

# 1

## Deploying Windows with Style: Windows Deployment Services (WDS), and Microsoft Deployment Toolkit 2008

This chapter is about getting Windows Operating Systems deployed with the least amount of pain and the best possible results. It is filled with tips and tricks, best practices, guidance, and notes from the field on how you can take advantage of the latest deployment tools and technologies.

Chances are that you are already working with image-based deployment. You might even be working with a third-party imaging solution and may be fine with that.

If you've already got it down to a science, then why should you read this chapter? Well, first of all, it's the first chapter—but beyond that, we think you should understand that the fundamental way to set up Windows has changed.

That means that new deployment tools and techniques have been made available. Some of those are free tools. Indeed, Windows now ships with its own imaging utility called ImageX (which we'll start exploring in the "Imaging Software Isn't About Speed" section). And the best part is, you can leverage these new tools to deploy Windows Vista and Windows XP if you want.

Let's start by busting a common myth:

"Ghost (or your current image tool) is faster than the new stuff Microsoft came out with."  
Really? Is it?

To prove it to you, I decided to put on my "Mythbusters" hat and put some science to the test.

*This chapter was written by Johan Arwidmark with Jeremy Moskowitz*



Check out the MythBusters TV show at <http://dsc.discovery.com/fansites/mythbusters/mythbusters.html>. Maybe they'll invite us on the show to prove our facts to you!

## It's All About Imaging

I spent a good few days in my Lab Center, measuring speed and efficiency when deploying different Windows operating systems (and deploying to a large number of machines) using different imaging tools.

The results may surprise you.



### Real World Scenario

#### Quick Tips Before I Begin

Before giving you the details, let me give you some real-world tips about Windows deployment.

#### Tip #1: You'll Want to Use Image-Based Deployment

Trust me on this: image-based deployment is the way to go, whether you're deploying Windows XP, Windows Vista, Windows Server 2003, or Windows Server 2008.

With Windows XP and Windows Server 2003 you had an option; you could do a scripted setup and also work with images. With Windows Vista and Windows Server 2008 you *don't* have a choice; everything is now image based and for a reason: working with images has been proven to be the easiest, fastest, and most cost-effective way to deploy Windows.

#### Tip #2: Don't Fret About Your Imaging Tool

Which imaging tool you use (whether it's Ghost, Acronis, or ImageX, which we'll examine in this chapter) doesn't really matter that much. This is because the process of applying an image to disk is just one of many steps during deployment of a Windows operating system. Other steps might be driver injection, changes to the Windows setup, or adding post-install applications.

## High-Level Imaging Process

In this section, I'll give you a high-level overview of Windows deployment processes and review the different terms and definitions involved.

First up are Windows deployment terms, where tools and technologies like Windows PE, Sysprep, and PXE will be explained. Then you'll learn about how the Windows setup process works, and last, we'll pull all the information together and match it to the deployment process.

## Windows Deployment Terms

With Windows Vista, Windows deployment tools and processes have changed a lot. The new deployment tools can be used to deploy not only Windows Vista, but also Windows XP, Windows Server 2003, and Windows Server 2008. We can now have common processes no matter what operating system we are deploying.

Let's get some definitions out of the way first. That way, we can share a common vocabulary as we work through the issues with imaging.

**Windows PE** Windows PE is a cut-down version of the Windows operating system that is used to start the deployment process, either by starting a tool for applying an image, or by starting the Windows setup engine. This operating system is also called a boot image.

**Sysprep** You use Sysprep to prepare the machine for disk cloning and restoration via a disk image when working with operating system images. Sysprep will, for example, remove unique identifiers for a PC, clean the driver cache, and clear the event logs. The Sysprep processes can be automated by creating a text-based answer file called `sysprep.inf`.

**PXE** PXE, or Preboot Execution Environment allows you to boot images over the network, for example, a Windows PE boot image.

**Mini Setup** After deploying an image, Windows setup will reboot and enter the mini setup phase. This is when, for example, drivers are installed and the machine is joined to the domain. The `sysprep.inf` answer file is used to automate this process. In Windows Vista the mini setup phase, or pass, is called Specialize.

**Unicast** Unicast is a protocol specification that delivers a set of packages to a single computer (destination).

**Multicast** Multicast sends the same package to multiple computers (destinations) without affecting network bandwidth.

**Windows System Image Manager (WSIM)** WSIM is an authoring tool for editing `unattend.xml` files. This is the replacement for Setup Manager, which was used with Windows XP or Windows Server 2003; this tool works only with Windows Vista and Windows Server 2008.

**Windows Imaging (WIM)** WIM is a format for the new Windows Vista and Windows Server 2008 standard images. You can also create WIM images of Windows XP or Windows Server 2003. WIM images have a high compression rate (3:1), which leads to smaller images. Within each WIM image you have single instancing, which means you can combine multiple images to the same WIM image with only that difference between the two images being added to the wim image.

**ImageX** ImageX is an imaging utility that is part of the Windows AIK (Windows Automated Installation Kit). It can be used to create (capture), deploy (apply), and edit WIM images.

## Windows Desktop and Server Setup Overview

We'll start with exploring the setup of Windows XP and Windows Server 2003. They can be deployed in numerous ways:

**Manually** If you choose to manually deploy Windows, you will boot the computer from the CD/DVD and answer all setup questions manually.

**Unattended** If you want to install from CD/DVD media, but don't want to answer all setup questions manually, you can create a text-based answer file called `unattend.txt` with all the setup answers, name it `winnt.sif`, store it on a floppy or CD media, and then the setup will use it. Using an answer file like this is called a scripted setup.

## 4 Chapter 1 • Deploying Windows with Style

**Network-based Scripted Setup** If you like the idea about answering all setup questions using an answer file, but don't want to deploy from CD/DVD you can copy all setup and answer files to a server share, start a boot media like Windows PE, connect to the server share, and start the setup.

**Network Image-based Setup** After installing the operating system using any of the preceding methods, you can run Sysprep and then capture the operating system to a network share using an imaging utility. This image can then be deployed unattended.

The setup of Windows Vista or Windows Server 2008 may look similar to the setup of Windows XP or Windows Server 2003, but under the hood, the setup is very much different. You can still automate the setup, but the setup is now exclusively image based. In fact, if you look in the `\sources` folder of a Windows Vista DVD, you will see a file called `install.wim`. This is a Windows Vista image that Microsoft prepared in their build lab.

The following are the setup options for Windows Vista and Windows Server 2008:

**Manually** If you choose to manually deploy Windows, you will boot the computer from the CD/DVD and answer all setup questions manually. The difference, compared with the setup of Windows XP and Windows Server 2003, is that you actually deploy a sysprepped image.

**Unattended** If you want to install from CD/DVD media, but don't want to answer all setup questions manually, you can create a text-based answer file called `unattend.xml` with all the setup answers, name it `autounattend.xml`, store it on a floppy, USB or CD media, and then the setup will use it.

**Network Image-based Setup** If you like the idea about answering all setup questions using an answer file, but don't want to deploy from CD/DVD you can copy all setup and answer files to a server share, start a boot media like Windows PE, connect to the server share, and start the setup. This method is similar to the Network Image-based setup of Windows XP and Windows Server 2003.

### Deployment Process

Here is a high-level overview of deploying Windows using a standard deployment process.

1. The deployment process begins when starting a boot image (Windows PE), either from a CD or over the network (PXE boot).
2. After booting the Windows PE boot image, it will connect to a server share containing the image(s) and other files such as drivers and applications.
3. An application or script will prompt for, or query a central configuration file or database for setup information such as what image to deploy, what computer name to use, and so on. This step will be explained in the "Microsoft Deployment Toolkit 2008" section.
4. Apply the image to disk.
5. Detect what hardware you are deploying to and copy the right drivers to disk.
6. Reboot into mini setup (or Specialize pass if deploying Vista).
7. Run post-install actions like installing applications or restoring user data and settings.

## Imaging Software Isn't about Speed

As you can see, the imaging utility is only used in step 4, so it's not that important what imaging utility you are using. But since this step is what most people associate with Windows deployment, let's dive into that step for a while. I did promise you some test results, didn't I?

