' node and Click the 'Descriptor region length' in the right pane and set it to 0x2000.
4. In the left pane, under the 'Descriptor Region' node and click to select the 'Component Section' node.
   a. All of the parameters of the 'Component Section' will appear in the list in the right pane.
   b. Double-click on 'Fast read clock frequency' parameter.
   Note: Verify that the SPI parts you are using in your platform design will support this setting.
   c. A dialog box will open, entitled 'Fast read clock frequency'.
   d. Select '33MHz' from the drop-down list.
   e. Click 'OK' and the dialog box closes.
   f. Double-click the 'Fast read support' parameter.
   g. A dialog box will open, entitled 'Fast read support'.
   h. Select the 'true' radio-button if it is not already selected.
   i. Click 'OK' and the dialog box closes.
   j. Double-click the 'Flash component 1 density' parameter.
   k. A dialog box will open, entitled 'Flash component 1 density'.
   l. Select the correct component size from the drop-down list and set it to the size of the flash part being used, according to the following criteria:

      Flash component density:
      - If the value for 'Number of Flash Components' is set to 1, then 'Flash Component 1 density' should be set to 4MB.
      - If the value for 'Number of Flash Components' is set to 2, then 'Flash Component 1 density' should be set to 2MB.
      - If you are using the Intel McCreary CRB (it has two flash components) or another platform that has two 2MB flash components, set each component to 2MB. Examine the target system board to determine the proper settings.

Note: The below steps m, n and o are applicable only if the 'Number of Flash Components' is set as 2.



m. Double-click on 'Flash component 2 density' parameter.
n. A dialog box will open, entitled 'Flash component 2 density'.
o. Select the correct component density from the drop-down list (Example: 2MB).
p. All the remaining parameters for the 'Component section' are retained as default values.

5. Expand the 'Descriptor Region' node and then expand the 'Master Access Section' node in the left pane.
   a. Click to select the 'CPU/BIOS' node under the 'Master Access Section' node.
      i. All the parameters of the 'CPU/BIOS' section are displayed in the right pane.
      ii. Double-click the 'Read access' parameter.
      iii. A dialog box opens, entitled 'Read access'
      iv. **Enter the value '0xFF' here.
      v. Click 'OK' and the dialog box closes.
      vi. Double-click on 'Write access' parameter.
      vii. A dialog box opens, entitled 'Write access'
      viii. **Enter the value '0xFF' here. (**Important:** the value 0xFF - full read/write access - is for engineering purposes only – for production, you will need to change this to WRITE access permission only.)
      ix. Click 'OK' and the dialog box closes.
   b. In the left pane, open the 'Master Access Section' node and click to select the 'Manageability Engine (ME)' node.
      i. All the parameters of the 'Manageability Engine (ME)' section are displayed in the right pane.
      ii. Double-click the 'Read access' parameter.
      iii. A dialog box opens, entitled 'Read access'
      iv. **Enter the value '0xFF' here.
      v. Click 'OK' and the dialog box closes.
      vi. Double-click the 'Write access' parameter.
      vii. A dialog box opens, entitled 'Write access'
      viii. **Enter the value '0xFF' here.
      ix. Click 'OK' and the dialog box closes.
   c. In the left pane, open the 'Master Access Section' node and click to select the 'GbE LAN' node.
      i. All the parameters of the 'GbE LAN' section are displayed in the right pane.
      ii. Double-click the 'Read access' parameter.
      iii. A dialog box opens, entitled 'Read access'
      iv. **Enter the value '0xFF' here.
      v. Click 'OK' and the dialog box closes.
      vi. Double-click the 'Write access' parameter.
      vii. A dialog box opens, entitled 'Write access'
      viii. **Enter the value '0xFF' here.
      ix. Click 'OK' and the dialog box closes.

Note:

- All the above steps are necessary and must be followed. If any of the steps are missed, there may be chances of locking down the flash chip, in which case further updating will not be possible and an SPI flash part replacement will be required.
- ****** - The value '0xFF' gives all the regions (Example: Intel® ME, GbE and BIOS regions) complete read/write access to a particular region (Example: BIOS region). This value set is only for the engineers/developers.

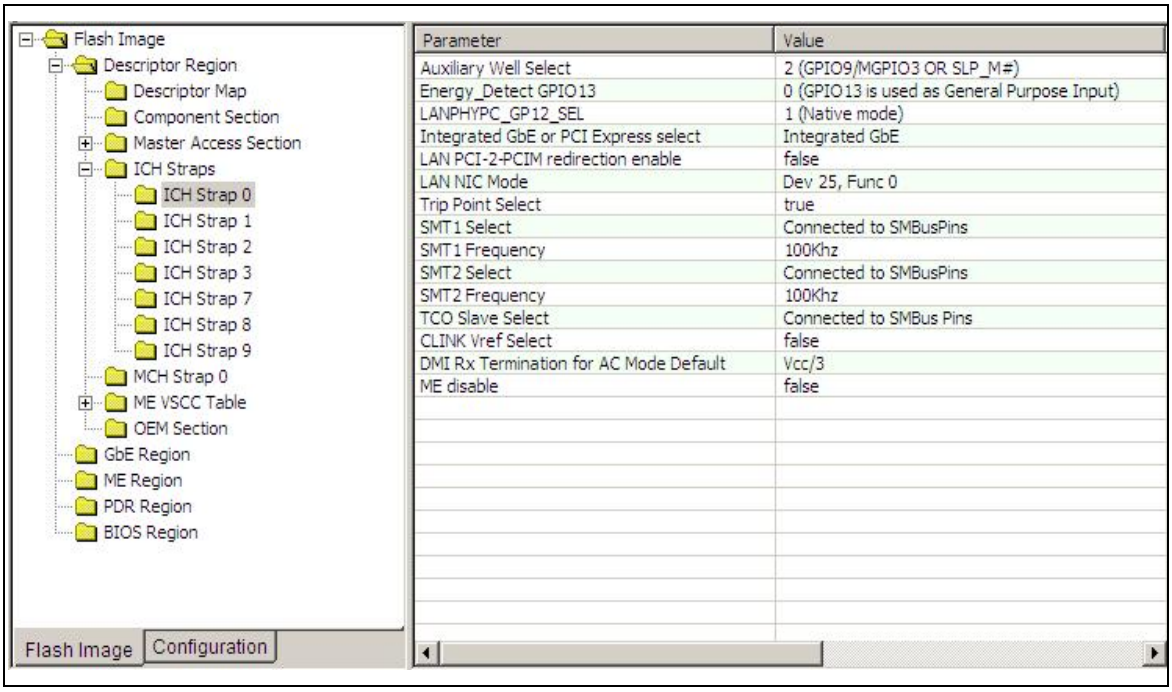
- For more details on the read/write access and how to determine the values for each permission setting, please refer to the document – 'Intel® flash image tool.pdf'.

6. Expand the 'ICH Straps' node and then select the 'ICH Strap 0' node in the tree on the left side of the main window.

   a. Ensure that the following options are configured:

	Option	Value
1.	Auxiliary Well Select	- '**2 (GPIO9/MGPIO3 OR SLP_M#)**'
2.	LANPHYPC_GP12_SEL	- '**1 (Native mode)**'
3.	Integrated GbE or PCI Express select	- '**Integrated GbE**'
4.	Trip Point Select	- '**true**'
5.	TCO Slave Select	- '**Connected to SMBus Pins**'

Figure 3. ICH Strap 0 Configurations:



Note: 'TCO Slave Select' settings For Intel CRB/EV platforms the recommendation is to set this strap to the '**Connected to SMBus Pins'**.

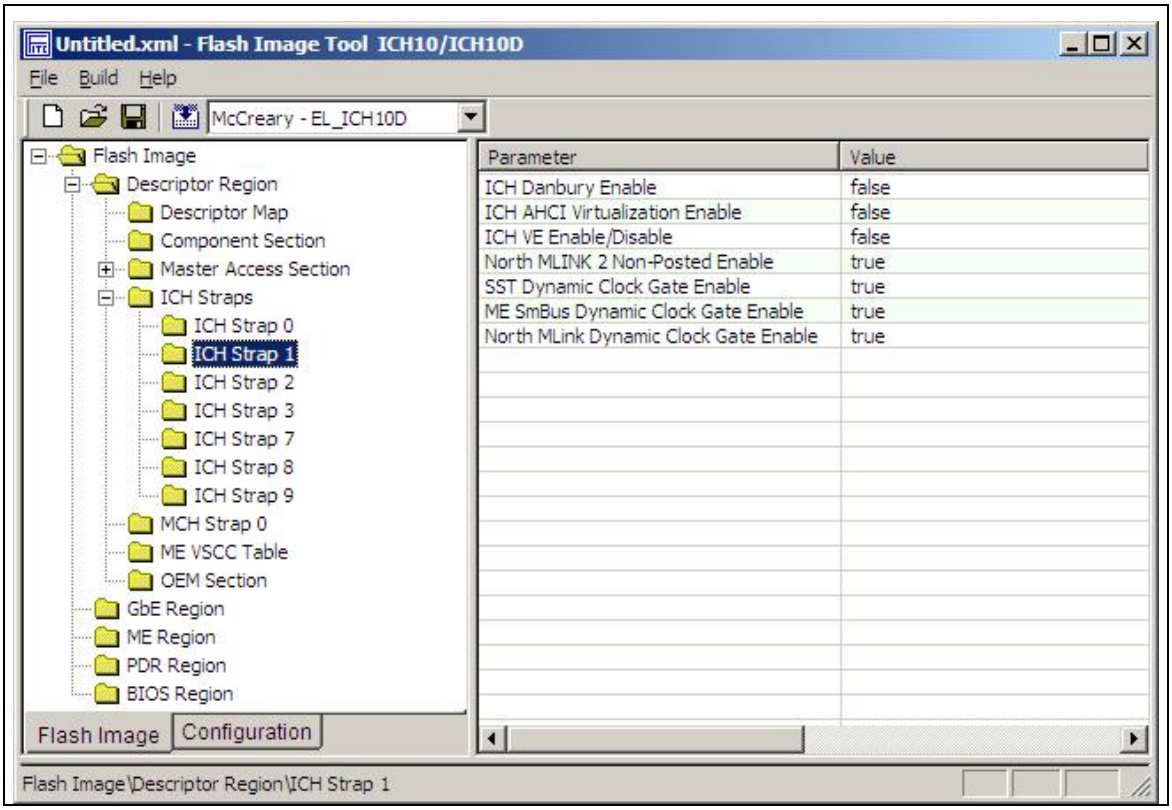
Note: 'Auxiliary Well Select' For Intel CRB/EV platforms the recommendation is to set this strap in ICH Soft Strap 0 to '**2 (GPIO9/MGPIO3 OR SLP_M#)**'. If this value is not set users may experience a D0 hang on resume from S3/S4 Moff state.

7. Click to select the 'ICH Strap 1' node in the tree on the left side of the main window.

   a. All the parameters of the 'ICH Strap 1' section are displayed in the right pane.
   b. Ensure that these options are configured as follows:

1.ICH Danbury Enable	'**false**'
2.ICH AHCI Virtualization Enable	'**false**'
3.ICH VE Enable/Disable	'**false**'
4.North MLINK 2 Non-Posted Enable	'**true**'
5.SST Dynamic Clock Gate Enable	'**true**'
6.ME SmBus Dynamic Clock Gate Enable	'**true**'
7.North MLink Dynamic Clock Gate Enable	'**true**'

Figure 4. ICH Strap 1 Configurations:



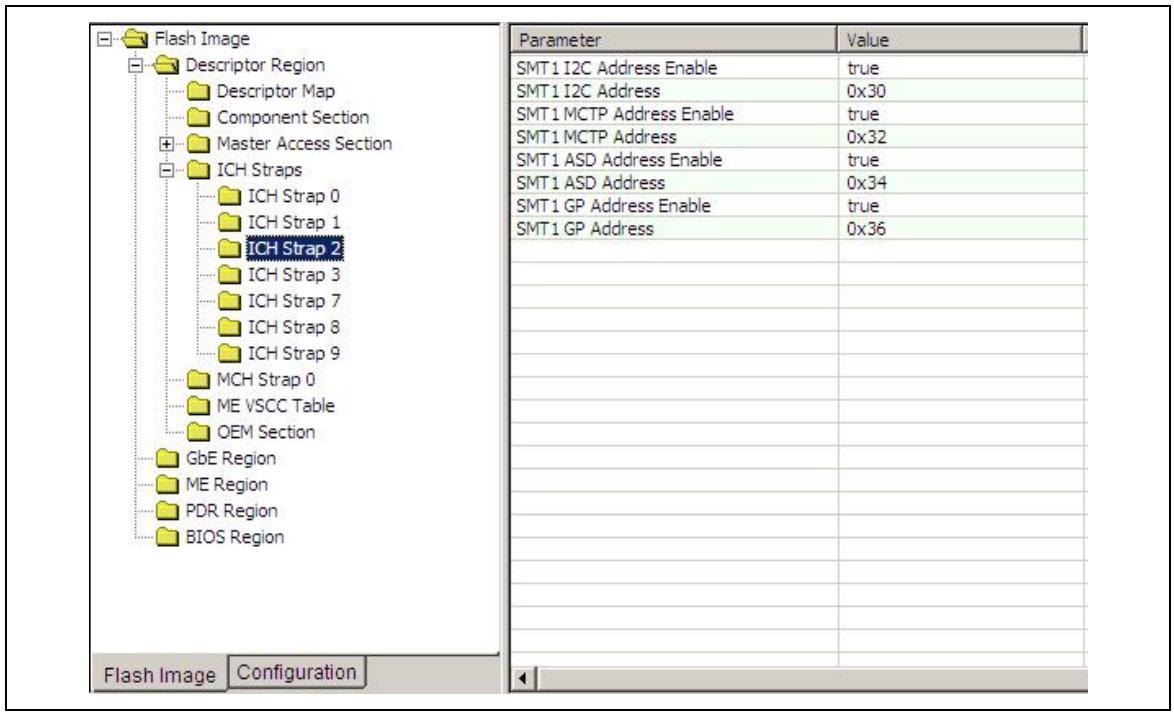
Note: The ICH VE Enable/Disable soft strap must be configured to **'true'** on silicon stepping prior to B0.

8. Click to select the 'ICH Strap 2' node in the tree on the left side of the main window.
   Ensure that all Address Enable options are set to '**true**'

   Configure the values for each of the addresses to the values shown:

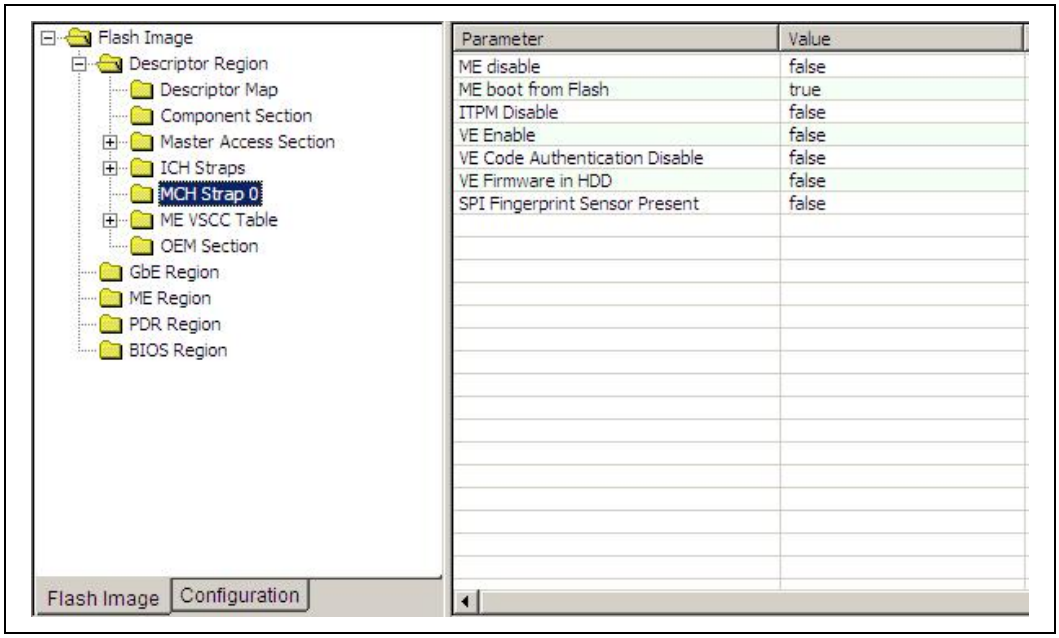
   a. I2C Address **0x30**
   b. MCTP Address **0x32**
   c. ASD Address **0x34**
   d. GP Address **0x36**

Figure 5. ICH Strap 2 Configurations:



9. Click to select the 'MCH Strap 0' node in the tree on the left side of the main window.

Figure 6. MCH Strap 0 Configurations:



a. All the parameters of the 'MCH Strap 0' section are displayed in the right pane.
b. Double-click on 'ME boot from Flash' parameter.
c. A dialog box opens, entitled 'ME boot from Flash'.
d. Select the 'true' radio button if it is not already selected.

