



Native VHD Boot: A Walkthrough of Common Scenarios

Microsoft Corporation

Published: November 2010

Author: Michael Waterman

Abstract

This guide describes the functionality and usability of native VHD boot, which is a feature in Windows® 7 and Windows Server® 2008 R2. Native VHD boot enables IT professionals to use a single disk format to service both virtual machines and physical client computers. This guide presents three scenarios of varying levels of complexity, and each scenario includes a series of step-by-step instructions. Collectively, these scenarios introduce you to the techniques and supporting deployment tools so that you will be able to apply native VHD boot in a useful and productive manner within your organization.

Microsoft

Copyright information

This document is provided “as-is”. Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

© 2010 Microsoft Corporation. All rights reserved.

Microsoft, Active Directory, BitLocker, Hyper-V, Windows, Windows Server, and Windows Vista are trademarks of the Microsoft group of companies.

All other trademarks are property of their respective owners.

Contents

Native VHD Boot: A Walkthrough of Common Scenarios	5
Introduction to native VHD boot	5
What is native VHD boot?	5
Goals	5
Available VHD types	6
Limitations	6
Example scenarios	7
Tools	7
Scenario 1: Configuring Native VHD Boot on a Single Computer.....	8
Scenario prerequisites	8
Step 1: Install the Windows AIK.....	9
Step 2: Create the VHD	9
Step 3: Apply the Windows image to the VHD.....	10
Step 4: Change the boot configuration data	11
Step 5: Detach the VHD.....	13
Scenario 2: Deploying Native Boot VHDs Using Windows Deployment Services	14
Scenario prerequisites	14
Step 1: Configure Windows Deployment Services	15
Step 2: Add a boot image.....	15
Step 3: Create the VHD	17
Step 4: Apply a Windows image to the VHD.....	17
Step 5: Add the VHD to Windows Deployment Services	19
Step 6: Create the answer file.....	20
Step 7: Associate the answer file	22
Scenario 3: Performing an Advanced Deployment of Native Boot VHDs	23
Scenario prerequisites	24
Step 1: Create a customized Windows PE image	24
Step 2: Create bootable media	25
Step 3: Copy the Windows image to the USB flash drive.....	27
Step 4: Boot to the USB flash drive from the client computer.....	27
Step 5: Partition the disk on the client computer.....	27
Step 6: Create the VHDs.....	29
Step 7: Apply the WIM files to the VHDs.....	31
Step 8: Configure the BCD store.....	31
Step 9: Add the Windows PE image to the client computer	34
Step 10: Create a RAM disk.....	34
Step 11: Create the BCD entry for the Windows PE image.....	35
Step 12: Prevent the VHD from expanding to its full size	36
Step 13: Configure the operating systems in the VHDs.....	39
Step 14: Create and configure differencing VHDs	39
Conclusion.....	40

Native VHD Boot: A Walkthrough of Common Scenarios

This guide describes the functionality and usability of native VHD boot, one of the many features of Windows® 7 and Windows Server® 2008 R2. Native VHD boot enables IT professionals to use a single disk format to service both virtual machines and physical client computers.

This guide presents three scenarios of varying levels of complexity and each scenario includes a series of step-by-step instructions. Collectively, these scenarios introduce you to the techniques and supporting deployment tools so that you will be able to apply native VHD boot in a useful and productive manner within your organization.

For a Web version of this document, see [Native VHD Boot: A Walkthrough of Common Scenarios](http://go.microsoft.com/fwlink/?LinkId=205691) in the Windows 7 Technical Library (<http://go.microsoft.com/fwlink/?LinkId=205691>).

Introduction to native VHD boot

What is native VHD boot?

Native VHD boot enables a physical computer to be booted directly from a virtual hard disk (VHD) with no requirement for virtualization software (such as Microsoft® Virtual PC or Hyper-V™). Native VHD boot is made possible with the addition of a mini-port driver that is responsible for all input/output (IO) operations to and from the disk. Earlier versions of Windows (such as Windows Vista®, Windows XP, and Windows 2000) do not support native VHD boot, and therefore, you cannot use them to implement the functionality described in this document.

With the introduction of native VHD boot, it is possible to achieve a single image deployment strategy for both physical and virtual machines. As long as the operating system that resides on the VHD is generalized and hosts Windows 7 Enterprise, Windows 7 Ultimate, or Windows Server 2008 R2 (all editions), the VHD can be used in both environments.

Goals

The main goals for introducing native VHD boot are to:

- Deploy a single image type in virtual and physical infrastructures.
- Manage virtual and physical infrastructures by using a single set of tools.
- Deploy operating systems to new computers in a rapid manner.
- Enable computers to boot from more than one operating system (multiboot or dual-boot) without requiring separate partitions.
- Create a system that has the flexibility of a virtual environment (such as running the Hyper-V role) but uses physical hardware.

Available VHD types

There are three types of VHDs that you can create:

- **Fixed-size VHD.** A fixed-size VHD is configured with a disk size that is allocated in full when the VHD is created. For example, a fixed-size VHD with a configured size of a 100 GB allocates the entire 100 GB at the moment the VHD is created. This configuration is best for performance and guarantees available disk space.
- **Dynamically expanding VHD.** A dynamically expanding VHD is created as a small file with an initial size of around 250 KB. As data is written to the VHD, it expands until it reaches its maximum size (which is configured when the file is created). Note that as a dynamically expanding VHD expands, the host volume could run out of space and cause the write operations to fail. Therefore, you should consider using fixed-size VHDs or configuring the VHD as discussed in Step 12 of [Scenario 3: Performing an Advanced Deployment of Native Boot VHDs](#).
- **Differencing VHD.** A differencing VHD is similar to a dynamically expanding VHD in respect to how it grows as data is written to it. However, differencing VHDs have a parent/child relationship between two disks. The child disk (the differencing VHD) holds the differences and grows when data is written to it. The parent disk remains read-only and it cannot be modified, which prevents data corruption. The parent file can be a fixed-size, dynamically expanding, or another differencing VHD. The latter will create a differencing disk chain.



Note

The maximum size of a VHD is 2040 GB (2 TB).

Limitations

The following are limitations with native VHD boot in Windows 7 and Windows Server 2008 R2:

- **BitLocker.** Although you can place a VHD on a partition that is enabled with BitLocker™ Drive Encryption, it is not possible to boot from that VHD. Enabling BitLocker on the operating system within the VHD is also not a supported configuration.
- **Manual image creation.** Creating VHD images is still a manual process. However, you can configure VHDs to align with the existing image engineering methods in your organization (creating a standard image, editing its configuration, and so on). For example, you can apply an image to a VHD with the same method as you would apply it to a hard disk drive.
- **No setup configuration during deployment.** There is no setup support while deploying a Windows image to a VHD. That is, you can only apply an image directly to a VHD; you cannot use setup to configure the operating system to be installed on a specific partition.
- **Location of differencing VHDs.** When using differencing VHDs, all VHD files (the parent files and the differencing VHD itself) must be stored on the same partition.
- **Supported editions.** You can only use VHDs that contain Windows 7 Enterprise or Windows 7 Ultimate edition for native VHD boot. However, all editions of Windows Server 2008 R2 are supported.
- **Hibernation.** You cannot hibernate an operating system that runs on a VHD. However, sleep functionality is available.
- **Windows Experience Index.** The Windows Experience Index cannot be calculated for VHDs that are configured for native boot because of the way disk performance is measured. This

means some features, such as the Aero® desktop experience, will not be automatically configured based on the index of the VHD's operating system. However, you can use the Winsat command-line tool to manually configure these indexes, except for the index for the disk data transfer rate.

Example scenarios

To help you gain insight into what can be achieved by using native VHD boot, the following sections detail three step-by-step scenarios. The scenarios range from a simple setup to an advanced deployment that utilizes various deployment tools.

- [Scenario 1: Configuring Native VHD Boot on a Single Computer](#). In this scenario, you will use the Disk Management MMC snap-in on an existing installation of Windows 7 to create a VHD, apply Windows 7 Enterprise to it, and then configure the boot configuration data (BCD) store so it can boot by using the VHD.
- [Scenario 2: Deploying Native Boot VHDs Using Windows Deployment Services](#). In this scenario, you will deploy a custom Windows 7 VHD by using Windows Deployment Services in Windows Server 2008 R2.
- [Scenario 3: Performing an Advanced Deployment of Native Boot VHDs](#). In this scenario, a locally installed version of Windows is not used—only VHDs are used. You will configure native VHD boot with two operating systems and have a local copy of the Windows Preinstallation Environment (Windows PE) for recovery purposes.

Scenario 2 is the only scenario that requires you to have a computer running Windows Server 2008 R2. You can perform all of the steps in Scenarios 1 and 3 on a computer running Windows 7 Enterprise or Windows 7 Ultimate.

Tools

The following tools are used throughout these scenarios:

- **BCDBoot**: A command-line tool for manually creating the boot configuration data (BCD) store. The BCD store contains the boot configuration parameters, and it is the successor to the Boot.ini.
- **BCDEdit**: A command-line tool for manually editing the BCD store.
- **Disk Management MMC snap-in**: A graphical tool for editing disk configurations.
- **DiskPart**: A command-line tool for editing disk configurations.
- **Windows AIK**: A toolkit for creating and maintaining images. Specifically, you will use the following tools:
 - **ImageX**: A command-line tool that is used to capture, create, modify, and apply images.
 - **Deployment Image Servicing and Management (DISM)**: A command-line tool that is used to apply updates, drivers, and language packs to an image.
- **Windows Deployment Services**: A server role in Windows Server 2008 R2 that deploys images over the network.
- **Windows PE**: A version of Windows with a very small footprint that is designed especially for deployment purposes.

Scenario 1: Configuring Native VHD Boot on a Single Computer

The goal of this scenario is to gain familiarity with the steps that are required to enable native VHD boot by using an existing installation of Windows 7. Alternatively, you can use Windows Server 2008 R2 in this scenario as the steps are the same for both operating systems.

Note

Only Windows 7 Enterprise or Windows 7 Ultimate can be used with native VHD boot.

In this scenario, you will create a VHD and apply an image to the new VHD by using tools from the Windows Automated Installation Kit (Windows AIK). Then, you will alter the BCD store so that you can boot to the operating system within the VHD. To summarize, you will take the following steps:

- Install the Windows AIK
- Create a VHD through the user interface
- Mount the VHD and create a partition on it
- Apply Windows 7 to the VHD
- Configure the BCD store

To illustrate, Figure 1 represents the disk layout and placement of the VHD disk for this scenario.

