Installing Windows Server 2008 and Windows Vista SP1 on UEFI Systems

July 25, 2008

Abstract

Windows Server® 2008 and Windows Vista® SP1 support Unified Extensible Firmware Interface (UEFI) on Intel Itanium-based and x64-based platforms. This paper provides guidelines for system manufacturers and information technology (IT) administrators to prepare and install Windows Server 2008 or Windows Vista SP1 on UEFI-based computers by using Windows® Setup or the deployment tools in the Windows OEM Preinstallation Kit (OPK) or the Windows Automated Installation Kit (WAIK).

*Hardware Support and Directions for Microsoft Windows Server* *Series*. This publication is the part of the Hardware Support and Directions for Microsoft Windows Server series, which shares the Microsoft intention and investment direction for support of specific hardware tech­nologies in current and future releases of the Windows Server operating system.

The series provides reference material for engineers who design servers or their core electronic components, technical decision-makers, system architects, and server product planners who want to have their systems support future releases of the Windows Server family of operating systems.

This information applies to the following operating systems:  
 Windows Server 2008  
 Windows Vista SP1

The current version of this paper is maintained on the Web at:   
 <http://www.microsoft.com/whdc/system/platform/firmware/uefiguide.mspx>

*Disclaimer: This is a preliminary document and may be changed substantially prior to final commercial release of the software described herein.*

*The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.*

*This White Paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.*

*Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.*

*Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document.* Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

*Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, email address, logo, person, place or event is intended or should be inferred.*

*© 2007–2008 Microsoft Corporation. All rights reserved.*

*Microsoft, BitLocker, Windows, Windows Server, and Windows Vista are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.*

*The names of actual companies and products mentioned herein may be the trademarks of their respective owners.*

Document History

| **Date** | **Changes** |
| --- | --- |
| July 25, 2008 | Updated document to include Windows Vista SP1 and better describe the Windows Setup installation procedure.  Changed title to reflect new contents. |
| May 9, 2007 | Originally published as “Imaging Guidelines for Windows Server 2008 on UEFI Systems” |

Contents

[Introduction 4](#_Toc204580640)

[Using Windows Setup with UEFI Systems 4](#_Toc204580641)

[How to Obtain an Installation DVD 5](#_Toc204580642)

[How to Prepare the System 5](#_Toc204580643)

[How to Use Windows Setup to Install Windows with UEFI Support 6](#_Toc204580644)

[Creating and Deploying a Windows Image for UEFI Systems 8](#_Toc204580645)

[Capturing a Windows Image with UEFI Support 8](#_Toc204580646)

[Deploying a Windows Image with UEFI Support 10](#_Toc204580647)

[Resources 12](#_Toc204580648)

[Appendix A: BCD System Store Settings for UEFI 14](#_Toc204580649)

[Windows Boot Manager Settings 15](#_Toc204580650)

[The Device Setting 15](#_Toc204580651)

[The Path Setting 16](#_Toc204580652)

[Other Settings 16](#_Toc204580653)

[Windows Boot Loader Settings 17](#_Toc204580654)

[The Device and OSDevice Settings 17](#_Toc204580655)

[The Path Setting 18](#_Toc204580656)

[Windows Memory Tester Settings 18](#_Toc204580657)

[The Device Setting 18](#_Toc204580658)

[The Path Setting 18](#_Toc204580659)

[BCD Settings for BIOS Systems 19](#_Toc204580660)

[Windows Boot Manager Settings 19](#_Toc204580661)

[Windows Boot Loader Settings 19](#_Toc204580662)

[Windows Resume Application Settings 19](#_Toc204580663)

[Windows Memory Tester Settings 20](#_Toc204580664)

[Appendix B: Clearing Duplicate Firmware Objects in BCD and NVRAM 21](#_Toc204580665)

# Introduction

Windows Server® 2008 and Windows Vista® SP1 include the following support for Unified Extensible Firmware Interface (UEFI) on Intel Itanium-based and x64-based computers:

* Disk partitions support the GUID partition table (GPT) disk format.
* Windows® Setup can initialize the system's hard drive to use the GPT format and create an EFI system partition.
* The system can install and service boot environment files on the EFI system partition, including the Windows Boot Manager, Boot Configuration Data (BCD), and related font files.
* The system includes EFI versions of the Windows Boot Manager and Windows resume boot applications.
* The system automatically manages nonvolatile RAM (NVRAM) boot entries that reference the partition that contains Windows Boot Manager.
* The system provides backup support for the EFI system partition.
* Windows BitLocker™ Data Encryption supports UEFI systems.

You can install Windows Server 2008 or Windows Vista SP1 UEFI-based platforms in either of the following ways:

* Use Windows Setup to install the operating system on a single UEFI-based computer.
* Use the tools in the OEM Preinstallation Kit (OPK) or the Windows Automated Installation Kit (WAIK) to create and deploy a Windows image on multiple UEFI-based computers.

This paper provides guidelines for information technology (IT) administrators who must install Windows Server 2008 or Windows Vista SP1 on UEFI systems. The primary focus of the paper is about how to configure an original equipment manufacturer (OEM) factory image or create and deploy a Windows image for internal corporate use.