Note: This option will not be selectable until you have selected a firmware image under the ME Region section. Additionally if By-pass firmware is selected this options will be automatically selected as 'false' and the option will be grayed and cannot be changed.

e. Click 'OK' and the dialog box closes.

Note:

- Step 6 is to enable 'ROM By-pass mode'. A ROM by-pass image has been provided in this kit (Firmware image).
- For more details on the ROM by-pass mode, please refer to the document 'ReleaseNote.pdf' (which can be downloaded from the same web page where you obtained the kit)

NOTE: ROM by-pass mode is only possible on a non-production fused MCH parts and is also dependent on the use of a Bypass enabled firmware image. This strap will have no affect if it is set on production fused MCH parts.

10. Click to select the VSCC Table node in the tree on the left side of the main window.

    a. Add the SPI flash component to your VSCC table:

    (Example 1 for SST SPI flash)

    i. Right click on VSCC Table to add entry name: SST 25VF016B
    ii. Left Click to select the entry name created
    iii. Configure Vendor ID: 0xBF
    iv. Configure Device ID 0: 0x25
    v. Configure Device ID 1: 0x41
    vi. Configure VSCC register value: 0x20092009

    (Example 2 for Blanchard SPI flash)

    vii. Right click on VSCC Table to add entry name: Intel Blanchard 4MB
    viii. Left Click to select the entry name created
    ix. Configure Vendor ID: 0x89
    x. Configure Device ID 0: 0x89
    xi. Configure Device ID 1: 0x11
    xii. Configure VSCC register value: 0x401ED81F

Figure 7. VSCC Table Configuration Example 1:

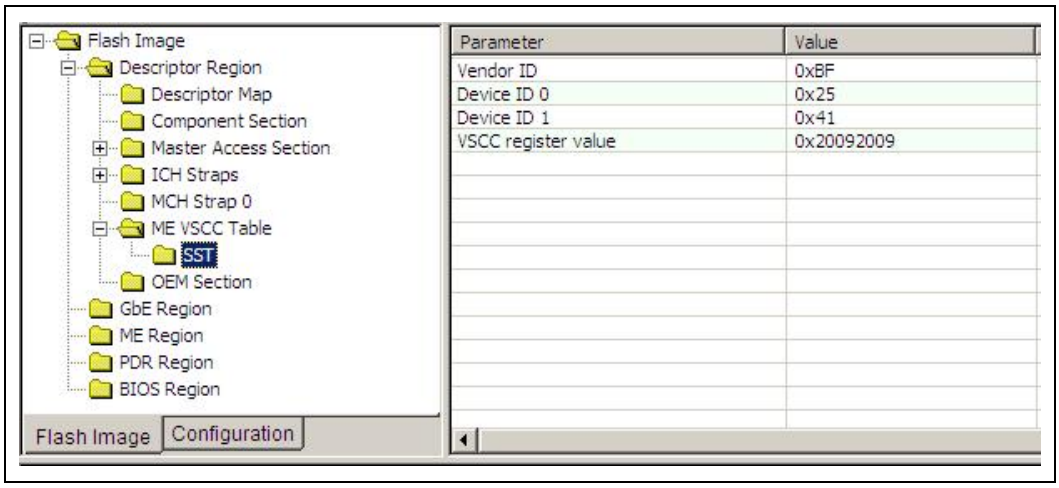
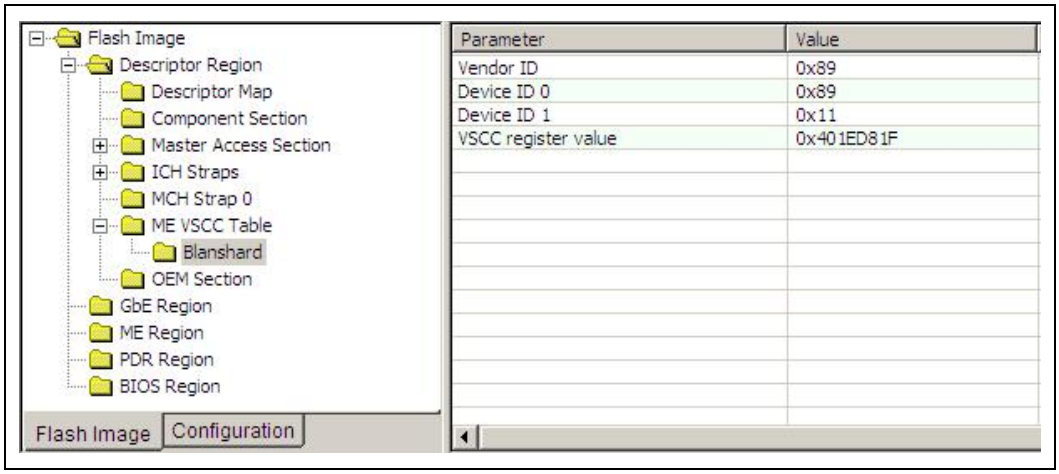
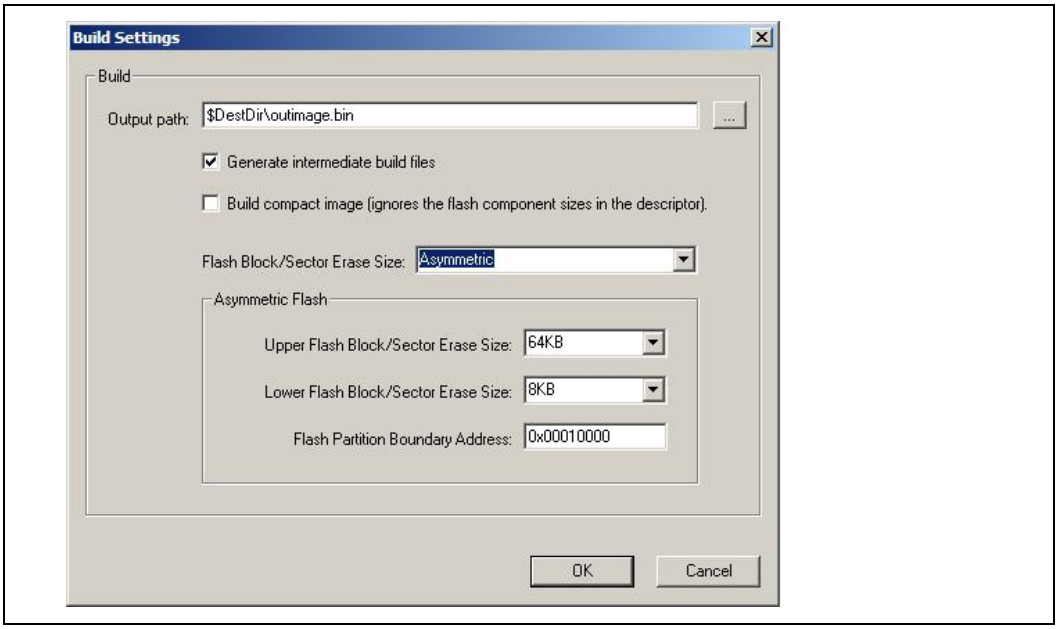


Figure 8. VSCC Table Configuration Example 2:



(**Note:** For Blanchard you will also need to make the following changes in your 'Build settings' from the 'Build' pull down menu.

a. Configure 'Flash Block/Sector Erase Size:' to Asymmetric
b. Make sure that the 'Upper Flash Block/Sector Erase Size:' is set to 64KB
c. Make sure that the 'Lower Flash Block/Sector Erase Size:' is set to 8KB.



Note: For values to be entered for the above setup, please refer to the *Intel® I/O Controller Hub 10 (ICH10) Family EDS, SPI ICH-SPI programming guide* and the SPI flash's datasheet. Vendor ID, Device ID 0 and Device ID 1 are all derived from the output of the JEDEC ID command which can be found the in flash's datasheet. In the ICH10 EDS, *22.2.8.2 VSCC0—Vendor Specific Component Capabilities 0* describes the 32 bit VSCC register value.

11. No modifications to the 'OEM Section' are needed – it should be left to default.
12.

5.4.2 ME region

1. Click to select the 'ME Region' node in the left pane of the main window.
2. All the parameters of the 'ME Region' section are displayed in the right pane.
3. Double-click the 'Binary file' parameter.
4. A dialog box opens, entitled 'Binary file'.
5. Click the browse button (…) and open the directory where the Intel® ME image is located (Example: E:\Digital_Office_5.0.0.xxxx\NVM Image\Firmware).

Note:

- There will be several Intel® ME images will be present in this directory.
- The first image (EL_ICH10_SKUx_BYP_ME_PreProduction.BIN) is the Intel® ME image to be selected for creating the flash image for Pre-Production Silicon using the flash image tool.
- The second image (EL_ICH10_SKUx_ME_PreProduction.BIN) is the Intel® ME image to be selected for creating the flash image for Production Silicon using the flash image tool.
- Bypass image files (Example: EL_ICH10_SKUx_BYP_ME_PreProduction.BIN) are generally only needed



on early pre-production A0 silicon stepping CRB boards and are usually not needed for later silicon steppings. Check the **Readme** and **Release Notes** included with each kit release for specific information on this.

- Remaining images (Example: EL_ICH10_SKUx_ME_UPD_PreProduction.BIN) are Intel® ME image files to be used only with the Firmware Update Tool (FWUpdate Tool). For more details on this tool, please refer to the document – 'Intel® AMT Tools User Guide.pdf', located in the folder: "Unzipped_folder"\iAMT Tools\ (Example: E:\Digital_Office_5.0.0.xxxx\Tools\iAMT Tools\Intel® AMT Tools User Guide.pdf)

1. Select the Intel® ME image (Example: EL_ICH10_SKUx_ME_PreProduction.BIN) from this location, and click 'Open'.
2. Click 'OK' and the dialog box closes.
3. The remaining parameters are left with default values.

5.4.3 QST Configuration

1. Click the 'ME Region' node in the left pane of the main window.
2. Double-click the 'Intel® QST config' file option.
3. A dialog box will appear, entitled 'This is the Intel® QST configuration binary file'.
4. Click the Browse (…) button and open the directory where the QST binary which was created previously in the '**Intel® QST Image Generation**' section of the guide is located. (Example: E:\Digital_Office_5.0.0.xxxx\Tools\QST Tools\QSTBinary.bin).
5. Click 'OK' and the dialog box closes.

Note: This is only needed if the image being created for the platform is using Intel® QST.

5.4.4 GbE region

1. Click the 'GbE Region' node in the left pane of the main window.
2. All the parameters of the 'GbE Region' are displayed in the right pane.
3. Double-click the 'GbE LAN region length and set it to 0x4000.
4. Double-click the 'Binary input file' parameter.
5. A dialog box will appear, entitled 'Binary input file'.
6. Click the Browse (…) button and open the directory where the GbE image is located. (Example: E:\Digital_Office_5.0.0.xxxx\NVM Image\GbE).
7. Select the GbE image file (Example: NAHUM2_NINEVEH_A0_1_1.bin) from this location and click on 'Open'.
8. Click 'OK' and the dialog box closes.
9. Double-click the 'MAC address' parameter.
10. A dialog box will appear, entitled, 'MAC address'.
11. Enter the 48-bit MAC address (Physical Address) for the host adapter on the target platform. (Example: 00 12 34 56 78 91)
12. Click 'OK' and the dialog box closes.

Note:

- The GbE MAC Address set in the steps above must be unique for all systems on a network.

13. The remaining parameters are left with default values.

5.4.5 BIOS region

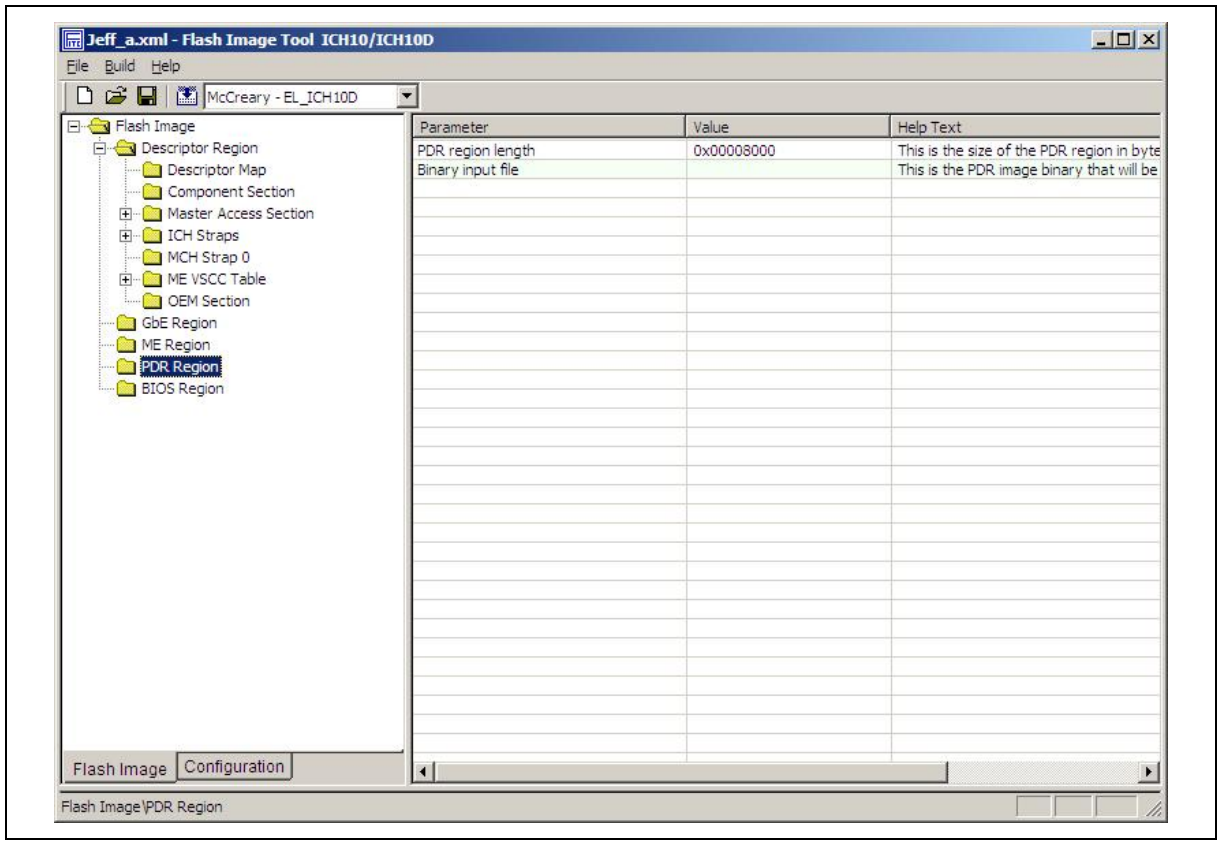
1) Click to select the 'BIOS Region' node in the left pane.
2) All the parameters of the 'BIOS Region' are displayed in the right pane.
3) Double-click the 'Binary input file' parameter.
4) A dialog box will appear, entitled 'Binary input file'.
5) Click to select the Browse (…) button and open the directory where the BIOS image (Intel Reference BIOS) is located. (Example: E:\Digital_Office_5.0.0.xxxx\NVM Image\BIOS).
6) Select the BIOS image file from this location and click **OPEN** (Example: "CGELIA33.ROM") where "xxx" is actually the revision number.
7) Click 'OK' and the dialog box closes.
8) The remaining parameters are left with default values.

5.4.6 Platform Data Region

Ensure that the Platform Data Region is enabled. This option is disabled by default in Fitc.

