

# Deploying UEFI-Aware Operating Systems on Dell™ PowerEdge™ Servers

---

By Anand Joshi, Bill Munger, Mark Shutt,  
Thomas Cantwell, and John Sieber



THIS WHITE PAPER IS FOR INFORMATIONAL PURPOSES ONLY, AND MAY CONTAIN TYPOGRAPHICAL ERRORS AND TECHNICAL INACCURACIES. THE CONTENT IS PROVIDED AS IS, WITHOUT EXPRESS OR IMPLIED WARRANTIES OF ANY KIND.

© 2009 Dell Inc. All rights reserved. Reproduction of this material in any manner whatsoever without the express written permission of Dell Inc. is strictly forbidden. For more information, contact Dell.

*Dell*, the *DELL* logo, and the *DELL* badge, and *PowerEdge* are trademarks of Dell Inc. *Microsoft*, *Windows*, *Windows Server*, and *Active Directory* are either trademarks or registered trademarks of Microsoft Corporation in the United States and/or other countries.; *SUSE* is a registered trademark of Novell, Inc., in the United States and other countries.

# Table of Contents

History .....	2
What is UEFI? .....	2
What UEFI has to Offer .....	3
How is Dell's UEFI implemented? .....	3
Differences between BIOS and UEFI Boot Modes .....	4
UEFI Boot Options .....	4
Automatic and Manual Boot Options .....	4
Booting to a Removable Media Device .....	5
UEFI Boot Manager .....	5
Boot Options .....	5
UEFI Boot Settings .....	5
System Utilities .....	6
Deploying a UEFI-Aware Operating System .....	7
Deploying a UEFI Operating System .....	7
Limitations .....	8
Large Operating System Partitions .....	8
Third Party Imaging Utilities .....	8
Transferring Hard Disks .....	8
UEFI Legacy Boot Controllers .....	8

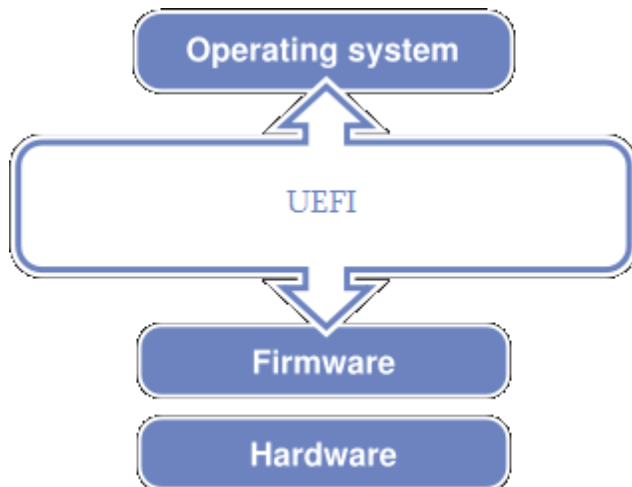
# History

In the mid-1990s during the development of Intel's Itanium-based system, PC BIOS limitations (for example, 16-bit processor mode, 1 MB addressable space, PC AT hardware dependencies, etc.) hindered platform development for Itanium. While the Itanium chip was not well received, the development exposed significant limitations to the original PC firmware architecture that led to the development of a new framework, called Extensible Firmware Interface (EFI).

In 2005, several leading tech companies came together to create the EFI Forum. Based on EFI architecture, the UEFI Forum's mission was to create the first ever industry standard firmware interface specification – Unified Extensible Firmware Interface (UEFI). Dell is represented on the board of directors of the UEFI Forum.

# What is UEFI?

The UEFI specification defines an interface between operating systems and platform firmware. The interface consists of data tables that contain platform-related information, plus boot and runtime service calls that are available to the operating system and its loader. Together, these provide a standard, modern environment for booting an operating system and running pre-boot applications.



In addition to the services, UEFI defines various protocols/APIs to access various hardware and the boot devices in the system. The UEFI specification also defines a driver model that is generic and can be adapted to any type of bus or device. The *UEFI Driver Model* is designed to support the execution of modular pieces of code, known as drivers, that run in the preboot environment. These drivers may manage or control hardware buses and devices on the platform, or they may provide some software-derived, platform-specific service.