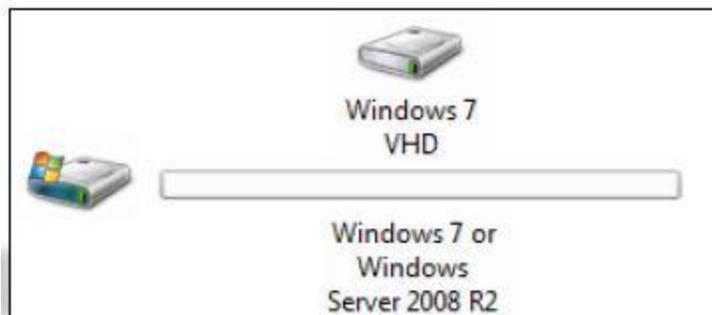


Figure 1 Disk and VHD configuration for this scenario

Scenario prerequisites

Before you proceed, ensure you have the following available:

- A computer running Windows 7 Enterprise, Windows 7 Ultimate, or Windows Server 2008 R2
- The installation media for Windows 7 Enterprise, Windows 7 Ultimate, or Windows Server 2008 R2

Note

If you have Windows 7 or Windows Server 2008 R2 in the .iso format, extract the .iso files as flat files or burn them to a DVD

- Media that contains the [Windows AIK](http://go.microsoft.com/fwlink/?LinkID=154426) installation files (<http://go.microsoft.com/fwlink/?LinkID=154426>)

Step 1: Install the Windows AIK

Before you create a VHD file, you need to install the Windows AIK on the local computer. This kit is full of utilities for deploying and maintaining images (.wim files and .vhd files). To install the kit, use the following procedure.

▶ To install the Windows AIK

1. Browse to the location where you stored the Windows AIK media and double-click **Startcd.exe**.
2. In the **User Account Control** dialog box, click **Yes**.
3. In the left pane of the **Welcome to the Windows Automated Installation Kit** page, click **Windows AIK setup**.
4. On the **Windows Automated Installation Kit Setup Wizard** page, click **Next**.
5. On the **License Terms** page, select **I agree**, and then click **Next**.
6. Select an appropriate installation folder or leave the default setting, and then click **Next**.
7. On the **Confirm Installation** page, click **Next** and wait for the installation to finish.
8. After the installation has finished, click **Close** to exit the Windows AIK setup.

When the installation is successful, you have installed all the tools that you will need. The primary tool you will use in this scenario is ImageX, which you will use to apply the image to the VHD. But before you can apply the image, you need to create the VHD as explained in the next step.

Step 2: Create the VHD

To store the VHDs that you will create, this example scenario uses a local folder, **C:\VHD Files**. However, you can use any name and location that you desire. Instead of storing VHDs in a subfolder, you could store them in the root of the drive. Use the following procedure to create a VHD by using the user interface.

▶ To create a VHD by using Disk Management

1. On the start menu, right-click **Computer**, and then click **Manage**.
2. In the **User Account Control** dialog box, click **Yes**.
3. In the left pane of the **Computer Management** console, click **Disk Management**.
4. In the right pane, click **More Actions**, and then click **Create VHD**.
5. On the **Create and attach Virtual Hard Disk** page, specify the following settings, and then click **OK**:
 - **Location:** C:\VHD Files\WIN7.vhd
 - **Virtual hard disk size:** 50 GB
 - **Virtual hard disk format:** Dynamically expanding

You should create a dynamically expanding VHD only when you want to save on the amount of initial disk space that you use. With this option, only a small VHD is created

initially, and it will grow dynamically as you add data to the VHD. However, using this type of file has an impact on performance because not all blocks are consistently written next to each other on the physical disk. For servers or workstations where performance is of concern, you should consider creating a fixed-size VHD.

 **Note**

If this is the first time that you have created a VHD, an indicator appears in the notification area to inform you that a driver is being installed.

6. The next thing you need to do is initialize the disk. In **Disk Management**, right-click the 50 GB disk that you just created (notice that the VHD is distinguished with a light blue icon), and then select **Initialize Disk**.
7. On the **Initialize Disk** page, ensure that the VHD you just created is selected and that **MBR** is the selected partition style. Click **OK**.
8. After the disk is initialized, you can create a partition on it. Right-click the unallocated space of the disk, and then click **New Simple Volume**.
9. In the **Welcome to the New Simple Volume Wizard**, click **Next**.
10. On the **Specify Volume Size** page, leave the default values (you will use the entire size of the disk as the system volume), and then click **Next**.
11. In the **Assign Drive Letter or Path** list, choose drive letter **V:**, and then click **Next**.
12. Under **Format this volume with the following settings**, specify the following options, and then click **Next**:
 - **File system:** NTFS
 - **Allocation unit size:** Default
 - **Volume label:** system
13. To create the partition, on the **Completing the New Simple Volume Wizard** page, click **Finish**.

Wait a few second for the partition to be created. Do not close the **Computer Management** console yet because you will need it later in this scenario.

Step 3: Apply the Windows image to the VHD

One of the limitations of native VHD boot is that it is only available with Windows 7 Enterprise, Windows 7 Ultimate, or Windows Server 2008 R2 (all editions). Windows image (.wim) files often contain more than one edition of Windows, so you first need to determine the edition (and its “index number”) that you want to use. This is especially true for the Install.wim file that is available on the installation media. In Step 1, you installed the Windows AIK, which contains the tools to determine the index number. In the following procedure, you will search for the index number (in this example Windows 7 Enterprise), and then you will apply Windows 7 Enterprise to the VHD. The tool you will use is new to Windows 7; it is called Deployment Image Servicing and Management (DISM). To perform these steps, use the following procedure.

▶ To apply a Windows image to the VHD

1. Click **Start**, point to **All Programs**, and then click **Microsoft Windows AIK**.
2. Right-click **Deployment Tools Command Prompt**, and then click **Run as administrator**.
3. In the **User Account Control** dialog box, click **Yes**.
4. Type the following DISM command to view information about the editions of Windows within the .wim file.

```
dism /Get-Wiminfo /Wimfile:"C:\Sources\Windows 7  
Enterprise\Sources\install.wim"
```

You will see that there is only one edition of Windows 7 Enterprise available in the .wim file. Remember the index number because you will use it in the next step to apply the data to the VHD.

Note

If you want more information about a specific edition of Windows 7 in the .wim, type `/Index:<Number>` at the end of the previous command.

5. Now that you have the index number, you can apply the data from the .wim file to the VHD. To do this, type the following ImageX command:

```
imagex /apply "C:\Sources\Windows 7  
Enterprise\Sources\install.wim" 1 v:
```

Wait for the command to finish. The whole process can take 10 minutes or longer depending on the performance of your computer.

6. Type `exit` in the Command Prompt window to close it.

Step 4: Change the boot configuration data

You have almost completed the steps that are required for booting to a VHD. The last task is to change the boot configuration data (BCD) store, which is essentially the successor to the Boot.ini file in previous versions of Windows. The default tool for editing the BCD store is the BCDEdit command-line tool.

Next, you will use BCDEdit to add a boot entry to the store. This will enable you to select the VHD when you start the computer. To do this, use the steps in the following procedure.

▶ To change the BCD

1. Open an elevated Command Prompt window (click **Start**, right-click **Command Prompt**, and click **Run as administrator**).
2. In the **User Account Control** dialog box, click **Yes**.
3. To take a first look at the BCD store, you will use BCDEdit to view (enumerate) the current Windows Boot Loader entries. To do this, type `bcdedit /enum`. The output will look similar to Figure 2:

```

C:\Windows\system32>bcdedit /enum

Windows Boot Manager
-----
identifier           {bootmgr}
device               partition=\Device\HarddiskVolume1
description          Windows Boot Manager
locale              en-US
inherit              {globalsettings}
default              {current}
resumeobject        {d89f3fa0-6416-11de-b070-b08b383b8f33}
displayorder        {current}
toolsdisplayorder   {mendiag}
timeout             30

Windows Boot Loader
-----
identifier           {current}
device               partition=C:
path                 \Windows\system32\winload.exe
description          Windows 7
locale              en-US
inherit              {bootloadersettings}
recoverysequence    {d89f3fa2-6416-11de-b070-b08b383b8f33}
recoveryenabled     Yes
osdevice             partition=C:
systemroot           \Windows
resumeobject        {d89f3fa0-6416-11de-b070-b08b383b8f33}
nx                  OptIn

```

Figure 2 Default BCD store

4. You could create a new entry manually, but it is easier to copy the settings from the current boot entry and use that as a starting point. To copy the current entry, type the following command:

```
bcdedit /copy {current} /d "Windows 7 VHD Boot"
```

The output will generate a GUID, which is associated with the new boot entry. Take note of this GUID because you will use it to change settings in the next step.

5. You need to change two crucial settings in the boot loader entry. These settings tell the Windows Boot Manager to look for the VHD during initialization. Type the following commands with the appropriate GUID to alter the **device** and **osdevice** settings of the new boot entry.

```
bcdedit /set {d89f3fa4-6416-11de-b070-b08b383b8f33} device
vhd="[locate]\VHD Files\WIN7.vhd"
```

```
bcdedit /set {d89f3fa4-6416-11de-b070-b08b383b8f33} osdevice
vhd="[locate]\VHD Files\WIN7.vhd"
```

The setting **[locate]** instructs Windows Boot Manager to locate the VHD without requiring you to specify a drive letter.