The two major imaging utilities today are ImageX from Microsoft and Ghost from Symantec. The end result is the same—they both apply an image to disk—but they work a bit differently.

Ghost is sector based, which means you can apply a Ghost image without having to partition the drive first. It's also fast both when capturing and deploying an image, and it supports offline editing of NTFS images (this feature is available in Ghost Solution Suite version 2.0 and later). However Ghost is not freeware.

ImageX is file based, which means you need to partition and format the drive first, then apply the image. It works like a very, very large file copy after the disk is partitioned. The bad news about ImageX is that it is very slow when capturing an image. However, it's pretty fast when *deploying* the image (see my results in Table 1.1). And ImageX is freeware. Like Ghost, it also supports offline editing of the image (which we'll explore in the "Servicing a WDS WIM Image" section a little later).

Table 1.1 shows my test results when capturing and deploying a 32-bit Windows XP image (fully patched and with some applications) and a 64-bit Windows Vista Enterprise (fully patched, no applications) with both Ghost and ImageX.

**TABLE 1.1** Ghost vs. ImageX

	Ghost		ImageX	
	Time to Capture in	Resulting Size	Time to Capture in	Resulting Size
<b>Creating (Capturing the 32-bit Windows XP Image), Initial Size: 4.4GB</b>				
	7 min	2GB	18 min	1.3GB
<b>Deploying Windows XP-32 bit image (plus some apps, fully patched), Initial Size: 4.4GB</b>				
One computer	4 min		4 min	
Five computers (unicast)	4 min		4 min	
Ten computers (unicast)	6 min		5 min	

## 6 Chapter 1 • Deploying Windows with Style

**TABLE 1.1** Ghost vs. ImageX (continued)

	Ghost		ImageX	
	Time to Capture in	Resulting Size	Time to Capture in	Resulting Size
<b>Creating (capturing the 64-bit Windows Vista Image), Initial Size: 10.3GB</b>				
	13 min	4.5GB	30 min	2.5GB
<b>Deploying the 64-bit Windows Vista image (plus some apps and fully patched), Initial Size: 10.3GB</b>				
One computer	7 min		5 min	
Five computers (unicast)	9 min		8 min	
Ten computers (unicast)	12 min		10 min	

Our Mythbusters test shows that ImageX is slower when creating the image, but as fast or faster than Ghost when deploying the image.

It's okay to consider using something other than Ghost. If you're happy with what you're using, that's good. But I think you might be happier with the Microsoft way of things. I'll emphasize that way of things here because it's about as fast (if not faster) and will make your deployments easier to manage overall.

## Windows Deployment Services (WDS)

Do you remember the high-level deployment overview earlier in the chapter? The first step was starting a boot image, followed by selecting an image. Windows Deployment Services (WDS) can help you with that. It's a core player and is frequently used in modern network-based deployment solutions. Later in the chapter, in the "Microsoft Deployment Toolkit 2008" section, we will dive deep into the Microsoft Deployment Toolkit 2008 (formerly known as Business Desktop Deployment, or BDD) and you will learn how these tools and pieces of infrastructure work better together and get some guidance on how to create and configure the deployment solution.

In a networked environment, and with a small investment of your time, WDS can help you better manage your Windows deployments. Its goal is to provide the administrator with the ability to roll out any number of Windows XP, Windows Vista, Windows 2003, and Windows Server 2008 configurations in a short amount of time. In a nutshell, you simply prepare your server for WDS, leverage the built-in hardware your client machine already has to connect to the network, answer a few questions (optional), and away the installation goes.

WDS is available in several forms. It is included in Windows Server 2003 SP2, as well as Windows Server 2008. Windows Server 2003 SP2 will upgrade your existing Windows Server 2003

so that it supports WDS. You can find it at [www.microsoft.com/technet/windowsserver/sp2.msp](http://www.microsoft.com/technet/windowsserver/sp2.msp). You can also get the update for Windows Server 2003 SP1 as part of the Windows Automated Installation Kit (WAIK) here: <http://tinyurl.com/37b42g>. Both versions will be covered in this chapter.

WDS is the descendant of RIS (Remote Installation Services). Under the hood, they've got some similarities and, if you're still using RIS, that's okay: WDS on Windows Server 2003 will upgrade your existing RIS server so it will keep on acting like an RIS server. However, we're not going to go into RIS here. If you'd like information on RIS, you'll have to chase down an older copy of this book (revision 3 and earlier).

Here, we'll assume you don't have RIS installed on your network and that you have installed Windows Server 2008 to get the WDS functionality (remember, WDS is available on Windows Server 2003, as well).

After you create your first WDS-based client machine, you can customize it with commercial or homegrown apps and save that configuration to the server as well, making that machine appear as another downloadable image.

## Inside WDS

WDS is built on multiple components and also relies on some additional network components.

### The Network Component Prerequisites

Before you can use WDS, you need to make sure several components are present on your network.

**DHCP Server** The DHCP server is the first place the client machines look to get a temporary TCP/IP address while the system is being installed.

**DNS Server** DNS server is also required, mainly because it is a key ingredient in Active Directory.

**Active Directory** You're not surprised, are you? Active Directory is also a requirement for WDS. It doesn't mean you need to (or should) run WDS on a domain controller, but rather that you need to run it on a member server in the domain. For test lab or educational purposes you can, of course, run WDS on a domain controller.



Unless you're a smaller company running Small Business Server as your only server, WDS should be running on a member server for performance and security reasons.

**WDS Server** At least one server in your environment needs to be running WDS. We'll set up this service later in the "Setting Up the WDS Server" section.

Although you can run all these services on just one server, in practice you probably wouldn't want to due to the potential heavy processor and disk load the WDS server will have to shoulder. Most real-world configurations run the DHCP and DNS servers on separate boxes but configure one or more specific servers solely for the purpose of running WDS server and dishing out WDS images.

For the sake of this example, however, you can use one server to run it all: DNS, DHCP, and WDS. In this case, we'll use DC01.

### What Does WDS Have That RIS Doesn't?

If you're wondering if you should make the switch from RIS to WDS, the answer is yes. But here's a quick rundown of what you'll get when you do:

- Native support for Windows PE as a boot operating system
- Native support for the Windows Imaging (.WIM) file format (which means support for Windows Vista and Windows Server 2008)
- An extensible and higher-performing PXE server component
- A new client menu for selecting boot operating systems

You'll get to see more details on these items as we progress during this chapter.

## WDS Components

The following WDS components work together to enable you to deploy Windows operating system images:

**Server Components** These consist of a shared folder (with the default name of `RemoteInstall`, shared as `Reminst`), which contains the images and files necessary for network boot, a Preboot Execution Environment (PXE) server, and a Trivial File Transfer Protocol (TFTP) server for network-booting clients.

**Client Components** One client component is the WDS client; it's a GUI that runs in the Windows PE and allows for image selections (attended or unattended). Another client component is the PXE boot ROM. If your network card has the PXE boot ROM code embedded, you can boot directly to the network. You might need to turn this feature on in the network card's BIOS or in the PC's BIOS—or both.



Don't try to deploy all your laptops with one PXE-capable docking station. WDS machines are registered in Active Directory based on a GUID. The GUID is either hard-coded or based on the MAC address. If you deploy all your machines with a single docking station, you'll have multiple machines appearing to have the same GUID. Active Directory requires that machines have a GUID to function properly.

## Setting Up the WDS Server

You can easily add the WDS to any Windows Server 2003 or Windows Server 2008 installation. (WDS is not available in Windows 2003 Server, Web edition.) In the following examples, you'll set up WDS on Windows Server 2008 and add a Windows Vista image for distribution.

Before you set it up, you'll need a decently sized NTFS partition for the WDS components and the WDS images. The size, of course, depends on how many images you want to support, but I consider 36GB to be a minimum. In the upcoming steps I'll use drive letter R: for my WDS server components and images, but you can use any drive letter. For most production environments you usually set it to D: or E:.



Although you can use the boot partition or a second partition on the same hard disk, this results in poor performance of both WDS and the system as a whole. My recommendation is to install WDS on a separate physical disk from that on which Windows resides.