There are 3 types of entities that can execute under UEFI environment:

**Applications:** Some examples of common EFI applications include the EFI shell, EFI shell commands, flash utilities, and diagnostic utilities. It is perfectly acceptable to invoke EFI applications from inside other EFI applications.

**Operating System Loader:** A special type of EFI application, called an operating system boot loader, provides the necessary initialization routines until the loader has set up enough of the operating system infrastructure to be ready to assume ownership of the system resources. Once the operating system assumes control, the EFI core frees all of its boot time services and drivers, leaving only the run-time services and drivers.

**Drivers:** EFI drivers differ from EFI applications in that the driver stays resident in memory unless an error is returned from the driver's entry point. The EFI core firmware, the boot manager, or other EFI applications may load drivers.

## What UEFI has to Offer

The primary goal of UEFI is to define an architecture that can scale with time and to offer structured coding environment allowing easy enablement of new technology. The main characteristics of UEFI are:

**Abstraction for the Operation System.** The UEFI specification provides an interface between the platform firmware and the operating system. The interfaces/API/protocols mark a clear boundary between the firmware and the operating system and the operating system loader.

**Abstraction for devices and related code.** UEFI abstracts interfaces that make it possible to build code which works on a range of underlying hardware devices without having explicit knowledge of the specifics for each device. The specification defines interfaces to platform capabilities such as standard bus types (PCI, USB, and SCSI). The list of supported bus types may grow over time. These defined interfaces and the ability to extend to future bus types are components of the UEFI *Driver Model*.

**Scalable platform environment.** The specification defines a complete solution for the firmware to describe all platform features and capabilities to the operating system during the boot process. These definitions cover a range of the contemporary platform designs but are simple enough to be able to extend in the future.

**Operating System Agnostic Rich Pre-Boot environment.** The UEFI specification defines extensible interfaces that enable creation of platform drivers. The drivers, analogous to operating system drivers, provide support for new devices and may provide enhanced platform capabilities, such as firmware update, platform configuration, diagnostics and deployment services. The existence of networking, USB, and file system capabilities adds to the richness of the pre-boot environment.

**GUID Partition Table.** The UEFI defines a new standard layout for the partition table, known as GUID Partition Table (GPT). GPT provides a more flexible mechanism for partitioning disks than the Master Boot Record (MBR) partitioning scheme that has been common to computers. MBR disks support only four partition table entries and the partition size is limited to 2TB. GPT scheme allows up to 128 primary partitions and can support larger than 2-terabyte partitions. Some near-term limitations to 2 terabyte support exist due to device support, but once devices fully support GPT/UEFI, this will no longer be an issue. See the "Limitations" section.

## How is Dell's UEFI implemented?

To support booting to both UEFI and non-UEFI aware operating systems, the Dell BIOS supports a Boot Mode option in the BIOS Setup Utility. The Boot Mode can be set to BIOS or UEFI.

Setting the **Boot Mode** to **UEFI** allows booting to a UEFI-aware operating system and provides the UEFI Boot Manager. This is the only change required to enable UEFI on the system. Setting the **Boot Mode** to **BIOS** provides support for all current and legacy operation systems.

## Differences between BIOS and UEFI Boot Modes

The following table highlights the differences between the boot modes.

Feature	Boot Mode	
	BIOS	UEFI
Operating System Support	Compatible with operating systems that do not support UEFI. All the current and legacy operation systems can be installed in this boot mode.	Must support UEFI
BIOS POST <F11> Boot Manager Hot Key	Enters BIOS Boot Manager	Enters UEFI Boot Manager
Boot Order Control	Via BIOS Setup Utility	Via UEFI Boot Manager
USB Emulation	Supported via BIOS Setup Utility	Not needed
Default Boot Order	Traditional Dell BIOS default boot order	None
Boot Options	Legacy BIOS boots to a drive. Support for multi-boot drive requires a boot loader.	In UEFI mode, there is a concept of a boot file. This allows multiple files to be added as a boot option on a single drive.

## UEFI Boot Options

The way boot options work in UEFI mode differs from that of the legacy BIOS. The UEFI boot option:

- Specifies a file on a drive as a boot target (vs. a drive as in legacy BIOS).
- Is automatically created by the operating system during installation and points to its boot file.
- Provides a predetermined boot path for removable media.
- Can be manually added by the user via the UEFI Boot Manager.
- Points to a specific boot file; hence a system format is not necessary to make a device bootable.