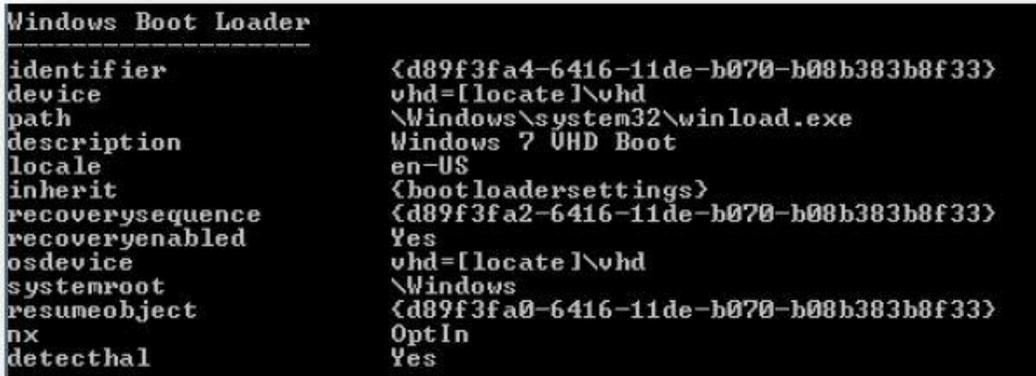
6. Because you will be using an installation of Windows 7 in the VHD, you also need Windows Boot Manager to detect the hardware abstraction layer (HAL) during setup. The HAL is basically an interface between the software and hardware of the computer. To enable a setting to make this possible, type the following command:

```
bcdedit /set {d89f3fa4-6416-11de-b070-b08b383b8f33} detecthal
Yes
```

 **Note**

If you want to set the VHD boot loader entry as the default boot configuration, type the following command: `Bcdedit /default {d89f3fa4-6416-11de-b070-b08b383b8f33}`

7. If you run the `bcdedit /enum` command again, you will see the new entry in the output, similar to Figure 3.



```
Windows Boot Loader
-----
identifier                {d89f3fa4-6416-11de-b070-b08b383b8f33}
device                    vhd=[locate]\vhd
path                      \Windows\system32\winload.exe
description                Windows 7 VHD Boot
locale                    en-US
inherit                   <bootloadersettings>
recoverysequence         {d89f3fa2-6416-11de-b070-b08b383b8f33}
recoveryenabled           Yes
osdevice                  vhd=[locate]\vhd
systemroot                \Windows
resumeobject              {d89f3fa0-6416-11de-b070-b08b383b8f33}
nx                        Opt In
detecthal                 Yes
```

Figure 3 The new boot entry that enables native VHD boot

8. You have now created a boot entry for the VHD that you created earlier. Type `exit` in the Command Prompt window to close it.

Step 5: Detach the VHD

The final step is to detach the VHD. This prevents data corruption, which can occur if you restart the computer while the VHD is attached. If the VHD remains attached, files could remain open or locked, and the data on the VHD would eventually be damaged. Just like attaching a VHD, you can detach the VHD by using the **Disk Management** user interface. Assuming that the **Computer Management** console is still open, use the following procedure to detach the VHD.

To detach the VHD

1. In **Disk Management**, right-click the disk that represents the attached VHD, and then click **Detach VHD**. (Remember that the disk is displayed with a light blue icon.)
2. On the **Detach Virtual Hard Disk** page, ensure that the **Delete the virtual hard disk file after removing the disk** check box is not selected, and then click **OK**.
3. Close the **Computer Management** console.

You have now completed all the steps to create a VHD, apply an operating system to it, and configure the BCD store so that users can select the VHD when they start the computer. All that remains is to restart the computer and start the installation.

Scenario 2: Deploying Native Boot VHDs Using Windows Deployment Services

In Scenario 1, we walked through how to create a VHD, apply an operating system to it, and edit the boot configuration so you can boot into the VHD when you start the computer. In this scenario, we will explore how to deploy VHDs by using Windows Deployment Services, a server role in Windows Server 2008 R2.

If you are already familiar with Windows Deployment Services, you will notice that there is little difference in how you deploy a .vhd file compared to a Windows image (.wim) file. The main difference is how you manage the VHD with Windows Deployment Services because it is currently not possible to use the Windows Deployment Services MMC snap-in to add VHD files to the server. Instead, you will use the WDSUtil command-line tool. In this scenario, you will add a VHD to the Windows Deployment Services server, assign an answer file, and then deploy the VHD to a client computer. To summarize, you will take the following steps:

- Configure Windows Deployment Services on a server, which includes adding the appropriate “boot image”
- Create and configure a VHD, similar to the steps you completed in Scenario 1, and then add it to the server
- Create an answer file and associate it with the VHD

When you are finished with this scenario, the client computer will have a disk layout with two partitions as shown in Figure 4: one that is the system partition where the boot files are stored, and one that is a data partition where the VHD is stored.



Figure 4 Disk layout of the client computer

Scenario prerequisites

Before you proceed, ensure you have the following available:

- A computer on which to install the VHD.
- A domain controller running Windows Server 2008 or Windows 2008 R2
- A DHCP server and a DNS server on the network
- Windows Deployment Services installed (but not configured) on a computer that is joined to a domain. This scenario assumes that the server has the following configuration:

- Windows Server 2008 R2 is installed on drive C.
- Drive D is set up as a separate disk (not a partition). You will use this disk to store data.
- The Windows Automated Installation Kit (Windows AIK) is installed. If not, see Step 1 of [Scenario 1: Configuring Native VHD Boot on a Single Computer](#).
- The installation files from Windows 7 Enterprise are copied to the following folder:
D:\Sources\Windows 7 Enterprise.

Step 1: Configure Windows Deployment Services

The first step is to complete the setup wizard to configure Windows Deployment Services. To do so, use the following procedure:

▶ To configure Windows Deployment Services

1. Click **Start**, click **Administrative Tools**, and then click **Windows Deployment Services**. This will open the Windows Deployment Services MMC snap-in.
2. In the MMC snap-in, click **Servers**. Windows Deployment Services will try to add any Windows Deployment Services servers that are already installed (if applicable). When your local server name is displayed, select it.
3. In the right pane, you will see a message that says the server needs to be configured. Right-click the server name, and then click **Configure**.
4. Read through the requirements on the **Before you begin** page, and then click **Next**.
5. The data for the Windows Deployment Services server (in our case the VHD and the answer files) need to be stored locally on the server. You configure this location on the **Remote Installation Folder Location** page. Specify to store the data at **D:\RemoteInstall**, and then click **Next**.
6. Before you bring your Windows Deployment Services server into production, you need to configure how the server will respond to clients that request service. On the **PXE Server Initial settings** page, you can select one of three choices. Choose **Respond to all client computers (known and unknown)** because the other options are more advanced and out of scope of this document.
7. Click **Next** to apply the initial settings.
8. On the final page, clear the check box next to **Add images to the server now**. There is no need to add a .wim file at this time because you will be deploying a VHD file instead. Click **Finish** to complete the setup.

Step 2: Add a boot image

Besides deploying a VHD, you need a way to boot the client computer to then install the VHD. Adding a “boot image” is a wizard-driven task in Windows Deployment Services. Boot images are files that contain the Windows Preinstallation Environment (Windows PE), a minimal version of Windows that is specially designed for deployment purposes. After you configure Windows Deployment Services (Step 1), you will see several folders on the left pane of the Windows Deployment Services MMC snap-in as shown in Figure 5.

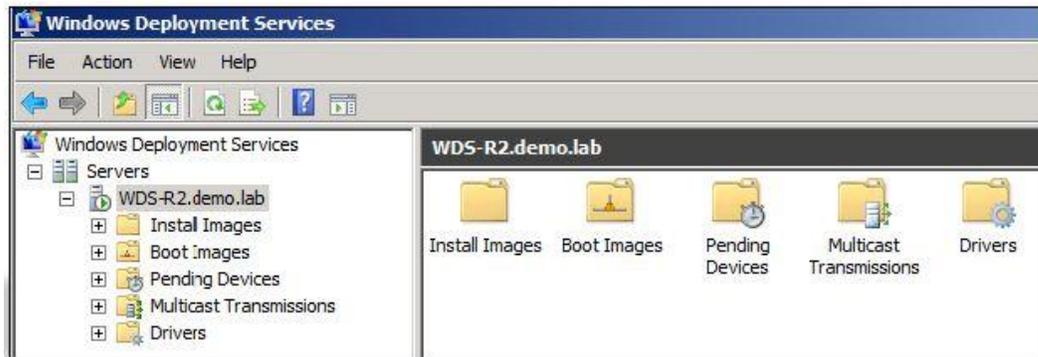


Figure 5 Windows Deployment Services folders

Following is an overview of these folders. The first two folders are those that are important to this scenario:

- **Install Images:** This folder contains the operating system images (.wim and .vhd files) that are available to deploy to client computers.
- **Boot Images:** This folder lists the boot images, which contain Windows PE.

The following three folders are out of scope for this scenario, but they are listed here in case you are interested in further configuring your server.

- **Pending Devices:** Depending on the server's configuration, client computers that request service appear in this folder. For more information, see [Create Pending Devices](http://go.microsoft.com/fwlink/?LinkId=205103) (<http://go.microsoft.com/fwlink/?LinkId=205103>)
- **Multicast Transmissions:** You can use this folder to configure and maintain multicast transmissions (where an image is transmitted to many clients at one time). For more information, see [Performing Multicast Deployments](http://go.microsoft.com/fwlink/?LinkId=205104) (<http://go.microsoft.com/fwlink/?LinkId=205104>).
- **Drivers:** In Windows Server 2008 R2, you can deploy drivers along with the operating system. You use this folder to configure and maintain drivers. For more information, see [Managing and Deploying Driver Packages](http://go.microsoft.com/fwlink/?LinkId=205105) (<http://go.microsoft.com/fwlink/?LinkId=205105>).

In the following procedure, you use the **Boot Images** folder to add a boot image to the server. Later, you will use the boot image when you deploy the VHD to the client computer.

▶ To add a boot image

1. Right-click the **Boot Images** folder, and then click **Add Boot Image**.
2. When you purchase Windows 7 or Windows Server 2008 R2, two files are included in the Sources folder of the installation media: Boot.wim and Install.wim. Boot.wim is a default version of Windows PE that you will use in this example, and you will use Install.wim later in this scenario. On the **Image File** page, click **Browse**, navigate to Boot.wim, and then click **OK**. Click **Next** to proceed.
3. On the **Image Metadata** page, leave the default text or alter it as appropriate, and then click **Next**.

4. On the **Summary** page, review your settings, and then click **Next**. The boot image is saved to the **RemoteInstall** folder and configured for use.
5. After this process completes, click **Finish**.

Step 3: Create the VHD

In Scenario 1, you used the user interface (Disk Management) to create a VHD. In this scenario, you will perform the same steps, but you will use the command line. Knowing both methods gives you the flexibility of choosing the way that best fits you. The command-line equivalent of Disk Management is DiskPart.