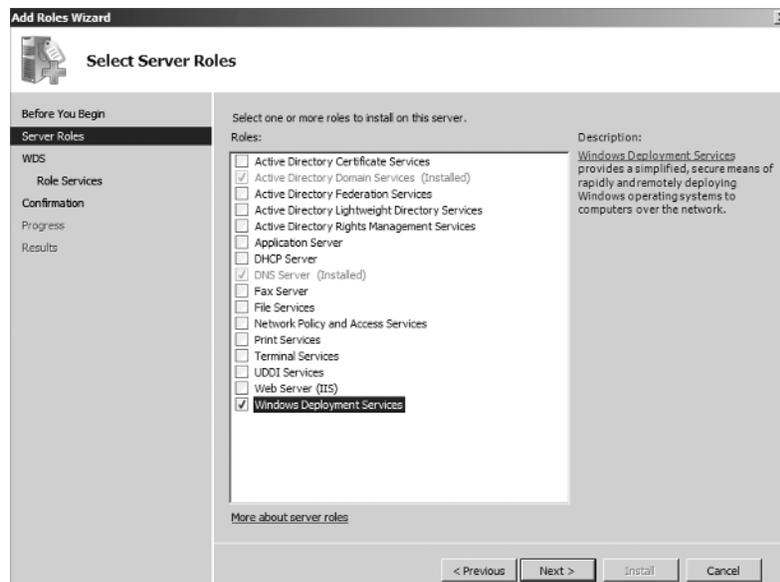
## Loading WDS

The following steps apply only to Windows Server 2008. For Windows Server 2003, you add WDS using Add/Remove Windows Components in the Control Panel.

If you did not add the WDS role when you created the server, you can load it now. Follow these steps:

1. Choose Start > Server Manager to open Server Manager.
2. Click Roles to open the Roles window.
3. Click Add Roles to start the Add Roles Wizard, as shown in Figure 1.1.
4. In the Roles list, click the Windows Deployment Services check box, click Next three times, click Install, and then click Close.

**FIGURE 1.1** Adding the WDS role is easy via Server Manager on a Windows Server 2008 server.



## Setting Up the WDS Server

Once WDS is installed and the server is rebooted, you're ready to fire up WDS and get started with initial setup.

1. Choose Start > All Programs > Administrative Tools > Windows Deployment Services.
2. In the WDS console, choose Expand Servers, right-click the DC01.corp.com server and select "Configure server." The first page of the WDS Configuration Wizard appears.
3. Walk through the wizard to answer the questions. Choose a folder on an NTFS volume that is not the system drive. In this example, I'm choosing R:\RemoteInstall. Click Next to open the "DHCP Option 60" screen. Note that you'll only see the "DHCP Option 60" screen if you have WDS and Microsoft DHCP on the same machine.
4. The WDS server needs to configure DHCP somewhat differently if the WDS machine is also a DHCP server. If it is, select "Do not listen on port 67" and "Configure DHCP option 60 to 'PXEClient'" when asked.
5. The WDS server must be configured to accept client connections. For our quick example, click the "Respond to all (known and unknown) client computers." For now, do not select the "For unknown clients, notify administrator and respond after approval."



Clicking the "Do Not Respond to Unknown Client Computers Requesting Service" check box lets you lock down a computer's GUID to a specific WDS server. Make this connection when you're manually adding a computer to Active Directory Users and Computers by selecting "This is a Managed Computer" and then entering the computer's 32-character-long GUID. You can find the computer's GUID in the computer's BIOS by using a WMI script (or you can use the MAC address of the network card instead) in the format of 20 zeros followed by the 12-character-long MAC address (hexadecimal); for example 000000000000000000000000A309CDE24601.

Once your server is initially configured, you need to introduce it to the images you want to deploy. Let's first get a grip on the kinds of image files available to us, and then we'll install the image types we need.

## Understanding WDS's Image Types

WDS has three types of images.

**Boot Image** When your target machine performs a PXE boot to connect to the WDS server, this is what will be run on the client to "get it going." This is a cut-down version of Windows called WinPE which helps you (stay with me here) load the big version of Windows in the "Install Image" section (coming up next).

**Install Image** This is the actual image you'll be downloading to your target machine. You might have one image for Sales, another for Marketing, and so on. However, a best practice is to have just one image which does it all. Note that images can be Windows XP, Windows Vista, Windows Server 2003, or Windows Server 2008.

**Legacy Image** If you want to make your Windows Server 2003 WDS machine act like RIS, this is the place to do it. If you've upgraded an existing RIS server with RIS-style images, they will appear here. Ultimately, you'll want to convert these to the .WIM file format. The WDS help file, under the topic "Legacy Images" gives tips for how to do just that.

To use WDS successfully, you will leverage these images. If you're going to use WDS in native mode, you'll start out by booting a boot image then loading an install image. If you're going to use WDS in legacy mode, you'll leverage legacy images only.

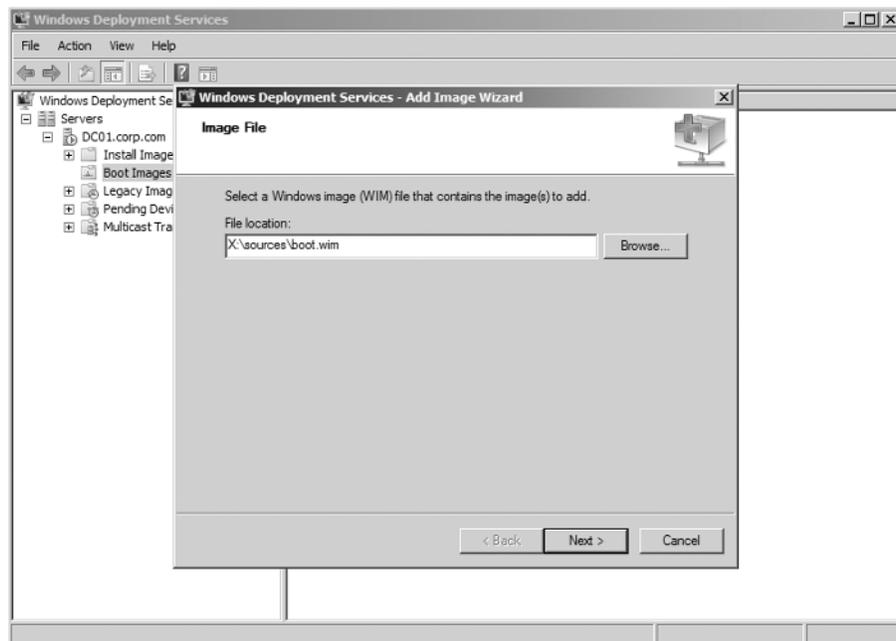


Only WDS in Windows Server 2003 supports legacy mode. See the "New Features for Windows Server 2008" section for more information on Windows Server 2008 support for legacy images.

## Adding the Boot Image

Let's start out by adding the boot image. Again, the boot image is a cut-down version of Windows to help you get started with your installation. Right-click the Boot Images node and select Install Image. The boot image you'll need is right on the Windows Server 2008 DVD in the \sources directory. Enter the path to the Windows Server 2008 DVD's \sources directory and specify boot.wim, as seen in Figure 1.2.

**FIGURE 1.2** The boot.wim file gets your clients started with WDS.



## 12 Chapter 1 • Deploying Windows with Style

The boot image you are using must match, or be newer than, the operating system image (install image) you want to deploy. That's why we are using the `boot.wim` file from Windows Server 2008 and not from Windows Vista.

Once you click Next, you'll be able to enter in the image name. I'm not wild about the default name Microsoft Windows Longhorn Setup (x86), so I've changed it to Vista Boot WIM (x86) so it's much clearer. Another perfectly good name is Windows Server 2008 Boot WIM (x86), since we are using that media.

### Adding the Install Image

Before we install the first image, you need to understand the concept of *image groups*. Image groups are like folders that contain similar images.

