Intel® DQ57TM UEFI 2.3.1 Development Kit

Getting Started Guide

February 2012

Version 1.3

Table of Contents

Preface

Introduction

Welcome to the *Intel® DQ57TM UEFI 2.3.1 Development Kit Getting Started Guide* (Getting Started Guide). This guide explains how to install a Unified Extensible Firmware Interface (UEFI) 2.3.1-compliant BIOS image on an *Intel® DQ57TM UEFI 2.3.1 Development Kit* (Development Kit) that support development, testing, and debugging of UEFI 2.3.1-compliant drivers and applications.

Scope

This guide explains the process of installing an *Intel® DQ57TM UEFI 2.3.1 Development Kit* BIOS image on an assembled or purchased *Intel® DQ57TM UEFI 2.3.1 Development Kit*.

Note: *Firmware developer platforms must be assembled from components found in the supported and recommended hardware components section Table 1-1.*

The guide covers:

- Overview of the *Intel® DQ57TM UEFI 2.3.1 Development Kit*
- Upgrading to the latest *Intel® DQ57TM UEFI 2.3.1 Development Kit* BIOS image
- Programming in a new *Intel® DQ57TM UEFI 2.3.1 Development Kit* BIOS image
- Verifying that the *Intel® DQ57TM UEFI 2.3.1 Development Kit* BIOS image is programmed correctly
- Backing up and erasing existing production BIOS from a motherboard using an SPI Flash programmer
- Troubleshooting information to help if errors occur

This guide assumes that you possess PC hardware assembly skills and familiarity with Microsoft Windows* environments. This guide does not explain PC assembly or Microsoft* operating systems.

Terminology

Note: *A complete list of acronyms is provided in the Appendix A: Acronyms and Glossary.*

Commonly used terms in this guide include:

Developer platform	An *Intel® DQ57TM UEFI 2.3.1 Development Kit* (the target PC). This is a desktop PC built using components from the Supported and Recommended Hardware Components list in *Section 1* of this guide. This PC is programmed with the *Intel® DQ57TM UEFI 2.3.1 Development Kit* BIOS image in order to enable UEFI 2.3.1 functionality.
Host PC	A functional PC with a Microsoft Windows XP* or Windows 7* operating system.
Target PC	An *Intel® DQ57TM UEFI 2.3.1 Development Kit* (the developer platform).
UDK 2010	*UEFI Development Kit 2010*, which supports the UEFI 2.3.1 and PI 1.2 specification.
UEFI	Unified Extensible Firmware Interface.

For More Information

For more information about the *Intel® DQ57TM UEFI 2.3.1 Development Kit* and the *Intel® UDK 2010*, visit the Intel Web site.

To purchase a pre-assembled *Intel® DQ57TM UEFI 2.3.1 Development Kit*, visit the website at Hard Drives Northwest*.

For more information about the *UEFI 2.3.1 Specification*, including features and advantages of using UEFI 2.3.1 over previous versions, visit the UEFI Forum website.

Section 1:
Overview

Introduction

The *Intel® DQ57TM UEFI 2.3.1 Development Kit* enables BIOS engineers to design, test, and debug Unified Extensible Firmware Interface (UEFI) drivers and applications on a UEFI 2.3.1-compliant system.

The *Intel® DQ57TM UEFI 2.3.1 Development Kit* is based on the Intel DQ57TM motherboard, supplemented with UEFI 2.3.1-compliant platform BIOS images, a firmware update utility, and user documentation. These items are detailed later in this section.

This guide explains how to install the developer platform UEFI BIOS image on an *Intel® DQ57TM UEFI 2.3.1 Development Kit*.

An *Intel® DQ57TM UEFI 2.3.1 Development Kit* can be purchased, or it can be assembled and configured, in a lab or home workshop.

Important: *An Intel® DQ57TM UEFI 2.3.1 Development Kit must use components on the supported hardware list (see Supported and Recommended Hardware Components section - Table 1-1).*

UEFI 2.3.1 Benefits

A UEFI 2.3.1 BIOS offers many benefits to developers and users:

- Support for large hard drives, such as 2.2 TB drives
- Secure boot, with the ability to validate the BIOS image on the machine
- Full networking stack, including IPv6 support
- Advanced menu system and setup screens, which are more convenient, robust, and allow for better manageability of machine features
- Flexible look-and-feel, so vendors can “brand” their machines more easily

Intel® DQ57TM UEFI 2.3.1 Development Kit

The *Intel® DQ57TM UEFI 2.3.1 Development Kit* offers several benefits for developers and their UEFI 2.3.1-based products:

- Supports Intel® UDK 2010 and UEFI 2.3.1 development and debug
- Long lifetime hardware platform support from Intel
- All hardware and software components are commercially available, without a nondisclosure agreement (NDA)
- Allows developers to build platforms on demand for development, debugging, or validation
- Can be purchased pre-assembled or built from commodity hardware

Intel® DQ57TM UEFI 2.3.1 Development Kit BIOS Images

The *Intel® DQ57TM UEFI 2.3.1 Development Kit* (developer kit) includes three UEFI 2.3.1-compliant BIOS images:

- `UDK2010_TM_B9_release.rom` – The *release* version of the BIOS, with debugging features disabled.
- `UDK2010_TM_B9_debug.rom` – The *debug* version of the BIOS, with debug output redirected to the serial port (**COM1**).
- `UDK2010_TM_B9_srcdbg.rom` – The *source level debug* version of the BIOS, enabling support for the Intel® UEFI Development Kit Debugger Tool using the serial port (**COM1**).

The BIOS images included in the developer kit were only validated against the supported hardware component list.

CAUTION: *Installing the BIOS image on an unsupported motherboard may render the motherboard unusable until it is re-flashed with a backup copy of the motherboard's original BIOS. Use only supported components with the Intel® DQ57TM UEFI 2.3.1 Development Kit.*

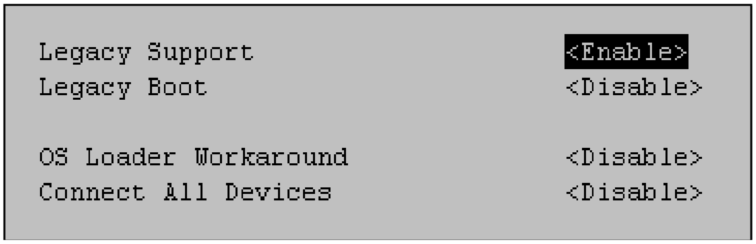
Enabling & Disabling Legacy BIOS Support

Starting with BIOS Revision SDV.TM.B8, support for legacy BIOS compatibility is controlled by options in the ***Device Manager -> Boot*** setup menu. This allows the Compatibility Support Module (CSM) to be enabled or disabled without re-flashing the BIOS. These options can be used in different scenarios:

- ***Legacy Support: Enable*** (default). This setting implements a *UEFI Class 2* system, providing interfaces for UEFI and legacy BIOS services

(INT 19h boot, INT 10h video, etc.) using the CSM. The CSM included in this BIOS image was provided by Insyde* Software.

- ***Legacy Support: Disable***. This setting implements a *UEFI Class 3* system. Legacy BIOS interfaces and legacy option ROMs are not supported due to the absence of the CSM. This setting may be required to utilize UEFI Secure Boot capabilities.



```
Legacy Support                  <Enable>
Legacy Boot                     <Disable>

OS Loader Workaround            <Disable>
Connect All Devices             <Disable>
```

Note: *Due to differences in the legacy video BIOS (with CSM) and UEFI GOP driver (without CSM), pre-boot video may only appear on one Intel® DQ57TM Desktop Board video output (DVI-I or DVI-D). The primary display output may change after upgrading the BIOS or changing the* ***Legacy Support*** *setting. If no video is displayed at boot, please switch output connectors.*

Supported and Recommended Hardware Components

Table 1-1 below describes the supported and recommended PC hardware components for the Intel® DQ57TM UEFI 2.3.1 Development Kit.

Table 1-1. Supported and recommended hardware components for the Intel® DQ57TM UEFI 2.3.1 Development Kit

Supported hardware and firmware	Notes
Intel® DQ57TM Desktop Board	**You must use the commercially available Intel DQ57TM Desktop Board**.
2nd generation Intel® Core™ i5-650 Processor with heat sink	**You must use a 2nd gen Intel Core i5-650 processor with heat sink.** This processor is available commercially.
Intel® DQ57TM UEFI 2.3.1 Development	**You must use one of the BIOS images from the Intel® DQ57TM UEFI 2.3.1 Development Kit.** The BIOS image replaces the Intel® DQ57TM Desktop Board BIOS and enables

Supported hardware and firmware	**Notes**
Kit BIOS image	UEFI 2.3.1 support. The BIOS images in the developer's kit have been validated to work only with the specific processor and motherboard listed above.
Serial cable	**You must use a DB-9 male to 10-pin IDC socket serial cable** for the serial port on the Intel DQ57TM Desktop Board.

Recommended hardware	**Notes**
4GB (2 x 2GB) DDR3 1333 memory	The Intel DQ57TM Desktop Board supports up to 16 GB RAM. You should use at least 2 GB of DDR3 memory in your development build.
500 W Power supply	A power supply that meets the requirements of the Intel DQ57TM Desktop Board is adequate. However, consider installing a minimum 500W power supply.
SATA HDD 500GB	You should install a hard drive that has at least 500 GB. Testing hard drives over 2.2TB requires the use of UEFI.
SATA DVD-RW Optical Disk Drive or other install media	Most developers use a DVD drive to perform this type of UEFI build. Make sure you have an appropriate DVD drive or other media (such as a network connection or USB drive) appropriate for installing the operating system.
Micro-ATX Chassis	A chassis is not required. However, if you want the build to be portable, consider using a micro-ATX or similar chassis.
USB keyboard and USB mouse	Including a USB keyboard and mouse with the developer's platform allows you to input UEFI shell commands and navigate BIOS menus.
Monitor	A monitor is recommended in order to view console output.

Installation Requirements

The following skills, components, and tools are required to install and update an *Intel® DQ57TM UEFI 2.3.1 Development Kit* with a UEFI 2.3.1-compliant BIOS image:

- PC assembly skills
- *Intel® DQ57TM UEFI 2.3.1 Development Kit*
- USB flash drive
- SPI flash programmer with test clip, and software utility (optional)

The rest of this discussion describes the tools and skills necessary to perform the procedure(s) in this guide.

PC Assembly Skills. You must have the skills necessary to assemble the hardware of a typical desktop PC, and be familiar with BIOS. This guide does not cover PC assembly. You must use a PC built with components from the supported hardware components list as described in Table 1-1. The BIOS images included in the *Intel® DQ57TM UEFI 2.3.1 Development Kit* are validated only for components on the supported hardware list.

Intel® DQ57TM UEFI 2.3.1 Development Kit. The developer kit includes BIOS images for the *Intel® DQ57TM UEFI 2.3.1 Development Kit*, a firmware update utility, and user documentation. The developer kit can be downloaded from the Intel UEFI Driver and Application Tool Resources webpage. The kit includes:

- **UEFI 2.3.1 compliant BIOS images** for the *Intel® DQ57TM UEFI 2.3.1 Development Kit*. During installation, the appropriate BIOS image replaces the existing *Intel® DQ57TM Desktop Board* BIOS and adds UEFI 2.3.1 support. The developer's kit includes three BIOS images:
    - `UDK2010_TM_B9_release.rom` – The *release* version of the BIOS, with debugging features disabled.
    - `UDK2010_TM_B9_debug.rom` – The *debug* version of the BIOS, with debug output redirected to the serial port (**COM1**).
    - `UDK2010_TM_B9_srcdbg.rom` – The *source level debug* version of the BIOS, enabling support for the Intel® UEFI Development Kit Debugger Tool using the serial port (**COM1**).
- **Firmware Update Utility for the UEFI Shell.** The firmware update utility (`FirmwareUpdate.efi`) is a UEFI Shell software tool provided by Intel, used to update the BIOS image on the *Intel® DQ57TM UEFI 2.3.1 Development Kit*. The firmware update tool is used to flash the platform BIOS and patch the MAC address for the motherboard.
- **User Documentation** for the *Intel® DQ57TM UEFI 2.3.1 Development Kit*, including:

- *Intel® DQ57TM UEFI 2.3.1 Development Kit Getting Started Guide (this document):* `Getting_Started-UDK2010_FIRMWARE_DEV_PLT_Guide.pdf`
- *Intel® DQ57TM UEFI 2.3.1 Development Kit Release Notes:* `ReleaseNotes.txt`
- Instructions for the *Firmware Update tool*: `ReadMe.txt`
- Software Tools License Agreement: `EULA.pdf`

USB Flash Drive. You will need a FAT-formatted flash drive to copy the firmware update tool from the host system to the target PC.

SPI Flash Programmer, Test Clip and Software Utility (Optional). The *Intel® DQ57TM UEFI 2.3.1 Development Kit* may require a hardware-based reprogramming of the SPI flash part, since the development and testing of pre-production products may corrupt the flash image. In this case the developer must use a third-party SPI flash programmer, test clip, and corresponding software application to transfer the BIOS image from the host PC to the flash device on the firmware developer platform (the *target PC*). SPI reflash instructions in this document are based on products from Dediprog Technology Co, Ltd.*, which have been verified to work with the supported hardware (see Table 1-1).

- *Dediprog SF100 SPI flash programmer*: Allows the developer to read and write to a flash device that is already soldered on a motherboard (no need to unsolder and reinstall the flash device).
- *Dediprog Test Clip*: Connects the third-party SPI flash programmer to the flash device on the developer platform motherboard (see Table 1-2).
- *Dediprog SF100 USB Software Tool Chain*: A software utility that controls the flash programmer hardware, allowing the developer to read, erase, and/or write a flash image to/from a flash device. For the procedure described in this guide, the software utility is used to load a BIOS image onto the flash programmer and transmit the image to the flash device on the motherboard. The Dediprog software utility works only with the Dediprog SPI flash programmer listed above. Table 1-2 summarizes the tools required to complete this installation.

Overview of SPI Flash Programmer Environment

The installation environment for the *Intel® DQ57TM UEFI 2.3.1 Development Kit* consists of a host PC, SPI flash programmer with test clip, software, and an *Intel® DQ57TM UEFI 2.3.1 Development Kit* (target PC) assembled from components on the list of supported hardware. Figure 1-1 below shows the installation environment for the *Intel® DQ57TM UEFI 2.3.1 Development Kit*.