**Note:** This paper assumes that you have access to the Windows OPK or the WAIK. This paper is intended to be an extension of the OPK or AIK documentation. You should refer to that documentation for details. For information on how to obtain the OPK or WAIK, see ”Resources” at the end of this paper.

# Using Windows Setup with UEFI Systems

Windows Setup provides the simplest way to install Windows Server 2008 or Windows Vista SP1 on a UEFI-based computer. This approach is most useful for installing Windows on, at most, a small number of computers. However, the Windows Setup procedure that is described in this section is also an essential part of preparing an image for deployment, as discussed later in this paper.

## How to Obtain an Installation DVD

To install Windows Server 2008 or Windows Vista SP1 on a UEFI system, you must have a Windows installation DVD. In particular, you cannot add UEFI support to the release to manufacturing (RTM) version of Windows Vista by installing the Windows Vista SP1 update package. You must do a complete installation of Windows Vista SP1 on a clean hard drive with no partitions.

You can use the Windows Server 2008 RTM installation DVD to install the operating system with UEFI support. If you plan to install Windows Vista SP1, you can obtain an installation DVD through several channels:

* Retail customers can purchase a Windows Vista SP1 installation DVD through the usual retail channels.
* OEMs that license the Windows OPK receive a Windows Vista SP1 update to the OPK that includes a master installation medium that can be duplicated for internal use.
* System builders can purchase a variation of the Windows Vista SP1 installation DVD that they can include with the systems that they assemble and sell.
* Volume license customers can download Windows Vista SP1 from a secure Web site as an .iso file, just as they download beta versions of Windows from Microsoft Connect. They can then use the .iso to create an installation DVD.

## How to Prepare the System

UEFI-based computers use a GPT disk partition format, which is distinct from the master boot record (MBR) format that is used for BIOS-based systems. UEFI computers require at least three GPT partitions:

* An EFI system partition (ESP) to store the boot applications and other related information.
* A Microsoft reserved (MSR) partition, which must be located between the ESP and the Windows operating system partitions.
* A Windows system partition.

Additional partitions might also be necessary for purposes such as supporting BitLocker. For more information about GPT disks, EFI, and hard-disk partitions, see ”Windows and GPT FAQ.” For a comparison of MBR and GPT disk partitioning, see ”How Basic Disks and Volumes Work.”

**Note:** Windows does not assign a driver letter to the ESP. This partition should not be used for any user or application data files.

You can use Windows Setup to install Windows 2008 or Windows Vista SP1 on any UEFI-compatible computer. However, if the hard drive that you plan to use already has an MBR disk format and an existing operating system, you must first delete all partitions on the selected hard drive. This operation deletes all data on the hard drive, so be sure to back up any important data. You then install Windows with UEFI support on a clean hard drive and create the required GPT partitions.

The simplest way to create the GPT partitions is to run Windows Setup in attended mode, as discussed in the next section. Setup automatically creates the ESP, MSR, and Windows system partitions before it installs Windows, as shown in Table 1.

Table 1. UEFI Partitions

| **Partition** | **Size** | **Type** | **Format** |
| --- | --- | --- | --- |
| ESP | 200 megabytes (MB) | EFI (hidden) | FAT32 |
| MSR | 128 MB | MSR |  |
| Windows system partition | Available disk size |  | NTFS |

To run Windows Setup in unattended mode, you must manually create the three partitions before you run Setup. The following **Diskpart** command creates the EFI, MSR, and Windows system partitions:

Diskpart

Select disk 0 // 0 being the disk to setup

Clean // wipe the disk

Convert gpt // convert disk to GPT

Create partition efi size=200 // EFI system partition

Assign letter=s // Any allowable letter

Format quick fs=FAT32 // Format the ESP

Create partition msr size=128 // Create the MSR partition

Create partition primary // Create Windows partition

Assign letter=c

Format quick fs=NTFS // Format primary partition

Exit

**Note:** The Setup and boot environment of Windows Vista RTM depend on the MBR disk format and BIOS firmware support, which do not support UEFI. To install Windows SP1 with UEFI support, you must do a complete installation of Windows Vista SP1 on a clean hard drive with no partitions.

## How to Use Windows Setup to Install Windows with UEFI Support

To use Windows Setup to install Windows Server 2008 or Windows Vista SP1 with UEFI support, you must boot the installation DVD from the UEFI firmware. The installation DVD contains a multi-entry boot catalog—as specified in *El Torito Bootable CD-ROM Format Specification*—with one entry for a BIOS-initiated boot and one for a UEFI-initiated boot. Both entries share a common Windows Setup image.

The platform firmware initiates the boot application and boots the system into the Windows preinstallation environment (Windows PE) for UEFI. Windows PE automatically runs Windows Setup, which creates the required GPT partitions and installs the boot files on the EFI partition before it installs the operating system. Windows Setup also initializes BCD entries for the EFI versions of Windows Boot Manager, Windows boot loader, Windows memory tester, and Windows resume application.

Different UEFI firmware implementations support different methods for handling the installation operation. For UEFI firmware based primarily on the Intel TianoCore framework:

1. Insert the installation DVD and restart the computer.

2. When the firmware starts to initialize, press F10 and then open the UEFI firmware boot device menu.

3. Select **EFI DVD/CD** as the boot device, which causes the system to display **Hit any key to boot from the DVD**.

After that message appears, you must press a key within 5 seconds to continue the installation.