Alternatively, you could use VHDs that you created by using Hyper-V. Hyper-V is a server role in Windows Server 2008 R2 that you can use to create VHDs and apply operating systems to them. After generalizing the VHDs, you can then use them in virtual environments or for deployment through technologies like Windows Deployment Services.



Note

You cannot use VHD files that you created with Virtual PC 2007.

To create a VHD by using the command-line, use the following procedure.

▶ To create a VHD by using DiskPart

1. To start DiskPart, open an elevated Command Prompt window (click **Start**, right-click **Command Prompt**, and click **Run as administrator**), and then type: `diskpart`
2. Type the following commands. These commands create a dynamically expanding VHD with a maximum size of 50 GB, save it as **D:\Sources\VHD Files\WIN7WDS.vhd**, mount it as drive V, and prepare the file system.

```
create vdisk file="D:\Sources\VHD Files\WIN7WDS.VHD"  
type=expandable maximum=50000  
select vdisk file="D:\Sources\VHD Files\WIN7WDS.VHD"  
attach vdisk  
Create partition primary  
Assign letter=v  
format fs=ntfs quick  
exit
```

Step 4: Apply a Windows image to the VHD

In Scenario 1, you applied a Windows image (.wim) file to a VHD with the ImageX command-line tool (Step 3). You will perform these same operations in this step.



Note

You can use ImageX on computers running Windows 7 or Windows Server 2008 R2 if you have the Windows AIK installed.

To apply the image to the VHD, use the following procedure.

▶ To apply the Windows image to the VHD

1. Click **Start**, point to **All Programs**, and then click **Microsoft Windows AIK**.
2. Right-click **Deployment Tools Command Prompt**, and then click **Run as administrator**.
3. In the **User Account Control** dialog box, click Yes.
4. Type the following DISM command to view the information about the editions of Windows in the .wim file.

```
dism /Get-Wiminfo /Wimfile:"D:\Sources\Windows 7  
Enterprise\Sources\install.wim"
```

There is only one edition of Windows 7 Enterprise available in the .wim file, so you will use an index of **1** in the next step.



Note

If you want more information about a specific edition of Windows 7 in the .wim, type `/Index:<Number>` at the end of the previous command.

5. Now that you have the index number, you can apply the data from the .wim file to the VHD using the following command:

```
imagex /apply:"D:\Sources\Windows 7  
Enterprise\Sources\install.wim" 1 v:
```

Wait for the command to finish. The whole process can take 10 minutes or longer depending on the performance of your computer.

6. You need to detach the VHD before you configure it with Windows Deployment Services. Instead of using Disk Management as you did in Scenario 1, you will use DiskPart. To detach the VHD by using Diskpart, type the following:

```
diskpart  
  
select vdisk file="D:\Sources\VHD Files\WIN7WDS.VHD"  
  
detach vdisk  
  
exit
```

Step 5: Add the VHD to Windows Deployment Services

In Step 2 of this scenario, you saw that after initializing the Windows Deployment Services server, several default folders were created. One of these folders is the **Install Images** folder, which is where VHD files that you add will be displayed. Before you can add a VHD to the **Install Images** folder, you need to create an “image group” within it. This creates a more structured way of maintaining your VHDs.

At this point, you could open the Windows Deployment Services MMC snap-in and use the user interface to add an image group (right-click the **Install Images** folder and click **Add Image Group**). However, because you have been using command-line tools, this is a good time to introduce WDSUtil. WDSUtil is more powerful than the MMC snap-in, and it exposes all the features of Windows Deployment Services. To create the image group and add the VHD to it, use the following procedure.

► To add the VHD to Windows Deployment Services

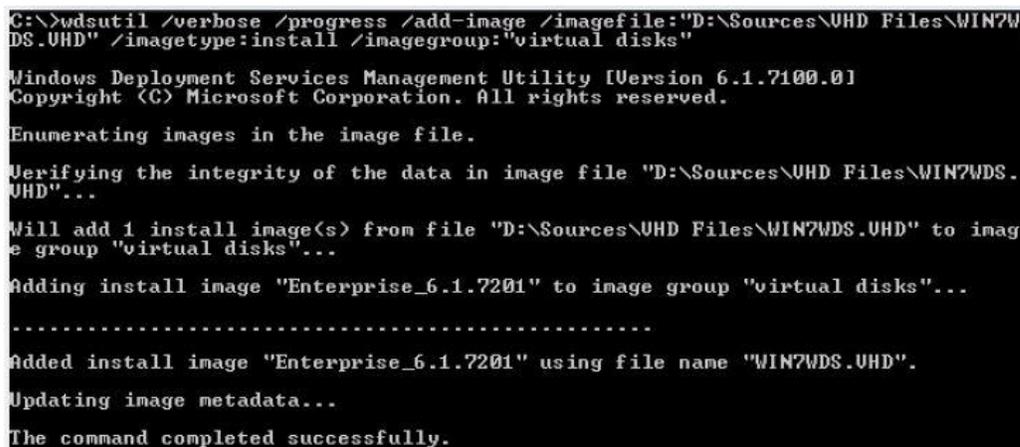
1. Type the following command to create an image group named **Virtual Disks**. The **/verbose** and **/progress** options display extra information during the copy process.

```
WDSUtil /add-imagegroup /imagegroup:"Virtual Disks"
```

2. Use the following command to add the VHD to the image group you just created.

```
Wdsutil /verbose /progress /add-image  
/imagefile:"d:\sources\vhd files\win7wds.vhd"  
/imagetype:install /imagegroup:"Virtual Disks"
```

This process will take some time; let it finish before you continue with the next step. The output will look similar to Figure 6.



```
C:\>wdsutil /verbose /progress /add-image /imagefile:"D:\Sources\UHD Files\WIN7WDS.UHD" /imagetype:install /imagegroup:"virtual disks"  
Windows Deployment Services Management Utility [Version 6.1.7100.0]  
Copyright (C) Microsoft Corporation. All rights reserved.  
Enumerating images in the image file.  
Verifying the integrity of the data in image file "D:\Sources\UHD Files\WIN7WDS.UHD"...  
Will add 1 install image(s) from file "D:\Sources\UHD Files\WIN7WDS.UHD" to image group "virtual disks"...  
Adding install image "Enterprise_6.1.7201" to image group "virtual disks"...  
.....  
Added install image "Enterprise_6.1.7201" using file name "WIN7WDS.UHD".  
Updating image metadata...  
The command completed successfully.
```

Figure 6 WDSUtil output while adding the VHD image

Step 6: Create the answer file

If you started a computer and booted to the network at this time, you would probably see the VHD that you added in the previous step. However, deploying VHDs by using the user interface is not a supported configuration. To deploy a VHD (particularly when you are deploying to more than one computer), you should create an answer file. An answer file is a text file that scripts the answers for a series of user interface dialog boxes, and it is commonly called Unattend.xml.

To create an answer file by using Windows System Image Manager (Windows SIM) from the Windows AIK, use the following procedure.

► To create the answer file

1. To open Windows SIM, click Start, click **All Programs**, click **Microsoft Windows AIK**, and then click **Windows System Image Manager**.
2. In the **Windows System Image Manager** window, right-click **Select a Windows image or catalog file**, and then click **Select Windows Image**.
3. You will notice that you cannot select the VHD file from the **Select a Windows Image** dialog box that appears. You do not want to open a VHD at this point, but instead you want to select a .wim image so that you can use the data within it. Browse to the location where you stored the Windows 7 media (**D:\Sources\Windows 7 Enterprise\Sources**), and select the catalog file named **Install_windows 7 ENTERPRISE.clg**. This catalog file contains the state of the settings and packages in the .wim file; the actual file is too large to work with directly.



Note

If an error message appears that indicates the catalog file is out of date, select the Install.wim file instead and a new catalog will be generated for you.

4. Now that you can use the data of the image, you can create the answer file. Right-click **Create or open an answer file**, and then click **New Answer File**.

An empty answer file is created with seven components added by default. These components represent the seven stages that a setup can go through. However, not all of the components are applicable at all deployment stages. In this scenario, you will focus on the **Windows PE** component.

5. In the left pane of Windows System Image Manager, expand **Components**, scroll down to **Microsoft-Windows-Setup**, and expand the node. Right-click **WindowsDeploymentServices**, and then click **Add Setting to Pass 1 WindowsPE**. This exposes all the options for the Windows PE component in the answer file. Before the settings are actually written to the file, you have to assign them a value. Use the following tables to create the answer file.

Windows International Core WinPE

Property	Setting and Value
Windows-International-Core-Winpe	UILanguage = en-US
SetupUILanguage	UILanguage = en-US

Windows Setup Neutral

Property	Setting and Value
DiskConfiguration	WillShowUI = OnError
Disk[DiskID="0"]	DiskID = 0 WillWipeDisk = True
CreatePartition[Order="1"]	Extend = False Order = 1 Size = 100 Type = Primary
CreatePartition[Order="2"]	Extend = True Order = 2 Type = Primary
ModifyPartition[Order="1"]	Active = True Format = NTFS Label = System Reserved Order = 1 PartitionID = 1
ModifyPartition[Order="2"]	Active = False Format = NTFS Letter = D Order = 2 PartitionID = 2

WindowsDeploymentServices

Property	Setting and Value
InstallImage	Filename = WIN7WDS.vhd ImageGroup = Virtual Disks ImageName = Enterprise_6.1.7200
InstallTo	DiskID = 0 PartitionID = 2
Credentials	Domain = <YourDomain> Password = <YourPassword> Username = <YourUsername>



Note

If you have worked with answer files, you might notice that the settings in the answer file deploy the VHD to drive D. This is because the drive letter that you assign here will be assigned to the partition where the VHD will be deployed. When deploying VHDs, the VHD needs to be on drive D so the operating system within the VHD can be assigned the drive letter C. This is not a hard technical requirement, but having the operating system on drive C prevents users from being confused and applications from failing—all because the expectation is often that the operating system is located on drive C.

To save your answer file, on the **File** menu, click **Save Answer File**, navigate to **D:\RemoteInstall\WDSClientUnattend**, and then save the file as **Autounattend.xml**.