1) Right Click on the PDR Region and select the Enable Region option.
2) Set PDR Region length = 0x8000

Note: This region is reserved for VA 3.0 and is not to be used for OEM storage.

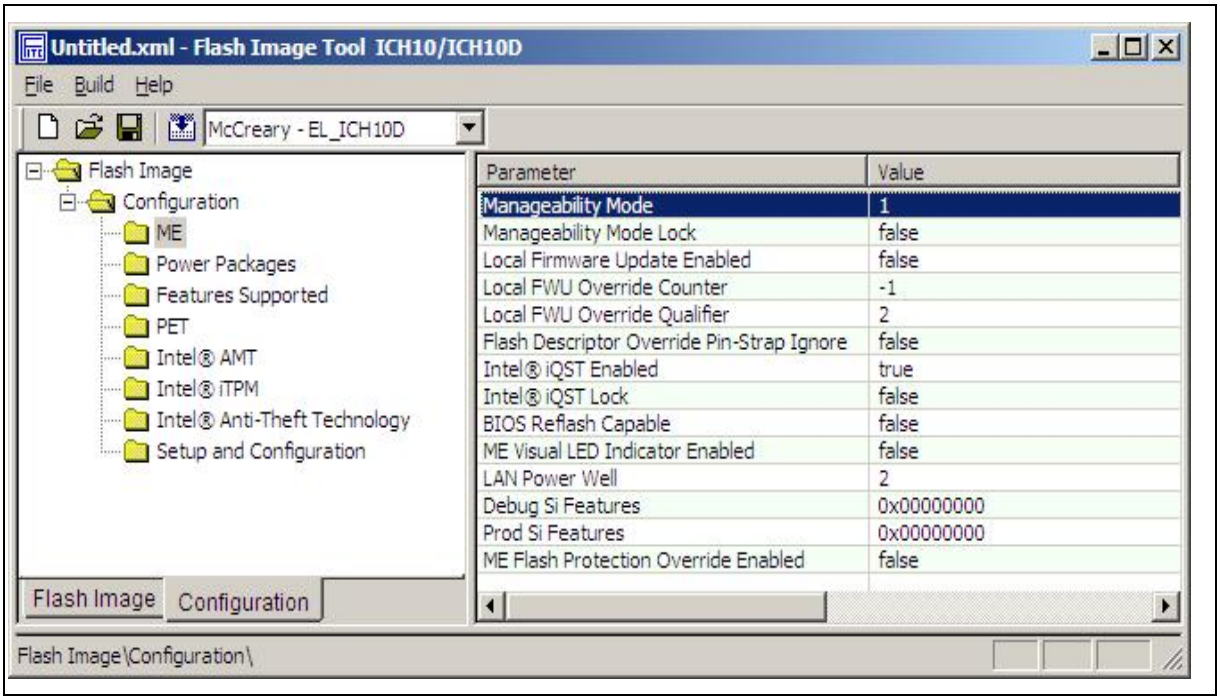




5.4.7 ME Parameters

The ME section allows the user to specify the manageability features. The parameter values can be found in the help text next to it. The default settings are shown in Figure 9 below.

Figure 9. ME Region



5.4.7.1 Temporary Firmware Update parameters

When **Local FWU Override Counter** has a value between 1 and 255, firmware updates are allowed even if updates are disabled in the ME BIOS Extension settings. After the flash is programmed, each time the machine restarts it causes **Local FWU Override Counter** to be decremented. When **Local FWU Override Counter** reaches 0, firmware updates are no longer allowed if they are not enabled by the ME BIOS Extension settings.

Note: The restart that takes place after the flash memory has been programmed also causes **Local FWU Override Counter** to be decremented. Therefore if you want to enable updating the firmware **N** times, you need to assign **Local FWU Override Counter** the initial value **N+1**.

If **Local FWU Override Counter** is set to -1 and **Local Firmware Override Qualifier** is set to 0, firmware updates are always allowed regardless of the settings in the ME BIOS extension

The following table shows the possible value combinations for the two variables. To enable local firmware updates, make sure both variables are assigned the correct values.

Firmware Override Update Variables

	Local FWU Override Qualifier = 0 (zero)	Local FWU Override Qualifier = 1 (one)	Local FWU Override Qualifier = 2 (two)
Local FWU Override counter = 0 (zero)	Local Firmware Updates NOT Allowed	Local Firmware Updates NOT Allowed	Local Firmware Updates NOT Allowed
Local FWU Override Counter = -1 (minus one)	Local Firmware Updates Allowed	Local Firmware Updates NOT Allowed	Local Firmware Updates Allowed only until ME is configured
Local FWU Override Counter = 0<n<255	Local Firmware Updates Allowed	Local Firmware Updates Allowed	Local Firmware Updates Allowed

5.4.7.2 Si features parameters

These options allow for Firmware status codes to be output to the MDDD ME Port80 module.

Si Features Options:

Parameter	Description	Default Value
Debug Si Features	Setting this option to a '1' value will configure the FW to output extended debugging FW status codes to the MDDD module.	0x00000000
Prod Si Features	Setting this option to a '1' value will configure the FW to output FW status codes to the MDDD module.	0x00000000

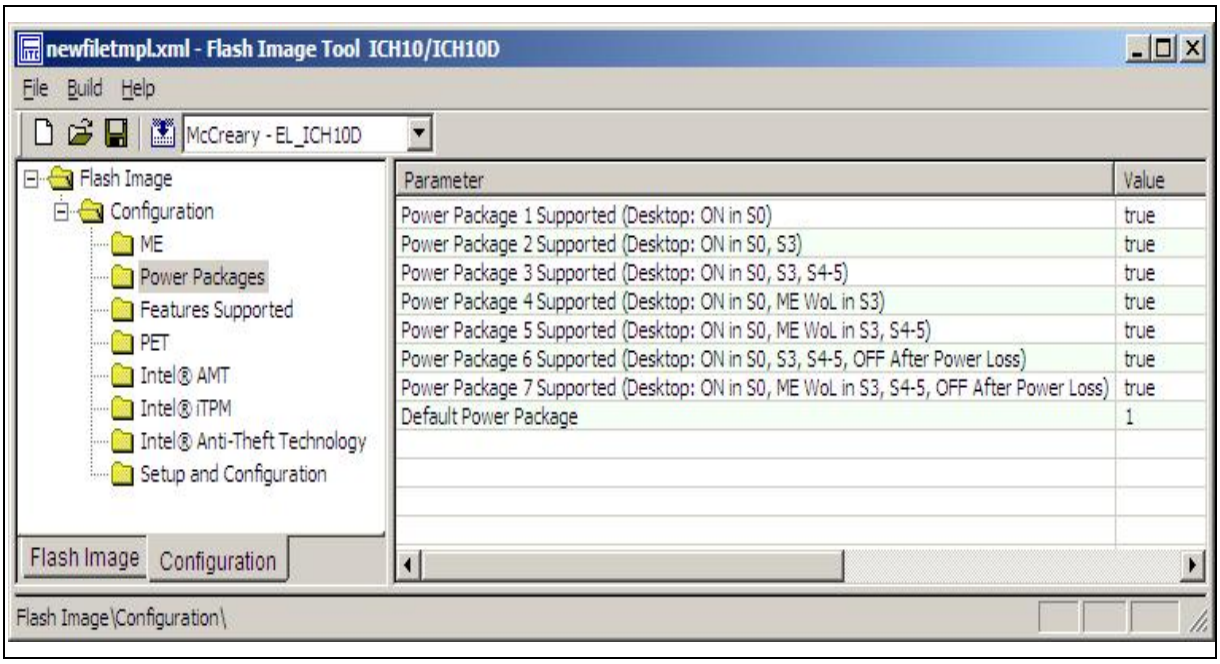
Note: These options should only be enabled if you are using the MDDD module for Firmware related debugging. Leaving these options enabled in a production level Firmware image will cause significant boot performance impact.

5.4.7.3 Power Packages

This section allows the user to specify which Power Packages will be supported on the platform and which package will be used by the ME as the default.

The default settings are shown below.

Power Packages Region

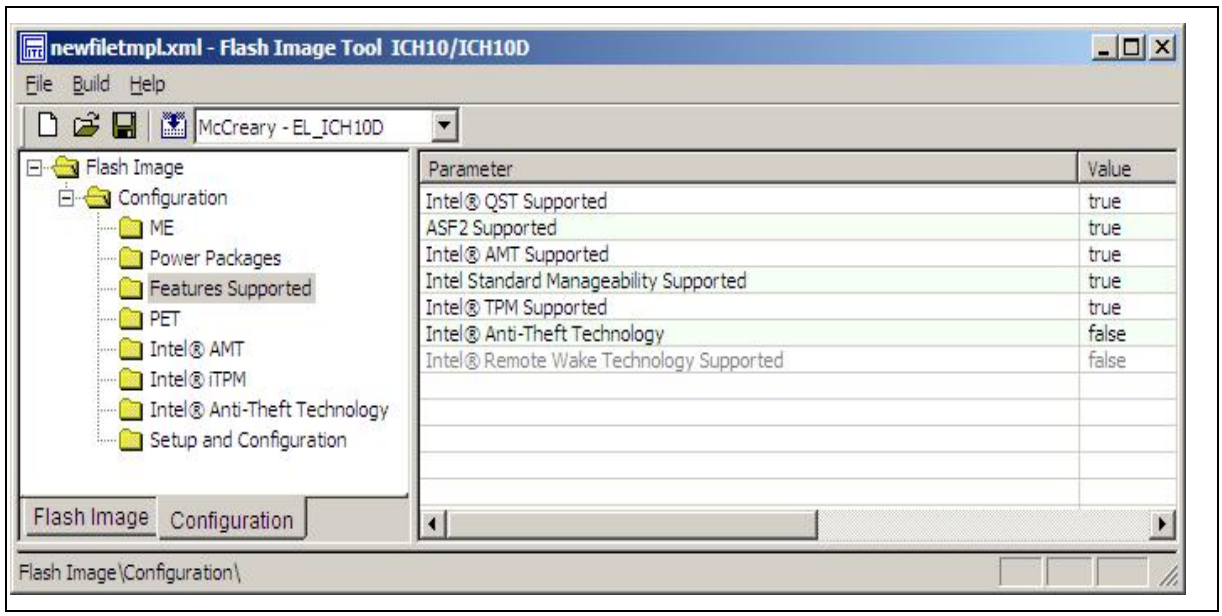


5.4.7.4 Features Supported

This section allows the user to specify which features will be supported on the in the Firmware image.

The default settings are shown below.

Feature Supported Region

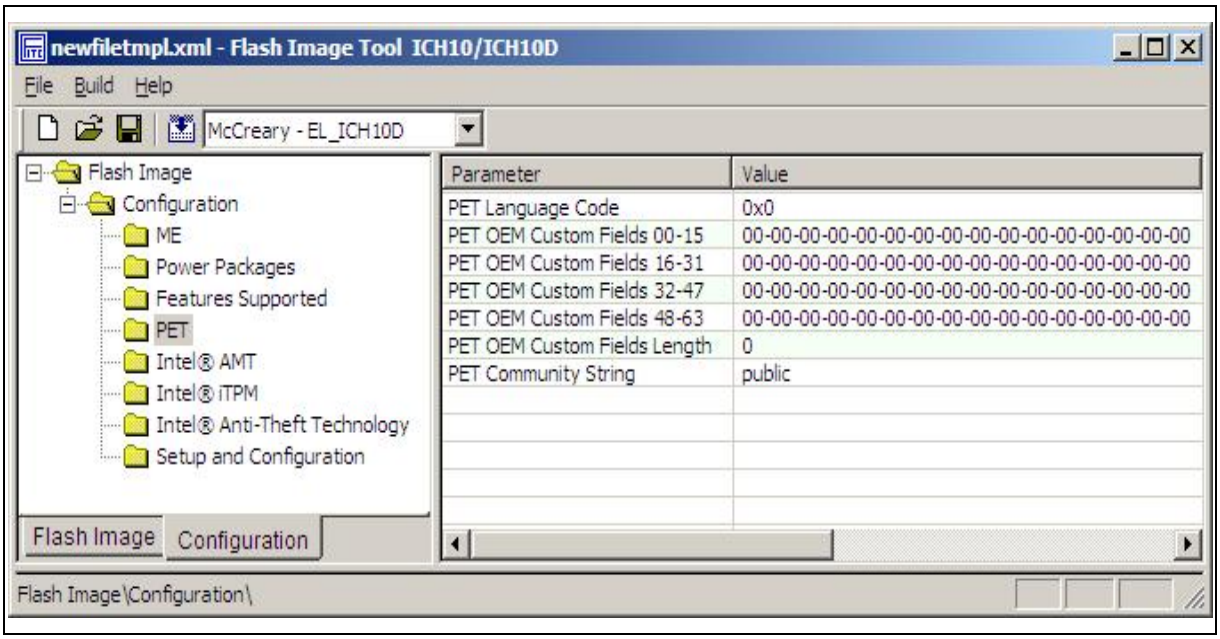


5.4.7.5 PET

This section allows the user to specify any OEM specific PET messages.

The default settings are shown below.

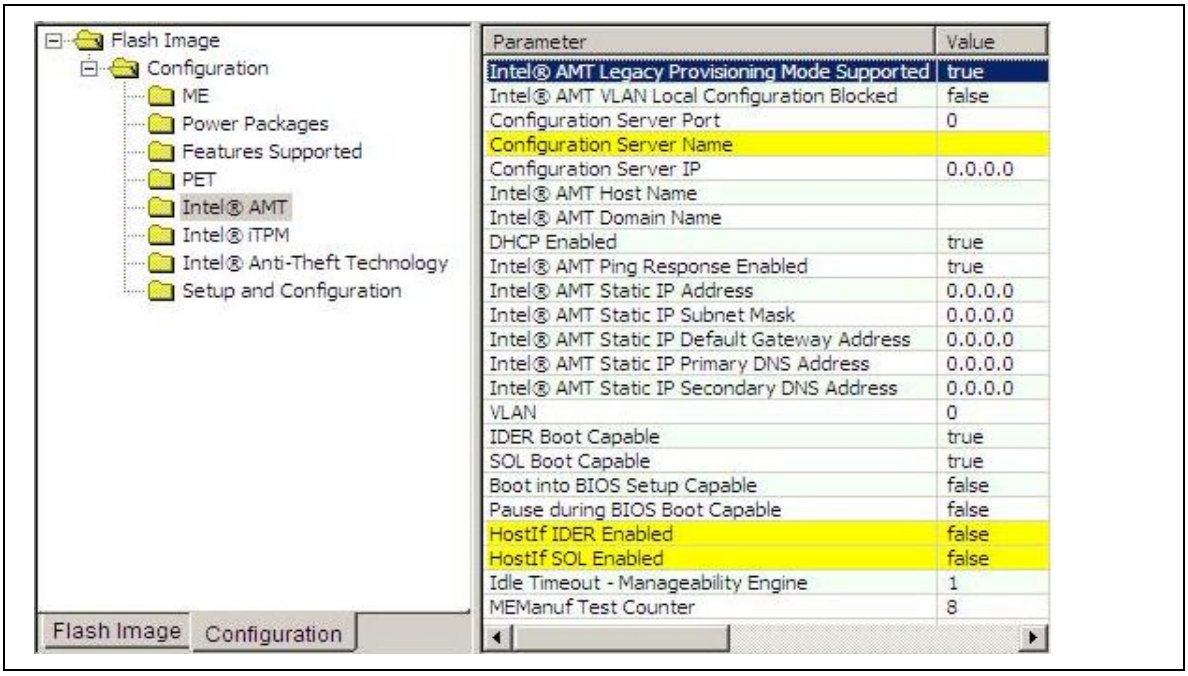
PET Region



5.4.7.6 AMT Section

The AMT section allows the user to specify the default AMT parameters. After the Intel® AMT system is Un-provisioned (full or partial) the values specified in this section will be used. The default settings are shown in Figure 10 below.

Figure 10. AMT Region



Unless specific options are needed or otherwise noted in the table below the default values do not need to be changed.