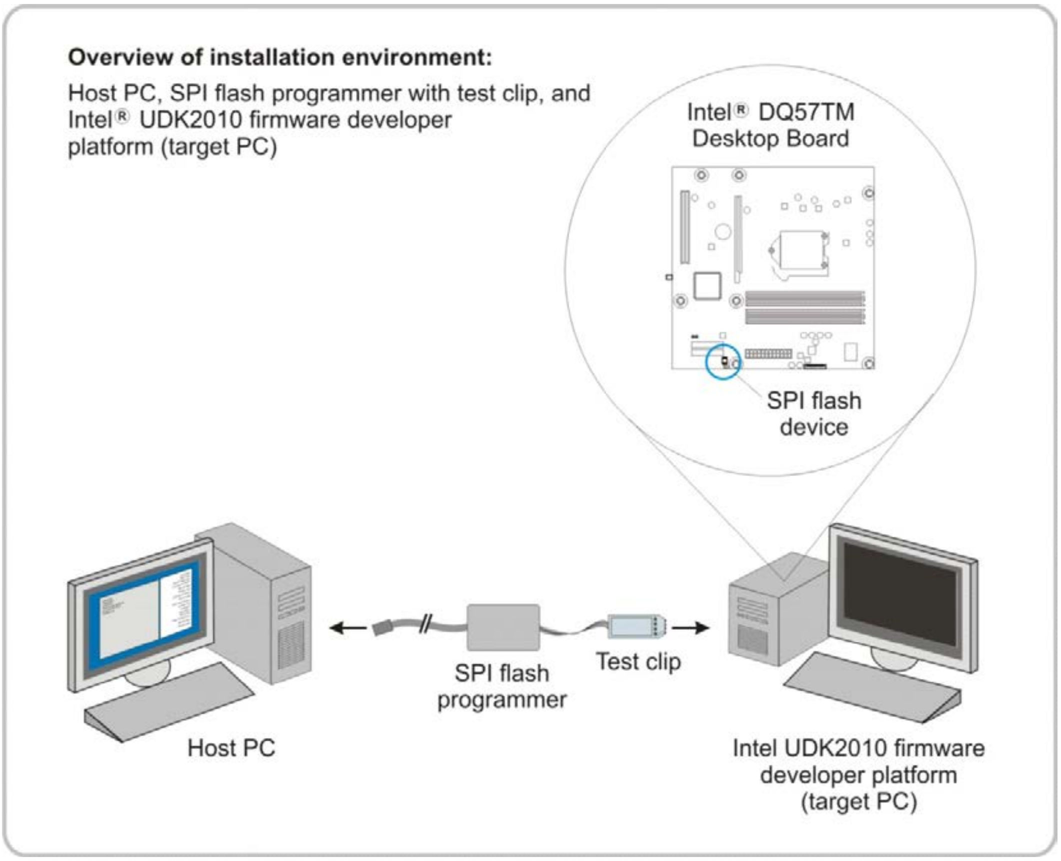

Overview of installation environment:
Host PC, SPI flash programmer with test clip, and Intel® UDK2010 firmware developer platform (target PC)
Intel® DQ57TM Desktop Board
SPI flash device
SPI flash programmer
Test clip
Host PC
Intel UDK2010 firmware developer platform (target PC)

Figure 1-1. Overview of the SPI Flash Programming Environment

The host PC is a Microsoft Windows XP* or Windows 7* class system. The host PC is connected to the target PC's SPI flash device, via a third-party SPI flash programmer (with test clip). Information about installation requirements, including third-party flash programmer products, is provided after Table 1-2, later in this section.

Table 1-2. Software and hardware tools necessary for BIOS installation

Firmware or firmware tool	**Description**
Intel® DQ57TM UEFI 2.3.1 Development Kit	The Intel® DQ57TM UEFI 2.3.1 Development Kit is provided by Intel, and contains: • Intel® DQ57TM UEFI 2.3.1 Development Kit BIOS images • Intel DQ57TM Firmware Update Utility: *FirmwareUpdate.efi* • Related documentation (Getting Started Guide, EULA, Release Notes, etc.) **You must use the firmware update utility included in the kit, and you must use one of the supported BIOS images** in order to complete the procedures in this guide. Multiple downloadable UEFI 2.3.1 BIOS images are provided in the kit. These images will work only with the Intel® DQ57TM Desktop Board and 2nd generation Intel® Core™ i5-650 Processor with heat sink (refer to Table 1-1, earlier in this section). You can download the zip file from the Intel Web site at: http://developer.intel.com/technology/efi/uefi-ihv.htm
USB flash drive	**You will need a FAT-formatted USB flash drive** to copy the firmware update tool.
DediProg* SF100 flash programmer, test clip, and software utility (optional)	**The third-party SPI flash programmer is only required** to install Intel® DQ57TM UEFI 2.3.1 Development Kit BIOS image on the Intel® DQ57TM motherboard if it has not been previously programmed. The test clip connects the programmer to the flash device on the motherboard. The software utility is required in order for you to use the flash programmer. The SPI flash programmer, test clip, and software utility are third-party products available from Dediprog Technology Co, Ltd. • DediProg Technology Co, Ltd., SF100 programmer • DediProg Technology Co, Ltd., ISP Test Clip (ISP-TC-8) • DediProg Technology Co, Ltd., SF100 USB software tool chain
Host system OS	**Microsoft Windows XP* or another appropriate Windows*** operating system compatible PC.

Overview of Firmware Upgrade/Installation

There are two procedures outlined in this document for upgrading or installing a BIOS image on the *Intel® DQ57TM UEFI 2.3.1 Development Kit*. Please review these procedures before attempting a firmware upgrade or installation.

Firmware Upgrade (software based)

This procedure uses the `FirmwareUpdate.efi` utility from the UEFI Shell to upgrade the BIOS and patch the MAC address. No hardware-based SPI programmer is required. This procedure is recommended for BIOS upgrades on properly booting systems that are running older versions of the Intel® DQ57TM UEFI 2.3.1 Development Kit BIOS.

1. Set motherboard in configuration mode (BIOS CFG Jumper, pin 2-3).
2. Power up the target PC and boot to the UEFI shell.
3. Use the UEFI Shell firmware update utility (`FirmwareUpdate.efi`) to patch the BIOS with the motherboard's MAC address (see Figure 2-1):

   ```
   FirmwareUpdate –f UDK2010_TM_B9_xxxx.rom –m 0011AA33CC55
   ```

   Notes: *a. The system will reset after the update has been applied.*
   b. Replace `UDK2010_TM_B9_xxxx.rom` with the proper ROM name.
   c. Replace `0011AA33CC55` with the motherboard's MAC address, which can be found on the motherboard.
   d. Upgrading the flash will restore Setup and Boot Manager settings to their default values. Any previous changes to Setup or Boot Manager values will be cleared in the upgrade process.

4. After the platform resets, verify that the BIOS functions correctly by entering setup, verifying the BIOS version string matches the expected value for the new BIOS version, and booting to the UEFI Shell.
5. Set motherboard back to normal mode (BIOS CFG Jumper, pin 1-2).

Once a BIOS update is verified, the *Intel® DQ57TM UEFI 2.3.1 Development Kit* is ready for use in UEFI development.

Firmware Installation (hardware based)

This procedure uses a hardware-based SPI programmer to flash the *Intel® DQ57TM UEFI 2.3.1 Development Kit* BIOS image then uses the UEFI Shell

`FirmwareUpdate.efi` utility to patch the MAC address. This procedure is recommended for BIOS upgrades on improperly booting systems or flashing the *Intel® DQ57TM UEFI 2.3.1 Development Kit* BIOS onto a retail motherboard for the first time.

1. Set motherboard in configuration mode (BIOS CFG Jumper, pin 2-3).
2. Download the Development Kit BIOS images, firmware update tool and user documentation.
3. Install the Dediprog software utility on the Host PC.
4. Prepare the Development Kit (or target PC) for the BIOS update.
5. Create a backup copy of the motherboard's original BIOS image.
6. Erase the existing BIOS from the motherboard.
7. Write the Development Kit BIOS image to the target PC.
8. Reassemble the target PC.
9. Power up the target PC and boot to the UEFI shell.
10. Use the UEFI Shell firmware update utility (`FirmwareUpdate.efi`) to patch the BIOS with the motherboard’s MAC address:

    ```
    FirmwareUpdate –m 0011AA33CC55
    ```

    Note: *Replace `0011AA33CC55` with the motherboard’s MAC address (Figure 2-1) which can be found near the ATX power connector.*

11. After the platform resets, verify that the BIOS functions correctly by entering setup and booting to the UEFI Shell.
12. Set motherboard back to normal mode (BIOS CFG Jumper, pin 1-2).

Once a BIOS update is verified, the *Intel® DQ57TM UEFI 2.3.1 Development Kit* is ready for use in UEFI development.

Important Notes

These instructions only apply to the *Intel® DQ57TM UEFI 2.3.1 Development Kit*. A platform can be assembled from the Supported and Recommended Hardware Components list Table 1-1, or purchased pre-assembled from Hard Drives Northwest (recommended).

For more information about ordering third-party Dediprog hardware and software tools, visit the Dediprog website.

Section 2:
Firmware Upgrade

Introduction

This procedure uses a UEFI Shell utility to upgrade the BIOS and patch the MAC address. This procedure is recommended for BIOS upgrades on properly booting systems that are running older versions of the *Intel® DQ57TM UEFI 2.3.1 Development Kit* BIOS.

For BIOS upgrades on improperly booting systems or a *first time flash* of the *Intel® DQ57TM UEFI 2.3.1 Development Kit* BIOS onto a retail motherboard for the first time, please refer to the Firmware Installation procedure in Section 3.

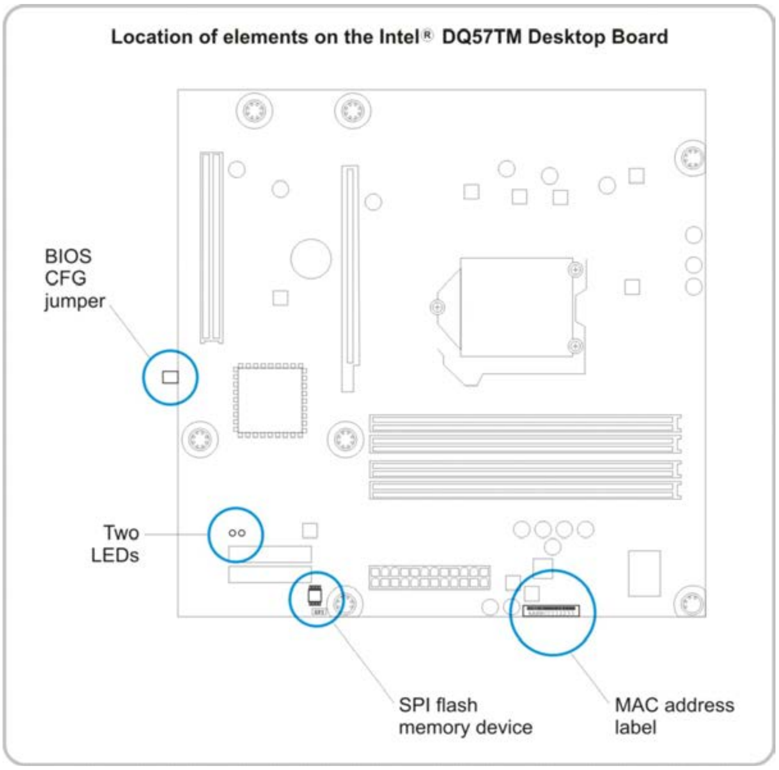


Figure 2-1. Intel® DQ57TM Desktop Motherboard – key elements

Figure 2-1 shows the location of the BIOS CFG jumper, the SPI flash memory device, and the MAC address label. The left LED is the green standby-power indicator LED; the right LED is the red Intel ME indicator LED.

Follow the steps below to upgrade the UEFI 2.3.1 BIOS on the target PC.

Step 1. Ready the Intel® DQ57TM UEFI 2.3.1 Development Kit

1. Make sure target PC is ready for this procedure. The *Intel® DQ57TM UEFI 2.3.1 Development Kit* must be built with hardware components listed in the supported hardware components list.
2. Set motherboard in configuration mode (BIOS CFG Jumper, pin 2-3).

   CAUTION: *If you try to install the supported BIOS image on an unsupported motherboard or processor, the motherboard may be unusable until it is reflashed with a backup copy of the motherboard's original BIOS. Use only supported components with the Intel® DQ57TM UEFI 2.3.1 Development Kit.*

   CAUTION: *Handle the motherboard properly during installation to prevent damage to the board and its flash memory.*

Step 2. Download Files

1. Go to the Intel® UEFI Driver and Application Tool Resources web page and download the latest *Intel® DQ57TM UEFI 2.3.1 Development Kit* release, saving the ZIP file to an appropriate destination.
2. Unzip the files and verify that these individual files are present:
   - `UDK2010_TM_B9_release.rom` – The *release* version of the BIOS, with debugging features disabled.
   - `UDK2010_TM_B9_debug.rom` – The *debug* version of the BIOS, with debug output redirected to the serial port (COM1).
   - `UDK2010_TM_B9_srcdbg.rom` – The *source level debug* version of the BIOS, enabling support for the Intel® UEFI Development Kit Debugger Tool using the serial port (COM1).
   - Firmware update tool: `FirmwareUpdate.efi`
3. Copy the firmware update tool and BIOS image (ROM) files to a FAT-formatted USB flash drive.

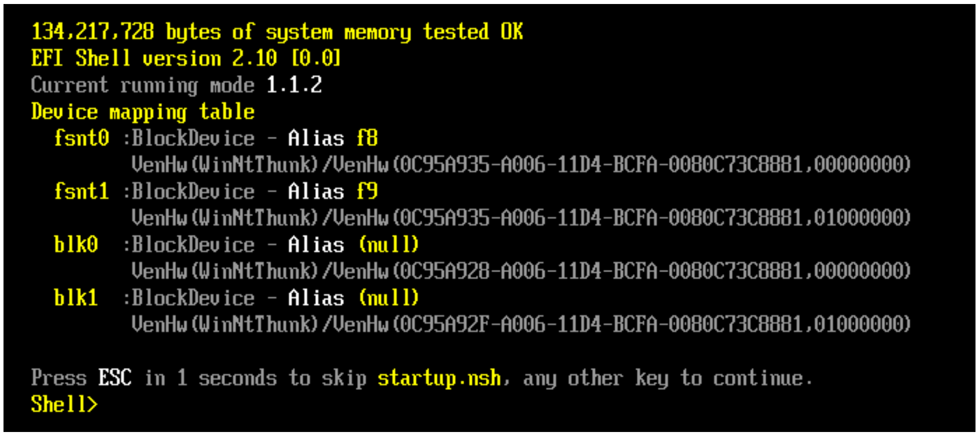
   Note: *Always use the latest version of* `FirmwareUpdate.efi` *included with the Intel® DQ57TM UEFI 2.3.1 Development Kit. Using older* *versions of the firmware update tool with the latest firmware may cause problems with the upgrade process.*

Step 3. Record the Motherboard MAC Address

1. Look on the motherboard and locate the small sticker that lists the MAC address for the board.
2. Write the MAC address on a piece of paper. Keep this paper available. You will need it later when writing and verifying the new BIOS.

Step 4. Boot the Target PC to the UEFI Shell

1. Check all connections between components on the target PC.
2. Connect the target PC to an appropriate power source.
3. Insert the USB key containing the firmware update tool & BIOS image into an open USB port on the target PC.
4. Power up the target PC.
5. During boot, in the initialization screen, press the appropriate key(s) to enter BIOS setup screens.
6. Navigate to the Boot Manager and load the built-in UEFI Shell.
7. Verify the system boots to the built-in UEFI Shell.



```
134,217,728 bytes of system memory tested OK
EFI Shell version 2.10 [0.0]
Current running mode 1.1.2
Device mapping table
  fsnt0 :BlockDevice - Alias f8
         VenHw(WinNtThunk)/VenHw(0C95A935-A006-11D4-BCFA-0080C73C8881,00000000)
  fsnt1 :BlockDevice - Alias f9
         VenHw(WinNtThunk)/VenHw(0C95A935-A006-11D4-BCFA-0080C73C8881,01000000)
  blk0  :BlockDevice - Alias (null)
         VenHw(WinNtThunk)/VenHw(0C95A928-A006-11D4-BCFA-0080C73C8881,00000000)
  blk1  :BlockDevice - Alias (null)
         VenHw(WinNtThunk)/VenHw(0C95A92F-A006-11D4-BCFA-0080C73C8881,01000000)

Press ESC in 1 seconds to skip startup.nsh, any other key to continue.
Shell>
```

Figure 2-2. UEFI Shell

Step 5. Flash the BIOS & Patch the MAC Address

1. Make sure you have the firmware update tool copied onto a FAT-formatted USB flash drive.

***Note:** The Intel®-supplied firmware update tool does not need to be installed on the target PC. This tool works from the USB drive. It is assumed that you have only one USB drive on the target PC (software-development platform).*

2. Connect the FAT-formatted USB flash drive to USB port on target PC.
3. Boot the system to the UEFI shell. If the system is already at the UEFI shell, reboot the system to make sure the USB flash drive is correctly mapped.
4. Use the UEFI **map** command to identify the file system associated with the attached USB flash drive. This example uses `fs1` as the file system:

```
Shell> map fs*:
```

5. Press **Enter**.
6. The system then displays the list of file systems, including USB drives. Look for the entry listed as Removable HardDisk that has USB listed in the device path. In this example procedure, the entry is `fs1`.
7. Select the file system associated with the USB flash drive by typing the following command:

```
Shell> fs1:
```

8. Press **Enter**.
9. Run the firmware update utility (`firmwareupdate.efi`) to update upgrade the BIOS and patch the MAC address for the integrated LAN device. In the following examples, replace the string `001122AABBCC` with the MAC address found on the label on your motherboard.