Step 7: Associate the answer file

The final step in this scenario is to associate the answer file that you created in Step 6 with the appropriate client computers by using Windows Deployment Services. Associating the answer file configures every applicable computer in exactly the same way. In this scenario, this means that every computer that contacts the Windows Deployment Services server will get two partitions and the VHD will be installed on drive D. You can associate the answer file in two ways:

1. With the architecture of the computer (x64-based, x86-based, or Itanium-based) by using the Windows Deployment Services MMC snap-in
2. With an individual computer by using WDSUtil.

This step explains both methods.

The following procedure explains how to associate the answer file with computers that are x86-based.

▶ To associate the answer file with x86-based computers

1. In the Windows Deployment Services MMC snap-in, right-click the server name, and select **Properties**.
2. On the **Client** tab, select the **Enable unattended installation** check box.
3. Click **Browse**, and then locate the answer file that you stored in the **WDSClientUnattend** folder. Click **Open** to return to the **Client** tab.
4. Click **OK** to close the **Properties** pages.

Associating the answer file with an architecture may work for the majority of your needs, but there are always exceptions where you may want to configure a specific computer with different settings. To associate the answer file with an individual computer, use the following procedure.



Note

Before you begin this procedure, make sure that you have the appropriate administrative credentials to create a computer object in Active Directory®.

► **To associate the answer file with an individual computer**

1. Open a Command Prompt window.
2. Type the following command with the appropriate UUID for the client computer. You can find the UUID in the BIOS of the computer and you will see it when the computer boots to the network. Alternatively, if the computer has already installed an operating system using Windows Deployment Services, you can find the UUID in the properties of the computer object in Active Directory.

```
wdsutil /Add-Device /Device:Computer1 /ID:{The computer's  
UUID} /WDSClientUnattend:WDSClientUnattend\WDSUnattend.xml
```



Note

For more information about this command, see [/add-Device](http://go.microsoft.com/fwlink/?LinkId=205106) (<http://go.microsoft.com/fwlink/?LinkId=205106>).

You have now completed all of the steps to deploy the VHD to computers by using Windows Deployment Services. You can now start the computer and boot to the network to install the VHD.

Scenario 3: Performing an Advanced Deployment of Native Boot VHDs

In this scenario, you will use more advanced deployment and configuration tools that are included in Windows 7 and Windows Server 2008 R2. You will make extensive use of command-line tools and you will create a customized Windows PE image, which will serve as an installation platform (instead of the Boot.wim file used in Scenario 2). After you finish the steps in this scenario, you will have a computer that can boot to two VHDs in addition to Windows PE. The following is a summary of the steps that you will take to accomplish this:

- Create a customized Windows PE image and boot into it from a USB flash drive to a client computer.
- Create two partitions on the local disk of the client computer.
- Copy the Windows PE image from the USB flash drive to the first partition on the client computer.
- Edit the Windows PE image so that when it is selected, it is loaded into memory.
- Create two VHDs on the second partition of the client computer to use as boot disks. No additional operating system will be installed on this drive.
- Edit the boot configuration data (BCD) store so you can select Windows PE, Windows 7, or Windows Server 2008 R2 when the computer boots.
- After the operating systems are installed, create and configure differencing VHDs.

To illustrate, Figure 7 represents the disk layout and placement of the VHDs of the resulting client computer.

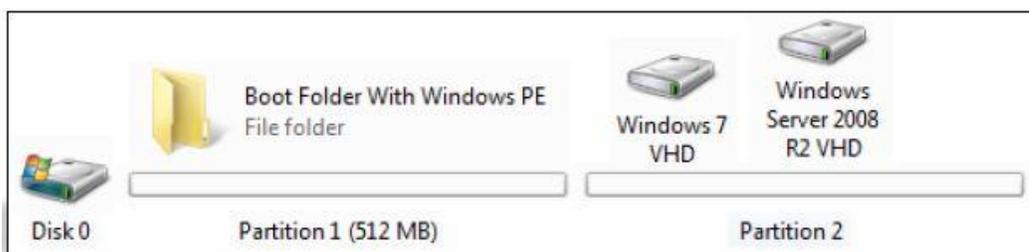


Figure 7 VHD layout for the client computer

As you can see, the first partition is slightly larger than the default 100 MB partition that Windows 7 or Windows Server 2008 R2 creates. The reason for this is that you will store the custom .wim file, which is larger than 100 MB, on that partition. Choosing a size of 512 MB ensures that there is sufficient disk space for additional tools.

Scenario prerequisites

Before you proceed, ensure you have the following resources available:

- A computer running Windows 7 Enterprise, Windows 7 Ultimate, or Windows Server 2008 R2.
- The Windows Automated Installation Kit (Windows AIK) installed on your computer. If not, see Step 1 of [Scenario 1: Configuring Native VHD Boot on a Single Computer](#).
- The installation media for Windows 7 Enterprise, Windows 7 Ultimate, or Windows Server 2008 R2.
- A computer on which to install the VHD.
- A USB flash drive with a minimum of 10 GB free disk space.

Step 1: Create a customized Windows PE image

The first step is to create a custom version of Windows PE. You might not be aware, but you already use Windows PE when you install Windows 7 or Windows Server 2008 R2. When you start the installation, the environment that you see is a version of Windows PE. In the case of Windows 7 and Server 2008 R2, the version is 3.0.

▶ To create a customized Windows PE image

1. On the Start menu, in the **Microsoft Windows AIK** folder, right-click **Deployment Tools Command Prompt**, and then click **Run as administrator**.
2. Create the initial directory structure for Windows PE by using `Copype.cmd`, a script that is installed with the Windows AIK. The following command creates the appropriate directory structure for a 64-bit version; if you want to create a 32-bit version, replace **amd64** with **x86**.

```
copype.cmd amd64 c:\winpe_x64
```

Specifically, this command will create the following directory structure:

```
\winpe_x64
\winpe_x64\ISO
\winpe_x64\mount
```

In the root of the **winpe_x64** folder, you will find a default Windows PE image (Winpe.wim) that has the architecture you used in the previous command (in this case x64). This file will serve as the base for the custom version of Windows PE that you will modify in the following steps.

3. Later, when you configure the client computer, you will make use of many command-line tools. However, these tools are not all available in Winpe.wim by default. To add them manually, you need to mount the image so that you can add, edit, or delete files. (Mounting an image is basically attaching it to a folder. When you open that folder, you will see the data of the image and you can alter the data.) The tool that makes this possible is Deployment Image Servicing and Management (DISM). This tool is new to Windows 7 and Windows Server 2008 R2, and it can handle most management tasks for maintaining .wim and .vhd images. In this example, to mount the Winpe.wim file that is stored in the folder named **C:\winpe_x64\mount**, type:

```
Dism /Mount-Wim /WimFile:C:\winpe_x64\winpe.wim /index:1
/MountDir:C:\winpe_x64\mount
```

4. Optionally, you can browse to **C:\winpe_x64\mount** in Windows Explorer to see the content of image that you just mounted.
5. In the root of **C:\winpe_x64\mount**, create a subfolder where you will copy the necessary tools. Later, you will use these tools to apply the Windows PE image to the VHDs and to create and configure the boot files. To run these tasks, type:

```
mkdir c:\winpe_x64\mount\tools
xcopy "c:\Program Files\Windows AIK\Tools\x64" /e
c:\winpe_x64\mount\tools
```

6. Close the Winpe.wim, and then type the following command to save the changes by using DISM:

```
dism /unmount-wim /Mountdir:c:\winpe_x64\mount /commit
```

 **Note**

The **/commit** option stores the changes to Winpe.wim. All the changes would be lost if you omitted this parameter.

Step 2: Create bootable media

After creating a custom Windows PE image, you are ready to configure it on a bootable device. You could simply create an .iso file of the image and burn it to a DVD, but in this scenario we will use a USB flash drive, which provides flexibility. Most USB flash drives will work as long as they can be used as a bootable device and they contain sufficient storage capacity.

► **To create bootable media**

1. When you boot to a USB flash drive, the computer searches for a Boot.wim file that is located in the **Sources** folder in the root of the boot device. To copy Winpe.wim to the **Sources** folder and rename it as Boot.wim, type:

```
copy c:\winpe_x64\winpe.wim c:\winpe_x64\ISO\sources\boot.wim
```

2. You are now ready to use DiskPart to prepare the USB flash drive so it can be used as a boot device.. To start DiskPart and display all of the available disks on the computer, type:

```
diskpart  
list disk
```

Your output will look similar to Figure 8.

```
c:\winpe_x64>diskpart  
Microsoft DiskPart version 6.1.7100  
Copyright (C) 1999-2008 Microsoft Corporation.  
On computer: MICHAELS-PC  
DISKPART> list disk  


| Disk ### | Status   | Size   | Free | Dyn | Gpt |
|----------|----------|--------|------|-----|-----|
| Disk 0   | Online   | 279 GB | 0 B  |     |     |
| Disk 1   | No Media | 0 B    | 0 B  |     |     |
| Disk 2   | No Media | 0 B    | 0 B  |     |     |
| Disk 3   | No Media | 0 B    | 0 B  |     |     |
| Disk 4   | No Media | 0 B    | 0 B  |     |     |
| Disk 5   | Online   | 14 GB  | 0 B  |     |     |

  
DISKPART>
```

Figure 8 Display of all local disks

3. In this example, disk 5 is the USB flash drive (circled in red). The following commands will select this disk, remove all the data from it, and create the partition where the files will be copied.

❗ **Important**

These actions will delete all data on the USB flash drive.

To perform these tasks, type:

```
select disk 5  
clean create partition primary active  
format quick fs=ntfs  
assign letter=u  
exit
```

4. After you create the partition on the USB flash drive, you need to copy the custom Windows PE files that you created in Step 1 to this partition. The files are stored in the **C:\winpe_x64\iso** folder. To copy these files to drive U, type:

```
xcopy c:\winpe_x64\iso\*. * /e u:\
```

You now have a bootable USB flash drive that contains a custom Windows PE image.

Step 3: Copy the Windows image to the USB flash drive

In this step, you will copy the Windows image (Install.wim), which contains all the bits for the Windows installations, onto the USB flash drive. The Install.wim file is located in the **Sources** folder of the installation media (a DVD or an extracted .iso file). Because you want to create a dual boot configuration between Windows 7 and Windows Server 2008 R2, you need the Install.wim from both operating systems. You will create two folders to separate the files: a **W2K8R2** folder to store the Install.wim for Windows Server 2008 R2, and a **WIN7** folder to store the Install.wim for Windows 7.