Intel® AMT Options:

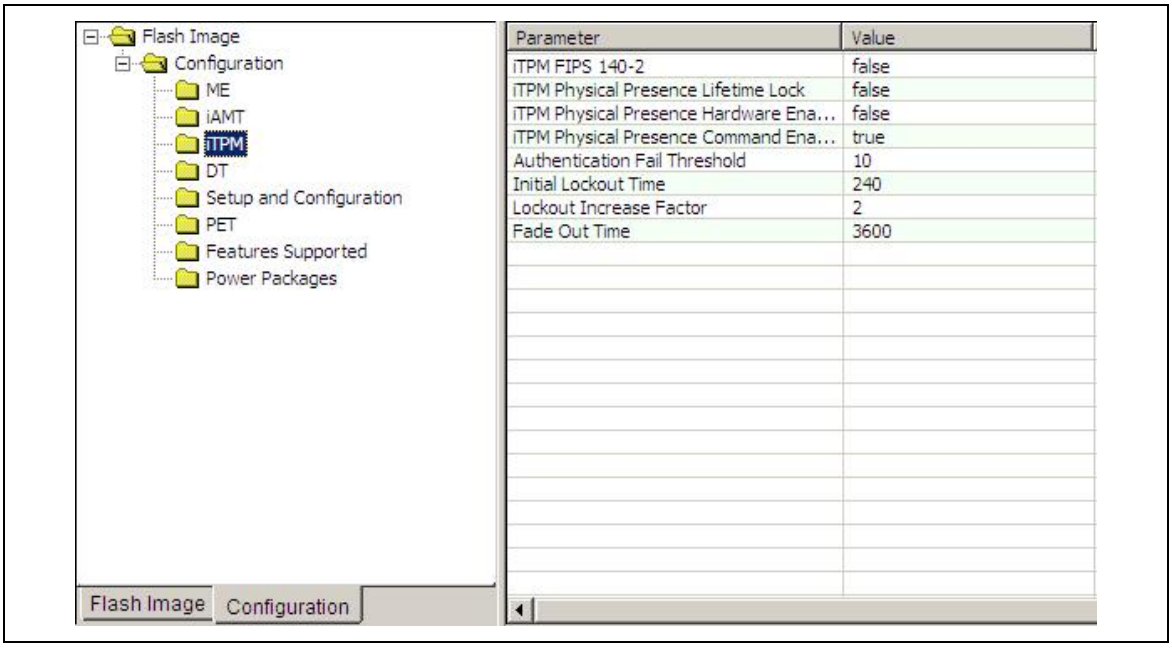
Parameter	Description	Default Value
Intel® AMT Legacy Provisioning Mode Supported	This setting determines if the MEBx will display the Legacy Provisioning Mode option.	True
Intel® AMT VLAN Local Configuration Blocked	This setting determines if the MEBx will display the VLAN Configuration Mode option.	False
Configuration Server Port	This setting will determine the port address for the Setup and Configuration Server. Note: To ensure proper platform Setup and Configuration operation set this value to **9971**	0
Configuration Server Name	This setting will determine the name of the Setup and Configuration Server. Note: To ensure proper Setup and Configuration operations set this value to ProvisionServer.	Not Set
Configuration Server IP	This setting configures a static IP address for the Setup and Configuration Server. Note: You should not configure this value unless you intend to run your AMT platforms on a Static IP address base network.	0
Intel® AMT Host Name	This setting allow for the configuration of the Host Name of the Intel® AMT platform.	Not Set
Intel® AMT Domain Name	This setting allow for the configuration of the Domain Name that the Intel® AMT platform is on.	Not Set
DHCP Enabled	This setting determines if the Intel® AMT platform is using DHCP.	True
Intel® AMT Ping Response Enabled	This setting determines with the Intel® AMT will respond to ping requests from remote systems.	True
Intel® AMT Static IP Address	This setting configures a static IP address for the Intel® AMT platform. Note: You should not configure this value unless you intend to run your AMT platforms on a Static IP address base network.	0

Parameter	Description	Default Value
Intel® AMT Static IP Subnet Mask	This setting configures the static IP Subnet Mask for the Intel® AMT platform. Note: You should not configure this value unless you intend to run your AMT platforms on a Static IP address base network.	0
Intel® AMT Default Gateway Address	This setting configures the static IP Default Gateway Address for the Intel® AMT platform. Note: You should not configure this value unless you intend to run your AMT platforms on a Static IP address base network.	0
Intel® AMT Static IP Primary DNS Server Address	This setting configures the static Primary DNS Static address for the Intel® AMT platform. Note: You should not configure this value unless you intend to run your AMT platforms on a Static IP address base network.	0
Intel® AMT Static IP Secondary DNS Server Address	This setting configures the static Secondary DNS Static address for the Intel® AMT platform. Note: You should not configure this value unless you intend to run your AMT platforms on a Static IP address base network.	0
VLAN	This setting configures the Virtual Local Area Network port address.	0
IDER Boot Capable	This setting determines if the Intel® AMT platform has IDE Redirection functionality enabled by default.	True
SOL Boot Capable	This setting determines if the Intel® AMT platform will support Serial over LAN functionality enabled by default.	True
Boot into BIOS Setup Capable	This setting determines if the Intel® AMT platform will support booting into the System BIOS Setup through Serial over LAN.	False
Pause during BIOS Boot Capable	This setting determines if the Intel® AMT platform will allow you to pause during Serial over LAN BIOS screen redirection.	False
HostIf IDER Enabled	This setting determines if the MEBx will display IDE Redirection menu option to the end user.	True

Parameter	Description	Default Value
HostIf SOL Enabled	This setting determines if the MEBx will display Serial over LAN menu option to the end user.	True
Idle Timeout – Manageability Engine	This setting determines default Idle Timeout value in minutes for the Manageability Engine before it will transition into Moff state with the appropriate ME Wake on LAN Power Packages selected.	1
MEManuf Test Counter	This setting determines how many times MEManuf full testing can be run on a platform before it is locked	8

5.4.7.7 iTPM Section

Figure 11. iTPM Region



The following is a list of the iTPM parameters along with their defaults that can be modified.

Table 5. iTPM Permanent Flags

Bit	Flag	Description	Default
0	FIPS	TRUE: This TPM operates in FIPS mode FALSE: This TPM does NOT operate in FIPS mode	FALSE
1	Physical Presence Lifetime Lock	FALSE: The state of either physicalPresenceHWEnable or physicalPresenceCMDEnable MAY be changed TRUE: The state of either physicalPresenceHWEnable or physicalPresenceCMDEnable MUST NOT be changed for the life of the TPM	FALSE
2	Physical Presence HW Enable	FALSE: Disable the hardware signal indicating physical presence TRUE: Enables the hardware signal indicating physical presence	FALSE
3	Physical Presence CMD Enable	FALSE: Disable the command indicating physical presence TRUE: Enables the command indicating physical presence	TRUE

Table 6. Dictionary Attack Flags

Bit	Field	Description	Default	Min	Max
0	Auth Fail Threshold	Number of failed auth attempts which will trigger lockout	10	1	100
1	Initial Lockout Time	Duration in seconds of first lockout period	240	16	65535
2	Lockout Increase Factor	Factor by which lockout period is multiplied with every additional failed auth	2	1	65535



Bit	Field	Description	Default	Min	Max
3	Fade Out Time	Every time this period (in seconds) passes with no auth failures, lockout time will be reduced (divided by Lockout Increase Factor)	3600	256	65535

5.4.7.8 Setup and Configuration Section

Figure 12. Setup and Configuration Region

Parameter	Value	Help Text
MEBx Password Policy	0	0 = Over network password change allowed only if it i
Provisioning Time Period	24	Set Remote Configuration Provisioning Time Period. V
Intel® AMT Configuration Mode	0	0 = Enterprise mode. 1 = Small Business mode.2 = RC
Remote Configuration Enabled	true	false = Disabled.true = Enabled.
PKI DNS Suffix		Set PKI DNS Suffix in dotted string format.
Config Server FQDN		Set Config server FQDN.
Remote Connectivity Service Capability	true	false = Not capable, true = Capable.
Remote Connectivity Service Enabler Id	000000...	Unique numeric ID of the party which enabled the serv
Remote Connectivity Service Enabler Name		Description of the party which enabled the service in r
Hash 0 Active	false	false = Not Active true = Active
Hash 0 Friendly Name		Enter Hash Name.
Hash 0 Stream		Enter raw hash string or certificate file.

The following is a list of the Setup and Configuration parameters along with their defaults that can be modified.

Table 5. Setup and Configuration parameters

Parameter	Description	Default Value
Provisioning Period	This option configures the base time period for Remote Configuration to take place set in 1 hour increments.	24 (24hours)
AMT Configuration Mode	This option the base mode an AMT platform will operate in. Small Business Mode = 0 Enterprise Mode = 1 Remote Connectivity Service = 2	0
Remote Configuration Enabled	This option Enables / Disables Remote Configuration capabilities on the AMT platform.	False
PKI DNS Suffix	This option allows manual configuration the PKI DNS address. When this value is not set the AMT system will automatically query the DNS for the address of the PKI Configuration Server. PKI DNS Address Example: 192.155.1.2	Not set
Config Server FQDN	This option configures the Fully Qualified Domain Name setting for AMT. When this value is not set the AMT system will automatically query the DNS for the appropriate Domain Name of the Configuration Server.	Not Set
Remote Connectivity Service capability	Remote Connectivity Service provisioning mode or not. When the value is "**true**", the platform will have Remote Connectivity service as an option for Intel® AMT Provisioning Mode. **Possible values:** RCS Capable : **true** RCS Non-Capable: **false**	False



Parameter	Description	Default Value
Remote Connectivity Service enabler ID	Specifies the unique ID of the party (e.g. OEM) which enabled the platform for Remote Connectivity Service mode. **Possible values:** All 16 byte values between “0x00 - 0xffffffff” are valid except for the borders - all zeros or all 0xFFs are invalid.	0
Remote Connectivity Service enabler name	Specifies the textual description (string) of the party (e.g. OEM) which enabled the platform for Remote Connectivity Service mode. **Possible values:** Textual string. Limited to 60 bytes	None
Config Server FQDN	This option configures the Fully Qualified Domain Name for the Setup and Configuration Server. When this value is not set the AMT system will automatically use ‘ProvisionServer’ as the Fully Qualified Domain Name for the Setup and Configuration Server.	Not set
Remote Configuration Hash entry section There are 23 Certificate Hash table options provided in the Firmware. By default the first 5 values are pre-populated with existing key values pre-populated all Firmware images.		
Hash ‘x’ Active	These values control whether the Hash keys are active / inactive.	False
Hash ‘x’ Friendly Name	These are the ‘Friendly Name’ assigned to each individual Hash certificate entry. Example: VeriSign Class 3 Primary CA-G1	First 5 values pre-populated. All other values are blank.
Hash ‘x’ Stream	These entries are for the individual Hash certificate keys. Example: 74 2C 31 92 E6 07 E4 24 EB 45 49 54 2B E1 BB C5 3E 61 74 E2	First 5 values pre-populated. All other values are blank.

5.4.7.9 Intel® Remote Connectivity Service

Intel® Remote Connectivity **Service** is a platform extension service which allows PC OEMs, managed service providers (MSP) and IT Outsourcers to reach or "connect" with end user PCs over the public internet and remotely manage them regardless of system state.

Intel® Remote Connectivity Service code and configuration data are embedded into the ME region.

Setting these parameters correctly greatly eases the effort required by the end user to enable Intel® Remote Connectivity Service.

Remote Connectivity Service parameters are located under "Configuration"->"Flash Image" -> "Setup and Configuration"

Remote Connectivity Service capability - Specifies if the platform allows configuration of Remote Connectivity Service provisioning mode or not. When the value is "**true**", the platform will have Remote Connectivity service as an option for Intel® AMT Provisioning Mode.

When the value is "**false**" Remote Connectivity Service code is completely disabled in the Firmware and the platform has two possible AMT Provisioning Modes – Enterprise and Small Business.

Remote Connectivity Service enabler ID parameter **-** specifies the unique ID of the party (e.g. OEM) which enabled the platform for Remote Connectivity Service mode.

Remote Connectivity Service enabler name parameter **-** specifies the textual description (string) of the party (e.g. OEM) which enabled the platform for Remote Connectivity Service mode.

***Note: when "Remote Connectivity Service capability" = false:**

FITc does not report dependency errors, therefore, although the tool permits configuration of the following parameters, the resulted image configuration will cause "out of order" behavior. Hence, it is NOT supported to configure ANY of the following parameters:

#1. Set "AMT Configuration Mode" = 2 ("Remote Connectivity Service")

#2. Change the default value of "Remote Connectivity Service enabler ID"

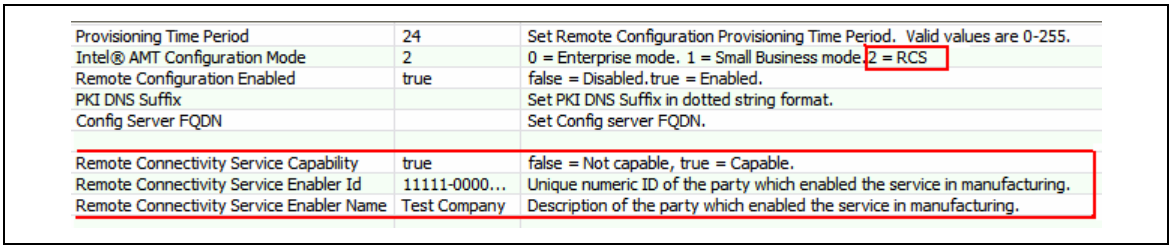
#3. Change the default value of "Remote Connectivity Service enabler name"

*Remote Connectivity Service parameters mentioned above are also available in the Fixed Offset Variables section and configurable using the FPT tool. **For Fixed Offset Variables details and**



instructions please refer to section 4.6 in System Tools User Guide

Figure 13. Intel® Remote Connectivity Service Section



Provisioning Time Period	24	Set Remote Configuration Provisioning Time Period. Valid values are 0-255.
Intel® AMT Configuration Mode	2	0 = Enterprise mode. 1 = Small Business mode 2 = RCS
Remote Configuration Enabled	true	false = Disabled.true = Enabled.
PKI DNS Suffix		Set PKI DNS Suffix in dotted string format.
Config Server FQDN		Set Config server FQDN.
Remote Connectivity Service Capability	true	false = Not capable, true = Capable.
Remote Connectivity Service Enabler Id	11111-0000...	Unique numeric ID of the party which enabled the service in manufacturing.
Remote Connectivity Service Enabler Name	Test Company	Description of the party which enabled the service in manufacturing.

Once the AMT configuration mode is set to Remote Connectivity Service no additional user configuration is required to interact with Intel® Remote Connectivity infrastructure in the Internet environment

To pre-configure Remote Connectivity Service parameters using FITC GUI please follow the below steps:

For details and instructions (GUI and CLI) of building the general image please refer to section 5.5

1. Confirm ME region is enabled
2. Confirm Intel® AMT is selected as Manageability Mode
3. Go to "Configuration"->"Flash Image" -> "Setup and Configuration"
    a. Set the value of "Remote Connectivity Service capability " parameter to "**true"**
    b. Set the value of "AMT Configuration mode " parameter to 2
    c. Set the value of "Remote Connectivity Service enabler ID" to ***your company** ID
    d. Set the value of "Remote Connectivity Service enabler description" to ***your company** ID textual description

***Each manufacturing party has its unique ID

5.4.8 Save configuration

The following steps are used to save the configuration settings you made in the previous sections:

1) Click 'File' on the top menu of the window, and then click 'Save'.
2) A dialog box will open which will take as input the name of the file under which the present configuration will be saved.

3) Use the pull-down to choose the folder where you want to place the file (Example “Unzipped_folder”, and enter the file name (Example: config.xml) and click the ‘Save’ button.
4) Close the main window.

The configuration settings are now saved under the filename (and the path) you specified earlier (Example: E:\Digital_Office_5.0.0.xxxx\config.xml)

Note:

- The purpose of saving your configuration to an .xml configuration file (as described above) is so that image layouts need not be recreated each time.

5.5 Building the flash image

An image can be created in two ways:

1) **GUI**

To create in image from a previously-saved configuration file:

- Double-click on ‘fitc.exe’.
- Click on ‘File’ in the top menu of the window and then click on ‘Open’.
- Select the configuration file, (Example: config.xml), and then click the ‘Open’ button. This will load the configuration file.
- Click on ‘Build’ in the top menu of the window and then click on ‘Build Image’. (Pressing <F5> after opening a stored configuration file will also execute the build process.)