After you press the key, the firmware uses the UEFI-specific boot information on the installation DVD to boot into the Windows PE for UEFI.

If the computer does not have a **Boot from EFI CD or DVD** option, you can use either of the following ways to boot into Windows PE for UEFI:

* Use Device Manager's **Firmware** option and select **Boot from file**.
* Use the EFI internal shell, as follows:

Start the EFI internal shell from the EFI boot menu and select the drive with the installation DVD. The following example assumes that the DVD drive is Fs0:

Shell> Fs0:

Start the EFI boot application. There are separate applications for x64 and Intel Itanium systems. For an x64 system, run the following command:

fs0:> \EFI\BOOT\BOOTX64.EFI

For an Intel Itanium system, run the following command:

Fs0:\EFI\BOOT\BOOTIA64.EFI

After the application starts, the system displays the following message:

Press any key to boot from CD or DVD…

Press a key to continue booting from the installation media.

Some EFI-compatible platforms support both UEFI and BIOS firmware, and it is not always apparent whether EFI or BIOS is the default DVD boot option. As UEFI firmware becomes more widely available, the UEFI firmware might use the EFI entry as the default DVD boot option. If you are not sure of the default boot behavior, you might be required to use the EFI boot options to explicitly start the correct EFI boot application, even if EFI Boot Manager supports booting from a DVD. Otherwise, Windows Setup might run in BIOS mode, which does not create the necessary GPT partitions.

After the installation is complete, you can check whether Windows used the EFI boot environment by running **Diskpart** and verifying that the disk partition table is the GPT format and that the ESP and MSR partitions are present.

# Creating and Deploying a Windows Image for UEFI Systems

You can simplify the process of installing Windows Server 2008 or Windows Vista SP1 on multiple UEFI-based computers by using the OPK or WAIK tools to create and deploy a custom Windows image.

To use the OPK or WAIK tools to create and deploy a customized Windows image, you must have the following systems, which must be network connected:

* A technician computer.

You install the OPK or WAIK tools, including the Windows system image manager (SIM), on the technician computer.

* A UEFI-compatible master computer.

You create a master installation by installing and customizing the selected Windows version on this computer. You then use the **ImageX** tools to create an image of the master installation for deployment.

* One or more UEFI-compatible destination computers.

You use Windows PE and the **ImageX** tools to deploy the customized Windows image to the destination computers.

## Capturing a Windows Image with UEFI Support

The first step in deploying Windows Server 2008 or Windows Vista SP1 for UEFI-based computers is to create an image of the disk partitions.

To create an image of the disk partitions

1. Use the Windows SIM on the technician computer to create an answer file that describes the details of your master installation. For details on how to create an answer file, see the OPK or WAIK documentation. Save the file to a DVD or other portable media.

2. Prepare a master installation by installing the appropriate version of Windows with UEFI support on the master computer and customizing the installation as required.

3. Capture an image of the master installation for deployment to the destination computers.

**Note:** To capture an image of a master installation with UEFI support, you must capture both the Windows system partition and the ESP. You cannot capture the MSR partition. Instead, you must manually re-create the MSR on the destination computers, as discussed in ”Deploying a Windows Image with UEFI Support.”

To prepare the master installation

1. Prepare the master computer, as described in the WAIK or OPK documentation.

2. Insert the appropriate Windows installation DVD and restart the computer. Then use Windows Setup to install the appropriate version of Windows with UEFI support on the master computer, as discussed earlier in this paper.

3. Optionally, configure other partitions based on your preinstallation design.

For example, you might want to configure additional partitions to customize the master installation for Windows Recovery Environment (Windows RE) or BitLocker, as described in the ”Preinstallation Design” section of the OPK.

4. Optionally, apply further customizations to the installed copy of Windows, such as custom drivers.

5. Generalize the image by running **Sysprep** from a Command Prompt window with elevated privileges:

Sysprep /generalize /OOBE /shutdown

This command generalizes the image and shuts down the system.

After the master installation is complete, you capture its image.

To capture an image of the master installation

1. Boot the master computer to Windows PE 2.0, either from a DVD or from the network by using a Pre-Boot eXecution Environment (PXE) boot.

**Note:** To capture the image, you are not required to boot into Windows PE for UEFI.

2. Assign a drive letter to the ESP by using **Diskpart**.

The following example assigns **s:** to the master computer's ESP:

Diskpart

Select disk 0 //select the disk

List volume // locate the 200MB hidden volume

Select volume 2 // assume vol 2 is the EFI part

Assign letter=s // assign any allowable letter

Exit // exit diskpart

Tip: You can script **Diskpart** commands by putting them in a text file and passing the file to **Diskpart**. For example:

diskpart /s mycommands.txt

3. Map a drive to the network share where the images are to be stored. This share could, for example, be on the technician computer.

The following example maps drive **n:** to the \\techcomp\images share on the technician computer:

net use n: [\\techcomp\images](file:///\\techcomp\images)

4. Capture the Windows system partition to a .wim file on the mapped network share by running **ImageX /capture**.

The following example captures an image of Window system partition— drive **c:** on the master computer—and places it on **n:** as a file that is named uefi\osimage.wim:

imageX /capture c: n:\uefi\osimage.wim "Windows Server 2008"

5. Capture the ESP to a separate .wim file on the same network share by running **ImageX /capture**.

The following example captures an image of the ESP—drive **s:** on the master computer, as defined in step 6—and places it on **n:** as a file that is named uefi\efisys.wim:

imageX /capture s: n:\uefi\efisys.wim "EFI partition"

You are now ready to deploy the image to destination computers.