To create the folders and copy the Install.wim files to them, type:

```
mkdir U:\WIM\W2K8R2
```

```
mkdir U:\WIM\WIN7
```

```
copy C:\W2K8R2\Sources\install.wim U:\WIM\W2K8R2
```

```
copy C:\WIN7\Sources\Install.wim U:\WIM\WIN7
```

Step 4: Boot to the USB flash drive from the client computer

You are now ready to begin configuring the client computer. Computers are different when it comes to how to boot from a USB flash drive. For example, some laptops use the F9 function key to enter the boot menu, but other computers have completely different methods.

When you know how to boot to the USB flash drive on your client computer, insert the USB flash drive, boot the computer, and select the USB flash drive from the boot menu. When the boot sequence is finished, you will see a Command Prompt window with a blinking cursor awaiting your commands.

Step 5: Partition the disk on the client computer

If you have installed Windows 7 or Windows Server 2008 R2 in the past, you may have noticed that Windows setup created a 100 MB partition. This partition is where Windows setup copies and configures the boot files for the operating system to prevent accidental damage. The partition does not receive a drive letter after the computer has booted. Another advantage of creating such

a partition is for those who want to use BitLocker Drive Encryption. Computers are already prepared for BitLocker, and they do not require additional disk configuration.

As discussed at the beginning of this document, you will create a somewhat larger partition than the default 100 MB partition to store the Windows PE image. You could use the USB flash drive that you created in the previous steps, but it is more convenient to have a local bootable copy of Windows PE, in case you need to recover or alter the underlying system. Figure 9 shows an overview of the intended partition layout.

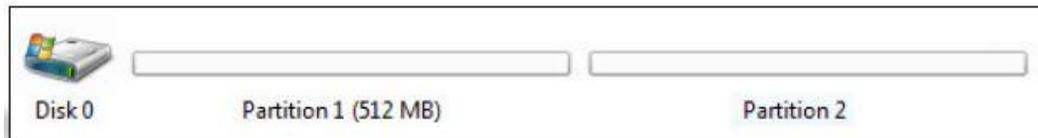


Figure 9 Partition layout

You will use the command-line tool DiskPart to partition the disk.

► **To partition the disk**

1. At an elevated command prompt, type `diskpart`, and then type the following command to determine the number of the disk that you want to target.

```
list disk
```

Locate the appropriate disk and remember the disk number. In most computers, it will be disk number 0, but this is not always the case.

 **Note**

Disk numbers always start with number 0, while partitions always start with 1.

2. Type the following commands to create a 512 MB partition, make it bootable, and assign drive letter C.

 **Important**

These actions will delete all data on the hard disk drive.

```
Select disk 0  
clean  
create partition primary size=512  
Active  
Assign letter=c  
format fs=ntfs quick
```

 **Note**

At this point you might be wondering why you assigned the drive letter C to this partition when you will want C to be the drive letter of the operating system. You can be assured that drive letters do not persist after you exit Windows PE and reboot into the operating system. You could have assigned drive letter Z because after the computer is rebooted, the drive letters are automatically assigned again.

3. After you create the bootable partition, you need to create a second partition to store the VHD files. In this example, you will use the remaining capacity of the disk, but you could assign a maximum partition size. In DiskPart, type the following commands to create the second partition and assign drive letter D to it.

```
create partition primary
assign letter=d
format fs=ntfs quick
exit
```

After partitioning the computer, your disk layout will look similar to Figure 10.

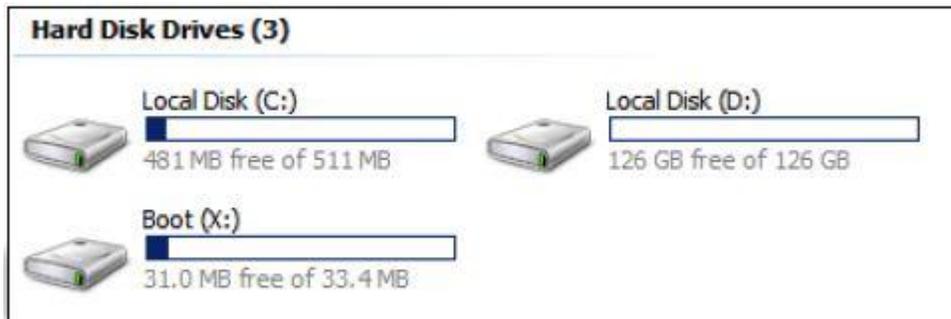


Figure 10 New partition layout

Step 6: Create the VHDs

The next step is to create the VHD files. Because Windows PE does not have a UI to create VHD files (like Disk Management), you will again use DiskPart to create and configure the VHD manually.

► To create the VHD

1. In DiskPart, type the following command to create the VHDs:

```
create vdisk file="D:\WIN7.vhd" maximum=102400
type=expandable
create vdisk file="D:\W2K8R2.vhd" maximum=102400
type=expandable
```

This command creates two VHD files: WIN7.vhd (which you will use for Windows 7) and W2K8R2.vhd (which you will use for Windows Server 2008 R2). The **type=expandable** parameter specifies to create a dynamically expanding VHD so the VHDs will grow as data is added to them (with a maximum size of 100 GB). For more information and options for creating VHDs with DiskPart, see [Create vdisk](http://go.microsoft.com/fwlink/?LinkId=205510) (<http://go.microsoft.com/fwlink/?LinkId=205510>).

2. Before you can continue editing the VHD, you have to tell DiskPart which disk you want to use (by using the **select** command) and bring it online (by using the **attach** command).

To perform these tasks, type:

```
select vdisk file="D:\WIN7.vhd"  
attach vdisk
```

3. From this point, it will be like you inserted a 100 GB physical disk that you can treat it like any other physical disk. If you run the **list disk** command, you will see the additional disk. To create a primary partition on the VHD, format it, and assign drive letter V, type:

```
create partition primary  
format fs=ntfs quick  
assign letter=v
```

4. Now use the same commands to configure the VHD that will host Windows Server 2008 R2. You cannot use the same drive letter, so you will type S:

```
select vdisk file="D:\W2K8R2.vhd"  
attach vdisk  
create partition primary  
format fs=ntfs quick  
assign letter=s
```

5. With both VHDs created, (type `exit`) to close DiskPart.

The disk layout that you created now looks like Figure 11.

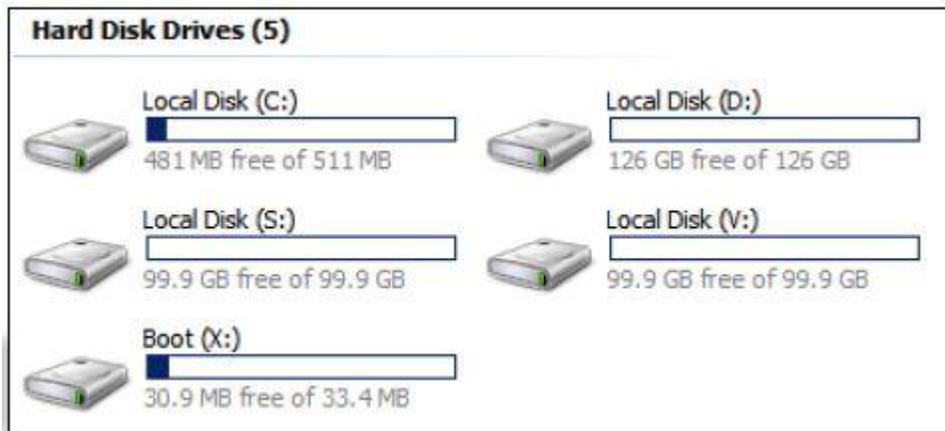


Figure 11 Physical disks and mounted VHDs (V and S)

Step 7: Apply the WIM files to the VHDs

After you create the VHDs, you need to apply the Install.wim files from Step 3 to the VHDs. The tool you will use is ImageX. Windows PE normally does not contain ImageX, but you added it to your Windows PE image with the Tools folder in Step 1.



Note

If you are a more advanced user, you could use a custom Windows image (.wim) file instead of using the Install.wim file from the installation media.

► To apply the Install.wim to the VHDs

1. Because a .wim image can contain more than one edition of Windows, you first need to determine the correct index number of the edition that you want to use. This is particularly important for Windows 7 because only Windows 7 Ultimate and Windows 7 Enterprise can be used for native VHD boot. To view the contents of the Install.wim, type the following commands:

```
cd \tools
imageX /info E:\WIM\WIN7\Install.wim
```

The output will display an overview of the Install.wim in XML format. To locate Windows 7 Ultimate, which is used in this scenario, find the following line:

```
<NAME>Windows 7 ULTIMATE</NAME>
```

2. After you locate this line, scroll up and locate the index number that is associated with it. Specifically, for Windows 7 Ultimate, you will see:

```
<IMAGE INDEX="4">
```

Remember this index number because you will use it in the next step.

3. To apply the data from Windows 7 Ultimate to the WIN7.vhd file, type:

```
imageX /apply E:\WIM\WIN7\Install.wim 4 V:\
```

Let the process finish before you continue.

4. Now you have applied Windows 7 Ultimate to WIN7.vhd, but you still need to apply Windows Server 2008 R2 to W2K8R2.vhd. Before you apply the Windows image, you will run ImageX again with the **/info** parameter to locate the appropriate index number. To view the output and then apply Windows Server 2008 R2 Standard (which has an index number of 1) to W2K8R2.vhd, run the following commands:

```
imageX /info E:\WIM\W2K8R2\Install.wim
imageX /apply E:\WIM\W2K8R2\Install.wim 1 S:\
```

Let the process finish before you continue to the next step.

Step 8: Configure the BCD store

Now that you have created the VHDs and applied the operating systems to them, you need to configure the files that are used to start the operating system (the boot files). Typically, when you install the Windows operating system on a computer, setup completes this step. However,

because you are building this system, you will need to create and manually configure the boot files.

You will work with the boot configuration data (BCD) store by using two command line tools: BCDBoot and BCDEdit. You will use BCDBoot to copy a default BCD store to the hard disk drive and make the necessary configuration changes. You will use BCDEdit to edit the BCD store that you create.