   Example: To upgrade to the release version of the BIOS …

   ```
   FirmwareUpdate –f UDK2010_TM_B9_release.rom –m 0011AA33CC55
   ```

   Note: Replace the filename in the examples with the proper BIOS image name (if different file is used).

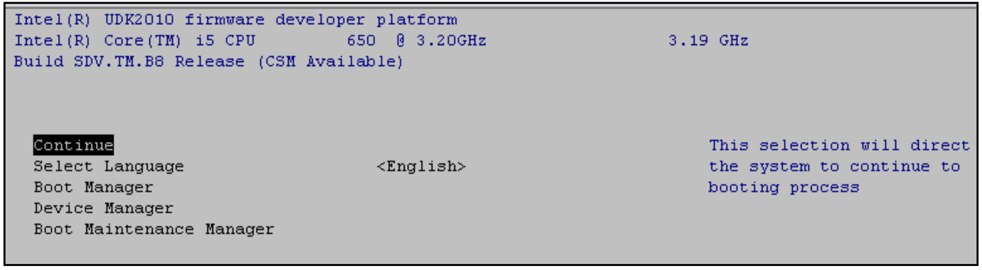
   ***Note:** Replace `0011AA33CC55` with the motherboard's MAC address, which can be found on a decal near the ATX power connector. The MAC address is case insensitive.*

   ***Note:** Upgrading the flash will restore Setup and Boot Manager settings to their default values. Any previous changes to Setup or Boot Manager values will be cleared in the upgrade process.*

10. Press **Enter**. The system automatically reboots when the update is complete.

Step 6. Verify the BIOS Upgrade

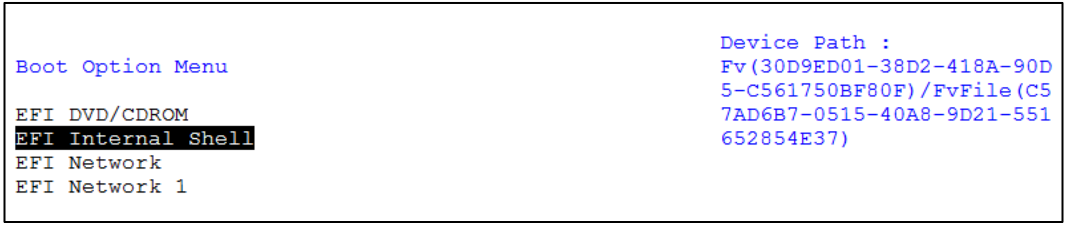
1. After system reset, enter the BIOS Setup menu and check the BIOS version string (3rd from the top) to verify it matches the upgrade.



```
Intel(R) UDK2010 firmware developer platform
Intel(R) Core(TM) i5 CPU         650  @ 3.20GHz               3.19 GHz
Build SDV.TM.B8 Release (CSM Available)


  Continue                                                      This selection will direct
  Select Language                  <English>                    the system to continue to
  Boot Manager                                                  booting process
  Device Manager
  Boot Maintenance Manager
```

2. After verifying the BIOS version string, select the **Boot Manager** and use the **Boot Option Menu** to load the UEFI Shell.



```
                                                    Device Path :
Boot Option Menu                                    Fv(30D9ED01-38D2-418A-90D
                                                    5-C561750BF80F)/FvFile(C5
EFI DVD/CDROM                                       7AD6B7-0515-40A8-9D21-551
EFI Internal Shell                                  652854E37)
EFI Network
EFI Network 1
```

Step 7. Verify the MAC Address Update

1. Verify that the MAC address update was successful by using the UEFI shell command `ifconfig` to display the onboard LAN MAC address:

```
Shell> ifconfig -l
```

2. Press **Enter**. Verify the `ifconfig` command returns the correct MAC address for the Intel® DQ57TM motherboard.
3. After completing the BIOS update, power off the motherboard and set the motherboard back to normal mode (BIOS CFG Jumper, pin 1-2).

If you run into issues, troubleshooting tips can be found in Section 3 - Firmware Installation (hardware based)

Section 3:

Firmware Installation

This section explains in detail how to install an Intel® DQ57TM UEFI 2.3.1 Development Kit BIOS image on an *Intel® DQ57TM UEFI 2.3.1 Development Kit* (target PC) using an SPI programmer. This section includes screen shots and troubleshooting information. For a basic software-based BIOS upgrade, first try the Firmware Upgrade procedures found in Section 2.

Procedure Summary

Required tools for installation	**Estimated time (inexperienced user)**	**Estimated time (experienced user)**
Dediprog SPI software tool chain (software application) SPI flash programmer tool and test clip Intel®-provided firmware update utility FAT-formatted USB flash drive	20 to 30 minutes	5 minutes

These instructions assume use of the following hardware and software:

- To support the Dediprog software tool, you must have a host PC with Microsoft Windows XP* or another appropriate Windows* operating system.
- *Intel® DQ57TM UEFI 2.3.1 Development Kit* (target PC) assembled with components from the supported hardware component list. See Table 1-1, earlier in this guide.
- *Intel® DQ57TM UEFI 2.3.1 Development Kit* BIOS image(s) and user documentation.
- *Intel DQ57TM Firmware Update Utility*: `FirmwareUpdate.efi`
- Third-party Dediprog tool, consisting of the Dediprog SF100 flash programmer, test clip, and the Dediprog software utility.
- FAT-formatted USB flash device.

Use these general steps for installing the UEFI 2.3.1-compliant BIOS image on the target PC:

1. Make sure you have an assembled, functional firmware developer platform built with the required hardware components listed in Table 1-1 (earlier in this guide). This developer platform is the target PC.
2. Download the *Intel® DQ57TM UEFI 2.3.1 Development Kit* BIOS image(s), firmware update tool, and user documentation to a FAT-formatted flash drive.
3. Install the Dediprog utility and device drivers on the host PC.
4. Prepare the target PC for the BIOS update.
5. Create a backup copy of the motherboard's original BIOS image.
6. Erase the existing BIOS from the motherboard.
7. Write the *Intel® DQ57TM UEFI 2.3.1 Development Kit* BIOS image to the flash device on the motherboard.
8. Reassemble the target PC.
9. Boot to the UEFI shell.
10. Patch the new BIOS image with the MAC address for the target PC's motherboard.
11. Verify that the new BIOS functions correctly by installing and executing a sample *UEFI* application.

You are now ready to begin the installation procedure for the *UEFI 2.3.1 BIOS*.

Step 1. Ready the Intel® DQ57TM UEFI 2.3.1 Development Kit

This guide assumes you have assembled or purchased a target PC –*Intel® DQ57TM UEFI 2.3.1 Development Kit* – comprised of hardware components compliant with the supported hardware list (See Table 1-1, earlier in this guide). PC assembly is not covered in this document.

CAUTION: *Installing the BIOS image on an unsupported motherboard may render the motherboard unusable until it is reflashed with a backup copy of the motherboard's original BIOS. Use only supported components with the Intel® DQ57TM UEFI 2.3.1 Development Kit.*

CAUTION: *Handle the motherboard properly during installation to prevent damage to the board and its flash memory.*

Step 2. Download Files

Before beginning the BIOS update, you must download the *Intel® DQ57TM UEFI 2.3.1 Development Kit* from the Intel® Web site. The kit consists of the UEFI 2.3.1-compliant BIOS images, the Intel®-provided firmware update utility, and user documentation. Follow these steps to download the necessary files:

1. Go to the Intel® UEFI Driver and Application Tool Resources web page.
2. Click on the link to download the *Intel® UDK2010 Firmware Developer Platform Kit* and save the ZIP file to an appropriate destination, such as a directory on the host system's hard drive.
3. Unzip the download and verify that these individual files are present:
    a. `UDK2010_TM_B9_release.rom` – The *release* version of the BIOS, with debugging features disabled.
    b. `UDK2010_TM_B9_debug.rom` – The *debug* version of the BIOS, with debug output redirected to the serial port (**COM1**).
    c. `UDK2010_TM_B9_srcdbg.rom` – The *source level debug* version of the BIOS, enabling support for the Intel® UEFI Development Kit Debugger Tool using the serial port (**COM1**).
    d. Firmware update tool: `FirmwareUpdate.efi`
4. Download the *Intel® DQ57TM Desktop Board product guide* and save it to your local file system:
5. Copy the firmware update tool *`FirmwareUpdate.efi`* to a FAT-formatted USB flash drive.

Step 3. Install the Dediprog software utility and device drivers

Make sure you install the Dediprog software utility and all necessary device drivers for the flash programmer before trying to run the application.

1. On the host PC, install the Dediprog software utility.
2. On the host PC, install all necessary Dediprog device drivers.

Step 4. Verify functionality of the target PC

Follow these general steps to verify functionality of the target desktop PC:

1. Check all connections between components on the target PC.
2. Connect the target PC to an appropriate power source.
3. Power up the target PC.
4. During boot, in the initialization screen, press the appropriate key(s) to enter BIOS.
5. Navigate to at least two BIOS menus to verify that the system is working properly.
6. Exit BIOS.
7. Power down the system.

Step 5. Prepare the Target PC

Throughout this procedure, refer to Figure 3-1 for the location of the SPI flash device, the MAC address, the LEDs, and the BIOS CFG jumper.

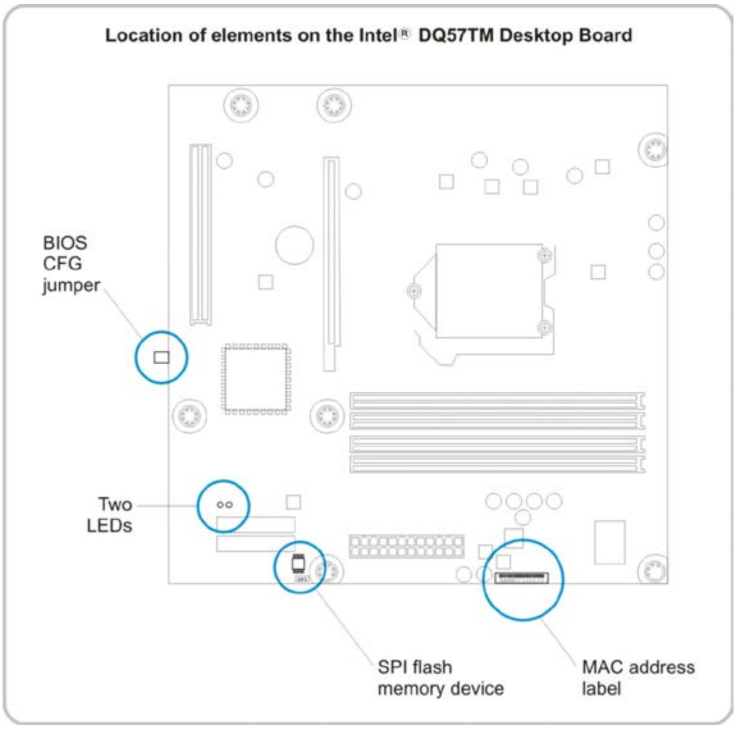


Figure 3-1. Location of BIOS CFG jumper, LEDs, SPI device, and MAC address.

Follow these steps:

> ***CAUTION:*** *To avoid damaging the motherboard and/or other components, AUX power to the machine must be OFF, and the power cord unplugged from AC power.*

CAUTION: *To avoid damaging the motherboard and/or other components, make sure you follow standard anti-static precautions, including the use of ground straps.*

CAUTION: *Overwriting the BIOS with a new image can be problematic while the management engine is active, because the management engine can create activity on the SPI bus.*

1. On the target PC, make sure power is off.
2. Disconnect the power cable from AC power.
3. There is a green standby power LED next to the SATA connector on the motherboard. Wait until this LED turns off. See Figure 3-2.

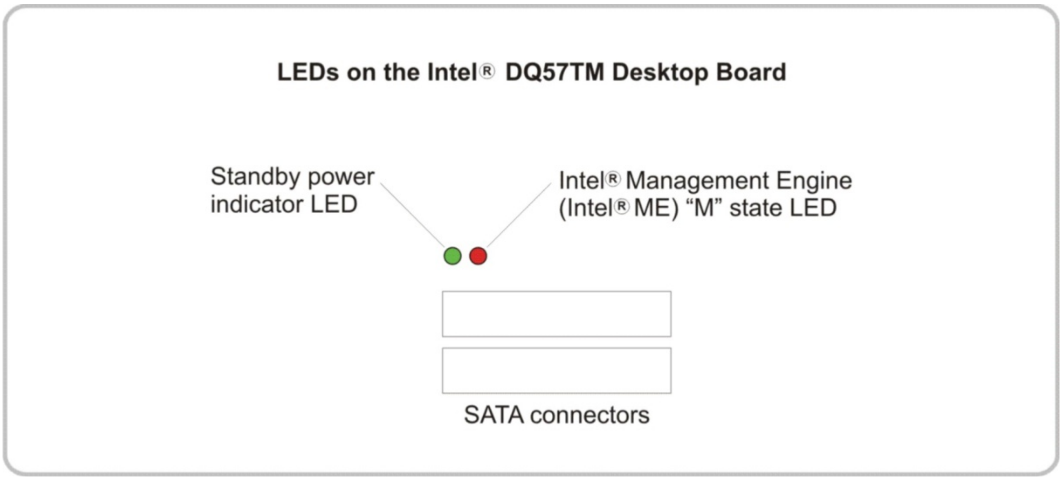

LEDs on the Intel® DQ57TM Desktop Board
Standby power indicator LED
Intel® Management Engine (Intel® ME) "M" state LED
SATA connectors

Figure 3-2. Location of the standby power indicator LED on the motherboard.

4. On the host PC, plug the USB end of the Dediprog cable into a USB port.

   Note: *Make sure the Dediprog flash programmer software utility and all necessary device drivers are installed before trying to program the BIOS. Refer to your third-party Dediprog product documentation for information about installing the software utility and device drivers.*

5. On the target PC, if the system has a SATA1 cable, remove the cable to make it easier to access the SPI.
6. Set motherboard in configuration mode (BIOS CFG Jumper, pin 2-3).
7. On the target PC, connect the Dediprog SF100 programmer by connecting the ISP-TC-8 alligator clip to the SPI flash device on the motherboard. Make sure pin 1 on the alligator clip is connected to

pin 1 on the flash device (see Figure 3-3). The white line on the SPI plug should line up with the white dot or arrow on the motherboard.