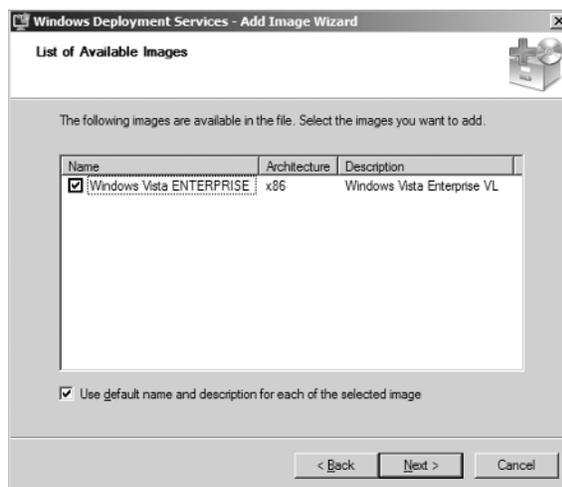
Here's the idea: within an image group, all the data is "single instance stored." That means if you have multiple images and they're very, very similar, you're only saving a copy of the differences. This is huge! Because the difference between the Sales and Marketing images might really be very small!

The other advantage of image groups is that you can specify who has access to different image groups. So, you can say that Server Installers have access to an image group for servers, but Desktop Installers have access just to an image group for desktops.

For our working examples, we're not going to have more than one image, so setting up multiple image groups isn't necessary. Follow these steps:

1. Right-click Install Images and select Add Install Image. You'll be prompted for your first image group. I'm naming mine VistaGroup, but you can call it anything you like.
2. In the Image File dialog, point WDS toward the file called `install.wim` in the `\sources` directory of the Windows Vista Enterprise DVD. When you do, you'll be presented with the "List of Available Images" screen, as seen in Figure 1.3.

**FIGURE 1.3** You can select which versions of Windows Vista you want to put in the image group.





Because the images in the image group are single instance stored, you're not burning 1.7GB for each Vista image. It's 1.7GB plus a little more for each image you add.

3. Click Next until the image is added. This could take a while to complete.

## Managing the WDS Server

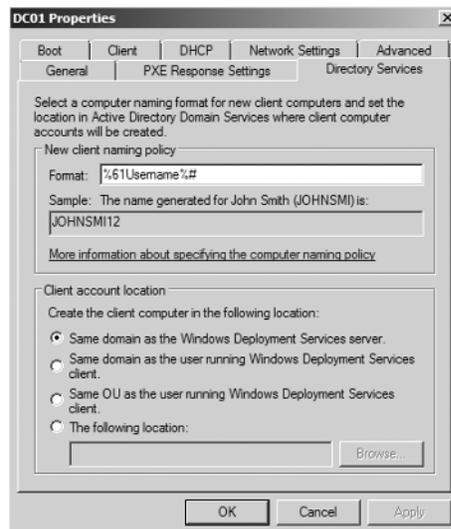
Before you roll out your first client, you might want to tweak WDS. To do this, right-click the server name (dc01.corp.com) inside the WDS management tool and select Properties. Inside, you'll find various tabs.



There is nothing configurable on the General tab. It is informational only.

**Directory Services** Space prevents us from diving into each option, but one you'll likely want to explore is Directory Services, as seen in Figure 1.4, where you can change the computer name of newly born computers as well as where to place newly born computers in Active Directory.

**FIGURE 1.4** You can customize some WDS defaults, such as the client's computer name and where to drop it into Active Directory.



## 14 Chapter 1 • Deploying Windows with Style

There are certainly lots of other options here, as well, including associating unattended installation files to your WDS client installations to make them hands off. This is done via the Windows System Image Manager (WSIM) tool to create the unattended files. Microsoft docs on the process can be found here: <http://tinyurl.com/y2sdbd>. The tool is found in the Windows Automated Installation Kit download (about 1200MB!) and can be found here: <http://tinyurl.com/37b42g>. You will find step-by-step guidance for associating unattended installation files in the “Beyond the Basics: Care and Feeding of WDS and Your Images” section later in this chapter.



In the past Windows Setup used multiple answer files, which are text files for different parts of the setup. Windows Setup used `unattend.txt` for scripted setups, `sysprep.inf` for automating Sysprep processes, `winbom.ini` in Windows PE and `cmdlines.txt` to perform operations during setup or Sysprep processes. All these text files have now been replaced by a single XML file (which is also a text file but a more structured one). And, to serve multiple purposes, it has been divided into different sections, or *configuration passes*. If you can't wait to learn more about this, you will find a section on how to create your own `unattend.xml` files in the section “Creating an Answer File Using WSIM.”

**PXE Response Settings** This tab instructs the server how to respond to known and unknown clients. A known computer is one that already has a record in Active Directory. When you create a new computer account in Active Directory, you can prestage it using the wizard when creating new computers. See Figure 1.5.

**FIGURE 1.5** Set how a server should or should not respond to client requests.



**Boot** This tab is the PXE program that's downloaded right after you press F12 for a network boot. You can use this tab to automate the PXE boot or select the default boot image.

**Client** Customizations in this area affect all clients when loading via WDS. You can basically customize the WDS Client installation here.

## WDS Specifics for Windows Server 2008

Windows Server 2008 brings a new version of WDS. In this section you will learn the new features of WDS in Windows Server 2008 compared with the Windows Server 2003 version.

Before digging into the details about the new features, let's peek into the architectural changes. WDS in Windows Server 2008 still has the same network requirements as the Windows Server 2003 version (remember: DHCP, DNS, and Active Directory).

**Server Components** The server components are pretty much the same as in the Windows Server 2003 version (a shared folder, a PXE server, and a TFTP server).

**Client Components** A new multicast client has been added to the client components (WDS client and PXE boot ROM).

**Management Components** The management components are same (WDS Console and WDSUTIL).



### Real World Scenario

#### Real-World Test Results

Here are some test results when deploying a 3.5GB Windows Vista image in a 100Mb connection (server connected using a gigabit connection):

- To 25 machines with multicast, 30 minutes
- To 25 machines with SMB (unicast), 30 minutes
- To 50 machines with multicast, 30 minutes
- To 50 machines with SMB (unicast), 43 minutes

Because we cannot multicast boot images and can only install images, this will be a limiting factor. Booting the Windows PE boot image takes about 50 seconds for 25 clients.

## New Features for Windows Server 2008

Here is a detailed walkthrough of the new features:

**Ability to Transmit Data and Images Using Multicast** WDS in Windows Server 2008 now has support for multicast, and not only that, but two different "versions" of multicast. We'll

**16** Chapter 1 • Deploying Windows with Style

explore the multicast features in the section titled “Utilizing Multicast Deployment with WDS and Windows Server 2008.”

**Increased Efficiency** The WDS keyword here is efficiency, not speed. Multicast is not the fastest method for deploying, say, 25 clients on a 100Mb network—it actually takes about the same amount of time as with SMB (unicast). The difference is network utilization. Multicast will load the network about 5 percent, SMB about 95 percent.

This means that you may no longer be restricted to deploying your machines at nonpeak hours.

**Standalone Server Support** When installing the WDS role in Windows Server 2008, you can choose to install only the transport server role service. This is for advanced scenarios; for example, when you don’t have Active Directory Services present or a DNS or DHCP.

The cons to this are that Windows Server 2008 contains only the PXE listener, which is the server-side component. You will need to write a custom PXE provider and register it with WDS (see the Windows Server 2008 SDK for more information).

**An Enhanced TFTP Server** WDS in Windows 2003 scales well up to 75 clients on a 1GB network; a slower network reduces scalability further. With multicast WDS in Windows Server 2008, WDS easily scales to 250 clients.

**Network Boot x64-based Computers with EFI** EFI is the next generation BIOS, or rather a firmware layer between BIOS and the operating system.

**Enhanced Diagnostics and Reporting** WDS now logs detailed information about its clients. The information is published to the event log. You will find the events under Application and Services Log > Microsoft > Windows > Deployment-Services-Diagnostics.

These logs can be parsed for creating metric reports and can answer questions such as, “Which images are used most frequently?”



Logging for standard WDS client deployments is not enabled by default. See the “Troubleshooting WDS” section for information on how to enable this.

**Ability to Deploy Windows Vista and Windows Server 2008** That says it all. You can now use the same processes and tools to deploy both servers and clients.

**No Support for RISEUP Images or OSChooser Screens** Support for legacy image types and the OSC files is gone. This means that you need to convert any existing legacy images to native WIM images before upgrading. It also means you need to use Windows PE 2.0 for boot images, which requires 512MB RAM on all your client machines.