### ***Automatic and Manual Boot Options***

In UEFI mode, boot options are automatically added for removable devices. These automatically added boot options cannot be deleted. Operating System installation automatically adds a boot option that points to the Operating System Boot loader. Boot options can also be added manually by using the UEFI Boot Manager. Whether options are added manually or by the operating system, they can be deleted using UEFI Boot Manager.

Multiple boot options per device, or per file, are allowed. You may want to have two boot options for the same file with different input parameters, such as a debug parameter. When installing a UEFI-aware operating system, the installation process will add a corresponding boot option.

### **Booting to a Removable Media Device**

To make a removable device bootable, the UEFI application simply needs to be renamed to BOOTx64.EFI (case insensitive) and placed in the \EFI\BOOT directory.

When a removable device, such as a USB key, is detected in UEFI Boot Mode, a boot option is automatically added to point to the following location:

```
\EFI\BOOT\BOOTx64 .EFI
```

### **UEFI Boot Manager**

The UEFI Boot Manager is entered by pressing <F11> during the pre-boot phase of system startup. In UEFI boot mode, it replaces both the BIOS Boot Manager and the boot configuration options in the BIOS Setup Utility <F2>. When the **Boot Mode** is set to **UEFI** the BIOS Setup Utility fields, **Boot Sequence** and **USB Flash Drive Emulation Type** are grayed out, replaced by enhanced options in the UEFI Boot Manager.

#### **Boot Options**

The front page of the UEFI Boot Manager lists boot options in order of boot priority. Selection of any of these options results in an attempt to boot. If the option is not bootable (for example, no media in an optical drive), an error message displays along with a prompt to enter the UEFI Boot Manager or the BIOS Setup Utility.

In cases where a boot option has been added manually or by an operating system installation and that option is no longer valid (for example, a hard-disk drive has been removed), the option will continue to be listed, but will be grayed out and display “Unavailable.” Boot Option entries for removable devices are automatically added and removed based in insertion and removal of the device.

#### **UEFI Boot Settings**

The **UEFI Boot Settings** menu can be entered from the front page. The following options are available from this menu:

Option	Description
<b>Add Boot Option</b>	<p>Provides a menu from which to select a boot option.</p> <p>For mass storage devices with a FAT32 file system, a menu displays to navigate to a file to select as the boot option. For PXE-capable devices, the device itself can be selected.</p> <p>When adding a new boot option, the <b>Description</b> and <b>Optional Data</b> can be entered. The <b>Description</b> identifies the device in the UEFI Boot Manager; the <b>Optional Data</b> is passed to the boot image, per the UEFI specification.</p>
<b>Delete Boot Option</b>	<p>Provides a menu of boot options that were manually and operating system installed and can be deleted. Boot options that were automatically added are not listed.</p>
<b>Enable/Disable Boot</b>	<p>Provides a menu of boot options that can be disabled which removes them from the boot option list. Disabled devices can be re-enabled from</p>

<b>Option</b>	this menu.
<b>Change Boot Order</b>	Allows the boot priority of the boot options to be modified.
<b>One-Time Boot From File</b>	Provides a menu similar to that of <b>Add Boot Option</b> , but does not add the boot option. Instead, it attempts to boot to the selected file or device.

### ***System Utilities***

The **System Utilities** menu can be entered from the front page. The following options are available from this menu:

<b>Option</b>	<b>Description</b>
<b>System Setup</b>	Enters the System Setup Utility. This can also be entered by pressing <F2> during the pre-boot phase of system startup.
<b>System Services</b>	Enters System Services. This can also be entered by pressing <F10> during the pre-boot phase of system startup.
<b>BIOS Boot Manager</b>	Enters the BIOS Boot Manager to provide the ability to boot to a non-UEFI aware operating system while the <b>Boot Mode</b> is set to <b>UEFI</b> . This is the same Boot Manager that would be entered by pressing <F11> during the pre-boot phase of system startup if the <b>Boot Mode</b> is set to <b>BIOS</b> .
<b>Reboot System</b>	Reboots system.



## Deploying a UEFI-Aware Operating System

Once the hardware and firmware support UEFI, the operating system needs to be redesigned to support UEFI. UEFI is still an emerging technology and standard, so there are only a few operating systems that have full support. Dell offers both legacy BIOS-mode and UEFI-mode.