To create an image without a previously-saved configuration file:

- Double-click on ‘fitc.exe’ and perform all steps as described in section 5.3.1.
- Click ‘Build’ in the top menu of the window and then click ‘Build Image’ (or press <F5>) after you have configured all the various sections.

2) **Command Line:** The flash image tool is also supported on the Windows* XP command line interface. The following syntax is used to execute the flash image tool on the command line:

- fitc.exe [xml_file] [/o <file>] /b

  Where:
  <xml_file> : The XML configuration file saved when configuring using the flash image tool.

  /o <file> : The path and filename where the image will be saved. This command overrides the ‘Output path’ in the XML file.

  /b : Automatically builds the flash image.



The UI will not be shown if this flag is specified. This causes the program to run in auto-build mode. If there is an error, an error message will be displayed and the build will not occur.

(Example: fitc.exe config.xml /o outimage.bin /b)

- The 'config.xml' file used in the command above is created and saved as explained in previous sections.

Note:

- The output of the build (a binary file) will be created in directory specified by the '$DestDir' environment variable, which you set during the initial setup of flash image tool earlier in this document. This directory will contain the image map (Example: outimage_Map.txt), the flash image created (Example: outimage.bin) and a folder named 'Int' containing four intermediate build binaries (Example: BIOS Region.bin, Descriptor Region.bin, GbE Region.bin and Intel ME Region.bin). The final flash image (Example: outimage.bin) is the file which will be programmed into the flash devices as explained later.
- Note: when 2 SPI parts are specified in the flash image tool (as with the McCreary CRB), the output directory will also contain 2 additional binary files (example: outimage(1).bin and outimage(2).bin). These files are for use in a stand-alone flash programming machine, and will not be used in this process. They can be ignored.
- For more information on the flash image tool, please refer to the document - 'Intel® flash image tool.pdf', located in the folder:

  "Unzipped_folder"\Tools\System Tools\flash image tool\

  (Example: E:\Digital_Office_5.0.0.xxxx\Tools\System Tools\flash image tool\Intel® Flash Image tool.pdf)

Result:

The result of using the flash image tool is the flash image file (Example: outimage.bin) created by integrating the component images, namely, the BIOS image, the GbE image and the Intel® ME image. The flash image file will be located in the directory defined by the $DestDir environment variable, as mentioned above. For McCreary, this image binary file will be 4MB in size.

5.6 Editing Image with FITC

The FITC tool can be accessed under the following directory:

"Unzipped_folder"\ Tools\System Tools\Flash Image Tool

(Example: E:\Digital_Office_5.0.0.xxxx\Tools\System Tools\Flash Image Tool\FITC.exe)

System Requirements for the FITC tool:

1. AMTNVM is a Windows* command line tool and is supported on the following operating systems:
   - Windows Server* 2003 Operating System
   - Windows* XP Operating System [SP1 or SP2]
   - Windows Vista* Operating System

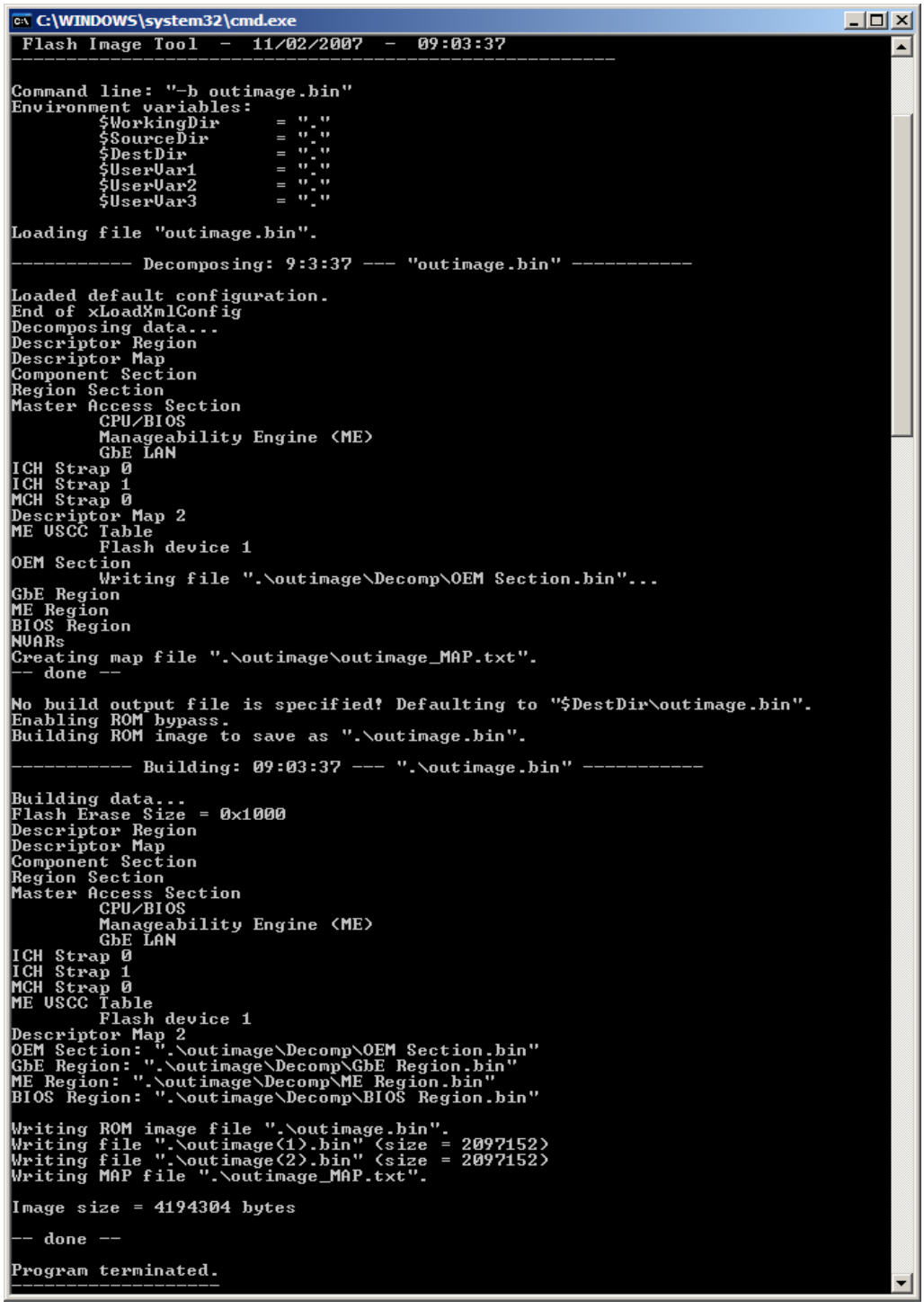
Steps to edit the image

Follow the below steps to configure the flash image file:

1. To parse the flash image file (generated by the flash image tool as described earlier), enter the following on the command line:
   - FITC /b <image file or xml file>

     (Example: FITC /b outimage.bin)
   - The 'outimage.bin' is the output of the FITC tool.

2. On successful execution of the command an NVARs.txt file is created in 'decomp' file folder under the 'outimage' folder specified folder and the following will be displayed:



If the command does not complete successfully a descriptive error message will appear.

3. Open the created NVARs.txt file to modify the values of the desired parameters in the file.

4. Enter values for the following fields (Example values have been given):
   - AMT Host Name (Example: IntelAMT)
   - AMT Domain Name (Example: amt.intel.com)
   - Advanced Fan Speed Control Enabled = 0
   - AMT Static IP Address (Example: 10.0.0.5)

   Leave all other values in the file as they are.

5. To create a new image based on the modified NVARs.txt file, enter the following on the command line:
   - FITC /b < image file or xml file > /nvars NVARs.txt

     (Example: FITC /b outimage.bin /nvars NVARs.txt)

     NOTE:
     - Make sure that when designating the modified NVARs.txt file that it is in the same location as your image file.
     - The existing outimage.bin is replaced with the modified outimage.bin.

6. On successful execution of the command, <outfile> (Example: outimage.bin) is created in the specified folder and the following message will be displayed:



```
C:\WINDOWS\system32\cmd.exe
 Flash Image Tool  -  11/02/2007  -  09:03:37
--------------------------------------------------

Command line: "-b outimage.bin"
Environment variables:
        $WorkingDir     = "."
        $SourceDir      = "."
        $DestDir        = "."
        $UserVar1       = "."
        $UserVar2       = "."
        $UserVar3       = "."

Loading file "outimage.bin".

----------- Decomposing: 9:3:37 --- "outimage.bin" -----------

Loaded default configuration.
End of xLoadXmlConfig
Decomposing data...
Descriptor Region
Descriptor Map
Component Section
Region Section
Master Access Section
        CPU/BIOS
        Manageability Engine (ME)
        GbE LAN
ICH Strap 0
ICH Strap 1
MCH Strap 0
Descriptor Map 2
ME VSCC Table
        Flash device 1
OEM Section
        Writing file ".\outimage\Decomp\OEM Section.bin"...
GbE Region
ME Region
BIOS Region
NVARs
Creating map file ".\outimage\outimage_MAP.txt".
-- done --

No build output file is specified! Defaulting to "$DestDir\outimage.bin".
Enabling ROM bypass.
Building ROM image to save as ".\outimage.bin".

----------- Building: 09:03:37 --- ".\outimage.bin" -----------

Building data...
Flash Erase Size = 0x1000
Descriptor Region
Descriptor Map
Component Section
Region Section
Master Access Section
        CPU/BIOS
        Manageability Engine (ME)
        GbE LAN
ICH Strap 0
ICH Strap 1
MCH Strap 0
ME VSCC Table
        Flash device 1
Descriptor Map 2
OEM Section: ".\outimage\Decomp\OEM Section.bin"
GbE Region: ".\outimage\Decomp\GbE Region.bin"
ME Region: ".\outimage\Decomp\ME Region.bin"
BIOS Region: ".\outimage\Decomp\BIOS Region.bin"

Writing ROM image file ".\outimage.bin".
Writing file ".\outimage(1).bin" (size = 2097152)
Writing file ".\outimage(2).bin" (size = 2097152)
Writing MAP file ".\outimage_MAP.txt".

Image size = 4194304 bytes

-- done --

Program terminated.
--------------------
```

Result:

- The modified/edited flash image (Example: outimage.bin) with certain fields modified has been created in the specified folder.

NOTE:

- The FITC tool is supported on the following operating systems: Windows* 2000 [SP4], Windows Server* 2003, Windows* XP [SP1/2].
- For more details on the parameters / fields which can be modified in the flash image file, please refer to the document – 'System Tools User Guide.pdf'.
- For more details on the usage of the FITC tool, please refer to the document – 'System Tools User Guide.pdf', located in the folder: "Unzipped_folder"\Tools\System Tools
- (Example: E:\Digital_Office_5.0.0.xxxx\Tools\ System Tools \ System Tools User Guide.pdf)

5.7 Programming the image onto the flash

Prior to manufacturing, there are two ways to load the flash image (Example: outimage.bin) onto the flash. These are described in the following two sections (5.9 and 5.10):

Note: If you are re-flashing a full image onto an AMT platform previously programmed you must make sure to clear the CMOS to ensure that the platform reloads the default parameters and to ensure that there are no erroneous settings remaining.

5.8 Using a flash programmer

An external Flash Programmer/Burner can be used to write the flash image file onto the flash. When you use the flash image tool (FITC) to generate a flash image, in addition to the final output binary (outimage.bin), the tool also creates two separate binary images, named "outimage(1).bin" and "outimage(2).bin".

These two binary images are generated for the purpose of using an external flash programmer to program the SPI devices without having them installed on the system board. Each of the two binary image files mentioned above corresponds to a separate SPI device.

Multiple commercial flash burners are available. The description of the use of these is beyond the scope of this document.



5.9 Using the flash programming tool

The flash programming tool is supported on the following operating systems - DOS, Free DOS, DRMK DOS, Windows* XP [SP2] and Windows* PE. If you are using a DOS environment to program the image in to the flash, follow the steps in section 8.4.1. If you are using the Windows* environment to program the image in to the flash, follow the steps in section 8.4.5.

5.9.1 DOS environment

5.9.1.1 Flash programming tool from VIP

The DOS flash programming tool can be accessed under the following directory:

"Unzipped_folder"\Tools\System Tools\Flash Programming Tool\DOS\

(Example: E:\Digital_Office_5.0.0.xxxx\Tools\System Tools\Flash Programming Tool\DOS \fpt.exe)

The files in the '\Flash Programming Tool\DOS\' folder are:

1. **fpt.exe:** This is the executable to write the flash image file into the flash.
2. **License.rtf**: This contains the Intel® Software License Agreement.
3. **Intel® Flash Programming Tool.pdf**: This document gives more details on the flash programming tool.

IMPORTANT: DOS4GW.exe is necessary to use the flash programming tool. DOS4GW can be downloaded from the following URL: http://www.scene.org/file.php?file=%2Fresources%2Fdos4gw.exe&fileinfo.

Download the 'dos4gw.exe' file and save this file to the same directory as 'fpt.exe':

"Unzipped_folder"\Tools\System Tools\Flash Programming Tool\DOS\

(Example: E:\Digital_Office_5.0.0.xxxx\Tools\System Tools\Flash Programming Tool\DOS\)

5.9.2 Load files onto the target platform

The Flash Programming Tool files are loaded onto the target platform as follows:

1. Create the final flash image file (Example: outimage.bin) and copy all the files along with the Flash Programming Tool (fpt.exe, fparts.txt and DOS4GW.exe) to a medium that is accessible to the target platform (for example, using a USB pen drive or other media that can be made accessible to DOS on the target platform).
2. Boot the target platform to DOS.

3. Change to the directory where the Flash Programming Tool and the flash image file (outimage.bin) are stored.

5.9.3 Query the flash devices

To ensure that the flash programming tool recognizes the SPI flash devices on the target platform before programming the flash image file onto the flash, execute the following at the DOS prompt:

- fpt /i

Under 'Flash Devices Found' in the output, all flash devices on the target platform should be listed.

Example: The following will be displayed:

--- Flash Devices Found ---

SST25VF016B ID: 0x00BF41 Size: 2048KB (16384Kb)

SST25VF016B ID: 0x00BF41 Size: 2048KB (16384Kb)

NOTE:

- Initially, if no descriptor is present on the SPI flash, only the first flash part is listed (even if two flash parts are present on the target platform). Once the flash descriptor is loaded, both the flash parts will be listed upon subsequent execution of the above-referenced command.

- If no flash parts enumerate, please verify that the SPI part you are using is listed in the fparts.txt file and that your BIOS is correctly setting up the VSCC register as defined by the latest Intel® I/O Controller Hub (ICH10) BIOS specification.

5.9.4 Program the image

Prior to programming the image into the platform you will first need to disable the Manageability Engine by removing memory from Channel-A DIMM 0 this will ensure that the Manageability Engine is not working with the SPI flash during image programming.

To program the final flash image file (Example: outimage.bin) onto the SPI flash, execute the following command at the DOS prompt.

- fpt /f:<filename>

  (Example: fpt /f:outimage.bin)



After having successfully programmed the flash image (Example: outimage.bin) onto the SPI flash devices, the following message will be displayed: **'Write Complete.'**

Unplug the power cord and the network cable, and then reattach them after 10 seconds. This ensures all capacitors on the board are fully discharged before the system starts up again.

Result:

- The flash image file (Example: outimage.bin) has now been successfully programmed onto the SPI device.

NOTE:

- For more details on this tool, please refer to the document - 'Intel® Flash Programming Tool.pdf', located in the folder:

  "Unzipped_folder"\Tools\System Tools\Flash Programming Tool\

  (Example: E:\iAMT_ASF2_QST_5.0.0.xxxx\Tools\System Tools\Flash Programming Tool\Intel® Flash Programming Tool.pdf)

5.9.5 Windows* environment

The Windows* version of the flash programming tool is a command line tool designed for use within the Windows* command shell environment (cmd.exe).