CAUTION: *Make sure the Dediprog tool is clipped correctly to the motherboard spring. If you clip the tool on backwards, the pinout will also be backwards and the tool will not function properly.*

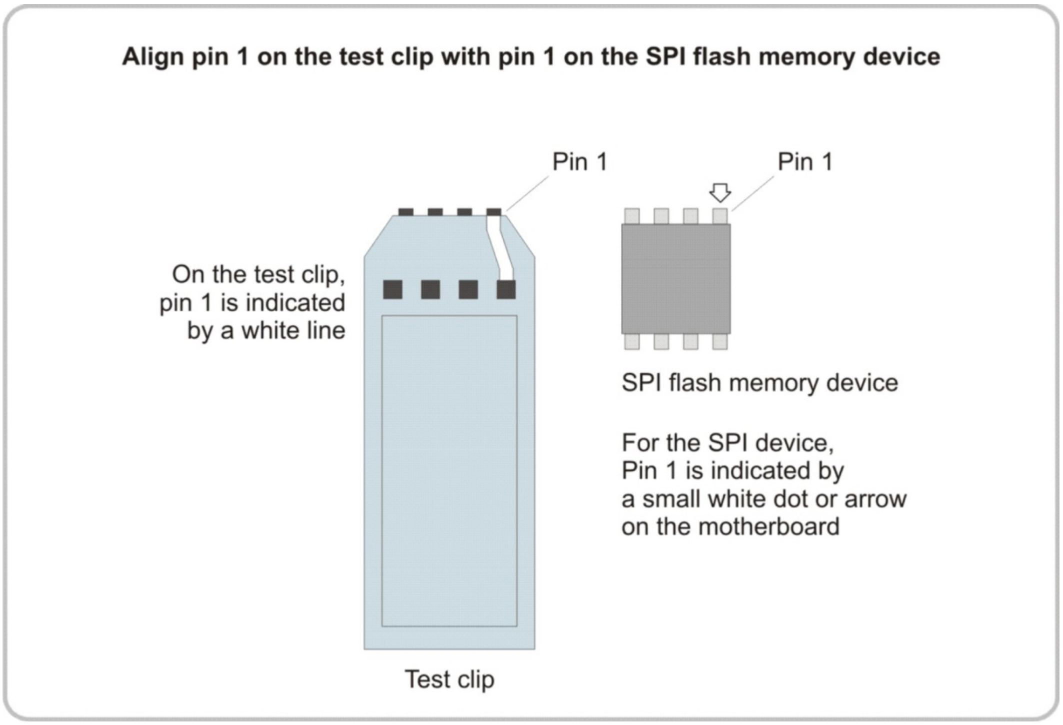

Align pin 1 on the test clip with pin 1 on the SPI flash memory device
Pin 1
Pin 1
On the test clip, pin 1 is indicated by a white line
SPI flash memory device
For the SPI device, Pin 1 is indicated by a small white dot or arrow on the motherboard
Test clip

Figure 3-3. Pin Alignment

8. Note the configuration of the BIOS CFG jumper.

   CAUTION: *Make sure the target PC is powered down and the power cable disconnected from AC power before moving the BIOS CFG jumper. Moving the jumper with the power on may result in unreliable computer operation.*

9. Grasp the tab on the BIOS CFG jumper, carefully remove the jumper, and set the jumper aside. For detailed information about removing the BIOS CFG jumper, refer to the Intel® DQ57TM motherboard product guide. Figure 3-1 shows the location of the BIOS CFG jumper.
10. Reconnect the power cable to the target PC, enabling AUX power.
11. Press and hold the power button on the target PC until the system powers up, then powers back off.

12. Locate the two LEDs near the SATA connector. The green standby power LED should now be lit, and the red Intel® ME status LED should be off.

If the green standby power LED is lit, you are ready to continue. If the red Intel® ME status LED is flashing, refer to troubleshooting, as described next.

Troubleshooting: Red LED is flashing

After you power the target PC back up, if red Intel® ME status LED flashes consistently (more than once) follow these steps:

1. Power down the target PC.
2. Disconnect the power cable from AC power.
3. Wait until the green standby power LED turns off.
4. Reconnect the power cable to AC power.
5. Press and hold the PC's power button until the system powers up, then powers back off.
6. Check the LEDs. The green standby power LED should now be lit, and the red Intel® ME status LED should be off.

Step 6. Record the MAC address

Make sure to note the MAC address of the target motherboard. You will need this information when installing the new BIOS. Follow these steps:

1. Look on the motherboard and locate the small sticker that lists the MAC address for the board. Figure 3-1 (earlier in this section) shows this location.
2. Write the MAC address on a piece of paper. Keep this paper available. You will need it later when writing and verifying the new BIOS.

   Note: *The MAC address is case insensitive.*

You are now ready to begin the backup process for the original BIOS.

Step 7. Backup the original BIOS

In this procedure, to help you recover from any errors, you will create a backup copy of the original motherboard BIOS.

Follow these steps to create a backup copy of the original motherboard BIOS:

CAUTION: *Do not perform any further steps until you have made a backup copy of the original motherboard BIOS. You might need to restore the original BIOS in the event of a catastrophic error.*

CAUTION: *To avoid damaging the motherboard and/or other components, make sure the PC is powered down at the start of this procedure, and the power cable is disconnected from AC power.*

Note: *The file types `.bin`, `.rom`, and `.fd` (firmware device) are all binary ROM images and are synonymous.*

1. On the target PC, do **not** power up the PC, reconnect AC power (AC power becomes available to the system). When AC power is applied, the green standby power LED next to the red SATA connector should light up. This indicates that power is applied to the board.
2. On the host system, to open the Dediprog application double-click the Dediprog icon on the desktop — see Figure 3-4. When the application opens, it should display either a screen of events or a Memory Type Ambiguity window. If you see the ambiguity window, there may be several SPI chips available from the manufacturer; however, only one chip from each manufacturer is supported by the Intel® DQ57TM UEFI 2.3.1 Development Kit.
   - If a list of events is displayed, the list should include detection of the connection between the host and target PCs (see Figure 3-5).
   - If a list of flash devices is displayed, select the Macronix* chip MX25L6445E or the Winbond* chip W25Q64BV (see Figure 3-6), as appropriate for the SPI device on your motherboard.

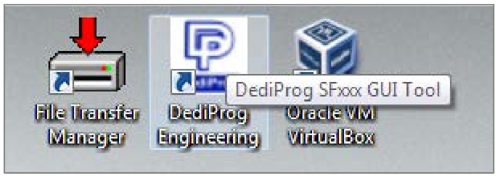
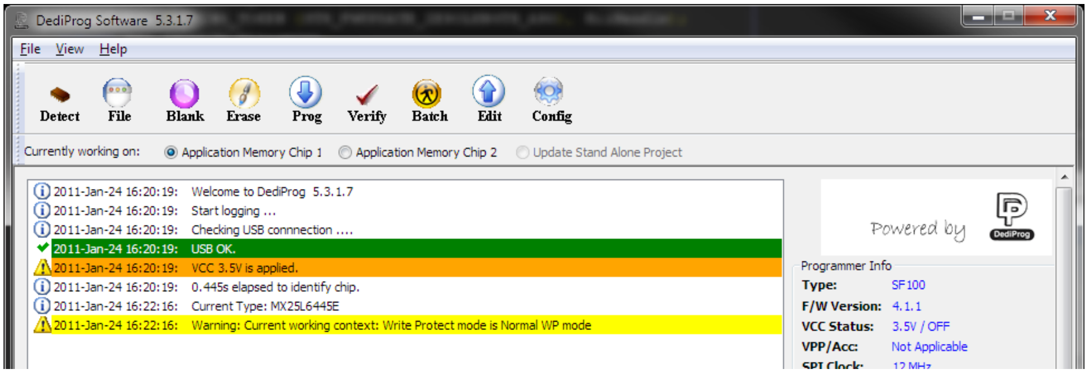


Figure 3-4. Launch the Dediprog application on the host system.



Currently working on: ● Application Memory Chip 1 ○ Application Memory Chip 2 ○ Update Stand Alone Project

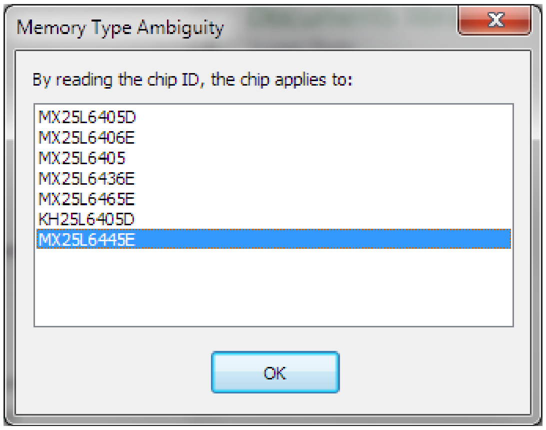
```
2011-Jan-24 16:20:19:   Welcome to DediProg  5.3.1.7
2011-Jan-24 16:20:19:   Start logging ...
2011-Jan-24 16:20:19:   Checking USB connection ....
2011-Jan-24 16:20:19:   USB OK.
2011-Jan-24 16:20:19:   VCC 3.5V is applied.
2011-Jan-24 16:20:19:   0.445s elapsed to identify chip.
2011-Jan-24 16:22:16:   Current Type: MX25L6445E
2011-Jan-24 16:22:16:   Warning: Current working context: Write Protect mode is Normal WP mode
```

Programmer Info

Type: SF100
F/W Version: 4.1.1
VCC Status: 3.5V / OFF
VPP/Acc: Not Applicable
SPI Clock: 12 MHz

Figure 3-5. Dediprog event log

The Dediprog application opens and identifies that it can communicate with the SPI flash chip on the target system. The event log should show both a green highlighted line and an orange highlighted line. The upper green line shows that the USB connection was successfully initiated between the host and target system. The orange line specifies the voltage being applied to the flash area in order to perform the read and write functions.



By reading the chip ID, the chip applies to:

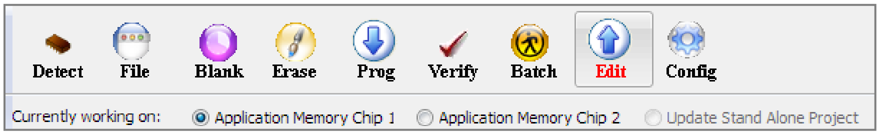
MX25L6405D
MX25L6406E
MX25L6405
MX25L6436E
MX25L6465E
KH25L6405D
MX25L6445E

Figure 3-6. Chips available from the manufacturer.

When the Dediprog application opens, it should display a list of available chips from the manufacturer.

3. Select the *chip* supported by your motherboard.

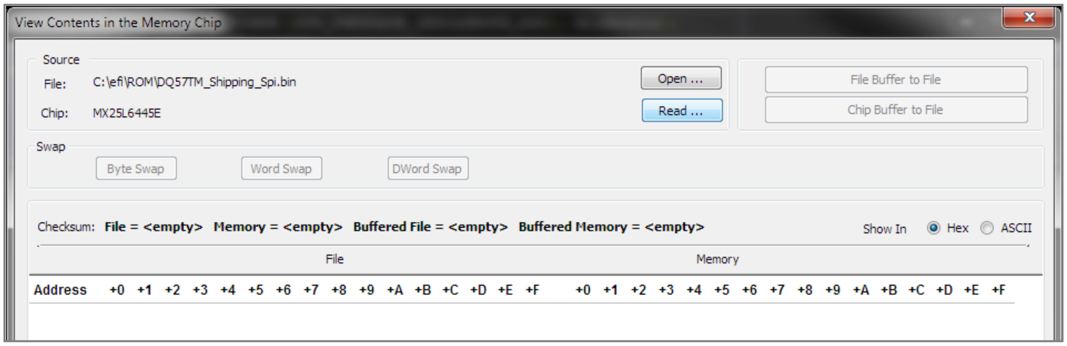
4. Click the **Edit** icon in the top menu bar (see Figure 3-7). The application then displays the View Contents window (see Figure 3-8).



Currently working on: ● Application Memory Chip 1 ○ Application Memory Chip 2 ○ Update Stand Alone Project

Figure 3-7. Icons List

5. Click **Edit**. When you click the Edit icon, the View Contents window is displayed (see Figure 3-8).
6. Click the **Read** button to read the contents of the flash image into memory. Dediprog immediately begins reading the contents of the flash image into a temporary memory location (see Figure 3-9). You will see a green progress bar indicating the status of the download. When the read operation is complete, Dediprog will display the contents of the file buffer (see Figure 3-10).


View Contents in the Memory Chip
Source
Open ...
Read ...
File Buffer to File
Chip Buffer to File
Swap
Byte Swap
Word Swap
DWord Swap
File
Memory
Address +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F

File: C:\efi\ROM\DQ57TM_Shipping_Spi.bin

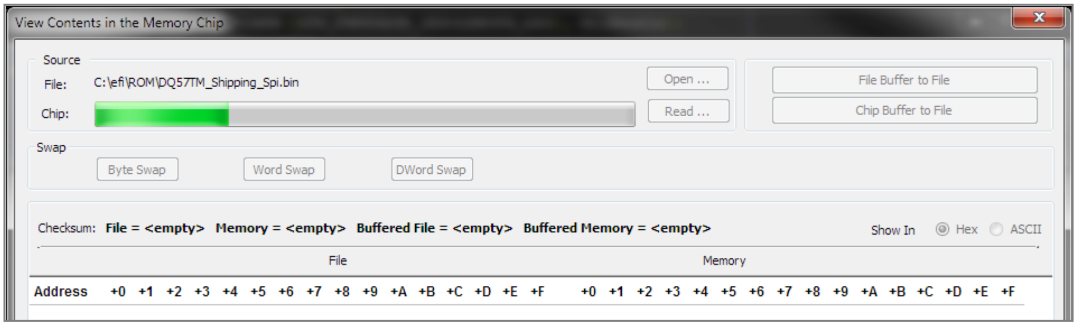
Chip: MX25L6445E

Checksum: **File = <empty> Memory = <empty> Buffered File = <empty> Buffered Memory = <empty>**

Show In ● Hex ○ ASCII

Figure 3-8. View Content in the Memory Chip display

7. Click **Read**. When you click the **Read** button, Dediprog reads the contents of the flash image into a temporary location.


View Contents in the Memory Chip
Source
Open ...
Read ...
File Buffer to File
Chip Buffer to File
Swap
Byte Swap
Word Swap
DWord Swap
File
Memory
Address +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F

File: C:\efi\ROM\DQ57TM_Shipping_Spi.bin

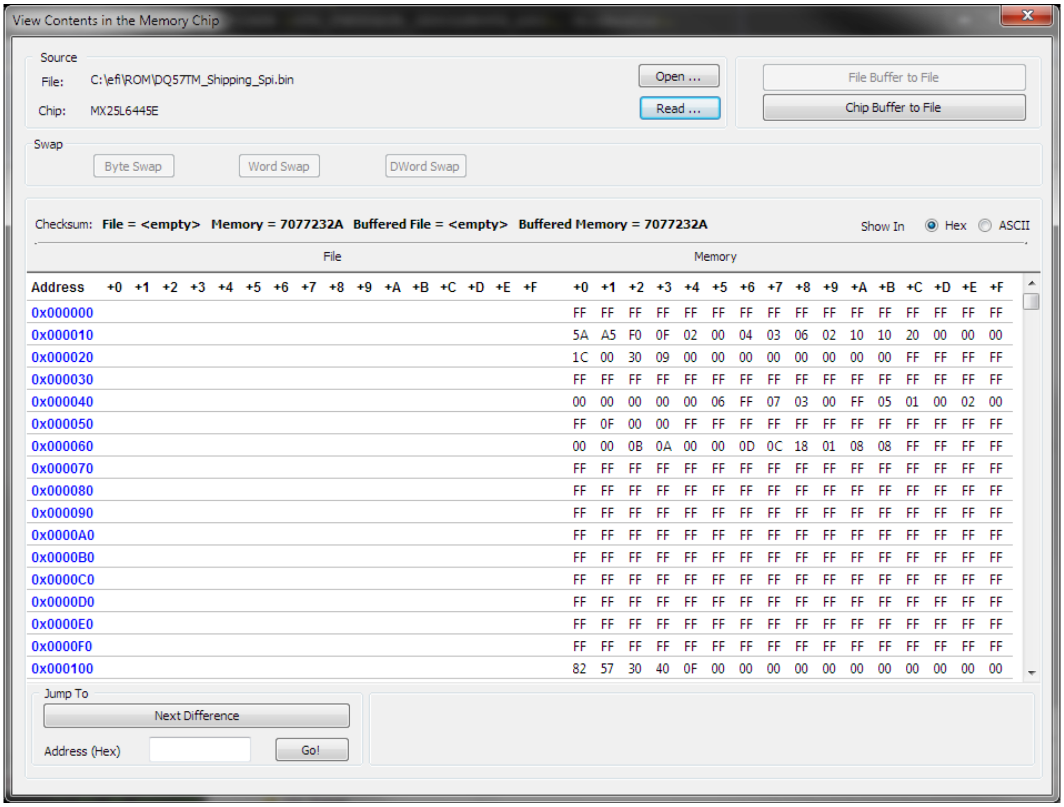
Chip:

Checksum: **File = <empty> Memory = <empty> Buffered File = <empty> Buffered Memory = <empty>**

Show In ● Hex ○ ASCII

Figure 3-9. Progress bar.