## Deploying a Windows Image with UEFI Support

To deploy the customized Windows image that you captured in the preceding section, you must configure the destination computers and use Windows PE to apply the EFI and Windows system images in a generalized state to the destination computer. You then make the image bootable by updating the destination computer's BCD settings and creating an EFI Boot Manager entry so that the computer automatically boots the image.

To deploy the image

1. Boot Windows PE on the destination computer by using the UEFI boot option, as described in “How to Use Windows Setup to Install Windows with UEFI Support” earlier in this paper. The remainder of this procedure uses the Windows PE Command Prompt window.

Note: You must boot into Windows PE for UEFI. If you start Windows PE by using the BIOS boot option, the BCDEdit operations that are described later in this procedure cannot correctly locate the BCD system store.

2. Create ESP, MSR, and Windows system partitions on the destination computer. They *must exactly match* the partition structure of the master computer, as shown in Table 1.

The following **Diskpart** command creates the correct ESP, MSR, and Windows system partition and assigns drive letters to the ESP (**s:**) and Windows system partitions (**c:**):

Diskpart

Select disk 0 // 0 being the disk to setup

Clean // wipe the disk

Convert gpt // convert disk to GPT

Create partition efi size=200 // EFI system partition

Assign letter=s // Any allowable letter

Format quick fs=FAT32 // Format the ESP

Create partition msr size=128 // Create the MSR partition

Create partition primary // Create Windows partition

Assign letter=c

Format quick fs=NTFS // Format primary partition

Exit

3. Map the network share on which the captured images are stored. This example assumes that you have mapped **n:** to the \\techcomp\images network share that was used as an example in the previous section.

4. Apply the ESP image to the destination computer's ESP by using **ImageX /apply**.

This step installs the master installation's EFI boot applications, EFI drivers, and other ESP boot files on the destination computer's ESP. The command requires a path to the ESP image file on the network share, an image number, and the path on the partition where the image is to be applied.

The following example applies the ESP image that was created in the previous section—efisys.wim—to the destination computer's ESP partition (**s:**):

imageX /apply n:\uefi\efisys.wim 1 s:\

5. Apply the Windows system image to the destination computer's Windows system partition by using **ImageX /apply**.

This step installs the files from the master installation's Windows system partition to the corresponding partition on the destination computer. The following example applies the Windows system image created in the previous section—osimage.wim—to the destination computer's Windows system partition (**c:**):

imageX /apply n:\uefi\osimage.wim 1 c:\

After you deploy the image, use BCDEdit from the Windows PE Command Prompt window to update the destination computer's BCD system store.

**Tip:** BCD uses globally unique identifiers (GUIDs) to identify the various boot entries. To simplify the command lines, BCDEdit provides readable standard identifiers for the commonly used boot application GUIDS. Table 2 lists the boot entry identifiers that are used in this paper.

Table 2. Boot Application Standard Identifiers

| **Boot application** | **Identifier** |
| --- | --- |
| Windows Boot Manager | {bootmgr} |
| Windows memory tester | {memdiag} |
| Firmware boot manager | {fwbootmgr} |
| Default Windows boot loader | {default} |

**Note:** The Windows resume application does not have a standard identifier. You must use an explicit GUID.

The following procedure updates those BCD settings that are necessary to make the deployed image bootable. For information on additional BCD settings, see Appendix A.

To update the BCD system store

1. Update the BCD settings for the Windows Boot Manager and memory tester boot applications so that they point to the destination computer's ESP. The following example assumes that **s:** is the ESP:

Bcdedit /set {bootmgr} device partition=s:

Bcdedit /set {memdiag} device partition=s:

When you change the settings, BCDEdit.exe creates an EFI NVRAM entry for Windows Boot Manager.

2. Update the default Windows boot loader entry to point to the destination computer's Windows system partition.

The following example updates the settings for the default Windows boot loader, assuming that **c:** is the Windows system partition:

Bcdedit /set {default} device partition=c:

Bcdedit /set {default} osdevice partition=c:

**Note:** **device** is set to the partition that contains the boot application. **osdevice** is set to the partition that contains the system root. If a computer has BitLocker enabled or includes multiple versions of Windows, **osdevice** and **device** are not necessarily set to the same partition.

3. Update the firmware boot display order to place Windows Boot Manager as the first boot option:

Bcdedit /set {FWbootmgr} displayorder {Bootmgr} /addfirst

4. Update the Windows resume application's **device** element to point to the partition that contains the system root.

Use **Bcdedit /enum** **all** to determine the Windows resume application's identifier GUID and then use that identifier to set the **device** element. The following example sets the Windows resume application's device element to **c:**. *<ID GUID >* is the Windows resume application's identifier GUID and the system root is on **c:** in the following example:

Bcdedit /set *<ID GUID>* device partition=c:

5. Restart the system to boot to the deployed image. Then use the Out of Box Experience Wizard to activate Windows.

The system is now ready to be distributed to an end user.