5.9.5.1 Flash programming tool from VIP

The Windows Flash Programming Tool can be accessed under the following directory:

- "Unzipped_folder"\Tools\System Tools\Flash Programming Tool\Windows\
- (Example: E:\Digital_Office_5.0.0.xxxx\Tools\System Tools\Flash Programming Tool \Windows\fptw.exe)

The files in the '\Flash Programming Tool\Windows\' folder are:

1. **fparts.txt:** This file 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 be added) to describe the flash part which is on the target system. Examine the target board before adding in the appropriate attribute values. Default values are already entered for SPI devices used with Intel Customer Reference Boards (CRBs).
2. **fptw.exe:** This is the executable to write the flash image file to the SPI flash devices.
3. **sseIdrvdll32e.dll**: Supported library file.
4. **ssePmxdll32e.dll**: Supported library file.

5. **ssepmxdrv.sys:** Supported system file.

5.9.6 Load files onto the target platform

Use the flash programming tool to load the files onto the target platform as follows:

1. Create the final flash image file (Example: outimage.bin) and copy all the files along with the Flash Programming Tool (fptw.exe, fparts.txt, sseIdrvdll32e.dll, ssePmxdll32e.dll and ssepmxdrv.sys) to a medium that can be made accessible to the target platform (for example, use a USB pen drive or other media that can be made accessible to Windows on the target platform).
2. Boot the target platform to Windows and open a command prompt window.
3. Change to the directory where the flash programming tool and the flash image file (outimage.bin) are located.

5.9.7 Query the flash devices

To ensure that the flash programming tool recognizes the SPI flash devices on the target platform before programming the flash image file onto the flash, execute the following at the command prompt:

- fptw /i

Under ‘Flash Devices Found’ in the output, all flash devices on the target platform should be listed.

Example: The following will be displayed:

--- Flash Devices Found ---

SST25VF016B ID: 0x00BF41 Size: 2048KB (16384Kb)

SST25VF016B ID: 0x00BF41 Size: 2048KB (16384Kb)

NOTE:

- Initially, if no descriptor is present on the SPI flash, only the first flash part is listed (even if two flash parts are present on the target platform). Once the flash descriptor is loaded, both the flash parts will be listed upon subsequent execution of the above-referenced command.

- If no flash parts enumerate, please verify that the SPI part you are using is listed in the fparts.txt file and that your BIOS is correctly setting up the VSCC



register as defined by the latest Intel® I/O Controller Hub (ICH10) BIOS specification.

5.9.8 Program the image

Prior to programming the image into the platform you will first need to disable the Manageability Engine by removing memory from Channel-A DIMM 0 this will ensure that the Manageability Engine is not working with the SPI flash during image programming.

To program the final flash image file (Example: outimage.bin) onto the flash devices, execute the following command at the command prompt.

- fptw /f <filename>

  (Example: fptw /f outimage.bin)

After having successfully programmed the flash image onto the SPI flash devices, the following message will be displayed: **'Write Complete.'**

Unplug the power cord and the network cable, and then reattach them after 10 seconds. This ensures all capacitors on the board are fully discharged before the system starts up again.

Result:

- The flash image file (Example: outimage.bin) has now been successfully programmed onto the SPI flash device.

NOTE:

- For more details on this tool, please refer to the document - 'Intel® Flash Programming Tool.pdf', located in the folder:
  - "Unzipped_folder"\Tools\System Tools\Flash Programming Tool\
  - (Example: E:\Digital_Office_5.0.0.xxxx\Tools\System Tools\Flash Programming Tool\Intel® Flash Programming Tool.pdf)

§

6 BIOS Setup

6.1 Set Parameters

The following steps are used for setting up parameters in the BIOS setup screen on the target platform (CRB) after you have programmed the image onto the flash devices. **Remember to always start with a fresh OS install on the platform any time you are enabling a new FW image.**

1. Press 'Del' during Setup message display at power-on to enter the BIOS Setup screen.
2. Once you are in BIOS Setup, press <F9> to load the default BIOS settings.
3. Update the time and date if it is incorrect.
4. Press F10 to save changes and exit BIOS setup. Disconnect the power cord for at least 10 seconds and then restart the system.

6.2 iTPM Parameters on BIOS

To modify the iTPM BIOS parameters:

Press **Del** during **Setup** message display at power-on to enter the BIOS Setup screen.

Load optimal settings

Select **Advanced.**

Select **Trusted Computing.**

The following options are available:

TGC/TPM SUPPORT (Yes/No) → Yes (Enable/Disable the TPM Device)

TPM Execute Command (Don't Change / Disabled / Enabled) → Enabled (Enable / Disable the TPM Device)

Clearing the TPM (Ok / Cancel) → Ok (Force Clear TPM data)

Save changes and exit BIOS setup.

§



7 Intel® Management Engine BIOS Extension Setup

7.1 Intel® Management Engine BIOS Extension screen

NOTE:

- The default power policy is Moff during Sx. To allow Intel ME® Firmware to go to M1 with network during Sx, the power policy settings as described in this section must be done.

The Intel® Management Engine BIOS Extension screen is used to enable Intel® AMT or ASF 2.0 on the target platform (CRB). Follow the below steps:

1. On rebooting the system (as mentioned in the previous section), after the initial boot screen, the following message will be displayed:

   'Press <Ctrl-P> to enter Intel® ME Setup'. Press 'Ctrl-P'

   NOTE:
   - Should press <Ctrl-P> as soon as the above message is displayed, as this message will be displayed for only a few seconds.
2. You will be prompted for the password if you enabled MEBx Password in BIOS Setup. If you left MEBx Password disabled in BIOS Setup, skip to step 5.
3. Enter 'admin' under 'Intel® ME Password' (you will change this later). Press Enter.
4. The Intel® Management Engine BIOS Extension screen will be displayed.

NOTE:

- The Intel® ME BIOS Extension screen can be used to enable either Intel® AMT or ASF 2.0, not both. If Intel® AMT is to be enabled on the target platform, then follow the steps in section 10.0.1. If ASF 2.0 is to be enabled on the target platform, then follow the steps in section 10.0.2.

7.1.1 Enabling Intel® AMT

This section describes how to enable Intel® AMT using the Intel ME BIOS Extension screen. This section describes how to set the Intel® AMT system in static mode only (for simplicity). The below steps are followed:

1. Select 'Change Intel® ME Password'. Press Enter.

Enter a new password under 'Intel® ME Password' and then press Enter.

The new password must be strong 7-bit ASCII characters excluding ':', ',' and '"' characters.

String length is limited to 32 characters.

Limitations:

Password Length: Passwords must comprise of at least 8 characters.

Password Complexity: Password must include:

a. At least one Digit character ('0', '1',...'9')

b. At least one 7-bit ASCII non alpha-numeric character, above 0x20, (e.g. '!', '$', ';').

c. Both lower-case Latin ('a', 'b',...,'z') and upper case Latin ('A', 'B',...'Z').

Note: '_' (underscore) and ' ' (space) are valid password characters but are not used in determination of complexity.

2. 'Verify password' will be displayed. Enter the new password just entered in the previous step. Press Enter.
3. Select 'Intel® ME Configuration'. Press Enter.
4. The following message will be displayed:

   'System resets after configuration changes. Continue: (Y/N)'.
5. Press 'Y'.
6. The following message will be displayed: 'Acquiring ME Configurations…'
7. Select 'Intel® ME State Control'. Press Enter.
8. Select 'Enabled'. Press Enter.
9. Select 'Intel ME Firmware Local Update'. Press Enter.
10. Select 'Enabled'. Press Enter.
11. Select 'Intel® ME Features Control'. Press Enter.
12. Select 'Manageability Feature Selection'. Press Enter.
13. Select 'Intel® AMT'. Press Enter.
14. Select 'Return to Previous Menu'. Press Enter.
15. Select 'Intel® ME Power Control'. Press Enter.
16. Select the desired power policy radio button.
17. Select 'Return to Previous Menu'. Press Enter.
18. Select 'Return to Previous Menu'. Press Enter.
19. The system shuts down. Pull out the power cord and reconnect after 5 seconds. Then, turn on the system.
20. After the initial boot screen, the following message will be displayed:

    'Press <Ctrl-P> to enter Intel® ME Setup'
21. Press 'Ctrl-P'.

NOTE:

- Should press <Ctrl-P> as soon as the above message is displayed, as this message will be displayed for only a few seconds.

22. The user will be prompted for the password.
23. Enter the new Intel® ME password (strong password required – see step 2 above). Press Enter.
24. Select 'Intel® AMT Configuration'. Press Enter.
25. The following message will be displayed: 'Connecting to Intel® AMT Client…'
26. The Intel® AMT Configuration screen is displayed.



27. Select ‘Host Name’. Press Enter.
28. Type in: IntelAMT. Press Enter.
29. Select ‘TCP/IP’. Press Enter.
30. The following message will be displayed:

    ‘Disable Network Interface: (Y/N)’.
31. Press ‘N’.
32. The following message will be displayed:

    ‘[DHCP Enabled] Disable DHCP: (Y/N)’.
33. Press ‘Y’.
34. The following is displayed: ‘IP address:’
35. Enter a static IP address. (Example: 192.168.0.15). Press Enter.

NOTE:

- If more than one Intel® AMT machine is being configured in static IP mode and all these Intel® AMT machines are on a single Local Area Network (LAN), then it should be ensured that each of the static IP addresses are unique. If more than one Intel® AMT device has the same static IP address, then a collision of IP addresses will occur on the network and the Intel® AMT machines will not respond correctly on the network.

36. The following is displayed: ‘Subnet mask:’
37. Enter the subnet mask. (Example: 255.255.255.0).Press Enter.
38. The following is displayed: ‘Default Gateway address:’
39. Press Enter. (Leave as default value).
40. The following is displayed: ‘Preferred DNS address:’
41. Press Enter. (Leave as default value).
42. The following is displayed: ‘Alternate DNS address:’
43. Press Enter. (Leave as default value).
44. The following is displayed: ‘Domain name:’
45. Enter the domain name. (Example: amt.intel.com). Press Enter.
46. Select ‘Provision Model’ from the menu. Press Enter.
47. The following message is displayed: ‘[Enterprise] Change to Small Business: (Y/N)’. Press ‘Y’.
48. Select ‘SOL/IDER’. Press Enter.
49. The following message is displayed: ‘[Caution] System resets after configuration changes. Continue (Y/N)’. Press ‘Y’.
50. The following message is displayed: ‘Username & Password’.
51. Select ‘Enabled’. Press Enter.
52. The following message is displayed: ‘Serial Over LAN’.
53. Select ‘Enabled’. Press Enter.
54. The following message is displayed: ‘IDE Redirection’.
55. Select ‘Enabled’. Press Enter.
56. Select ‘Return to Previous Menu’. Press Enter.
57. Select ‘Exit’. Press Enter.
58. The following message will be displayed:

    ‘Are you sure you want to exit? (Y/N):’
59. Press ‘Y’.
60. The system will boot to OS.

7.1.2 Enabling ASF 2.0

This section describes how to enable ASF 2.0 using the Intel® ME BIOS Extension screen. The below steps are followed:

1. Select 'Change Intel® ME Password'. Press Enter.

Enter a new password under 'Intel ME Password' and then press Enter.

The new password must be strong 7-bit ASCII characters excluding ':', ',' and '"' characters.

String length is limited to 32 characters.

Limitations:

Password Length: Passwords must comprise of at least 8 characters.

Password Complexity: Password must include:

a. At least one Digit character ('0', '1',...'9')

b. At least one 7-bit ASCII non alpha-numeric character, above 0x20, (e.g. '!', '$', ';').

c. Both lower-case Latin ('a', 'b',...,'z') and upper case Latin ('A', 'B',...'Z').

Note: '_' (underscore) and ' ' (space) are valid password characters but are not used in determination of complexity.

2. 'Verify password' will be displayed. Enter the new password just entered in the previous step. Press Enter.
3. The following message will be displayed:

'System resets after configuration changes. Continue: (Y/N)'.

4. Press 'Y'.
5. The following message will be displayed: 'Acquiring Intel® ME Configurations...'.
6. Select 'Intel® ME State Control'. Press Enter.
7. Select 'Enabled'. Press Enter.
8. Select 'Intel® ME Features Control'. Press Enter.
9. Select 'Manageability Feature Selection'. Press Enter.
10. Select 'ASF'. Press Enter.
11. Select 'Return to Previous Menu'. Press Enter.
12. Select 'Intel® ME Power Control'. Press Enter.
13. Select 'Intel® ME State upon Initial Power-On'. Press Enter.
14. Select 'ON'. Press Enter.
15. Select 'Intel® ME ON in Host Sleep States'. Press Enter.
16. Select 'ALWAYS'. Press Enter.
17. Select 'Return to Previous Menu'. Press Enter.
18. Select 'Return to Previous Menu'. Press Enter.
19. The system shuts down. Pull out the power cord and reconnect after 10 seconds.
20. Turn on the system and boot to the OS.

§



8 Installing drivers

8.1 Driver identification

NOTE: Before the drivers mentioned in this section are installed, make sure to install the chipset INF file (This is required for proper installation of the Intel® ME Interface and LMS_SOL drivers).

The following drivers must be installed on the target platform before continuing. Some of the McCreary/ICH10 tools used will require these drivers to be installed.

The drivers to be installed are:

1. Intel® ME Interface driver - The setup file for installing the Intel® ME Interface driver can be accessed from under the following directory:
   - "Unzipped_folder"\Drivers\HECI\
   - (Example: E:\iAMT_ASF2_QST_5.0.0.xxxx\Drivers\HECI\Setup.exe)

2. GbE driver -Copy the GbE driver onto the target platform (CRB) from the following location:
   - Pro_Lan_Drivers_xx.x.zip

   And update the GbE driver on the target platform.

   NOTE: The above zip component can be downloaded from the same web page where you obtained the kit.

NOTE:
- The LMS_SOL driver is to be installed only if the downloaded kit contains firmware for Intel® AMT.

3. LMS_SOL driver - The setup file for installing the LMS_SOL driver can be accessed from under the following directory:
   - "Unzipped_folder"\Drivers\LMS_SOL\
   - (Example: E:\Digital_Office_5.0.0.xxxx\Drivers\LMS_SOL\Setup.exe)

NOTE:

- Section 11 is only applicable if the downloaded kit has firmware for Intel® AMT and Intel® AMT has been enabled as described in section 10.0.1
- If the downloaded kit has firmware for ASF 2.0, and if ASF has been enabled as described in section 10.0.2, then for further details on configuring the client system, please refer to the documents under the following directory:

  "Unzipped_folder"\Tools\ASF 2.0 Tools\ASF Agent\DOCS\GUIDE\

  (Example: E:\Digital_Office_5.0.0.xxxx\Tools\ASF 2.0 Tools\ASF Agent\DOCS\GUIDE\)

  - For details on the ASF Tools and their installation instructions, please refer to: 'readme.pdf' document, which can be downloaded from the same web page where you downloaded the kit.
  - Please note that the Linux section under these documents is not relevant for McCreary/ICH10, as McCreary/ICH10 does not provide ASF Linux support. Only old Intel® NICs, like Tabor, Tekoa, Ophir, ESB2, etc, support ASF on Linux.

§



9 Basic Intel® AMT functional demonstration

9.1 Ping verification

Ping verification tests the network connection between the target platform (CRB/Intel® AMT device) and other systems on the network. This is done as follows:

1) Connect the system under test (client system/Intel® AMT machine) to the management console (any other system on the network) through a Local Area Network. This connection can be done in two ways:
   - Connect an Ethernet cross cable between the client system and the management console.
   - Or, use a hub/switch to connect the client system and the management console.

NOTE:

- When connecting the Intel® AMT system to a Local Area Network (LAN) with multiple Intel® AMT machines on the LAN (all configured in static IP mode), ensure that each of the machines have a unique static IP address (set in the 'Intel® AMT Configuration Screen' as described in section 10.0.1)
- If Intel® AMT device is configured in static IP mode, then the host machine (OS) of the Intel® AMT machine must also be configured in static IP mode. The host static IP address must be different from the Intel® AMT static IP address (set up in the 'Intel® AMT Configuration screen' as described in section 10.0.1). Also, the host static IP address must be unique across the LAN when multiple Intel® AMT systems are connected to the LAN.
- To configure the host machine of the Intel® AMT system in static IP mode, follow the below steps:
  - Boot the host system through Windows* operating system.
  - Click on 'Start' on the Windows* tool bar, and select 'Control Panel'.
  - The 'Control Panel' window is displayed. Select 'Network Connections' and double click on it.
  - Right-Click on 'Local Area Connection' and select 'Properties'.
  - In the 'General' tab, select 'Internet Protocol (TCP/IP)' and click on 'Properties'.
  - Select the option 'Use the following IP address'.
  - Enter a static IP address under 'IP address'. Example: 192.168.0.20

    NOTE: This IP address must be different from the static IP address set in the 'Intel® AMT Configuration Screen'.