## Installing and Managing Clients via WDS

You’re almost ready to start rolling out your clients. Remember that your clients need network cards that are PXE-boot ROM-capable. To use the NICs that have the ROM code built right onto the card, watch for the PC to flash “Hit F12 for Network Boot” upon reboot. If your

computer doesn't flash that message (or something similar), you'll need to check the network card's BIOS, the PC's BIOS, or both to see if the PXE feature is disabled.

## Installing Your First Client

Once you're sitting down at your target machine, you're ready to install your first client.

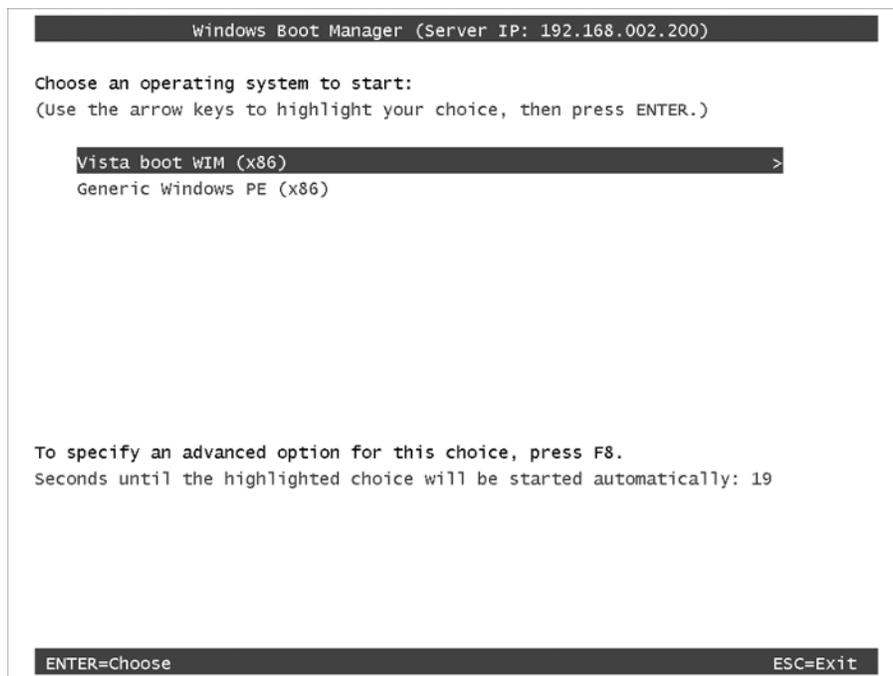


Running WDS on your workstations completely formats the first hard drive.

To use WDS to install a client, follow these steps:

1. Turn on the target system.
2. When prompted, immediately press F12 to see the boot menu, as seen in Figure 1.6.

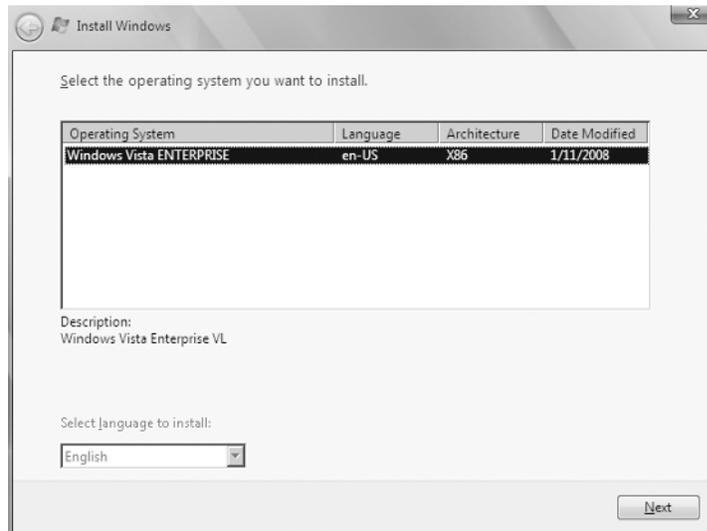
**FIGURE 1.6** Start WDS by pressing F12 and selecting which boot option you want.



3. Select the boot WIM you want. (Maybe you have created the “Vista Boot WIM with Custom Network Drivers” using the information in the “Booting New Clients with VMware” Real World Scenario).

**18** Chapter 1 • Deploying Windows with Style

4. It will take some time to download the `boot.wim`. That's because the `boot.wim` is about 120MB, so be patient. When the first information screen appears, press Enter to open the Client Installation Wizard Logon screen.
5. The first WDS screen you'll see is called WDS and you simply select a Locale (like English/United States) and a Keyboard Input Method (US). Click Next.
6. Enter a valid username, password, and domain. You'll have to enter it as `DOMAIN\username` or `username@domain.com`. Just about any username and password combination will work, like Frank Rizzo and our default password (if you chose to use it) of `P@ssw0rd`. In other words, users don't need to be (nor should they be) Domain Administrators to perform this function.
7. You can then select the Windows Vista image you want to install, as seen in Figure 1.7.

**FIGURE 1.7** Select the Windows Vista image you want to deploy and select Next.

8. You'll then be able to partition the hard drive if you like; click Next, and you're off to the races.

If all goes well, the computer will be left at a logon prompt, waiting for the user to log on for the first time.

### Converting Existing RIS Images to WDS Images

If you chose to upgrade your RIS server to a WDS server, you'll see your old RIS images in a folder called `Legacy Images`, as you saw in Figure 1.2 (though in that figure, we're not expressly showing the contents of the `Legacy Images` folder).



## Real World Scenario

### Booting New Clients with VMware

I do a lot of testing using VMware Workstation and Server. It's a great cheap way to emulate a lot of computers quickly. It's also great for when I need to take screenshots of the boot process. However, if you are not using the latest VMware Workstation release (version 6.0 as of this writing) when you try to spawn a new Windows Vista machine via WDS, it turns out that the network card that VMware emulates isn't part of the boot .wim (the file that's downloaded before you kick off WDS).

There are two options to getting older VMware Products to work with WDS: configure VMware to emulate an Intel 1000 MT adapter that is supported by default by Windows PE 2.0, or add the VMware Network Drivers to Windows PE 2.0 (boot .wim).

#### Implementing Option 1: Changing a VMware Client to Use Intel 1000 MT Adapters

Add the following information to your VMX file:

```
ethernet0.virtualDev = "e1000"
```

Remember that the latest release of VMware Workstation will automatically add this value if you select Windows Vista as your operating system when you're creating the virtual machine.

To add new or updated drivers you need the WAIK (Windows Automated Install Kit, found at <http://tinyurl.com/37b42g>), which can help you jam in the VMware network card driver. The tool you'll leverage is the `imagex.exe` utility, which can crack into an existing .WIM file, mount it in the file system, and allow you to put new drivers right into the file. Then, once the driver injection is completed, you can sew it up and use those drivers!

#### Implementing Option 2: Adding the VMware Network Drivers to Windows PE

1. Install the WAIK to the default location.
2. Create a new directory on your hard drive. This will be a temporary directory to "mount" the boot .wim file. For this example, I will use C:\6000 (the build number for Windows Vista). While you're here, create a subfolder under that called Mount (so the final path is C:\6000\Mount).
3. Copy the boot .wim file from the Windows Server 2008 DVD (in the \sources directory) to the C:\6000 folder.
4. The point is to jam in the network drivers you'll need that WDS doesn't natively support. In my case, I need the VMware drivers. To get them, leverage an ISO reading program (such as VCD, DaemonTools, WinISO, and so on) to mount the C:\Program Files\VMware\VMware Workstation\Windows.iso file. You could also grab these files while telling VMware Workstation (or Server) that you want to "Install VMware Tools" from the VM menu. You'll then be able to see these files to copy (and you don't actually have to reinstall VMware tools if you already have them installed).