# Resources

BCDEdit Commands for Boot Environment

<http://www.microsoft.com/whdc/system/platform/firmware/bcdedit_reff.mspx>

Boot Configuration Data in Windows Vista

<http://www.microsoft.com/whdc/system/platform/firmware/bcd.mspx>

[“El Torito” Bootable CD-ROM Format Specification](http://www.phoenix.com/NR/rdonlyres/98D3219C-9CC9-4DF5-B496-A286D893E36A/0/specscdrom.pdf)

[http://www.phoenix.com/NR/rdonlyres/98D3219C-9CC9-4DF5-B496-A286D893E36A/0/specscdrom.pdf](http://www.phoenix.com/NR/rdonlyres/98D3219C-9CC9-4DF5-B496-A286D893E36A/0/specscdrom.pdf%20)

How Basic Disks and Volumes Work (TechNet Web)

<http://technet2.microsoft.com/windowsserver/en/library/bdeda920-1f08-4683-9ffb-7b4b50df0b5a1033.mspx?mfr=true>

Microsoft OEM Preinstallation Kits

<http://www.microsoft.com/oem/sblicense/OPK/default.mspx>

UEFI Support and Requirements: Windows Server 2008http://www.microsoft.com/whdc/images/shared/common/space.gif

[http://www.microsoft.com/whdc/system/platform/firmware/uefireg.mspx](http://www.microsoft.com/whdc/system/platform/firmware/uefireg.mspx%20)

UEFI Information and Resources on WHDC Web

[http://www.microsoft.com/whdc/system/platform/firmware/default.mspx](http://www.microsoft.com/whdc/system/platform/firmware/default.mspx%20)

Windows and GPT FAQ (WHDC Web)

<http://www.microsoft.com/whdc/device/storage/GPT_FAQ.mspx>

Windows Automated Installation Kit (AIK)

<http://www.microsoft.com/downloads/details.aspx?FamilyID=c7d4bc6d-15f3-4284-9123-679830d629f2&DisplayLang=en>

# Appendix A: BCD System Store Settings for UEFI

For a typical deployment scenario, you create a custom BCD store that meets your requirements and then import that custom store into the BCD system store on your destination computers.

Windows Server 2008 and Windows Vista SP1 include a template for a UEFI-compatible BCD store—BCD-template. It includes all entries that are required for a UEFI system, with appropriate settings. You can easily customize that template to create a BCD store that meets your specific requirements and import that custom store into the BCD system store on your destination computers. BCD-template is located in \Windows\System32\config.

To create a custom BCD store, make a copy of BCD-template and then modify that copy to suit your particular requirements. For the following example, the copy is named newbcd.

**Note:** By default, BCDEdit operates on the system store. To modify the elements in a nonsystem store, you must include the store's name in the BCDEdit command. For example, the following command enumerates the contents of newbcd:

bcdedit /store newbcd /enum all

After you have finished modifying newbcd, you import newbcd to the system store. This action replaces the boot configuration that was defined by the current BCD system store with a configuration that was defined by newbcd. The following BCDEdit command imports newbcd into the system store:

bcdedit /import newbcd

To use the newbcd configuration for your destination computers, copy newbcd to those computers and use BCDEdit to import newbcd into each computer's BCD system store.

An alternative approach to creating a new BCD system store is to modify the existing store and export it to a temporary store. The following command exports the BCD system store to newbcd:

bcdedit /export newbcd

You can then copy newbcd to the other destination computers and import it into their BCD system stores, as discussed earlier.

The remainder of this section discusses the various BCD settings that you can or must modify. On UEFI systems, this includes settings for the following boot applications:

* Windows Boot Manager
* Windows boot loader
* Windows resume application
* Windows memory tester

The following sections describe the available settings for each of these boot applications in detail and how to modify appropriately for UEFI systems.

**Note:** For simplicity, the BCDEdit examples in this section modify the BCD system store. To modify another store, such as a copy of BCD-template, include the store name in the command line.

## Windows Boot Manager Settings

Windows Boot Manager ({bootmgr}) manages the boot process. UEFI-based systems contain a firmware boot manager—bootmgfw.efi—that loads an EFI application that is based on variables that are stored in NVRAM.

The BCD settings for Windows Boot Manager's **device** and **path** elements must point to the firmware boot manager. BCD-template for Windows Server 2008 includes the following settings for Windows Boot Manager:

Windows Boot Manager

--------------------

identifier {bootmgr}

device partition=\Device\HarddiskVolume1

path \EFI\Microsoft\Boot\bootmgfw.efi

description Windows Boot Manager

The following sections discuss the **device** and **path** settings, as well as several other optional settings.

### The Device Setting

For UEFI systems, Windows Boot Manager's **device** element must be set to the ESP volume letter. To determine the correct volume letter, use the **Diskpart** command to view the disk partitions. The following example assumes a system with a single hard drive with multiple partitions. The ESP has been assigned a drive letter of **s:**, as used in the examples earlier in this paper.