A green progress bar indicates the status of the download. The filename shown in the upper left area is simply the name of the previous file opened for programming, and does not reflect the current download.



File: C:\efi\ROM\DQ57TM_Shipping_Spi.bin

Chip: MX25L6445E

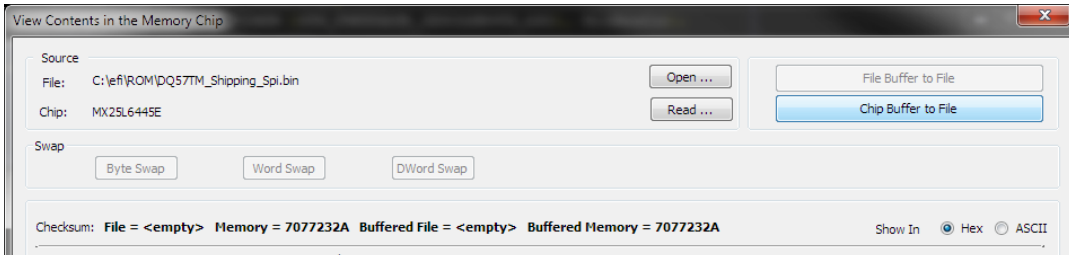
Checksum: **File = <empty> Memory = 7077232A Buffered File = <empty> Buffered Memory = 7077232A**

```
Address   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F
0x000000                                                    FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0x000010                                                    5A A5 F0 0F 02 00 04 03 06 02 10 10 20 00 00 00
0x000020                                                    1C 00 30 09 00 00 00 00 00 00 00 00 FF FF FF FF
0x000030                                                    FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0x000040                                                    00 00 00 00 00 06 FF 07 03 00 FF 05 01 00 02 00
0x000050                                                    FF 0F 00 00 FF FF FF FF FF FF FF FF FF FF FF FF
0x000060                                                    00 00 0B 0A 00 00 0D 0C 18 01 08 08 FF FF FF FF
0x000070                                                    FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0x000080                                                    FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0x000090                                                    FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0x0000A0                                                    FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0x0000B0                                                    FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0x0000C0                                                    FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0x0000D0                                                    FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0x0000E0                                                    FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0x0000F0                                                    FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0x000100                                                    82 57 30 40 0F 00 00 00 00 00 00 00 00 00 00 00
```

Figure 3-10. Read is completed.

When the read operation is complete, Dediprog displays the contents of the file buffer.

8. Select *Chip Buffer to File* (see Figure 3-11). This initiates a *save as* operation.



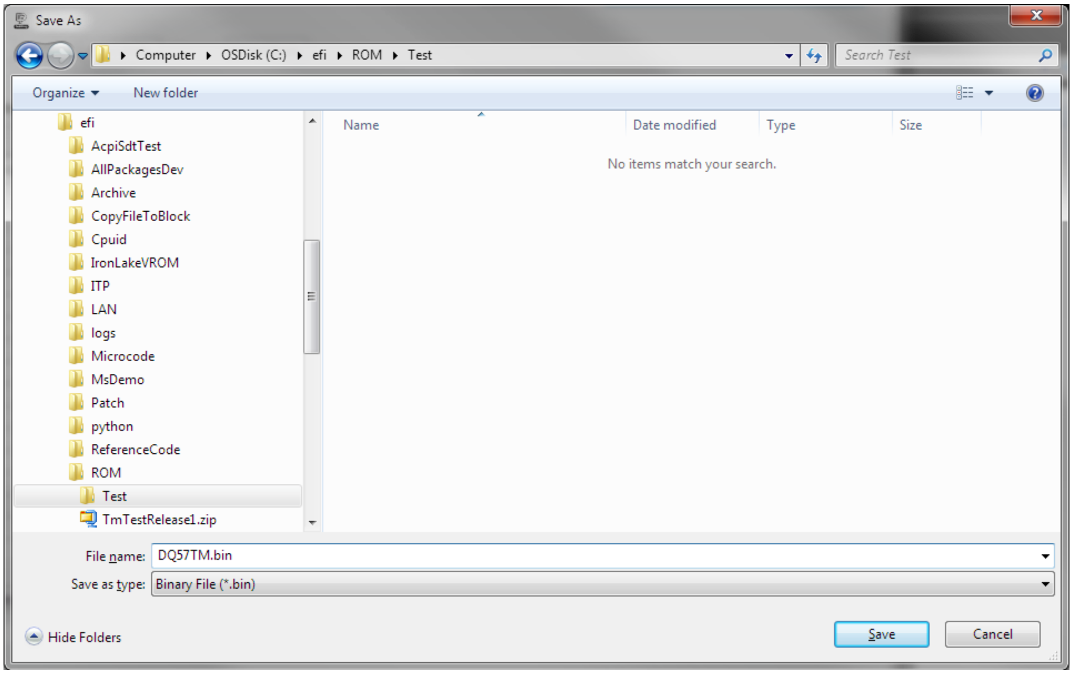
File: C:\efi\ROM\DQ57TM_Shipping_Spi.bin

Chip: MX25L6445E

Checksum: **File = <empty> Memory = 7077232A Buffered File = <empty> Buffered Memory = 7077232A**

Figure 3-11. Initiate a save operation by selecting Chip Buffer to File.

9. Enter a destination directory and filename into which to save the file on the hard drive (see Figure 3-12). In this guide, the example destination directory is c:/temp.



Computer ▸ OSDisk (C:) ▸ efi ▸ ROM ▸ Test

No items match your search.

File name: DQ57TM.bin

Save as type: Binary File (*.bin)

Figure 3-12. *Browse to a destination directory and filename, to which to save the new BIOS image.*

10. To save the file, click **Save**. The file is then saved, and the event log displays a line showing that the save is complete.
11. Close the window by clicking the **red X** in the upper right corner.

Now that you have a backup copy of the original BIOS, you are ready to erase the existing BIOS from the motherboard

Troubleshooting

There are a few typical problems you might encounter after connecting the SPI programmer cable and opening the Dediprog application.

Common errors are:

- **Cannot identify the target device.** This error is typically seen when:
  - *You have not connected the SPI programmer cable correctly on the motherboard*. Make sure you have a good connection to the SPI flash device, and make sure you have connected the cable correctly, with pin 1 on the clip connected to pin 1 on the SPI

device. The white line on the clip should line up with the small arrow or dot on the motherboard (see Figure 3-3, earlier in this section). If you connect the cable backwards, Dediprog will not be able to recognize the target device.

  - *There is activity on the SPI bus.* If you have correctly installed the SPI programmer cable but still cannot identify the flash area, there is probably activity on the SPI bus.

- **Unable to detect the Dediprog device.** This error is typically seen when:
  - *You forgot to connect the USB end of the SPI programmer cable to the host system.* Check your connections, and reseat the USB cable firmly.
  - *You have not connected the cable to an appropriate USB port.* Check your connections, make sure the cable is plugged into an appropriate USB port, and reseat the USB cable firmly.
  - *You have not installed device drivers for Dediprog device.* Make sure that you have installed the required device drivers before trying to run the Dediprog utility.
- **Red Intel® ME status LED is flashing.** This error is typically seen when there is activity on the SPI bus.
  1. Power down the target PC.
  2. Disconnect the power cable from AC power.
  3. Wait until the green LED turns off.
  4. Reconnect the power cable to AC power.
  5. Press and hold the PC's power button until the system powers up, then powers back off.
  6. Check the LEDs. The green LED should now be lit, and the red LED should be off.

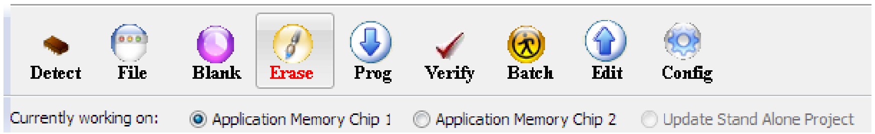
Step 8. Erase the existing BIOS

In this procedure, you will erase the original BIOS from the motherboard.

> ***CAUTION:*** *This procedure assumes you are experienced in erasing and restoring a BIOS. The Dediprog tool does not request confirmation before erasing a BIOS.*

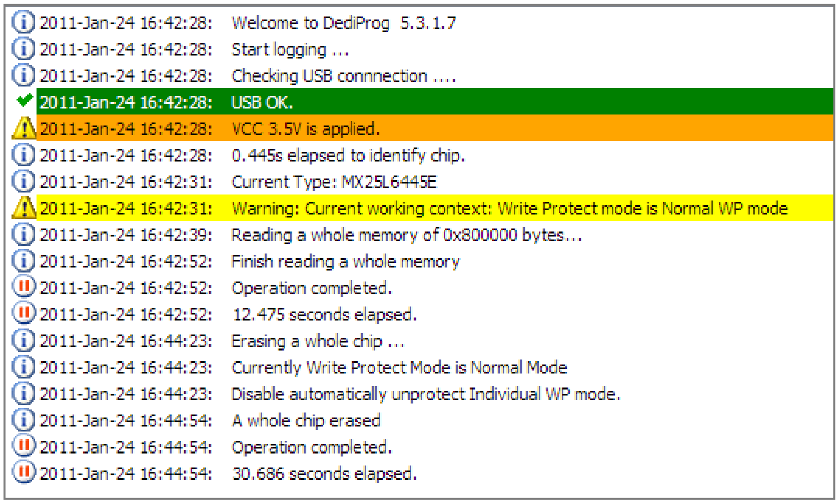
1. On the host system, click the **Erase** icon in the top menu bar (see Figure 3-13). You are not prompted to confirm the delete.

Dediprog immediately begins erasing the flash area. The green progress bar at the bottom of the screen shows the status of the erase. When the erase is complete, the event log will show an operation completed message. (see Figure 3-14).



Currently working on: ● Application Memory Chip 1 ○ Application Memory Chip 2 ○ Update Stand Alone Project

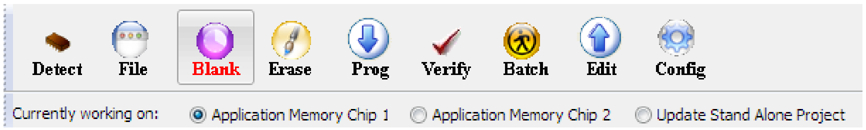
Figure 3-13. Click on the Erase icon to begin erasing the existing flash image.



```
2011-Jan-24 16:42:28:   Welcome to DediProg  5.3.1.7
2011-Jan-24 16:42:28:   Start logging ...
2011-Jan-24 16:42:28:   Checking USB connection ....
2011-Jan-24 16:42:28:   USB OK.
2011-Jan-24 16:42:28:   VCC 3.5V is applied.
2011-Jan-24 16:42:28:   0.445s elapsed to identify chip.
2011-Jan-24 16:42:31:   Current Type: MX25L6445E
2011-Jan-24 16:42:31:   Warning: Current working context: Write Protect mode is Normal WP mode
2011-Jan-24 16:42:39:   Reading a whole memory of 0x800000 bytes...
2011-Jan-24 16:42:52:   Finish reading a whole memory
2011-Jan-24 16:42:52:   Operation completed.
2011-Jan-24 16:42:52:   12.475 seconds elapsed.
2011-Jan-24 16:44:23:   Erasing a whole chip ...
2011-Jan-24 16:44:23:   Currently Write Protect Mode is Normal Mode
2011-Jan-24 16:44:23:   Disable automatically unprotect Individual WP mode.
2011-Jan-24 16:44:54:   A whole chip erased
2011-Jan-24 16:44:54:   Operation completed.
2011-Jan-24 16:44:54:   30.686 seconds elapsed.
```

Figure 3-14. Event log after an erase operation.

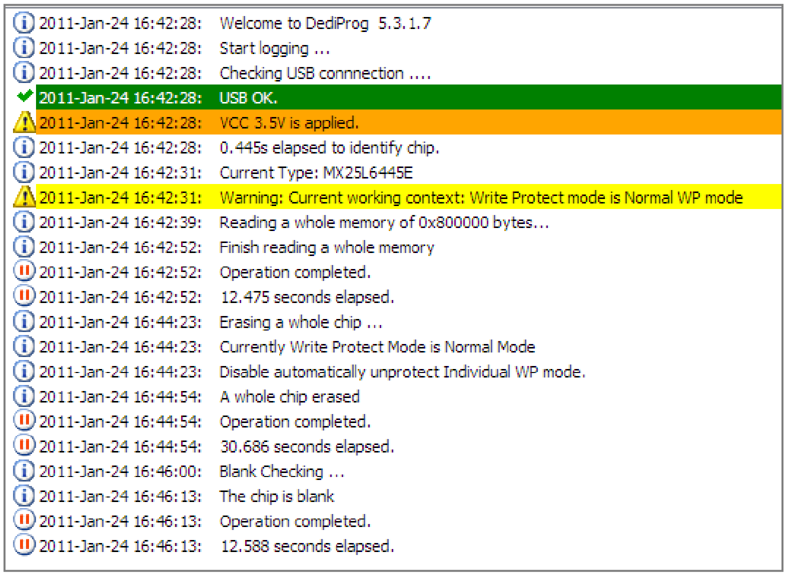
2. Click the Blank icon in the top menu bar (see Figure 3-15). The blank feature checks to make sure the flash area is erased (blank).



Currently working on: ● Application Memory Chip 1 ○ Application Memory Chip 2 ○ Update Stand Alone Project

Figure 3-15. Verify that the erase was successful.

3. Click the **Blank** icon to make sure the flash area has been erased.
4. Look at the Dediprog event log. There should be an entry listing ***A whole chip erased*** (see Figure 3-16).



```
2011-Jan-24 16:42:28:   Welcome to DediProg  5.3.1.7
2011-Jan-24 16:42:28:   Start logging ...
2011-Jan-24 16:42:28:   Checking USB connnection ....
2011-Jan-24 16:42:28:   USB OK.
2011-Jan-24 16:42:28:   VCC 3.5V is applied.
2011-Jan-24 16:42:28:   0.445s elapsed to identify chip.
2011-Jan-24 16:42:31:   Current Type: MX25L6445E
2011-Jan-24 16:42:31:   Warning: Current working context: Write Protect mode is Normal WP mode
2011-Jan-24 16:42:39:   Reading a whole memory of 0x800000 bytes...
2011-Jan-24 16:42:52:   Finish reading a whole memory
2011-Jan-24 16:42:52:   Operation completed.
2011-Jan-24 16:42:52:   12.475 seconds elapsed.
2011-Jan-24 16:44:23:   Erasing a whole chip ...
2011-Jan-24 16:44:23:   Currently Write Protect Mode is Normal Mode
2011-Jan-24 16:44:23:   Disable automatically unprotect Individual WP mode.
2011-Jan-24 16:44:54:   A whole chip erased
2011-Jan-24 16:44:54:   Operation completed.
2011-Jan-24 16:44:54:   30.686 seconds elapsed.
2011-Jan-24 16:46:00:   Blank Checking ...
2011-Jan-24 16:46:13:   The chip is blank
2011-Jan-24 16:46:13:   Operation completed.
2011-Jan-24 16:46:13:   12.588 seconds elapsed.
```

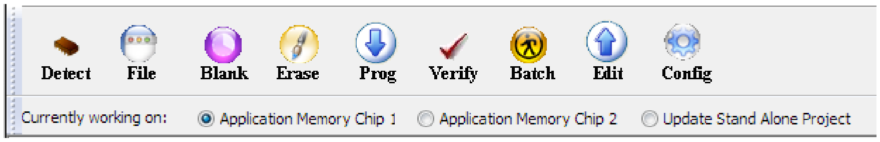
Figure 3-16. The event log should display a message verifying that the flash device is blank.