► To create a BCD store

1. You need to point BCDBoot to a copy of the Windows operating system and tell it where to copy the files. To copy the boot files from the WIN7.vhd file to drive C, which will be the partition that hosts the boot files, type:

```
bcdboot v:\windows /s c:
```

2. If you start the computer at this point, the operating system that was used to copy the boot files from (Windows 7) would automatically start. To validate this, type the following BCDEdit command:

```
BCDEdit -store c:\Boot\BCD /enum /v
```

The output will be similar to Figure 12. You will see that the **default** value (the GUID) under **Windows Boot Manager** matches the value for the Windows 7 entry. Note or copy this GUID because you will use it in the following procedure.

```
X:\tools>bcdedit -store c:\Boot\BCD /enum /v

Windows Boot Manager
-----
identifier          {9dea862c-5cdd-4e70-acc1-f32b344d4795}
device              partition=C:
description         Windows Boot Manager
locale              en-us
inherit             {7ea2e1ac-2e61-4728-aaa3-896d9d0a9f0e}
default             {24761cb1-3f82-11de-8235-0003ffa37a4a}
resumeobject       {24761cb0-3f82-11de-8235-0003ffa37a4a}
displayorder       {24761cb1-3f82-11de-8235-0003ffa37a4a}
toolsdisplayorder  {b2721d73-1db4-4c62-bf78-c548a880142d}
timeout            30

Windows Boot Loader
-----
identifier          {24761cb1-3f82-11de-8235-0003ffa37a4a}
device              vhd=[D: ]\WIN7RC.vhd, locate=custom:12000002
path               \Windows\system32\winload.exe
description         Windows 7
locale              en-us
inherit             {6efb52bf-1766-41db-a6b3-0ee5eff72bd7}
osdevice           vhd=[D: ]\WIN7RC.vhd, locate=custom:22000002
systemroot         \Windows
resumeobject       {24761cb0-3f82-11de-8235-0003ffa37a4a}
nx                 OptIn
detecthal          Yes
```

Figure 12 The default BCD

After you complete the previous procedure, a boot entry for Windows Server 2008 R2 is still missing. However, because you created a default store by using BCDBoot, you can add additional entries to it to support booting Windows Server 2008 R2. The following procedure shows how to use BCDEdit to complete this task.

► **To add a boot entry for Windows Server 2008 R2 to the BCD store**

1. Type the following command to make a copy of the default boot entry for Windows 7 (use the **identifier** GUID from the previous step). Optionally, you can use the **/d** parameter to specify a custom description for the new entry.

```
BCDEdit -store c:\Boot\BCD /copy {24761cb1-3f82-11de-8235-0003ffa37a4a} /d "Windows Server 2008 R2"
```

This command will generate a GUID, which you should note or copy. You will use it in the following steps to modify the entry.

2. There are now two parameters that you need to change before you can boot Windows Server 2008 R2: **OSDevice** and **Device**. The **Device** parameter references the location where the OSLoader is located (Winload), and the **OSDevice** parameter references the location of the operating system. Type the following commands by using the GUID for the Windows Server 2008 R2 boot entry that you just created.

```
BCDEdit -store c:\Boot\BCD /set {4a453550-3f84-11de-8235-0003ffa37a4a} device VHD=[D:]\W2K8R2.vhd
```

```
BCDEdit -store c:\Boot\BCD /set {4a453550-3f84-11de-8235-0003ffa37a4a} osdevice VHD=[D:]\W2K8R2.vhd
```

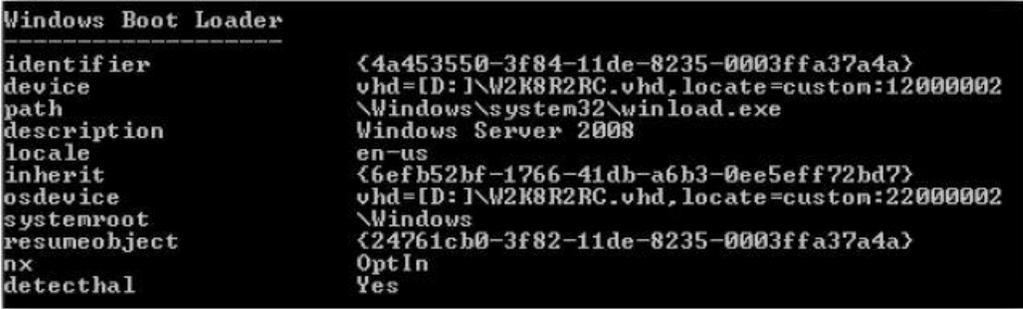
 **Note**

Instead of using **[D:]** in these two commands, you could use **[Locate]**, in which case the Boot Manager will search all drives for the VHD and use the first match.

3. Type the following command to view the new entry for Windows Server 2008 R2.

```
BCDEdit -store c:\Boot\BCD /enum /v
```

Figure 13 shows an example of the output.



```
Windows Boot Loader
-----
identifier          {4a453550-3f84-11de-8235-0003ffa37a4a}
device              vhd=[D:]\W2K8R2RC.vhd, locate=custom:12000002
path                \Windows\system32\winload.exe
description         Windows Server 2008
locale              en-us
inherit             {6efb52bf-1766-41db-a6b3-0ee5eff72bd7}
osdevice            vhd=[D:]\W2K8R2RC.vhd, locate=custom:22000002
systemroot          \Windows
resumeobject        {24761cb0-3f82-11de-8235-0003ffa37a4a}
nx                  OptIn
detecthal           Yes
```

Figure 13 New boot entry for Windows Server 2008 R2

Step 9: Add the Windows PE image to the client computer

One of the requirements for this scenario was to have the Windows PE image available locally on the physical hard disk drive. To configure this, you must copy the Winpe.wim file that you created earlier to the smaller partition. This way, it will remain hidden during the normal operations of Windows 7 or Windows Server 2008 R2. To copy the required files to the hard disk drive, type the following two commands:

```
copy e:\sources\boot.wim c:\Boot\winpe.wim
```

```
copy e:\boot\boot.sdi c:\boot\boot.sdi
```

In the next step, you will use the Boot.sdi file to create a RAM disk. A RAM disk is essentially RAM that you configure to act as a disk. However, because the disk is made of RAM, its contents are lost when the computer is turned off.

Step 10: Create a RAM disk

Typically when you boot a computer, you cannot edit the boot disk because the files that are loaded from the boot disk are locked. Therefore, to alter the partition layout of the hard disk drive, the operating system that starts from the hard disk drive must be loaded into memory first. This way, no handles remain open and you can make changes to the hard disk drive as needed. Windows PE can be loaded into memory, which is called a RAM disk. However, this is not default behavior so you need to configure the boot sequence to enable this functionality. To create a RAM disk, use the following procedure.

► To create a RAM disk

1. To make sure that you do not already have a RAM disk entry available in the BCD store, run the following BCDEdit command:

```
BCDEdit -store c:\Boot\BCD /enum all /v
```

The **all** parameter displays all boot entries, including those that are normally hidden.

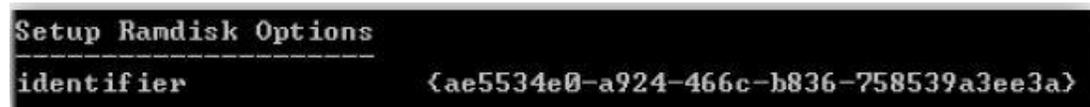
2. If you followed the steps in this scenario, there will not be an entry for a RAM disk so you will need to manually configure the BCD. To create the entry for the RAM disk, type:

```
BCDEdit -store c:\Boot\BCD /create {ramdiskoptions}
```

3. If the previous command ran successfully, an entry for the RAM disk will now be available in the BCD store. To view the entry, type:

```
BCDEdit -store c:\Boot\BCD /enum all /v
```

Figure 14 displays output similar to what you should see. The identifier to the RAM disk entry is another GUID, and you will use this GUID to alter the boot entry.



```
Setup Ramdisk Options
-----
identifier          {ae5534e0-a924-466c-b836-758539a3ee3a}
```

Figure 14 The entry for the RAM disk

4. Now you can specify the following three settings to alter the boot entry:
 - **Description.** The text that you enter for the description will be displayed in the boot menu as an option that you can choose to boot from.
 - **Ramdisksdidevice.** This setting specifies the partition letter where the binary file that creates the RAM disk (Boot.sdi) resides.
 - **Ramdisksdipath.** This setting points to the binary file.

To specify these settings, type the following three commands:

```
BCDEdit -store c:\Boot\BCD /set {ae5534e0-a924-466c-b836-758539a3ee3a} description "Windows PE"
BCDEdit -store c:\Boot\BCD /set {ae5534e0-a924-466c-b836-758539a3ee3a} ramdiskdidevice partition=C:
BCDEdit -store c:\Boot\BCD /set {ae5534e0-a924-466c-b836-758539a3ee3a} ramdiskdipath \boot\boot.sdi
```

Type the following command to display the custom options for the RAM disk entry:

```
BCDEdit.exe -store c:\Boot\BCD /enum /v
```

Figure 15 displays output similar to what you should see.



```
Setup Ramdisk Options
-----
identifier          {ae5534e0-a924-466c-b836-758539a3ee3a}
description         Windows PE
ramdiskdidevice     partition=C:
ramdiskdipath       \boot\boot.sdi
```

Figure 15 Custom RAM disk settings

Step 11: Create the BCD entry for the Windows PE image

You will create a new BCD entry for the Windows PE image. This will enable you to boot Windows PE from the hard disk drive directly to the RAM disk that you created in the previous step.

▶ To create a BCD entry for Windows PE

1. You could manually create all the required entries, but it is easier to copy a previously created boot entry and use that as a starting point. To copy the boot entry for Windows 7, type:

```
BCDEdit -store c:\Boot\BCD /copy {24761cb1-3f82-11de-8235-0003ffa37a4a} /d "Windows PE"
```

2. You may have noticed that you ran the same command as the command that you used to create the Windows Server 2008 R2 boot entry. Just as before, you will use the GUID for this boot entry as a reference.