The following **Diskpart** commands select disk 0 and then list the details of the volumes on that disk, including their drive letter. It shows Volume 2 as the ESP:

DISKPART> select disk 0

DISKPART> list volume

Volume ### Ltr Label Fs Type Size Status Info

---------- --- ------ ----- ---------- ------- --------- ------

Volume 0 D NTFS Partition 103 GB Healthy

Volume 1 C NTFS Partition 49 GB Healthy Boot

Volume 2 S FAT32 Partition 200 MB Healthy System

If the ESP does not have an assigned drive letter, assign one by using the **Diskpart** **assign** command. The following example assumes that ESP is volume 2 and assigns it **s:** as the drive letter:

Diskpart

select disk 0

list volume

select volume 2 // assuming vol 2 is ESP

assign letter=s

After you have determined the ESP volume, set Windows Boot Manager's **device** element to the corresponding drive letter. The following example sets **device** to the **s:** drive:

Bcdedit /set {bootmgr} device partition=s: // ESP

### The Path Setting

The **device** element specifies the volume on which Windows Boot Manager is located. The **path** element specifies where on that volume the Windows Boot Manager application is located. For UEFI systems, **path** must point to the firmware boot manager, whose path is \EFI\Microsoft\Boot\bootmgfw.efi.

BCD-template has the correct setting, which you can confirm by enumerating the values in the store, as follows:

bcdedit /store bcd-template /enum all

To explicitly set **path** to \EFI\Microsoft\Boot\bootmgfw.efi, use the following command:

Bcdedit /set {bootmgr} path \efi\microsoft\boot\bootmgfw.efi

### Other Settings

You should set Windows Boot Manager to be the first item in the EFI firmware's display order, as shown in the following example:

Bcdedit /set {fwbootmgr} displayorder {bootmgr} /addfirst

You should also specify the topmost Windows boot loader application in Windows Boot Manager's display order. Windows Boot Manager runs the topmost boot loader by default if the user does not choose a preferred boot loader. The following example shows how to put a specified Windows boot loader at the top of the display order:

Bcdedit /set {bootmgr} displayorder {*<ID GUID>*} /addfirst

*<ID GUID>* is the identifier for the specified Windows boot loader object and is discussed in the next section.

**Note:** A multiboot system with multiple installed operating systems has multiple instances of the Windows boot loader, each with its own identifier. The default Windows boot loader ({default}) can be set to any of these identifiers.

## Windows Boot Loader Settings

A BCD store has at least one instance, and optionally multiple instances, of the Windows boot loader. Each instance is represented by a separate BCD object and loads one of the installed versions of Windows with a configuration that is specified by the object's elements. Each Windows boot loader object has its own identifier, and the object's device and path settings must point to the correct partition and boot application.

The BCD-template for Windows Server 2008 has a single Windows boot loader object that has the following settings:

Windows Boot Loader

-------------------

identifier {9f25ee7a-e7b7-11db-94b5-f7e662935912}

device partition=C:

path \Windows\system32\winload.efi

description Microsoft Windows Server 2008

locale en-US

inherit {bootloadersettings}

osdevice partition=C:

systemroot \Windows

The identifier for this Windows boot loader is {9f25ee7a-e7b7-11db-94b5-f7e662935912}. You can use this GUID on your system or let BCDEdit generate a new GUID for you.

To simplify BCDEdit commands, you can specify one of the Windows boot loaders in the BCD system store as the default loader. You can then use the standard identifier—{default}—in place of the full GUID.

The following example specifies the Windows boot loader for EFI as the default boot loader, assuming that it uses the identifier GUID from BCD-template:

Bcdedit /default {9f25ee7a-e7b7-11db-94b5-f7e662935912}

### The Device and OSDevice Settings

The following two elements specify two key locations:

* **device** specifies the partition that contains the boot application.
* **osdevice** specifies the partition that contains the system root.

For the Windows boot loader for EFI, both elements are usually set to the Windows system partition drive letter. However, if BitLocker is enabled or a computer has multiple installed versions of Windows, **osdevice** and **device** might be set to different partitions.

BCD-template sets both elements to **c:**, which is the typical value. You can also explicitly set the **device** and **osdevice** values, as shown in the following example. The example also assumes that you have specified Windows boot loader for EFI as the default boot loader object:

Bcdedit /set {default} device partition=c:

Bcdedit /set {default} osdevice partition=c:

### The Path Setting

A Windows boot loader's **device** element specifies the volume on which the loader is located. The **path** element specifies where on that volume the boot loader is located. For UEFI systems, **path** must point to the Windows boot loader for EFI, whose path is \Windows\system32\winload.efi.

BCD-template has the correct value, which you can confirm by enumerating the values in the store. You can also explicitly set the **path** value, as shown in the following example:

Bcdedit /set {default} path \windows\system32\winload.efi

## Windows Memory Tester Settings

The Windows memory tester ({memdiag}) runs memory diagnostics at boot time. The BCD settings for the application's device and path elements must point to the correct application.

**Note**: Intel Itanium computers do not include a Windows memory tester boot application and do not require {memdiag} settings.

BCD-template for Windows Server 2008 has the following settings:

Windows Memory Tester

---------------------

identifier {memdiag}

device partition=\Device\HarddiskVolume1

path \boot\memtest.exe

description Windows Memory Diagnostic

The following sections discuss the **device** and **path** settings.

### The Device Setting

For UEFI systems, the Windows memory tester's device element must be set to the ESP drive letter. The following example assumes that the ESP is drive **s:**, as used in earlier examples:

Bcdedit /set {bootmgr} device partition=s: // ESP

### The Path Setting

The **path** element specifies where—on the volume specified by **device**—Windows Test Manager is located. For UEFI systems, **path** must point to the EFI version of the application—\efi\microsoft\boot\memtest.efi.