Now that you have verified that the flash device is erased, you are ready to write the new BIOS image, as described in the next procedure.

Step 9. Write the new BIOS image

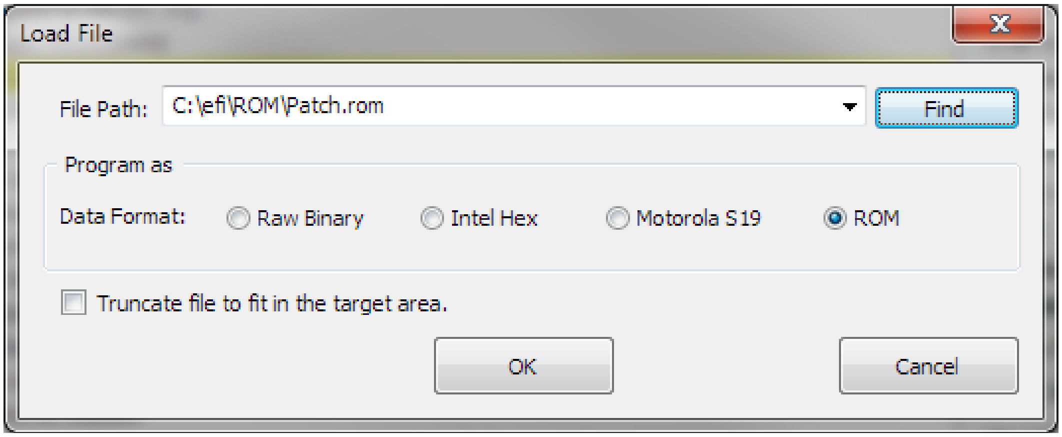
In this procedure, you will write the *Intel® UDK 2010 Firmware Developer* BIOS image to the Intel® DQ57TM Desktop Board, using the Dediprog SPI flash programmer. Follow these steps:

1. On the host system, in the Dediprog application, click the **File** icon in the top menu bar (see Figure 3-17). A dialog box will open (see Figure 3-18).



Currently working on: ● Application Memory Chip 1 ○ Application Memory Chip 2 ○ Update Stand Alone Project

Figure 3-17. Click the File icon.



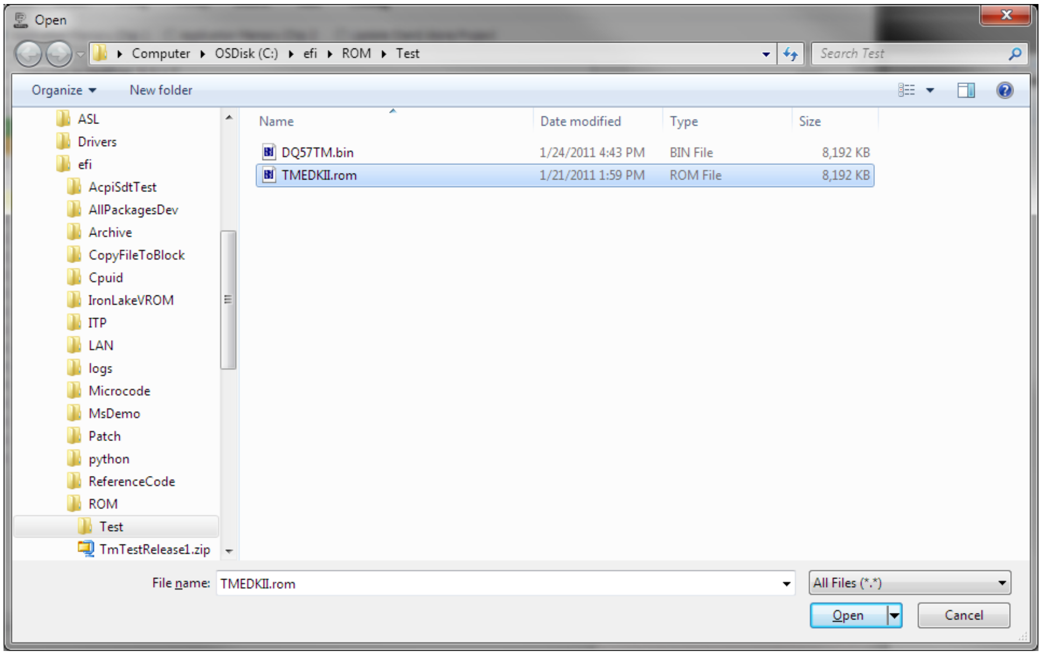
File Path: C:\efi\ROM\Patch.rom

Data Format: ○ Raw Binary ○ Intel Hex ○ Motorola S19 ● ROM

☐ Truncate file to fit in the target area.

Figure 3-18. Load File dialog box. (In this example, the last file loaded was `patch.rom`*.)*

2. In the dialog box, click **Find**.
3. Browse to the location where you saved the extracted UEFI 2.3.1 BIOS images, and select the appropriate BIOS image file to program. This is the BIOS image you downloaded from the Intel® Web site. Figure 3-19 shows an example pathname. Figure 3-20 shows the event log after the file is loaded into the memory buffer.



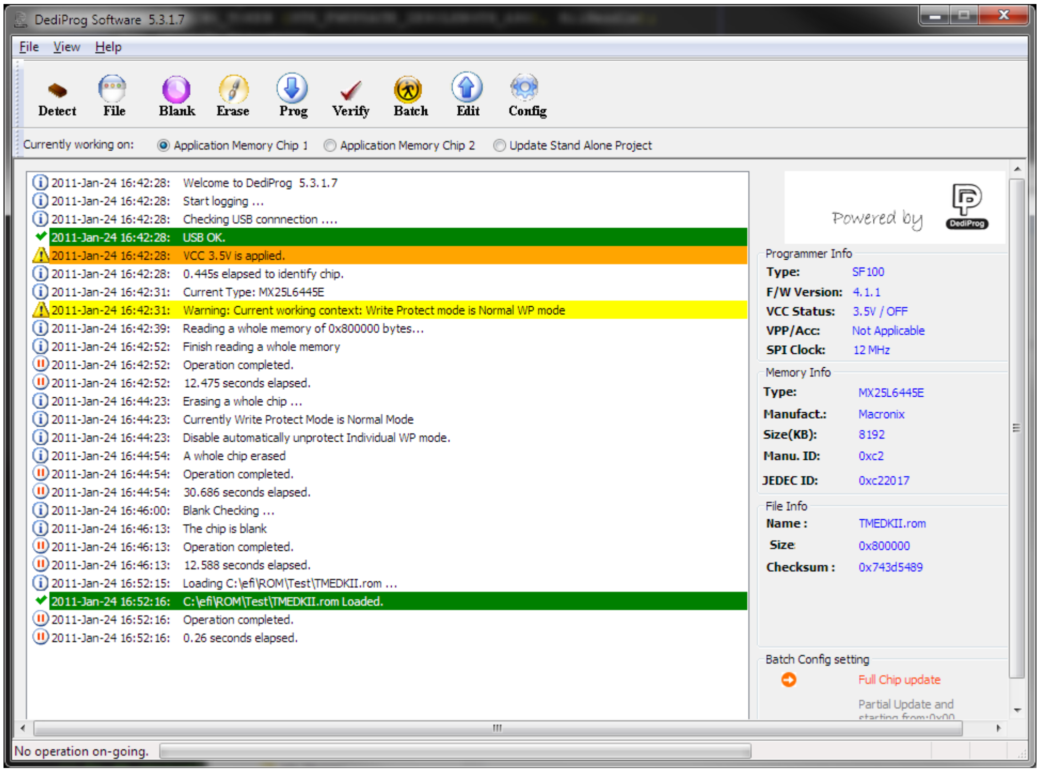
Name	Date modified	Type	Size
DQ57TM.bin	1/24/2011 4:43 PM	BIN File	8,192 KB
TMEDKII.rom	1/21/2011 1:59 PM	ROM File	8,192 KB

File name: TMEDKII.rom

All Files (*.*)

Figure 3-19. Select the file to load.

When you click **Open**, the Load File dialog box is displayed again.



Currently working on: ● Application Memory Chip 1 ○ Application Memory Chip 2 ○ Update Stand Alone Project

```
2011-Jan-24 16:42:28:   Welcome to DediProg  5.3.1.7
2011-Jan-24 16:42:28:   Start logging ...
2011-Jan-24 16:42:28:   Checking USB connnection ....
2011-Jan-24 16:42:28:   USB OK.
2011-Jan-24 16:42:28:   VCC 3.5V is applied.
2011-Jan-24 16:42:28:   0.445s elapsed to identify chip.
2011-Jan-24 16:42:31:   Current Type: MX25L6445E
2011-Jan-24 16:42:31:   Warning: Current working context: Write Protect mode is Normal WP mode
2011-Jan-24 16:42:39:   Reading a whole memory of 0x800000 bytes...
2011-Jan-24 16:42:52:   Finish reading a whole memory
2011-Jan-24 16:42:52:   Operation completed.
2011-Jan-24 16:42:52:   12.475 seconds elapsed.
2011-Jan-24 16:44:23:   Erasing a whole chip ...
2011-Jan-24 16:44:23:   Currently Write Protect Mode is Normal Mode
2011-Jan-24 16:44:23:   Disable automatically unprotect Individual WP mode.
2011-Jan-24 16:44:54:   A whole chip erased
2011-Jan-24 16:44:54:   Operation completed.
2011-Jan-24 16:44:54:   30.686 seconds elapsed.
2011-Jan-24 16:46:00:   Blank Checking ...
2011-Jan-24 16:46:13:   The chip is blank
2011-Jan-24 16:46:13:   Operation completed.
2011-Jan-24 16:46:13:   12.588 seconds elapsed.
2011-Jan-24 16:52:15:   Loading C:\efi\ROM\Test\TMEDKII.rom ...
2011-Jan-24 16:52:16:   C:\efi\ROM\Test\TMEDKII.rom Loaded.
2011-Jan-24 16:52:16:   Operation completed.
2011-Jan-24 16:52:16:   0.26 seconds elapsed.
```

Programmer Info

Type: SF100
F/W Version: 4.1.1
VCC Status: 3.5V / OFF
VPP/Acc: Not Applicable
SPI Clock: 12 MHz

Memory Info

Type: MX25L6445E
Manufact.: Macronix
Size(KB): 8192
Manu. ID: 0xc2
JEDEC ID: 0xc22017

File Info

Name : TMEDKII.rom
Size 0x800000
Checksum : 0x743d5489

Batch Config setting

Full Chip update
Partial Update and starting from:0x00

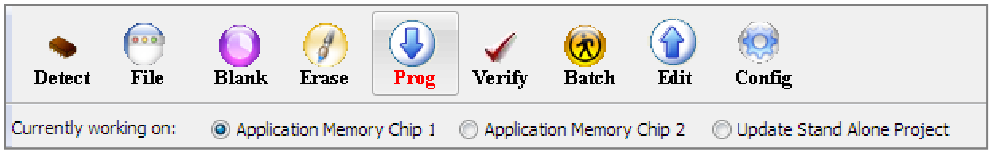
No operation on-going.

Figure 3-20. *Event log after the file is loaded into the buffer.*

The event log shows that the BIOS image has been loaded into the memory buffer and is ready for the program operation.

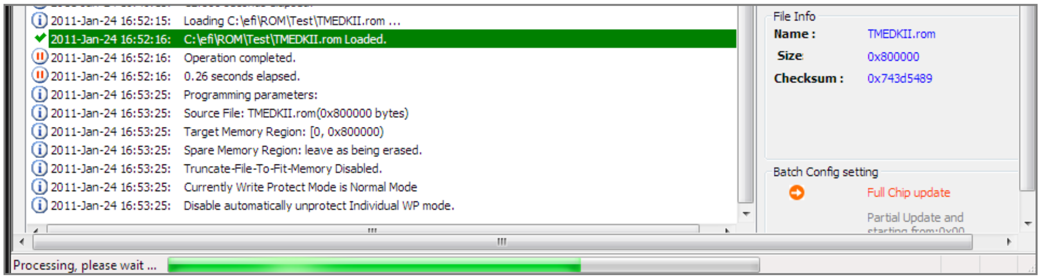
4. Uncheck the box for **Truncate file to fit in the target area**.
5. Now click the **Prog** (program) icon in the top menu bar to write the specified file to the Intel® DQ57TM motherboard (see Figure 3-21). Dediprog immediately begins writing to the flash area the file currently loaded into memory. Dediprog does not request confirmation for this write function.

   The green progress bar (see Figure 3-22) indicates the status of the write operation. When the write is complete, the Dediprog event log will indicate that the firmware image has been written to the motherboard (see Figure 3-23).



Currently working on: ● Application Memory Chip 1 ○ Application Memory Chip 2 ○ Update Stand Alone Project

Figure 3-21. *Click Prog to begin the write process.*



```
2011-Jan-24 16:52:15:   Loading C:\efi\ROM\Test\TMEDKII.rom ...
2011-Jan-24 16:52:16:   C:\efi\ROM\Test\TMEDKII.rom Loaded.
2011-Jan-24 16:52:16:   Operation completed.
2011-Jan-24 16:52:16:   0.26 seconds elapsed.
2011-Jan-24 16:53:25:   Programming parameters:
2011-Jan-24 16:53:25:   Source File: TMEDKII.rom(0x800000 bytes)
2011-Jan-24 16:53:25:   Target Memory Region: [0, 0x800000)
2011-Jan-24 16:53:25:   Spare Memory Region: leave as being erased.
2011-Jan-24 16:53:25:   Truncate-File-To-Fit-Memory Disabled.
2011-Jan-24 16:53:25:   Currently Write Protect Mode is Normal Mode
2011-Jan-24 16:53:25:   Disable automatically unprotect Individual WP mode.
```

File Info

Name : TMEDKII.rom
Size 0x800000
Checksum : 0x743d5489

Batch Config setting

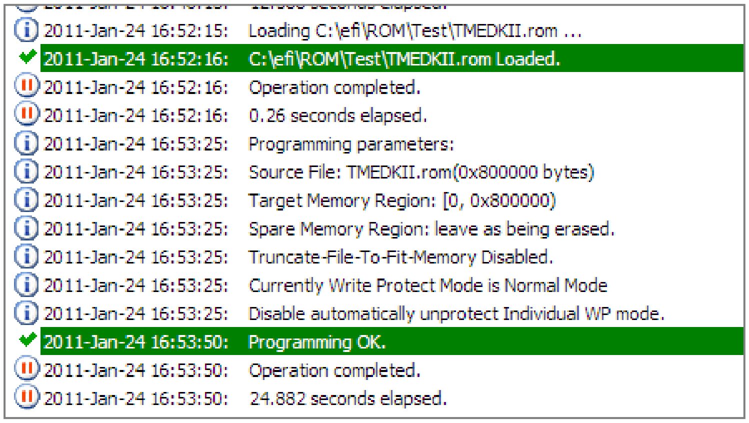
Full Chip update

Partial Update and

Processing, please wait ...