- Enter the subnet mask (255.255.255.0) under 'Subnet mask'.
- Click 'OK' and then on 'Close' to save the parameters entered as above.

2) Boot the Management Console system into Windows*.
3) Ensure that the Management Console is configured in static IP mode:
   a) On the host system, click on 'Start' on the Windows* tool bar, and select 'Control Panel'.
   b) The 'Control Panel' window is displayed. Select 'Network Connections' and double click on it.
   c) Right-Click on 'Local Area Connection' and select 'Properties'.
   d) In the 'General' tab, select 'Internet Protocol (TCP/IP)' and click on 'Properties'.
   e) Select the option 'Use the following IP address'.
   f) Enter a static IP address under 'IP address'. Example: 192.168.0.10

      NOTE: This IP address must be different from the static IP address set on the Intel® AMT machine (in the 'Intel® AMT Configuration Screen') and the host IP address of the Intel® AMT machine.
   g) Enter the subnet mask (255.255.255.0) under 'Subnet mask'.
   h) Click 'OK' and then on 'Close' to save the parameters entered as above.

4) Open the command line interface on the management console.
5) Ping the Intel® AMT device (target platform).

   ping <IP-address of Intel® AMT machine>

   (Example: ping 192.168.0.15)
6) Ping the host operating system from the management console.

   ping <IP-address of the host OS>

   (Example: ping 192.168.0.20)
7) Ping the management console from the host operating system.

   ping <IP-address of the management console>

   (Example: ping 192.168.0.10)

- If the client machine is connected to the management console through the network, then the ping is successful and returns the bytes of data transferred.
- If the ping returns: 'Request timed out', then either one of the following might be the cause:
  a) The system under test (client machine) is not connected to the network properly, or
  b) The user has not set the IP parameters of the Intel® AMT system (target platform/CRB) correctly.
- **Troubleshooting:** If the 'ping verification' is not working, then try the following:
  - Ping <IP-address of Intel® AMT host OS>, i.e., 192.168.0.20 from the management console. Ping the management console from the host OS



of the client machine, i.e., ping 192.168.0.10. This will help verify network connectivity. If this fails check your network setup.

- Reboot both the systems, client machine (Intel® AMT system) and the management console, and try again.
- Configure the network driver with static IP. This IP address must be different from the Intel® AMT Static IP address (configured in the 'Intel® AMT Configuration screen' as described in section 10.0.1). But must be configured on the same subnet.
- Verify all IP address configurations and try again.

NOTE: When pinging the Intel® AMT device, make sure the management console and the system under test (client machine/target platform operating system) are operating in the same segment, i.e., the management console, Intel® AMT device and the host OS of the Intel® AMT device should be configured in static IP mode or all the three of them should be configured in DHCP mode.

9.2 Test Intel® AMT using the WebUI

To ensure that Intel® AMT is enabled on the host system (client system), the user should try to access Intel® AMT via the WebUI application. This may be done from any system on the network with one of the following browsers installed:

- Microsoft* Internet Explorer 6 SP1 or newer
- Netscape* Navigator 7.1 or newer.
- Mozilla* Firefox* 1.0 or newer.
- Mozilla 1.7 or newer.

The Web UI cannot be accessed locally (from the host operating system of the Intel® AMT machine). You can access Intel® AMT remotely from another machine.

To access Intel® AMT via the WebUI, follow the steps given sequentially:

1) Open a web browser (any of above listed, and supported on the management console), and in the address box, enter one of the following:
   - If the network can resolve the client computer name to an IP address (i.e., a DNS server is installed or the name is listed in the hosts file), then: http://host_name:16992 (Example: http://clientname1:16992).
   - If the client system has a static IP address, then:

     http://host_name:16992 (Example: http://192.168.0.15:16992).
2) A Login web page will be displayed. Click to select the 'Log On…' button, and a login dialog box is displayed.
3) In the 'User name' box enter 'admin', and in the 'Password' box enter the same password as changed in the 'Intel® Management Engine BIOS Extension' screen setup.

Once the username and password have been successfully entered, the web browser opens the Intel® AMT Web User Interface page and the web interface can be used for browsing through the various options present.

§

10 Basic iTPM functional demonstration

Prior to testing the iTPM, the iTPM driver must be installed. Refer to the iTPM Windows* XP Driver Installation Guide.

10.1 Windows* XP Operating System Verification

To verify that the iTPM has been recognized under Windows* XP:

Click **Start**, then **Control Panel**.

Click **System**, then **Hardware** and select **Device Manager**.

Click on **System Devices** and verify that **Trusted Platform Module** is listed.

10.2 Windows Vista* Operating System Verification

There are two methods to verify that the iTPM has been recognized under Windows Vista*:

10.2.1 Method 1

Click **Start**, then **Run**.

Type **tpm.msc** and press **ENTER**.

The **TPM Management on Local Computer** window will be displayed. This will indicate whether the iTPM has been recognized and its state (enabled, active, owned).

10.2.2 Method 2

Click **Start**, then **Settings**, then **Control Panel**.

Click **System and** select **Device Manager**.

Click on **Security Devices** and verify that **Trusted Platform Module 1.2** is listed.

§



11 Intel® AMT Tools

This section describes the various tools you can use to test and verify Intel® AMT functionality, configuration and behavior. These tools are typically included in the kit or available separately from VIP. Any tool included in a kit release should supersede any version available from any other source. Always consider the tools and documentation supplied with the kit as the latest revision above all others.

Requirements:

AMTVTR runs on Windows XP SP1/2, XP 32/64, Vista 32bit/64bit, and Server 2003. It is a command-line executable and can be run on a remote computer connected to an Intel® AMT device.

11.1 Intel® AMT System Validation Tool – Local (AMTVTL)

The AMTVTL tool is used to check the functionality of a local Intel® AMT system. It is a command line executable and must be run locally on an Intel® AMT machine. AMTVTL tool needs the following services to be installed:

- Intel® AMT Local Manageability Service (LMS)
- Intel® ME Interface driver.

The tool checks that the following components are working:

- LMS

- Intel® ME interface driver

- Firmware Intel® ME Interface network interface (LME)

- Storage Manager in firmware.

11.1.1 AMTVTL from VIP

The AMTVTL tool can be accessed from under the following directory:

“Unzipped_folder”\Tools\iAMT Tools\iAMTVTL\

(Example: E:\Digital_Office_5.0.0.xxxx\Tools\iAMTVTL \iAMTFeaturesLocal\AMTVTL.exe)

11.1.2 Executing AMTVTL

On the command line interface (in the local Intel® AMT system), execute the following command:

- AMTVTL.exe [options]

Options is one or more of the following:

- user <username>—the username used to authenticate for access to Intel® AMT features on the local machine. If Kerberos authentication is used, the username and password should not be entered.

-pass <password>—the password corresponding to the username used to authenticate for access to Intel® AMT features on local machine. If Kerberos authentication is used, the username and password should not be entered.

-TLS—must be entered if TLS mode is enabled. If TLS is used, the hostname provided must be the same as the hostname used in the TLS certificate in the firmware.

-host <hostname>—the host name as it is appears in the MEBX. If TLS/Kerberos is used, the TLS certificate name must be used as the hostname.

-cert <certificate>—the common name of the client certificate is provided here. The certificate is used only when Mutual Authentication is enabled and in TLS mode. If TLS is used the hostname provided must be the same as the hostname used in the TLS certificate in the firmware.

-eoi is used for legacy support for local test. If EOI is NOT specified, WS-Management will be used.

-feat <feature>—specifies the feature to be tested. Feature can only be one of the following options:

Ls—local storage



Ap— agent presence. This test should be used in conjunction with AMTVTR. The –aptool option from AMTVTR should be executed before using this option

Gi—general information test. General information test returns:

- BIOS Version
- FW Version
- Build Number
- Configuration mode
- TLS mode
- Hardware Crypto Status
- Network Interface Status
- SOL settings
- IDE-R settings
- FW Update settings
- Configuration State

All—test all of the features mentioned above, except agent presence test (ap)

AMTVTL will return success or failure based on the availability of the tested features. In case of failure, AMTVTL will give a meaningful error message, such as the following:

PTSDK_STATUS_HARDWARE_ACCESS_ERROR: The Library has identified a HW Internal error.

11.2 MEInfo tool

The MEInfo Tool provides simple aliveness test of the Intel® ME firmware, returns data about Intel® ME and also compares the value of a feature given as input with the Intel® ME actual features value.

11.2.1 MEInfo DOS tool

11.2.2 MEInfo DOS tool from VIP

The MEInfo DOS tool can be accessed from under the following directory:

"Unzipped_folder"\Tools\System Tools\MEInfo\DOS\

(Example: E:\Digital_Office_5.0.0.xxxx\Tools\System Tools\MEInfo\DOS\MEInfo.exe)

11.2.3 Executing MEInfo

Execute the following command on the DOS prompt:

- MEInfo.exe [feat <feature name> <value>]

  Here:

  feat : To be used if only one feature is requested.

  <feature name> : The feature name.

  <value> : The feature value.

The following are examples of executing MEInfo.

1) Execute MEInfo without any optional parameters:

- Example: MEInfo



- Following is a sample of what MEInfo will display after executing the above command successfully:

  Intel® MEInfo Version: 5.0.0.1034

  BIOS Version: VVCBI837.86P

  Intel® AMT code versions:

Flash:	5.0.0
Netstack:	5.0.0
AMTApps:	5.0.0
AMT:	5.0.0
Sku:	0
Vendor ID:	8086
Build Number:	1034
Recover Version:	5.0.0
Recovery Build Num:	1034
Legacy mode:	False
Link status:	Link up
Hardware SKU:	AMT
Cryptography fuse:	Enabled
Flash Protection:	Enabled
Last reset reason:	Firmware Reset
Setup and Configuration:	Completed
AMT mode:	5.0
Bios Mode:	Post Boot
Dedicated Mac Address:	00-11-22-33-44-55
Host Mac Address:	00-11-22-33-44-44

- On error, an error message is printed and a non-zero error level is returned.

2) Execute MEInfo with the optional parameters specified – Here a definite feature should be requested

a) Example: MEInfo feat “Dedicated Mac Address” 00-11-22-33-44-55
   - The output displayed is: Success. The values are identical.

b) Example: MEInfo feat “Dedicated Mac Address” 00-aa-22-33-44-55

- The output displayed is:

  Failed. The values are not identical.

  Dedicated Mac Address: 00-11-22-33-44-55

NOTE:

- The MEInfo DOS tool is supported on the following operating systems: MS-DOS 6.22, Windows* 98 DOS, Free DOS, DRMK DOS.
- For more details on this tool, please refer to the document - ‘Intel® AMT Tools User Guide.pdf’, located in the folder: “Unzipped_folder”\Tools\System Tools\

  (Example: E:\Digital_Office_5.0.0.xxxx\Tools\iAMT Tools\Intel® AMT Tools User Guide.pdf)

11.2.4 MEInfo Windows tool

The MEInfo Windows tool requires that the following be installed on the Intel® AMT system:

Intel® ME Interface driver

11.2.5 MEInfo Windows tool from VIP

The MEInfo Windows tool can be accessed from under the following directory:

“Unzipped_folder”\Tools\System Tools\MEInfo\Windows\

(Example: E:\Digital_Office_5.0.0.xxxx\Tools\System Tools\MEInfo\Windows\MEInfoWin.exe)

11.2.6 Executing MEInfoWin

Execute the below on the Windows* command line:

- MEInfoWin.exe [feat <feature name> <value>]



Here:

feat : To be used if only one feature is requested.

<feature name> : The feature name.

<value> : The feature value.

The following are examples of executing MEInfo.

1) Execute MEInfo without any optional parameters:

- Example: MEInfo

- Following is a sample of what MEInfo will display after executing the above command successfully:

Intel® MEInfo Win Version: 5.0.0.1034

BIOS Version: VVCBI837.86P

Intel® AMT code versions:

Flash:	5.0.0
Netstack:	5.0.0
AMTApps:	5.0.0
AMT:	5.0.0
Sku:	0
Vendor ID:	8086
Build Number:	1034
Recover Version:	5.0.0
Recovery Build Num:	1034
Legacy mode:	False
Link status:	Link up
Hardware SKU:	AMT
Cryptography fuse:	Enabled
Flash Protection:	Enabled
Last reset reason:	Firmware Reset
Setup and Configuration:	Completed
AMT mode:	5.0
Bios Mode:	Post Boot
Dedicated Mac Address:	00-11-22-33-44-55
Host Mac Address:	00-11-22-33-44-44
LMS version:	5.0.0.1034

HECI version: 5.0.0.1034

- On error, an error message is printed and a non-zero error level is returned.

2) Execute MEInfo with the optional parameters specified – Here a definite feature should be requested

a) Example: MEInfoWin feat “Dedicated Mac Address” 00-11-22-33-44-55
   - The output displayed is: Success. The values are identical.

b) Example: MEInfoWin feat “Dedicated Mac Address” 00-aa-22-33-44-55

   - The output displayed is:

     Failed. The values are not identical.

     Dedicated Mac Address: 00-11-22-33-44-55

NOTE:

- The MEInfo Windows tool is a command line executable and is supported on the following operating systems: Windows* 2000 [SP4], Windows* XP [SP 1/2].
- This tool should be run on Windows* OS using Administrator privileges.
- Before running this tool Intel® ME Interface driver must be installed.
- For more details on this tool, please refer to the document - ‘Intel® AMT Tools User Guide.pdf’, located in the folder: “Unzipped_folder”\Tools\System Tools\

  (Example: E:\Digital_Office_5.0.0.xxxx\Tools\System Tools\Intel® AMT Tools User Guide.pdf)

11.3 AMTVTR tool

Intel® AMT system Validation Tool Remote (AMTVTR) is used to check the functionality of a remote Intel® AMT feature which may not be available through the Web GUI. Note that the Web GUI, AMT system Validation Tool local, and AMT system Validation tool Remote do not cover checking of all features of an Intel® AMT device.



11.3.1 AMTVTR tool from VIP

The AMTVTR tool can be accessed from under the following directory:

"Unzipped_folder"\Tools\iAMT Tools\ iAMTVTR\

(Example: E:\Digital_Office_5.0.0.xxxx\Tools\iAMT Tools\ iAMTVTR\AMTVTR.exe)

11.3.2 Executing AMTVTR

On the command line interface (in the management console), execute the following command:

- Usage: AMTVTR.exe [options]

  Where [options] can be:

  -host <Hostname/IP>—the hostname (hostname displayed in the MEBX) or the IP address (Intel® AMT IP address) of the remote Intel® AMT machine (example, 134.176.185.2). If TLS mode is used, the hostname must be used. If TLS is used, the hostname provided must be the same as the hostname used in the TLS certificate in the firmware.

  -user <username>—the username to authenticate for access to Intel® AMT features on a remote machine. If Kerberos authentication is used, the username and password should not be entered.

  -pass <password>—the password corresponding to the username to authenticate for access to Intel® AMT features on a remote machine. If Kerberos authentication is used, the username and password should not be entered.

  -TLS—must be entered if TLS mode is enabled.

  -cert <certificate>—the common name of the client certificate. The certificate is used only when Mutual Authentication is enabled and in TLS mode.

-EOI – Used to test legacy commands. If this option is not used WS-Management protocol will be used. This can not be used with "-dash" option.

-DASH – Specifies DASH protocol to be used instead of WS-Management. If this option is not specified WS-Management protocol will be used. This can not be used with "-eoi" option.

-feat—run the feature to be tested. Only one out of the following options may be used with each call to the program:

- ALL— check all features
- STO— tests the Storage interfaces. This also tests writing to and reading from the flash.

Note: If there are any EACL entries with Enterprise Name = Intel2, using the -sto option will erase these entries.

• HWINV—enumerates the hardware inventory and retrieves hardware details.

Note: When additional asset data information is not captured by the Intel® AMT device due to container space limitations, an appropriate warning will be displayed.