BCD-template has the correct path value, which you can confirm by enumerating the values in the store. You can also use BCDEdit to explicitly set the **path** value, as shown in the following example:

Bcdedit /set {memdiag} path \efi\microsoft\boot\memtest.efi

## BCD Settings for BIOS Systems

For comparison with UEFI settings, the following sections summarize the relevant BCD element settings for BIOS systems and provide the BCDEdit commands to configure a BCD system store for a BIOS system. You could, for example, use these BCDEdit commands to convert a BCD system store from a UEFI configuration to a BIOS configuration.

**Note:** This section assumes that the Windows system partition also contains the BIOS boot applications. In other words, **device** and **osdevice** are set to the same drive letter. On a system with BitLocker enabled or with multiple versions of Windows installed, **device** and **osdevice** might be set to different partitions.

### Windows Boot Manager Settings

For a BIOS system, Windows Boot Manager should have the following settings:

* **path** is set to \bootmgr.exe or nothing.
* **device** is set to the active partition.

The following example shows how to set Windows Boot Manager's **path** and **device** elements for a BIOS system:

Bcdedit /deletevalue {bootmgr} path

Bcdedit /set {bootmgr} device partition=c:

### Windows Boot Loader Settings

For a BIOS system, Windows boot loader objects should have the following settings:

* **device** should be set to the partition that contains the Windows boot loader boot application.
* **path** should be set to \windows\system32\winload.exe.
* **osdevice** should be set to the partition that contains the system root.

**device** and **osdevice** are usually set to the same partition, typically **c:**. However, on systems with BitLocker enabled or with multiple installed Windows versions, the two elements might be set to different partitions.

The following example shows how to set the Windows boot loader's **path**, **device**, and **osdevice** elements for a BIOS system. It assumes that **c:** contains the system root and the boot applications:

Bcdedit /set {default} path \windows\system32\winload.exe

Bcdedit /set (default} device partition=c:

Bcdedit /set {default} osdevice partition=c:

### Windows Resume Application Settings

For a BIOS system, the Windows resume application should have the following settings:

* **device** should be set to the partition that contains the Windows resume application.
* **path** should be set to \windows\system32\winresume.exe.
* **filedevice** should be set to the same partition as **device**.

The following example shows how to set the Windows resume application's **path**, **device**, and **filedevice** elements for a BIOS system. <ID GUID> represents the Windows resume application's identifier GUID, and the example assumes that **c:** contains the system root and the boot applications:

Bcdedit /set *<ID GUID>* device partition=c:

Bcdedit /set *<ID GUID>* path \windows\system32\winresume.exe

Bcdedit /set *<ID GUID>* filedevice partition=c:

### Windows Memory Tester Settings

For BIOS systems, Windows memory tester should have the following settings:

* **device** should be set to the partition that contains the Windows memory tester application, typically **c:**.
* **path** should be set to \boot\memtest.exe.

The following example shows how to set the Windows memory tester's **path** and **device** elements for a BIOS system. It assumes that **c:** contains the boot applications:

Bcdedit /set {bootmgr} device partition=c:

Bcdedit /set {memdiag} path \boot\memtest.exe

# Appendix B: Clearing Duplicate Firmware Objects in BCD and NVRAM

On some UEFI systems, the firmware creates NVRAM entries for local devices such as CD-ROM or hard drivers when the devices power up. When you run commands such as **bcdedit /set** or **bcdedit /enum**, BCDEdit opens the BCD system store and compares the firmware namespace objects in NVRAM with the corresponding objects in the store. If any firmware namespace objects are not represented in the BCD system store, BCDEdit creates appropriate objects and adds them to the store.

When BCDEdit closes the BCD system store, it compares the firmware namespace objects in the store with the objects in NVRAM. If any of the BCD objects are not represented in NVRAM, BCDEdit creates appropriate NVRAM entries.

If you run **bcdedit /import**, BCDEdit adds copies of all firmware namespace objects in the BCD system store to NVRAM. If you run bcdedit /import multiple times, the system's NVRAM might eventually contain multiple duplicate entries for devices such as the CD‑ROM and hard drives.

To check whether your system has duplicate NVRAM entries, run the following BCDEdit command, which enumerates the firmware namespace objects in the BCD system store:

Bcdedit /enum firmware

The following example is similar to the output from **bcdedit /enum firmware**:

Firmware Boot Manager

---------------------

identifier {fwbootmgr}

displayorder {bootmgr}

{93cee840-f524-11db-af62-aa767141e6b3}

{93cee841-f524-11db-af62-aa767141e6b3}

{93cee842-f524-11db-af62-aa767141e6b3}

{93cee844-f524-11db-af62-aa767141e6b3}

{93cee843-f524-11db-af62-aa767141e6b3}

timeout 2

Windows Boot Manager

--------------------

identifier {bootmgr}

device partition=\Device\HarddiskVolume1

path \EFI\Microsoft\Boot\bootmgfw.efi

description Windows Boot Manager

locale en-US

inherit {globalsettings}

default {current}

displayorder {current}

toolsdisplayorder {memdiag}

timeout 30

Firmware Application (101fffff)