Figure 3-22. Progress of the write.

The green progress bar at the bottom of the screen indicates the status of the write operation.

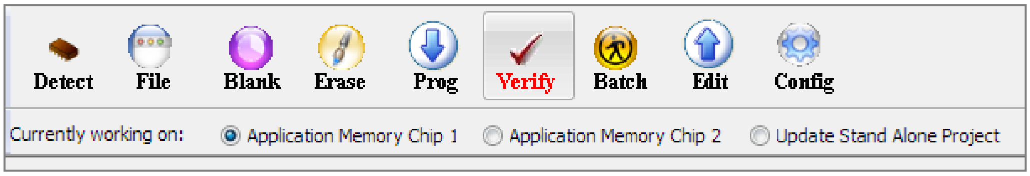


```
2011-Jan-24 16:52:15:   Loading C:\efi\ROM\Test\TMEDKII.rom ...
2011-Jan-24 16:52:16:   C:\efi\ROM\Test\TMEDKII.rom Loaded.
2011-Jan-24 16:52:16:   Operation completed.
2011-Jan-24 16:52:16:   0.26 seconds elapsed.
2011-Jan-24 16:53:25:   Programming parameters:
2011-Jan-24 16:53:25:   Source File: TMEDKII.rom(0x800000 bytes)
2011-Jan-24 16:53:25:   Target Memory Region: [0, 0x800000)
2011-Jan-24 16:53:25:   Spare Memory Region: leave as being erased.
2011-Jan-24 16:53:25:   Truncate-File-To-Fit-Memory Disabled.
2011-Jan-24 16:53:25:   Currently Write Protect Mode is Normal Mode
2011-Jan-24 16:53:25:   Disable automatically unprotect Individual WP mode.
2011-Jan-24 16:53:50:   Programming OK.
2011-Jan-24 16:53:50:   Operation completed.
2011-Jan-24 16:53:50:   24.882 seconds elapsed.
```

Figure 3-23. Event log after the write.

The Dediprog event log indicates that the firmware image has been written to the board.

6. Click the **Verify** icon in the top menu bar (see Figure 3-24). Dediprog then compares the previously saved file to the file written into flash memory.

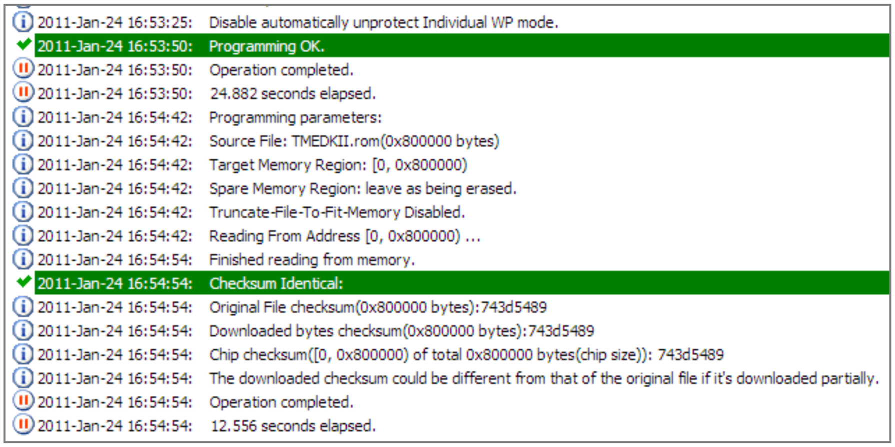


Currently working on: ● Application Memory Chip 1 ○ Application Memory Chip 2 ○ Update Stand Alone Project

Figure 3-24. Click Verify.

Verify the write via a comparison of the file written to flash memory versus the file you loaded into the buffer.

7. Check the event log. If the BIOS image was correctly written to the motherboard, the event log should state ***Checksum identical*** (see Figure 3-25).



```
2011-Jan-24 16:53:25:   Disable automatically unprotect Individual WP mode.
2011-Jan-24 16:53:50:   Programming OK.
2011-Jan-24 16:53:50:   Operation completed.
2011-Jan-24 16:53:50:   24.882 seconds elapsed.
2011-Jan-24 16:54:42:   Programming parameters:
2011-Jan-24 16:54:42:   Source File: TMEDKII.rom(0x800000 bytes)
2011-Jan-24 16:54:42:   Target Memory Region: [0, 0x800000)
2011-Jan-24 16:54:42:   Spare Memory Region: leave as being erased.
2011-Jan-24 16:54:42:   Truncate-File-To-Fit-Memory Disabled.
2011-Jan-24 16:54:42:   Reading From Address [0, 0x800000) ...
2011-Jan-24 16:54:54:   Finished reading from memory.
2011-Jan-24 16:54:54:   Checksum Identical:
2011-Jan-24 16:54:54:   Original File checksum(0x800000 bytes):743d5489
2011-Jan-24 16:54:54:   Downloaded bytes checksum(0x800000 bytes):743d5489
2011-Jan-24 16:54:54:   Chip checksum([0, 0x800000) of total 0x800000 bytes(chip size)): 743d5489
2011-Jan-24 16:54:54:   The downloaded checksum could be different from that of the original file if it's downloaded partially.
2011-Jan-24 16:54:54:   Operation completed.
2011-Jan-24 16:54:54:   12.556 seconds elapsed.
```

Figure 3-25. Event log after the write.

The Dediprog event log indicates that the firmware image has been written to the board.

You are now ready to remove the Dediprog cable as described in the next procedure.

Step 10. Exit Dediprog and reassemble the target PC

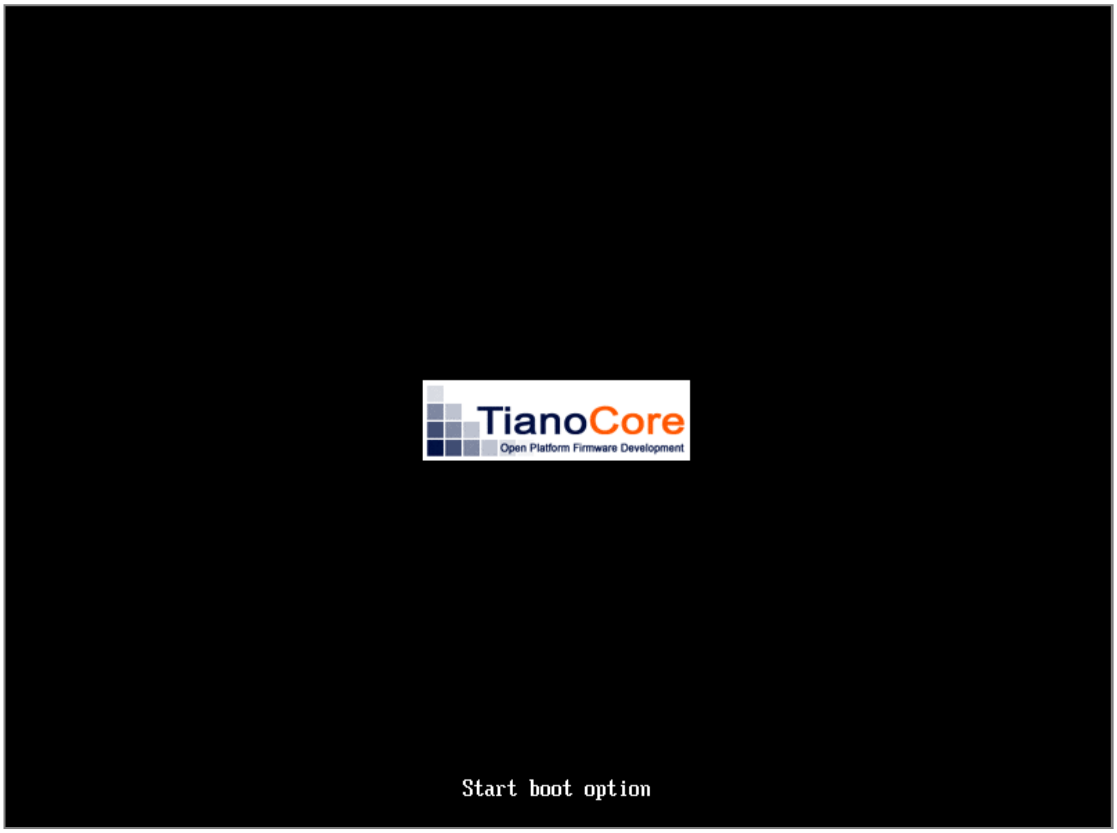
Once you confirm that the new BIOS flashed, you can disconnect the Dediprog SPI programmer from the motherboard. Follow these steps:

> ***CAUTION:*** *To avoid damaging the motherboard and/or other components, make sure the PC is powered down at the start of this procedure, and the power cable is disconnected from AC power.*

> ***CAUTION:*** *To avoid damaging the motherboard and/or other components, make sure you follow standard anti-static precautions, including the use of ground straps.*

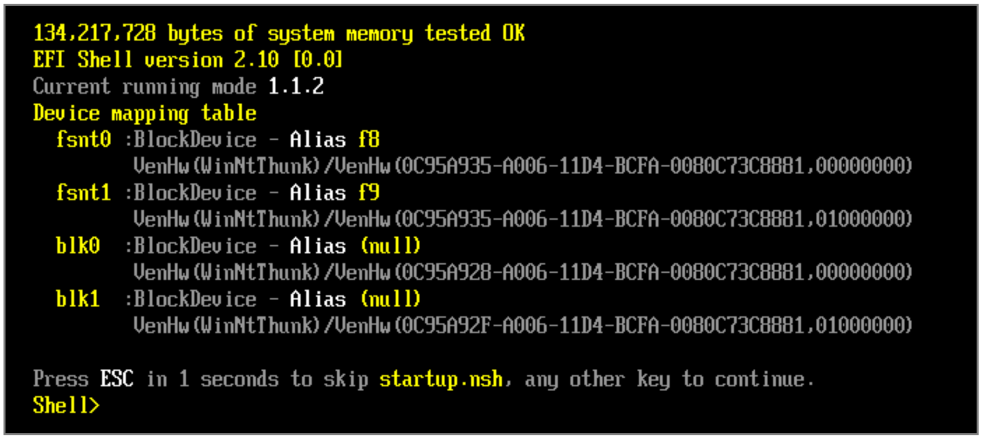
1. On the target PC, make sure power to the PC is powered OFF, and disconnect the power cable from AC power. Wait for the green standby power LED (near the SATA connector) to turn off.
2. On the host system, in the Dediprog application, click **Exit**.

3. On the host system, unplug the Dediprog programmer from the USB socket.
4. On the target system, unclip the Dediprog programmer from the SPI flash device.
5. If you disconnected the SATA1 cable earlier, reconnect the cable.
6. Replace the BIOS CFG jumper, as described in your motherboard's product documentation. Make sure the jumper is in its original orientation.
7. Reconnect the target PC to AC power.
8. Power up the target PC to verify that the new BIOS is installed correctly. The TianoCore* logo should be displayed (see Figure 3-26), and the system should automatically boot to the UEFI shell (see Figure 3-27).



```
Start boot option
```

Figure 3-26. Tianocore.org splash screen.



```
134,217,728 bytes of system memory tested OK
EFI Shell version 2.10 [0.0]
Current running mode 1.1.2
Device mapping table
  fsnt0 :BlockDevice - Alias f8
         VenHw(WinNtThunk)/VenHw(0C95A935-A006-11D4-BCFA-0080C73C8881,00000000)
  fsnt1 :BlockDevice - Alias f9
         VenHw(WinNtThunk)/VenHw(0C95A935-A006-11D4-BCFA-0080C73C8881,01000000)
  blk0  :BlockDevice - Alias (null)
         VenHw(WinNtThunk)/VenHw(0C95A928-A006-11D4-BCFA-0080C73C8881,00000000)
  blk1  :BlockDevice - Alias (null)
         VenHw(WinNtThunk)/VenHw(0C95A92F-A006-11D4-BCFA-0080C73C8881,01000000)

Press ESC in 1 seconds to skip startup.nsh, any other key to continue.
Shell>
```

Figure 3-27. The UEFI shell displays its top-level screen.

You are now ready to update the BIOS with the MAC address for your target PC.

Troubleshooting: UEFI shell does not automatically load

The UEFI should automatically come up when you power up the system, unless you have another USB device connected to the target PC. Make sure you disconnect any USB flash drives on the target PC before booting to the UEFI shell.

Step 11. Patch firmware image with MAC address for motherboard

This procedure shows how to use UEFI shell and the firmware update tool to update the MAC address to the correct value on the target PC motherboard.

1. Make sure you have the firmware update tool copied onto a FAT-formatted USB flash drive.

   Note: *The Intel®-supplied firmware update tool does not need to be installed on the target PC. This tool works from the USB drive. It is assumed that you have only one USB drive on the target PC (software-development platform).*

2. Connect the FAT-formatted USB flash drive to the USB port on the target PC.

3. Boot the system to the UEFI shell. If the system is already at the UEFI shell, reboot the system to make sure the USB flash drive is correctly mapped.
4. Use the UEFI shell **map** command to identify file system associated with the attached USB flash drive. This example uses `fs1` as the file system:

   ```
   Shell> map fs*:
   ```

5. Press **Enter**.
6. The system displays the list of file systems, including those associated with USB drives (see Figure 3-28). Look for the entry listed as Removable HardDisk that includes `USB` in the device path. In this example procedure, the entry is `fs1`.
7. Select the file system associated with the USB flash drive by typing the following command (see Figure 3-29). In this example, type `fs1`.

   ```
   Shell> fs1:
   ```

8. Press **Enter**
9. Run the Intel® firmware update tool *`FirmwareUpdate.efi`* to update the MAC address for the integrated LAN device on the target PC. Use the **-m** option to specify the MAC address for your the target PC. In the following example, replace the string *`001122AABBCC`* with the MAC address you copied earlier from the label on your motherboard. See Figure 3-30.

   ```
   fs1:\> firmwareupdate -m 001122AABBCC
   ```

   CAUTION: *Make sure you enter the correct MAC address in the command line. If you enter the wrong MAC address, the BIOS image will not work on the target PC.*

   Note: *The MAC address is case insensitive.*

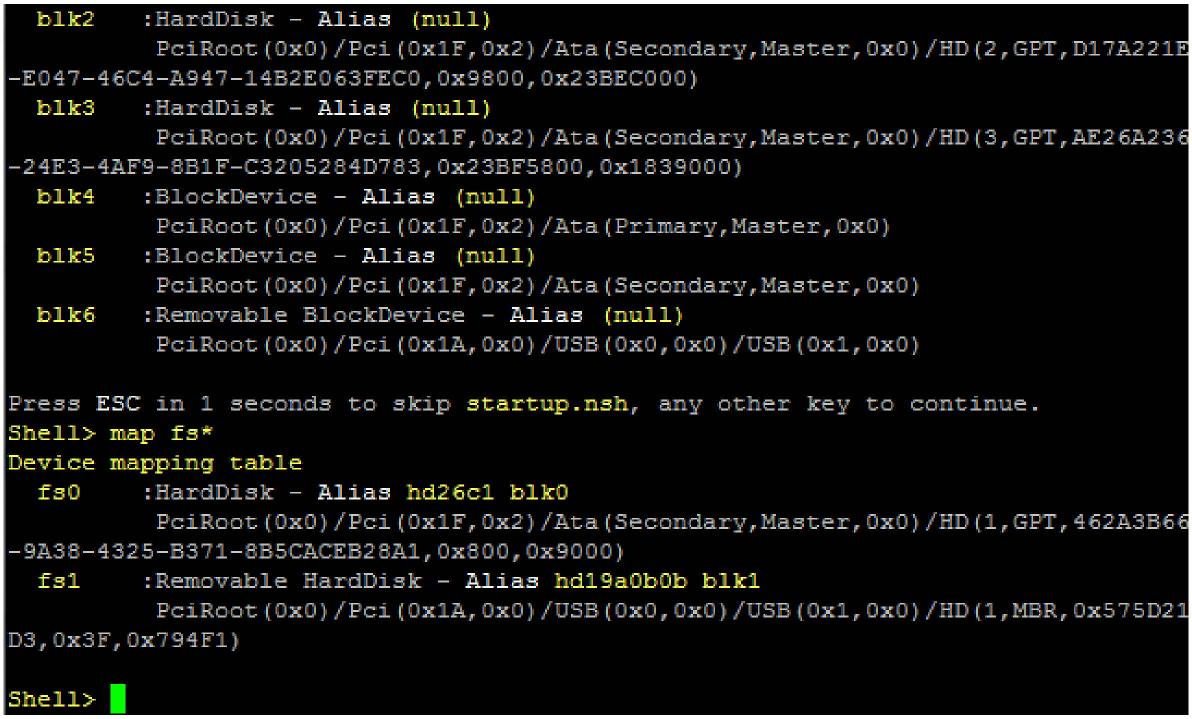
10. Press **Enter**. The system automatically reboots when the update is complete.
11. Boot to the UEFI shell again.