• SD—checks the System Defense feature. Enumerates System Defense filters and policies.

• EM—reads the event log records and log timestamp. **Note:** A warning will be displayed in a case where no event entries exist.

• RC—get the current power states of the Intel® AMT device

• NA—get TCP/IP parameters

• SA—get firmware and configuration mode

• AP—enumerate console watch dog timers

• WC—test wireless profiles that are currently added.

• SI – Simple Identity (only supported in DASH)

• ALOG – Audit Log test

boot —using this option in conjunction with the all or the rc option, the system will perform a power cycle test.

Aptool – Launches agent presence demonstration. This option needs to be executed before running the AP option with AMTVL



-cleanup – Removes the Agent presence policy created by AMTVTR

Redirect – Launches an interactive IDE-R and SOL demo. SOL and IDE-R must be enabled in the MEBX for this option to function.

Sysd – Launches an interactive system defense demo. This demo will create a system defense policy on the specified interface (LAN or WLAN)

-interface [LAN | WLAN| both] – Specifies which interface to enforce the system defense policy

11.4 AMTVTR for testing System Defense features

On the command line interface (in the management console), execute the following command:

- Follow the steps in the below example for testing System Defense functionality:

  a) The system defense demo will create a system defense profile that will block all packets.
  b) C:\ AMTVTR –host <Host name / IP Address> –user admin –pass Admin`12 –sysd

  c) This will prompt the user to enable the profile. After the profile is enabled, all packets will be blocked, but the ME will still remain active.

  d) The profile will remain even after system reset occurs. To remove the profile the user should disable when prompted or use the cleanup option.

  e) C:\ AMTVTR –host <Host name / IP Address> –user admin –pass Admin`12 –sysd -cleanup

§

12 Supplemental notes – CRB

Please refer to these supplemental notes for important jumper settings and locations, as well as BIOS implementation information you need to know in order to bring up the CRB successfully.

12.1 BIOS Implementation

When you first receive your CRB system it will be set to load BIOS from the SPI flash.

Important note:
A small image called HALTROM comes loaded on the SPI flash when you receive the CRB. This HALTROM code is necessary to enable the system to boot. So remember: **if you erase the SPI flash, the system will no longer boot** because when you erase the flash, you also erase the HALTROM code. This HALTROM code is present in all the released firmware image builds, but you cannot program an image onto the flash if the system won't boot! If you corrupt or erase the flash, you can replace the SPI chips with pre-programmed replacement parts **if** your board has socket-based SPI parts. If the SPI parts are hard-wired, you would need to have the board re-worked/replaced to remedy this.

12.2 JUMPERS

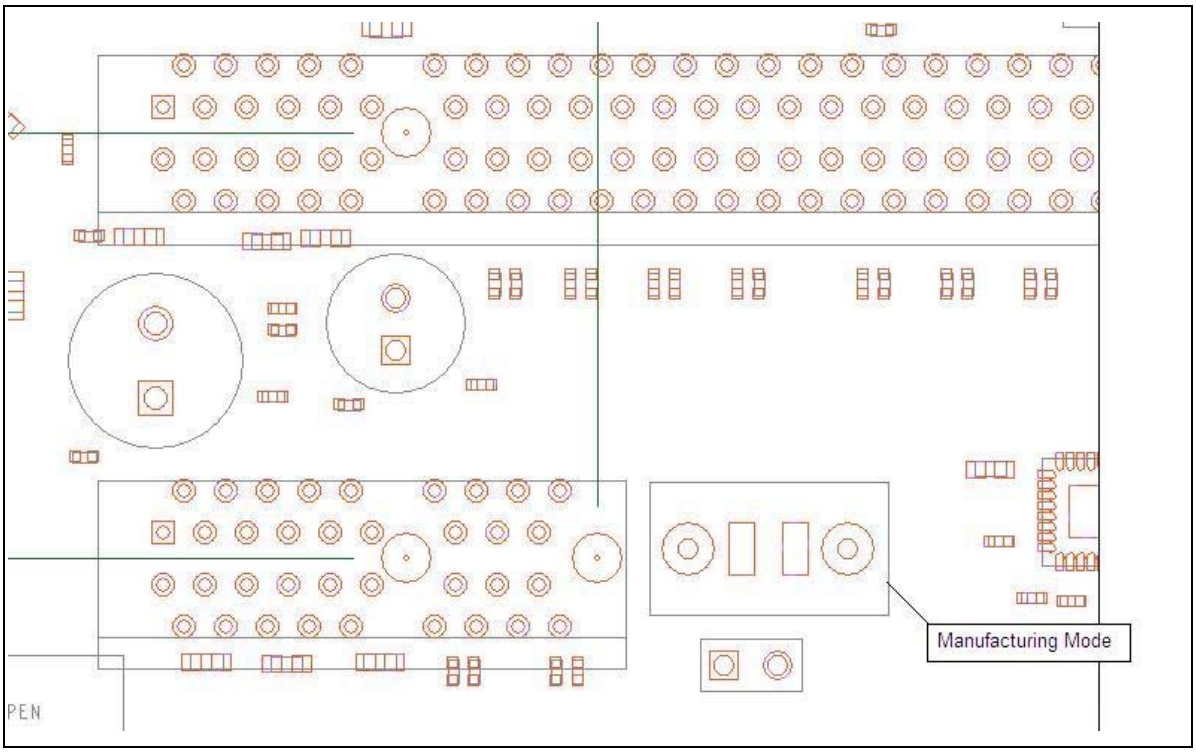
There are three jumpers on the board that affect your bring-up effort. These jumpers are identified in the following subsections.

12.2.1 Manufacturing Mode (Flash Override) jumper

The Manufacturing Mode (Flash Override) jumper permits or denies CPU write access to the SPI flash devices at the hardware level. When this jumper is open, the CPU will not have write access to the flash (if those permissions are not present in FW). Normally these permissions are set in the FW image, so this jumper would normally only be set in the event you need to override the FW permission settings to write a new image to flash. Close this jumper only if you need to override FW permissions to load a new ME FW image onto the

flash part (when permissions set in FW would otherwise prevent you from doing so). This jumper should be removed (left open) under normal operating conditions, wherein access permissions to the flash are controlled by the FW.

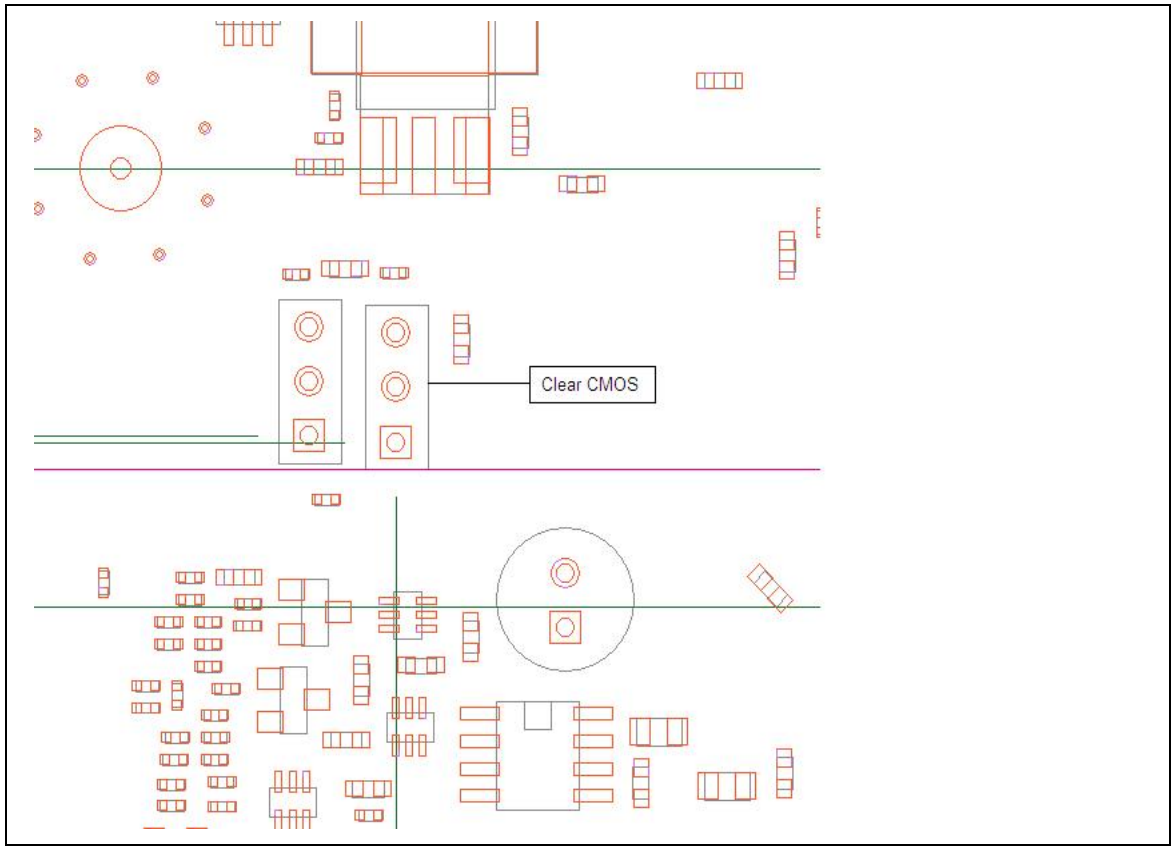
Figure 14. Manufacturing Mode jumper location:



12.2.2 Clear CMOS jumper

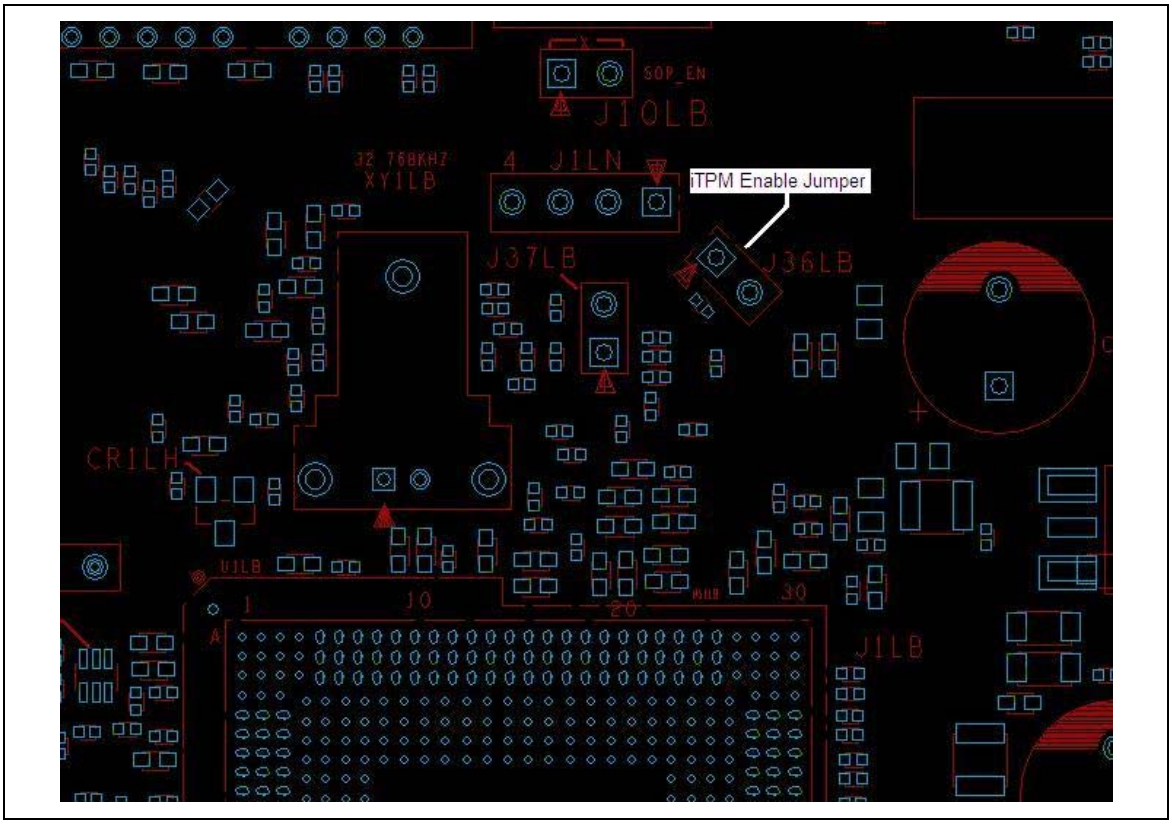
If you program an image onto the flash and later you need to boot from the BIOS on the on-board FW hub, such as in troubleshooting boot issues, you should always clear CMOS in addition to changing the BIOS relocation jumper to load BIOS from the FW hub. This is because the CMOS may still contain settings from the BIOS that loaded from flash. Closing the CMOS Clear Jumper will reset CMOS to default settings. It is always good practice to clear CMOS as a first step to troubleshooting boot issues and any time you change the BIOS relocation jumper.

Figure 15. CMOS Clear jumper location:



12.2.3 iTPM jumper

Figure 16. iTPM Enable jumper location:



12.3 Testing SOL and IDE-R features

- Follow the steps in the below example for testing SOL and IDE-R functionality:
    - Open a command prompt window (first) on the management console. At the command prompt, execute the following command:

      AMTVTR.exe <IP address of ME> <login> <password> This command will give you the options to select to run SOL and IDE-R
    - The following message will be displayed
    - a: Open SOL Session
    - b: Close SOL Session
    - c: Open IDER Session
    - d: Close IDER Session

- e: Regular boot
- f: SOL boot
- g: SOL boot to BIOS setup
- h: IDER Floppy boot
- i: IDER CD boot
- j: SOL + IDER Floppy boot
- k: SOL + IDER CD boot.
- m: enable Redirection listener
- n: disable Redirection listener
- l: Display Menu Option
- x: Exit

SOL

- Press m to enable listener
- Press a to open a SOL session
- Open Telnet terminal window with serial port at COM3 in host S0
- Type commands in Telnet terminal screen and you should see them on the Management Console.

IDE-R

- Press m to enable listener
- Use c, h for IDER floppy boot
- Use c, I for IDER CD boot
- Open device manager on host in S0 and “REFRESH” the device manager, you will see a “Intel virtual floppy or CDROM”
- Go to my computer and you can access the CDROM or floppy connected to the Management Console.

SOL / IDEr Installation Notes:

Summary:
The LMS/SOL installer will abort with an error message when you attempt to run it on systems which do not yet have the required hardware devices present. The error message which is produced in that scenario is shown below.

Customers should expect to see this behavior if they attempt to install device drivers when the required hardware devices are not present. For OEMs who need to pre-install just the software applications, a new command-line switch is now available with currently-shipping versions of the installer which allows installing just the software applications without scanning for hardware. When you specify this switch on the command line no drivers are installed. Only the software applications and services are installed.



Details:

- If the related hardware devices are not present in the system when you run the LMS/SOL installer, the program will produce the following error message:

  "*The system does not meet the minimum requirements.*"

  The installer will abort without installing the device drivers because the necessary hardware devices are not present. This protects against having device drivers on the system which cannot function.

- **This behavior is by-design and should be expected.**

- Intel has recently added a new switch (-nodrv) which can be specified when you run the LMS/SOL installer. When the -nodrv switch is specified, the software applications and services will be installed, but the installer will not scan for the necessary hardware and no device drivers will be installed. This switch is available with currently-shipping versions of the LMS/SOL installer and is useful for OEMs who wish to pre-install the software applications, but wish to install device drivers at some other time in their process.

12.4 Final checklist

This section provides a summarized checklist of the principal steps required to bring up an Intel® AMT firmware release on the CRB platform, arranged in the sequence that they should be performed. Please use this checklist to ensure no steps were missed as you went through the detailed procedures outlined herein above.

	Build FW Image using Fitc (Page 10)
	Set optimal defaults in system BIOS
	Shut down and close the Manufacturing Mode jumper
	Program the image onto the SPI Flash using Fpt (Page 20)
	Verify the image (Fpt /v)
	Shut down and close the BIOS Relocation jumper
	Restart the system to boot BIOS from the Flash
	Enter BIOS setup and enable AMT
	Restart the system and use CTRL-P to enter MEBx
	Set desired ME and AMT settings
	Restart the system and begin AMT testing

§