## 20 Chapter 1 • Deploying Windows with Style

5. From that ISO, copy the following: <root>\Program Files\VMWare\VMWare Tools\Drivers\vmxnet\win2k. Copy all the files in that folder to a new directory: C:\6000\vmxnet.
6. Now it's time to use imagex.exe from the WAIK. Start the Windows PE tools command prompt (a standard Windows command prompt but with the path set to the different WAIK tools, like ImageX). Type: **imagex /mountrw c:\6000\boot.wim 2 c:\6000\mount**. Note the number 2 between boot.wim and c:\6000\mount. This designates the second index within the boot.wim.
7. In the command prompt, execute the next two lines, which will jam in the drivers and commit the drivers into the WIM:
 

```
peimg /inf=c:\6000\vmxnet\*.inf c:\6000\mount\windows
imagex /unmount /commit c:\6000\mount
```
8. Now your boot.wim will have the network drivers it needs inside the second image (the one Windows Vista uses when setup runs).
9. Now, use the WDS server manager to right-click the Boot Images folder and select Add Boot Image. Import your newly updated boot.wim (c:\6000\boot.wim) into your WDS server and give it a unique name, such as Vista Boot WIM with Custom Network Drivers (x86).

Before upgrading to the Windows Server 2008 version of WDS, you need to first upgrade your RIS Server to WDS (Windows Server 2003 version) and convert your existing legacy images. WDS needs to be in native mode in order to be upgraded to the Windows Server 2008 version.

Existing RIPREP images can be converted offline. RISETUP images need to be deployed and recaptured (which of course you can do with RIPREP images, as well). We will deal with how to do this in this section.

### Converting Existing RIPREP Images to .WIM Offline

If you chose to upgrade your RIS server to a WDS server, you'll see your old RIS images in a folder called Legacy Images. Some of these could be RIPREP images, which are RIS images you made that already had the base applications inside the image. There's an alternate way to convert these to WIM format.

The command line `wdsutil.exe` (which is on your Windows Server 2003 WDS server) can convert an RIS image into a .WIM file. Your legacy RIS images are likely in a directory called:

```
<driveletter>\RemoteInstall\Setup\English\Images\<imagename>\i386\templates\
<something>.sif
```

For example:

```
R:\RemoteInstall\Setup\English\Images\WindowsXPSP2\
\i386\templates\myrippedimage.sif
```

There will be additional SIF files for every RIS image with an answer file.

So, to convert the RIPREP image directly to WIM format, we'll leverage the `wdsutil /convert-riprepimage` command.

Now, a quick word about the `WDSutil.exe` command before we actually run the thing. It depends upon some libraries and DLLs in the WAIK tools, but those .DLLs aren't in the path in the standard Windows Command Prompt, so you need to start the conversion from the `C:\Program Files\Windows AIK\Tools\Servicing` folder. Once you started the Command Prompt, here are two example commands that should do the trick:

```
Cd /d "Program Files\Windows AIK\Tools\Servicing"
wdsutil /verbose /progress /convert-RIPREPIImage /FilePath:"R:\remoteinstall\
setup\english\images\WinXPSP2ProImage\i386\Templates\riprep.sif" /
DestinationImage /FilePath:"R:\Windows XP SP2 Pro Eng X86.wim" /Name:"Windows
XP Pro SP2 Eng x86 Image" /Description:"Windows XP Pro SP2 Eng x86 Converted
RIPREP Image"
```



The command lines beginning with `wdsutil` should be entered as one command, they have been wrapped for readability.

Again, this trick only works for RIPREP images, not flat (RISETUP) images where you just gave it the CD. For flat images, you need to deploy the image, run Sysprep, and then run the Capture Wizard.

## Capturing Windows XP and Windows Vista Machines for WDS Deployment

You might have a Windows XP machine (or Windows Vista machine for that matter) that you want to deploy via WDS. But what about your applications? How you will get those onto the target systems?

You need to choose to do one of the following:

- Put your applications inside your WDS image. This is called a *thick image*.
- Have technologies like Group Policy Software Installation (GPSI) and/or assign application using tools from Microsoft Deployment Toolkit 2008. This is called a *thin image*.

Microsoft Deployment Toolkit 2008 (which is discussed in greater detail later in the “Microsoft Deployment Toolkit 2008” section in this chapter) allows you to assign applications per role, per hardware type, or per location (IP subnet).

You may also combine the two application deployment options; for example, you can have a few applications inside the image, like antivirus software and maybe some updates to the operating system like Windows Media Player 11 or Internet Explorer 7.0, and then deploy the other applications afterward.

On the one hand, it's certainly faster to load an application, such as Office 2003 or Office 2007, inside the WDS image and then deploy the image all at once rather than deploying a base WDS image and then using GPSI to shoot down Office 2007. The GPSI features have the added ability to upgrade packages and perform magic such as applying .MSP (Microsoft Patch) and .MST (Microsoft Transform) files into packages; these abilities are lost if the applications are embedded inside the WDS image.

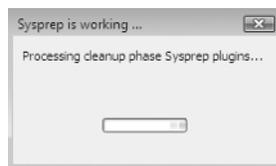
## 22 Chapter 1 • Deploying Windows with Style

Therefore, you'll need to analyze each application to determine if it's better to embed it inside the WDS image or deploy the package afterward. In my experience, in almost all cases it's better to deploy the applications after the image is installed. This makes it much easier to handle application upgrades and so on, or to remove applications from the deployment. If you do choose to embed applications in your WDS images, I'm presenting that information here. Again for the record, however, *I encourage you start working with thin images.*

After you install the applications on your target machine, you need to first run Sysprep on it, which prepares the machine for imaging. However, you won't be imaging it; you'll be capturing it, using the tools in the next steps.

We don't have room to go into all the Sysprep steps here, but you can see a screenshot of Windows Vista when it is being Sysprepped in Figure 1.8. However, if you want a quick Sysprep primer, check out the WAIK documentation.

**FIGURE 1.8** You need to have a Windows Vista machine (or Windows XP) Sysprepped in order to capture the installed machine to an image.



### Creating Your Capture Boot Image

To get started, you need to first create a *capture image* and introduce that into your WDS boot images. The idea is that once you press F12, then boot from the capture image, you'll be able to push any machine up (Windows XP, Windows Server 2003, Windows Vista, or Windows Server 2008) that you want to deploy via WDS. Again, these machines can have no applications or a zillion applications preloaded (although, as I've said, I'm not a fan of preloading applications).

The capture boot is required so that the target machine can find a WDS server and push the existing machine up into an image. You'll leverage an existing boot image to create your capture boot image. Simply right-click one of your boot images and select Create Capture Boot Image as seen in Figure 1.9.

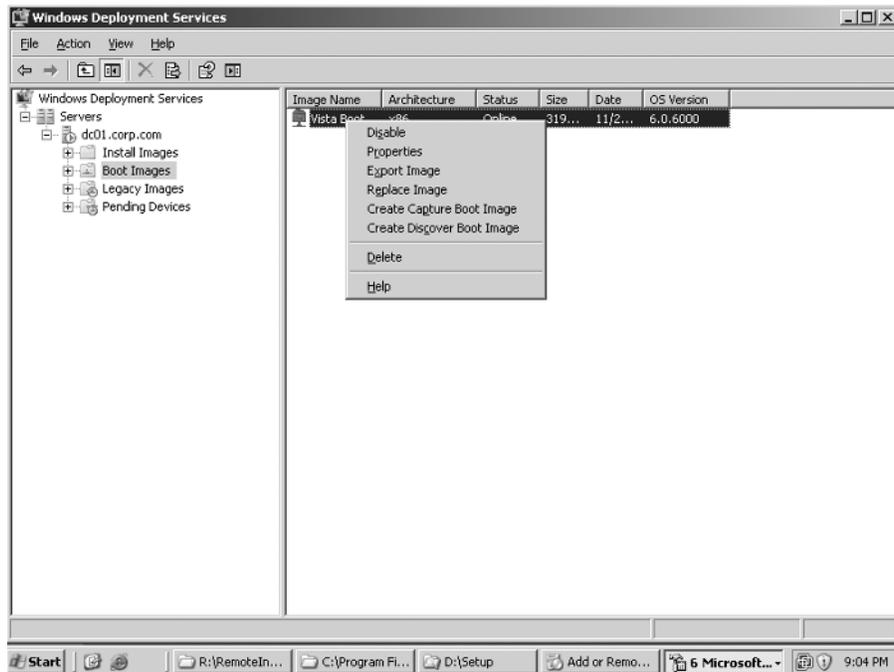
You'll be prompted for a name and location. Select any temporary location for now. Once the capture boot image is completely written, right-click Boot Images and suck it in from this temporary location. It's a bit convoluted, but it gets the job done.