12. Verify that the update was successful by using the UEFI shell command **ifconfig** with the -l (lowercase L) option. Enter the following command to display the MAC address for the integrated LAN device (see Figure 3-31):

    ```
    fs1:\> ifconfig -l
    ```

13. Press **Enter**. Because you just updated the MAC address, the **ifconfig** command should return the MAC address for your Intel® DQ57TM Desktop Board.

14. Verify that the MAC address is correct for your motherboard.

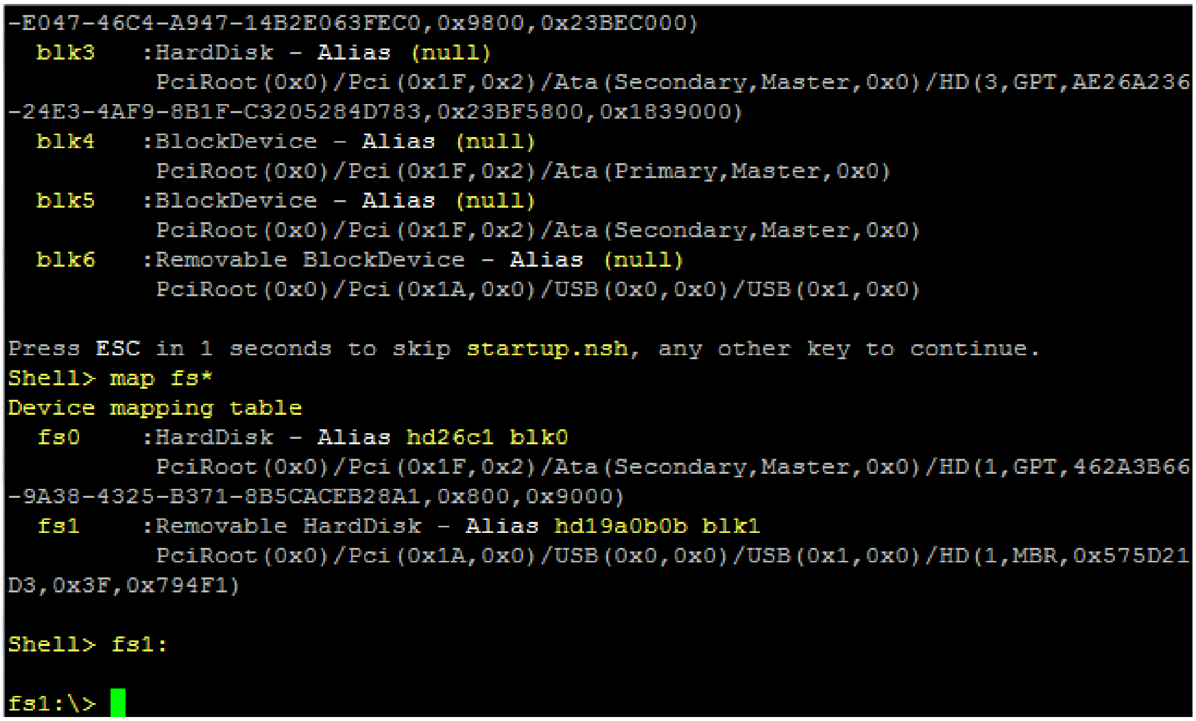


```
  blk2    :HardDisk - Alias (null)
           PciRoot(0x0)/Pci(0x1F,0x2)/Ata(Secondary,Master,0x0)/HD(2,GPT,D17A221E
-E047-46C4-A947-14B2E063FEC0,0x9800,0x23BEC000)
  blk3    :HardDisk - Alias (null)
           PciRoot(0x0)/Pci(0x1F,0x2)/Ata(Secondary,Master,0x0)/HD(3,GPT,AE26A236
-24E3-4AF9-8B1F-C3205284D783,0x23BF5800,0x1839000)
  blk4    :BlockDevice - Alias (null)
           PciRoot(0x0)/Pci(0x1F,0x2)/Ata(Primary,Master,0x0)
  blk5    :BlockDevice - Alias (null)
           PciRoot(0x0)/Pci(0x1F,0x2)/Ata(Secondary,Master,0x0)
  blk6    :Removable BlockDevice - Alias (null)
           PciRoot(0x0)/Pci(0x1A,0x0)/USB(0x0,0x0)/USB(0x1,0x0)

Press ESC in 1 seconds to skip startup.nsh, any other key to continue.
Shell> map fs*
Device mapping table
  fs0     :HardDisk - Alias hd26c1 blk0
           PciRoot(0x0)/Pci(0x1F,0x2)/Ata(Secondary,Master,0x0)/HD(1,GPT,462A3B66
-9A38-4325-B371-8B5CACEB28A1,0x800,0x9000)
  fs1     :Removable HardDisk - Alias hd19a0b0b blk1
           PciRoot(0x0)/Pci(0x1A,0x0)/USB(0x0,0x0)/USB(0x1,0x0)/HD(1,MBR,0x575D21
D3,0x3F,0x794F1)

Shell> 
```

Figure 3-28. Executing the map command in the UEFI shell.



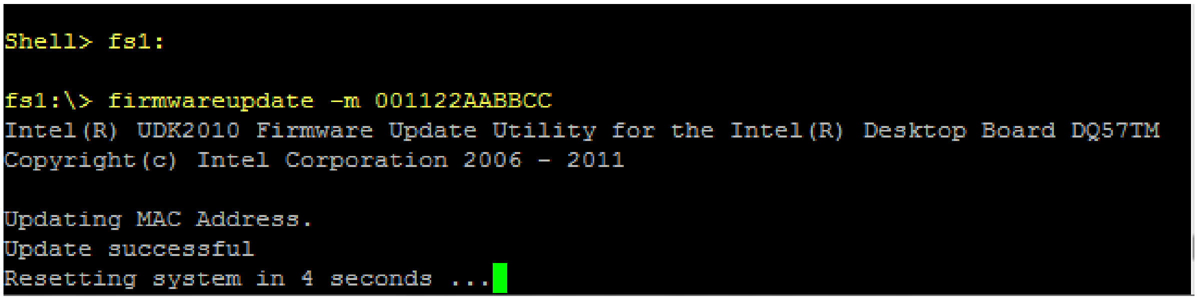
```
-E047-46C4-A947-14B2E063FEC0,0x9800,0x23BEC000)
  blk3    :HardDisk - Alias (null)
          PciRoot(0x0)/Pci(0x1F,0x2)/Ata(Secondary,Master,0x0)/HD(3,GPT,AE26A236
-24E3-4AF9-8B1F-C3205284D783,0x23BF5800,0x1839000)
  blk4    :BlockDevice - Alias (null)
          PciRoot(0x0)/Pci(0x1F,0x2)/Ata(Primary,Master,0x0)
  blk5    :BlockDevice - Alias (null)
          PciRoot(0x0)/Pci(0x1F,0x2)/Ata(Secondary,Master,0x0)
  blk6    :Removable BlockDevice - Alias (null)
          PciRoot(0x0)/Pci(0x1A,0x0)/USB(0x0,0x0)/USB(0x1,0x0)

Press ESC in 1 seconds to skip startup.nsh, any other key to continue.
Shell> map fs*
Device mapping table
  fs0     :HardDisk - Alias hd26c1 blk0
          PciRoot(0x0)/Pci(0x1F,0x2)/Ata(Secondary,Master,0x0)/HD(1,GPT,462A3B66
-9A38-4325-B371-8B5CACEB28A1,0x800,0x9000)
  fs1     :Removable HardDisk - Alias hd19a0b0b blk1
          PciRoot(0x0)/Pci(0x1A,0x0)/USB(0x0,0x0)/USB(0x1,0x0)/HD(1,MBR,0x575D21
D3,0x3F,0x794F1)

Shell> fs1:

fs1:\>
```

Figure 3-29. Executing the FS1 command in the UEFI shell.

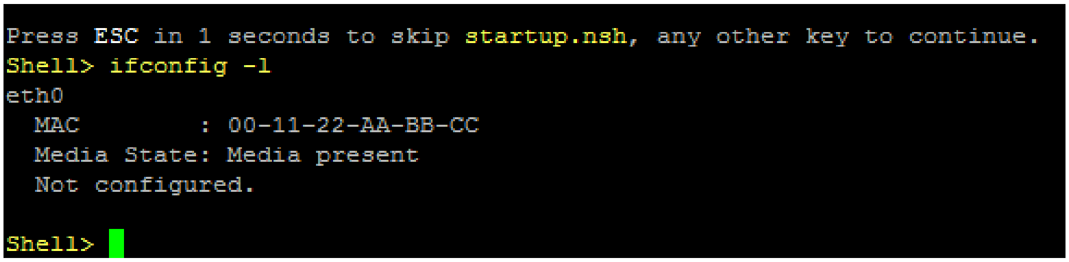


```
Shell> fs1:

fs1:\> firmwareupdate -m 001122AABBCC
Intel(R) UDK2010 Firmware Update Utility for the Intel(R) Desktop Board DQ57TM
Copyright(c) Intel Corporation 2006 - 2011

Updating MAC Address.
Update successful
Resetting system in 4 seconds ...
```

Figure 3-30. Executing the Firmware Update Application.



```
Press ESC in 1 seconds to skip startup.nsh, any other key to continue.
Shell> ifconfig -l
eth0
  MAC        : 00-11-22-AA-BB-CC
  Media State: Media present
  Not configured.

Shell>
```

Figure 3-31. The ifconfig command returns the MAC address.

By using the UEFI shell to verify the patched MAC address, you have also verified that the system is operational.

You have now successfully completed the BIOS update. Power off the motherboard and set the motherboard back to normal mode (BIOS CFG Jumper, pin 1-2).

Troubleshooting

It is possible that the value returned by the `ifconfig` command will not match the MAC value entered with the `firmwareupdate` command, or will not match the MAC address on the motherboard MAC address label. In this case, it is likely that:

- You wrote down the wrong MAC address from the motherboard MAC address label.
- You mistyped the MAC address when using the `firmwareupdate` command.

To resolve this issue:

1. Verify that you have copied the MAC address correctly from the motherboard MAC address label.
2. Rerun the firmware update utility (repeat steps 9 and 10 from the patch procedure), and make sure to enter the MAC address exactly as listed on the MAC address label.
3. Verify that the updated MAC address matches the MAC address on the motherboard label (repeat steps 11 through 14 from the patch procedure).

For More Information

For more information about the Intel® UDK 2010, visit the Intel® Web site.

For more information about the UEFI Specification, visit the UEFI home page.

Appendix A: Acronyms and Glossary

The following terms are used within this document:

API	Application program interface
BIOS	Basic input-output system
CFG	Configuration. For example, the BIOS CFG jumper on the motherboard.
Component	An executable image or library binary file built in EDK.
DEC	Driver execution code
DxE	Driver execution environment
ECC	Error correcting code. An ECC is an error detection correction feature of certain DRAM memory.
EDK II	UEFI developer's kit, 2nd Generation; the second series of the UEFI Developer's kit based on new design techniques and tools.
EFI	Generic term that refers to one of the versions of the Extensible Firmware Interface (EFI) or Unified Extensible Firmware Interface (UEFI) specification: EFI 1.02, EFI 1.10, UEFI 2.0, UEFI 2.1, UEFI 2.2, UEFI 2.3 or UEFI 2.3.1.
FAT	File Allocation Table, a file system commonly used by UEFI, MS-DOS and Microsoft Windows. This is a common file system format for USB flash drives.
FCE	Firmware configuration editor
FD	Firmware device; A persistent physical repository that contains firmware code and/or data and that may provide NVS.

FFS	Firmware file system; a binary storage format that is well suited to firmware volumes. The abstracted model of the FFS is a flat file system.
FMMT	Firmware module management tool
FV	Firmware volume
GC	Generation command (also garbage collection; the contextual use defines the specific meaning).
GUI	Graphical user interface
GUID	Globally unique identifier. A 128-bit value used to name entities uniquely. An individual without the help of a centralized authority can generate a unique GUID. This allows the generation of names that will never conflict, even among multiple, unrelated parties.
HII	Human interface infrastructure. This is a repository of configuration and translation information for localization. Typically used with boot manager and shell to provide a localized user interface.
IPF	Intel® Itanium™ processor family
KLOC	Thousand lines of code. This term is used as a weighted defect density measurement metric.
Module	A module is either an executable image or a library instance.
MRD	Market requirements document (this document)
NFG	Not functionally good
NVC	Non-volatile storage; A medium which continues to retain its contents when power is removed or has a battery back-up when main power is removed.
OS	Operating system
Package	A package is a container. It may hold a single file or a collection of files for any given set of modules. Packages may be described as one of the following types of modules: • Source modules, containing all source files and descriptions of a module

- Binary modules, containing UEFI Sections or a Framework File System and a description file specific to linking and binary editing of features and attributes specified in a Platform Configuration Database
- Mixed modules, with both binary and source modules

Multiple modules may be combined into a package, and multiple packages may be combined into a single package.

PC	Personal computer
Protocol	An API named by a GUID
PCD	Platform configuration database
PEI	Pre-UEFI initialization. The primary goals of PEI are to discover boot mode, launch modules that initialize main memory. It also discovers and launches DXE core and conveys platform information into DXE.
PEIM	PEI module; Set of drivers usually designed to initialize memory and the CPU so that DXE phase can run. Usually the first set of code run starting from reset.
PPI	A PEIM-to-PEIM interface that is named by a GUID. A PEIM-to-PEIM Interface is an interface that allows a PEIM to invoke another PEIM during the boot process.
PRD	Product requirements document –may be referred to as “The Implementation Plan” and drives product development. The PRD may be modified as negotiated requirements development during the course of project implementation. The PRD is an extension of the archived MRD.
SDK	Software development kit
SKU	Stock-keeping unit
SPI	Serial peripheral interface
UDK	UEFI development kit
UI	User interface

UML	Unified modeling language. Unified modeling language is a standardized general-purpose modeling language in the field of software engineering.
UEFI	Unified Extensible Firmware Interface (see: http://www.uefi.org/home/). UEFI is a firmware technology replacement for legacy BIOS. UEFI is an evolution of EFI, developed as a replacement of legacy BIOS to streamline the booting process and act as the interface between a PC operating system and its platform firmware. UEFI replaces only BIOS functions, but also offers a rich extensible pre-OS environment with advanced boot runtime services.