To further modify the entry, type the following three commands. Note that there is one difference to the commands that you ran previously: you need to configure the Boot Loader to first copy the Winpe.wim file to the RAM disk, and then boot it from there. You accomplish this by using the GUID that immediately follows the comma (this is the GUID for the RAM disk that you created in the Step 10).

```
BCDEdit -store c:\Boot\BCD /set {76247ac0-3f8f-11de-8235-0003ffa37a4a} device ramdisk=[C:]\boot\winpe.wim,{ae5534e0-a924-b836-758539a3ee3a}

BCDEdit -store c:\Boot\BCD /set {76247ac0-3f8f-11de-8235-0003ffa37a4a} osdevice ramdisk=[C:]\boot\winpe.wim,{ae5534e0-a924-b836-758539a3ee3a}

BCDEdit -store c:\Boot\BCD /set {76247ac0-3f8f-11de-8235-0003ffa37a4a} winpe yes

BCDEdit -store c:\Boot\BCD /set {76247ac0-3f8f-11de-8235-0003ffa37a4a} detecthal on
```

This was the last step you needed to achieve one of the primary goals of this scenario: no operating system is installed on the local hard disk drive and you can boot Windows PE from the local hard disk drive which will load directly into memory.

Step 12: Prevent the VHD from expanding to its full size



Note

You can skip this step if you are deploying fixed-size VHDs instead of dynamically expanding VHDs.

During boot time, a dynamically expanding VHD will expand to its full size to guarantee space on the computer and to prevent data corruption. If the computer cannot allocate sufficient space from the physical disk that the VHD resides on, it will generate a stop error message and prevent the operating system from booting.

When you have a single VHD configured on a computer (as in Scenario 1), this is typically not an issue. However, when you have multiple VHDs (as in this scenario), and you do not want to use all of the space at one time, you can prevent this expansion by adjusting a registry key. This is sometimes called “thin provisioning”.

In this step, you will learn how to configure this in two ways: after booting into the operating system, and from within Windows PE before the operating system has been booted (which is currently the case if you are following the steps in this scenario). Before we walk through the steps, the following example explains why you may want to consider preventing VHDs from expanding.

Imagine a situation where the Windows 7 VHD grows to a size of 30 GB and the underlying physical disk has a total size of 120 GB. In theory, this would leave sufficient disk space. After all, 30 GB + 5 GB = 35 GB, so 85 GB would remain. Unfortunately, this is not the default behavior of the native VHD boot. Normally, the Windows Server 2008 R2 VHD would expand to a full size of a 100 GB when it is selected to start. This would require 130 GB of disk space (100 GB for Windows Server 2008 R2 + 30 GB for the Windows 7 VHD). In this case, booting Windows Server 2008 R2 would generate a stop error message and prevent you from booting the system because it lacks 10 GB, and it cannot expand to its full size.

Therefore, in scenarios where multiple VHDs reside on the same hard disk drive and disk space is an issue, you can prevent this behavior in two ways:

- **After the operating system has booted.** For VHDs that you have already booted into, you can prevent this expansion by using Registry Editor. To do this, browse to the following location and change the value to **4**:

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\FsDepends\Parameters] "VirtualDiskExpandOnMount"=dword:00000004
```

As you can see in Figure 16, the disk will expand to its full size after you restart the computer. This allows you to make more efficient use of the underlying storage space.

 pagefile.sys	13-5-2009 0:39	System file	1,048,576 KB
 W2K8R2RC.vhd	13-5-2009 0:17	VHD File	5,117,355 KB
 WIN7RC.vhd	13-5-2009 0:39	VHD File	5,311,955 KB

Figure 16 Changes after applying the registry key

- **Before the operating system has booted.** You can also configure this setting from within Windows PE when the operating system has not already been booted into. You need to mount the registry from the operating system that resides on the VHD, apply the appropriate setting, and then save the registry file. Use the following procedure to configure this functionality on your Windows 7 VHD.

▶ To prevent the VHD from expanding by using Windows PE

1. In the Windows PE command prompt, type **Regedit**.
2. Click the **HKEY_LOCAL_MACHINE** hive.
3. Click **File**, and then click **Load Hive**.
4. Browse to the appropriate drive letter. In our example, this is **V**.
5. In the **V:\Windows\System32\Config** folder, select the **SYSTEM** file (shown in Figure 17), and then click **Open**.

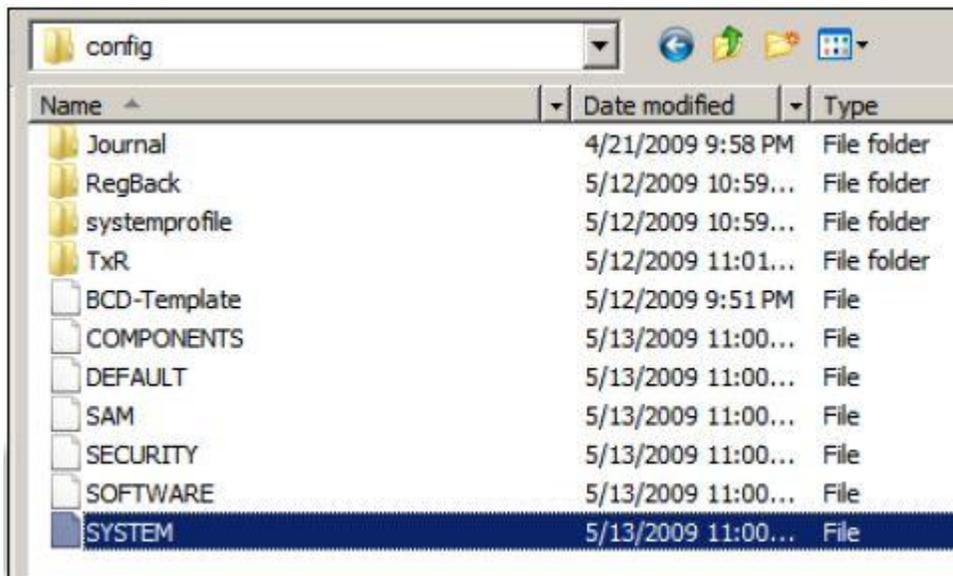


Figure 17 The SYSTEM file

6. In the **Load Hive** dialog box, type **TEMP**.
7. Browse to the following key:


```
HKEY_LOCAL_MACHINE\TEMP\ControlSet001\services\FsDepends\Parameters
```
8. Change the **VirtualDiskExpandOnMount** setting to a value of **4**.
9. In the left pane under **HKEY_LOCAL_MACHINE**, click **TEMP**.
10. Click the **File** menu, and then click **Unload Hive**.
11. On the **Confirm Unload Hive** page, click **Yes** to save your changes.
12. Repeat these steps to change the registry value for the VHD that hosts Windows Server 2008 R2.

 **Notes**

For increased performance during native VHD boot, Windows will place the paging file (Pagefile.sys) outside the VHD on the host volume. Therefore, when you only have a single VHD on a computer, you can use the following calculation to determine the maximum size that the VHD should be:

$$\langle \text{Maximum size of the VHD} \rangle = \langle \text{Size of the physical disk} \rangle - \langle \text{Size of Pagefile.sys} \rangle$$

Step 13: Configure the operating systems in the VHDs

In this step, you will boot into the VHDs and finish configuring them.

► To complete the configuration

1. To prevent corruption, you should again detach the VHDs before restarting the computer. To detach both VHDs, type:

```
diskpart
select vdisk file="D:\WIN7.vhd"
detach vdisk
select vdisk file="D:\W2K8R2.vhd"
detach vdisk
exit
```

2. Type **exit** in the Command Prompt window and remove the USB flash drive.
3. You are now ready to install the operating systems that you applied to the VHDs. Restart the computer, select Windows 7 from the boot menu, and continue the installation. When the installation has finished, configure the operating system as desired.
4. Repeat the same process for the Windows Server 2008 R2 VHD (boot into the image and configure the operating system as appropriate).
5. When you are satisfied with both configurations, boot back into Windows PE and continue with the next step.

Step 14: Create and configure differencing VHDs

By using differencing VHDs with native VHD boot, you can restore an operating system to its default configuration within a few minutes. Differencing VHDs with native VHD boot are the same identical to those you may have used with Hyper-V or Microsoft Virtual PC. The VHDs have a parent/child relationship where all new blocks are written to the child disk (the differencing VHD), leaving the parent disk in its preconfigured state. This enables you to quickly restore the operating system to its original state by replacing the differencing VHD and re-enabling the parent/child relationship.

In the previous steps, you installed and configured Windows 7 and Windows Server 2008 R2. Now that you have booted back into Windows PE, you can create the differencing VHDs. Essentially, you will follow the same steps that you performed previously: create two differencing VHDs, one for Windows 7 and one for Windows Server 2008 R2, and then alter the BCD store so that you can boot to the differencing VHDs.

► To configure booting to a differencing VHD

1. To create differencing VHDs that have a parent/child relationship with the VHDs you created previously, type:

```
diskpart
create vdisk file="D:\WIN7_Diff0.vhd" parent="D:\WIN7.vhd"
create vdisk file="D:\W2K8R2_Diff0.vhd"
parent="D:\W2K8R2.vhd"
exit
```

2. To boot to the differencing VHDs, you need to update the BCD store to reflect these intended changes. To do this, type the following commands, and use the GUIDs for the Windows 7 VHD boot entry and the Windows Server 2008 R2 VHD boot entry that you created in Steps 8 and 9. You can run the `bcdedit /enum all /v` command to view the appropriate GUIDs.

- **Windows 7**

```
BCDEdit -store c:\Boot\BCD /set {24761cb1-3f82-11de-8235-0003ffa37a4a} device VHD=[D:]\W2K8R2_Diff0.vhd
```

```
BCDEdit -store c:\Boot\BCD /set {24761cb1-3f82-11de-8235-0003ffa37a4a} osdevice VHD=[D:]\W2K8R2_Diff0.vhd
```

- **Windows Server 2008 R2**

```
BCDEdit -store c:\Boot\BCD /set {4a453550-3f84-11de-8235-0003ffa37a4a} device VHD=[D:]\W2K8R2_Diff0.vhd
```

```
BCDEdit -store c:\Boot\BCD /set {4a453550-3f84-11de-8235-0003ffa37a4a} osdevice VHD=[D:]\W2K8R2_Diff0.vhd
```

This concludes the steps that are required for this advanced deployment scenario.

Conclusion

With the addition of native VHD boot, you have a powerful new tool available that enables you to have a single file format that can be used in virtual and physical environments. Hopefully, the scenarios described in this document have provided enough insight so you can start enjoying the benefits of this technology.