### Operating Systems that support UEFI -

- 1) Microsoft® Windows® Vista, SP1 and Windows Server® 2008 – X64 versions only.
- 2) SUSE® Linux Enterprise Server, release 11, will contain support
- 3) General Linux support is available, but ask your operating system vendor if they fully support UEFI or what their plans are.

## Deploying a UEFI Operating System

For Microsoft operating systems, press <F11> and change the **Boot Manager** to **UEFI**. Once this is done, simply boot from the Windows Server 2008 or Vista DVD and follow the normal installation steps. From this perspective, everything will be the same as BIOS-mode installation.

What HAS changed? –

The primary visible changes are to the hard drive setup although this is not visible to the customer during installation. BIOS-mode setup installs MBR and has a single boot partition. UEFI uses GPT. For a Microsoft Windows installation, three partitions are created:

Partition	Type	Format
EFI System Partition (ESP)	EFI (hidden)	FAT32
Microsoft Reserved Partition (MSR)	MSR	
Windows operating system partition	Primary	NTFS

The ESP contains the EFI-capable Windows Boot Manager, boot configuration data (BCD), and related font files

The MSR is used to store metadata outside of the Windows volume. The MSR partition must exist between the ESP and the Windows operating system partitions.

These partitions are visible with the Diskpart utility – DO NOT change or remove these partitions.

**IMPORTANT:** You *cannot* switch between BIOS-mode and UEFI-mode without fully wiping the drive, either by using the Diskpart **Clean** command or through the hard disk controller and reinitializing the hard drive. Once you transition to UEFI, Dell recommends that you leave the system in UEFI-mode.

Once the operating system is installed, you can install applications and use the system normally with a few caveats. There are applications that may try to interact with the hard drive directly (imaging tools) – these may not work as

expected. Dell recommends that you contact your application vendors for all mission-critical applications to ensure they work in the UEFI environment.

## **Limitations**

This section explains potential problems which may occur while installing a UEFI-aware operating system and how to avoid them.

### ***Large Operating System Partitions***

An operating system installer relies on disk controller firmware to perform disk read/write functions prior to the disk driver is available. Certain legacy disk controllers use older firmware (such as Interrupt 0x13 handlers) to execute read/write functions in UEFI mode and do not provide native access disks in UEFI mode. The UEFI layer provides the abstraction driver to access the disks using the old firmware. The most notable restriction is that the legacy interrupt 0x13 functions are unable to access logical block addresses higher than 2 terabytes (TB). For this reason, Dell strongly recommends using a partition or virtual disk smaller than 2TB for the partition where the operating system resides. Failure to use a partition size of less than 2TB for the operating system virtual disk may cause undesired behavior and will eliminate the inherent GPT partition redundancy. Data partitions (non-system partitions) can be larger than 2TB since the disk driver is available when the operating system accesses these partitions.

### ***Third Party Imaging Utilities***

UEFI uses a partitioning scheme known as a GPT. This scheme assigns a Globally Unique Identifier (GUID) to each partition in a disk. Be aware that imaging utilities (such as Norton Ghost) need to re-create the GPT in order to have unique GUIDs for partitions which are copied from one disk to another. Otherwise different disks will use the same GUID for multiple partitions, which may cause conflicts for software which uses the GUIDs to distinguish each partition.

### ***Transferring Hard Disks***

UEFI Boot Options are stored in non-volatile memory on a system. When an operating system installer executes, it creates a boot option for the operating system in the system's non-volatile memory. If the hard disk which contains the operating system is removed and transferred to another system, this new system will not have the boot option in its non-volatile memory, and a user will not be able to boot to the operating system.

Dell has addressed this problem by keeping a copy of the operating system boot option on the hard disk with the operating system. When the hard disk is transferred to another Dell system, the system uses the copy of the operating system boot option to create a boot option in its non-volatile memory. This allows a user to boot to the operating system. The process is automatic; no user interaction is necessary to create the boot option on the second system.

### ***UEFI Legacy Boot Controllers***

The UEFI layer provides the abstraction driver to access the disks using the interrupt 0x13 functions. The following Dell UEFI Legacy boot controllers operate using legacy abstraction supported in UEFI mode:

Dell PERC 6/i

Dell SAS 6/iR

Future controllers will provide native UEFI drivers.