-------------------------------

identifier {93cee840-f524-11db-af62-aa767141e6b3}

description Primary Master CDROM

Firmware Application (101fffff)

-------------------------------

identifier {93cee841-f524-11db-af62-aa767141e6b3}

description Harddisk 4

Firmware Application (101fffff)

-------------------------------

identifier {93cee842-f524-11db-af62-aa767141e6b3}

description Internal EFI Shell

Firmware Application (101fffff)

-------------------------------

identifier {93cee843-f524-11db-af62-aa767141e6b3}

description Floppy

Firmware Application (101fffff)

-------------------------------

identifier {93cee844-f524-11db-af62-aa767141e6b3}

description Acpi(PNP0A03,0)/Pci(1F|1)/Ata(Primary,Master)/CDROM(Entry1)

If you have used bcdedit /import more than once, NVRAM and the BCD system store might contain multiple entries for the same device. You might also find multiple firmware entries for the same device if you import a BCD store from a master computer to a target computer.

The following example is similar to how the bcdedit /enum output might appear for a system with both multiple entries:

Firmware Boot Manager

---------------------

identifier {fwbootmgr}

displayorder {bootmgr}

{93cee840-f524-11db-af62-aa767141e6b3}

{93cee841-f524-11db-af62-aa767141e6b3}

{93cee842-f524-11db-af62-aa767141e6b3}

{93cee844-f524-11db-af62-aa767141e6b3}

{93cee843-f524-11db-af62-aa767141e6b3}

{8b87c5a0-f2f2-11db-9717-f87ee6ea6002}

{8b87c5a1-f2f2-11db-9717-f87ee6ea6002}

{8b87c5a2-f2f2-11db-9717-f87ee6ea6002}

{8b87c5a3-f2f2-11db-9717-f87ee6ea6002}

{8b87c5a4-f2f2-11db-9717-f87ee6ea6002}

timeout 2

Firmware Application (101fffff)

-------------------------------

identifier {93cee840-f524-11db-af62-aa767141e6b3}

description Primary Master CDROM

Firmware Application (101fffff)

-------------------------------

identifier {8b87c5a0-f2f2-11db-9717-f87ee6ea6002}

description Primary Master CDROM

...

The BCD system store in this example has duplicate firmware namespace objects for its devices. One set of objects has identifiers starting with 93, and the duplicate set has identifiers that start with 8b. The example shows the details for one of the devices: the Primary Master CDROM.

You can use BCDEdit to remove multiple or duplicate entries from NVRAM and the BCD system store. The operation requires a series of BCDEdit commands, so you might find it useful to create a command script that uses the correct object identifiers for the multiple object entries that you want to remove.

To remove duplicate entries

1. Save a copy of the current BCD system store by running the following command:

bcdedit /export savebcd

Savebcd might be useful later for recovery.

2. Make a copy of Savebcd to use for the BCDEdit delete operations:

copy savebcd newbcd

3. Enumerate the firmware namespace objects in the system BCD store and save the output to a text file:

bcdedit /enum firmware > enumfw.txt

4. Open enumfw.txt in Notepad and delete everything except the list of firmware GUIDs.

5. Save the edited enumfw.txt to a new command file—removedups.cmd.

6. Insert the following BCDEdit command line in front of each identifier in removedups.cmd file:

Bcdedit /store newbcd /delete

This command deletes all the duplicate firmware IDs. For the example used in this procedure, removedups.cmd should look like the following sample:

bcdedit /store newbcd /delete {93cee840-f524-11db-af62-aa767141e6b3}

bcdedit /store newbcd /delete {93cee841-f524-11db-af62-aa767141e6b3}

bcdedit /store newbcd /delete {93cee842-f524-11db-af62-aa767141e6b3}

bcdedit /store newbcd /delete {93cee843-f524-11db-af62-aa767141e6b3}

bcdedit /store newbcd /delete {93cee844-f524-11db-af62-aa767141e6b3}

bcdedit /store newbcd /delete {8b87c5a0-f2f2-11db-9717-f87ee6ea6002}

bcdedit /store newbcd /delete {8b87c5a1-f2f2-11db-9717-f87ee6ea6002}

bcdedit /store newbcd /delete {8b87c5a2-f2f2-11db-9717-f87ee6ea6002}

bcdedit /store newbcd /delete {8b87c5a3-f2f2-11db-9717-f87ee6ea6002}

bcdedit /store newbcd /delete {8b87c5a4-f2f2-11db-9717-f87ee6ea6002}

**Note:** On systems where the EFI firmware initializes NVRAM entries for local devices, you can delete all firmware namespace object GUID entries, but do not delete the {bootmgr} entry.

7. Add the following command to the end of removedups.cmd, and then save the file:

bcdedit /import newbcd /clean

The /import /clean option deletes all NVRAM entries and then re-initializes NVRAM based on the firmware namespace objects in the specified BCD store—newbcd.

8. Run removedups.cmd to remove the duplicate entries from the newbcd store, reinitialize NVRAM, and import newbcd into the BCD system store.

9. Reboot the system.

During the reboot, the EFI firmware reinitializes the NVRAM with the firmware object GUIDs that correspond to the attached devices.

10. Use bcdedit /enum **firmware** to verify that all duplicate firmware namespace objects have been removed.