When complete, you should see your new capture image as listed in Boot Images.

### Leveraging Your Captured Boot Image

Let's assume you have a Windows XP machine you want to make as an available install image. Again, you have to have pre-Sysprep the machine in order to go on to the next steps. Without having Sysprepped your Windows Vista machine (or Windows XP, for that matter), the Capture Wizard (described next) won't be able to see the partition.

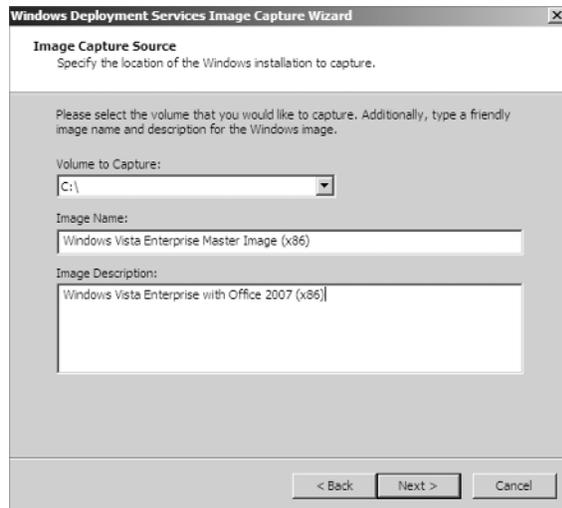
Once you Press F12 and select your new capture boot image, you'll be presented with the screen seen in Figure 1.10.

**FIGURE 1.9** Right-click an existing boot image to create a capture boot image.**FIGURE 1.10** The WDS Capture Wizard can upload a Sysprepped Windows Vista machine.

## 24 Chapter 1 • Deploying Windows with Style

When you click Next, you'll be able to select a drive letter to capture, as well as name the image. I'll be calling my image Windows Vista Enterprise Master Image (x86), as seen in Figure 1.11.

**FIGURE 1.11** You can name your image anything you like.



You'll then be able to specify the name of the server and the image group to plunk this in, and you're ready to go.



The WDS capture utility (different from ImageX) creates the WIM image locally first and then copies it to the server. This is to address slow or high latency networks.

This image will then be available for deploying using WDS.

## Utilizing Multicast Deployment with WDS and Windows Server 2008

The number one new feature for WDS in Windows Server 2008 has to be the multicast feature. And Microsoft didn't only give you multicast, they gave you two versions of multicast! (Don't worry, I cover both versions.)

Multicast should be used when you require many concurrent client installations and/or when you want to use the bandwidth efficiently. With multicast, you can limit the multicast deployments to a small fraction of your overall bandwidth.

Multicast does have some system requirements, and as I mentioned, the Microsoft implementation gives you two different multicast methods to choose from.

**Requirements** Multicast is a very powerful ally. To use it on your network, your routers (if you have any) need to support multicasting. The needed feature is support for Internet Group Management Protocol (IGMP). Without this feature your routers might treat the multicast packages as broadcast packages and your network may become very busy or flooded.

You also need a boot image that is only available from Windows Server 2008. You cannot use the Windows Vista boot image for multicast.

**Auto-cast** Auto-cast is the first of the multicast methods; it's like an ongoing stream of an image, present on the network as long as clients request it. A multicast client can join into this stream at any time. This is really neat when you see it in action.

Here's an example: You start the download for client 1, and, as suspected, it will start downloading data from the stream at the initial mark (0 percent). Then, later if you want to start client 2 downloading, but client 1 is now at 20 percent, client 2 just starts at 20 percent, finishes at 100 percent, and then goes back and gets the first 20 percent it missed! Wow!

**Scheduled-cast** This is more like the multicast standard that has been around for years. You define how many clients you want to deploy and when the server "sees" that all clients have checked in the multicast deployment process starts.

## Setting Up Auto-Cast Multicast Transmission

Creating an auto-cast multicast transmission is easy. You can choose between the management GUI, the WDS console, or the command-line management utility, WDSUTIL. In this example we will use the WDS console.

### Adding the Boot Image

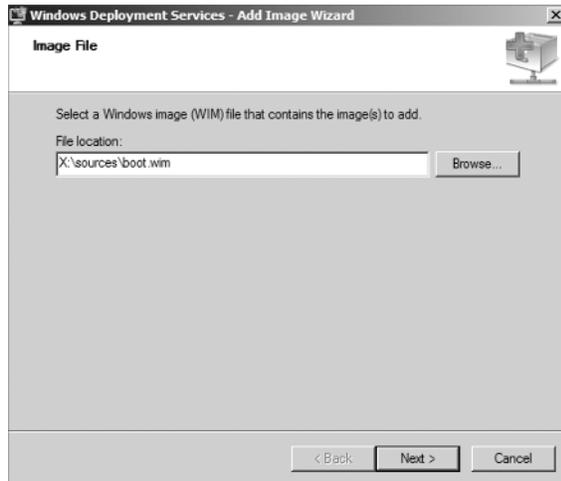
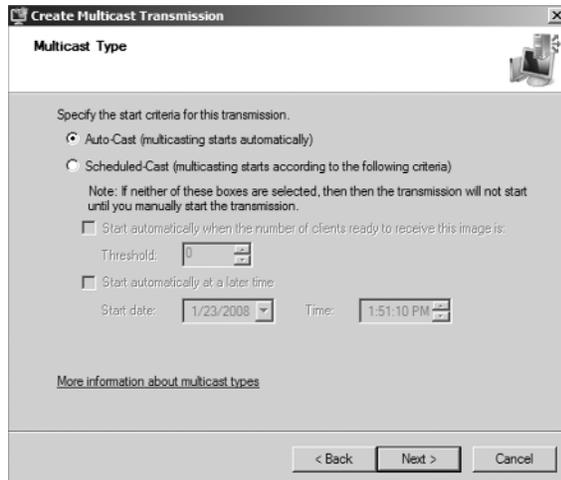
Remember, you'll need a Windows Server 2008 boot image for the multicast support. If you have not added a Windows Server boot image already, now is the perfect time to do it.

We already did something similar when we set up the WDS server earlier in this chapter, but this time, we'll use the boot image from Windows Server 2008. Since Windows Server 2008 and Windows Vista can use the same setup engine, this makes things easier. You will find the boot image in the same location on the DVD: `\sources\boot.wim`, as shown in Figure 1.12.

### Creating the Transmission

In order to start deploying Windows using multicast, we need to create a transmission on the server that the client can connect to. Follow these steps:

1. Right-click the Multicast Transmissions node and select Create Multicast Transmission. This will start the Create Multicast Transmission Wizard.
2. On the Transmission Name page, enter a name (I entered Windows Vista Enterprise Eng x86) and click Next.
3. On the Multicast Type page, select the "Auto-Cast (multicasting starts automatically)" option and click Next. See Figure 1.13.
4. On the Task Complete page, click Finish.

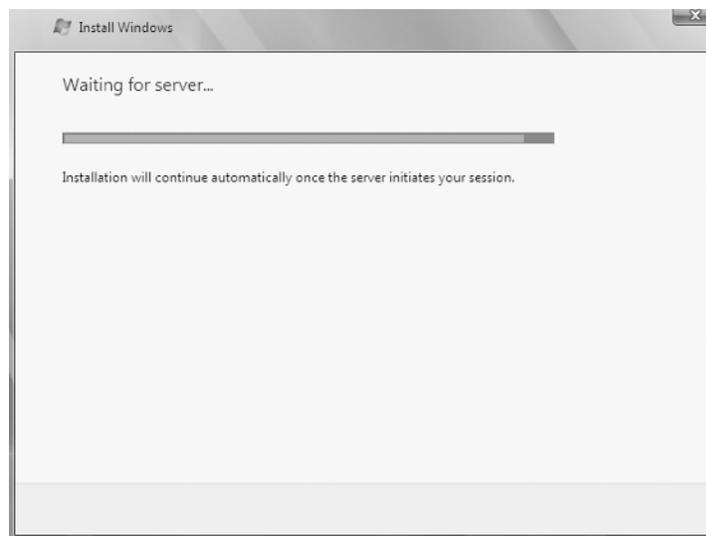
**26** Chapter 1 • Deploying Windows with Style**FIGURE 1.12** Adding a Windows Server 2008 boot image**FIGURE 1.13** Selecting Multicast Type**Installing Your First Client Using Multicast Transmission**

Now it is time to start deploying our clients, as described in the following steps. The server transmission is set up and we only need to PXE-boot the client and it will download the boot

image from the server. The boot image will then connect to our transmission and start downloading our install image.

1. Turn on the target system.
2. When prompted, immediately press F12 to see the boot menu.
3. Select the boot WIM you want.
4. When the first information screen appears, press Enter to open the Client Installation Wizard Logon screen.
5. The first WDS screen you'll see is "Windows Deployment Services." Select a Locale (like English/United States) and a Keyboard Input Method (US). Click Next.
6. Enter a valid username, password, and domain. You'll have to enter it as **DOMAIN\username** or **username@domain.com**.
7. Select the Windows Vista image you want to install.
8. You'll then be able to partition the hard drive if you like. Click Next. When the client connects to the multicast stream, you will see the screen shown in Figure 1.14. After that, if all goes well, the computer will be left at a logon prompt, waiting for the user to log on for the first time.

**FIGURE 1.14** Connecting to the multicast transmission



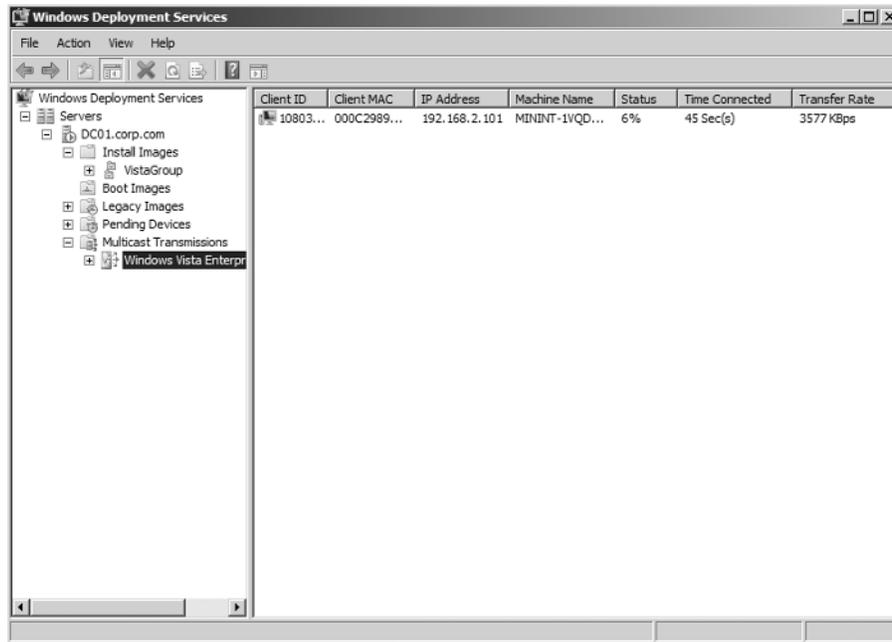
### Monitoring Your Multicast Transmission from the Server

After the client connects to the multicast transmission, it will download the image locally and start the Windows Vista Setup, so using multicast actually requires a bit more disk space than unicast does. (Unicast expands the image directly from the server.)

## 28 Chapter 1 • Deploying Windows with Style

You can see how your client progresses in the WDS console. You will see information like the MAC address, IP address, status (percentage complete), and transfer speed, as seen in Figure 1.15.

**FIGURE 1.15** One multicast client



### Installing More Clients Using the Existing Multicast Transmission

What if you want to install more than one client? That's easy! Just repeat the steps in the "Installing Your First Client Using Multicast Transmission" section. Then the second client will connect to stream at the current status. Figure 1.16 shows two clients downloading the image from the server using the same transfer speed. With multicast, it's the slowest client that determines the speed.

### Managing Your Multicast Transmission from the Server

I just told you that the slowest client determines the speed of the download. So what if we have one super-slow client affecting the overall transmission to the other clients?

We can either disconnect the client that is slowing things down, or we can bypass the multicast transmission for that client. This is done by right-clicking the client's entry in the console and selecting "Bypass multicast," as seen in Figure 1.17.

This will not stop the download but will instead configure that client to download the image using unicast.

FIGURE 1.16 Two multicast clients using auto-cast mode

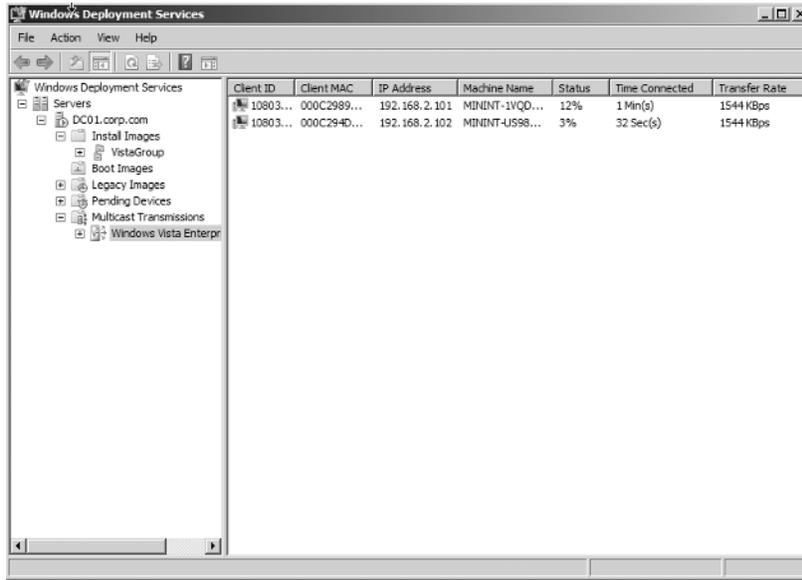
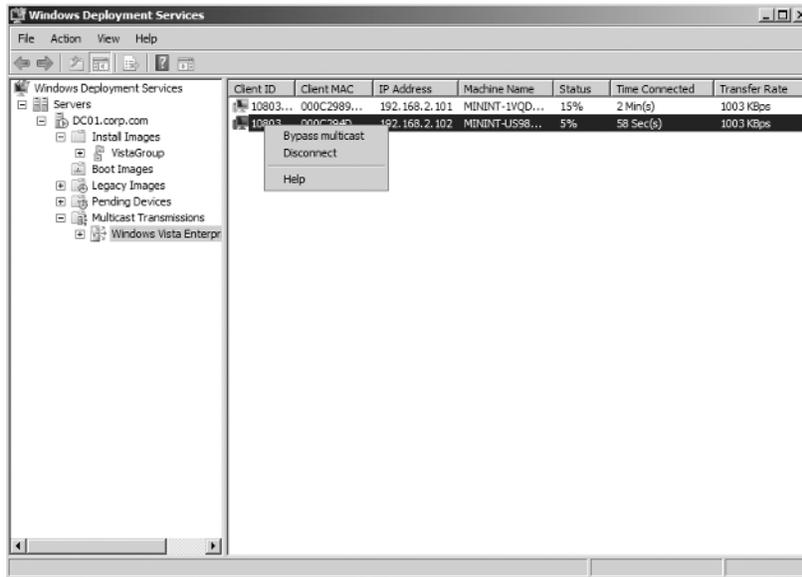


FIGURE 1.17 Bypassing a client



## Configuring Multicast Transmission

Now that you have learned to create, monitor, and manage your transmissions on a per-client basis, it is time to configure the actual transmission.

Remember, clients join a transmission of a stream when they're installed during a multicast session. By using the transmission options, you can control all the connected clients at one time. Right-clicking the transmission gives you these options:

- Starting it. This option is only valid if the transmission hasn't started, such as when you are using the schedule-cast method. You can also enable clients to join in.
- Deleting it. This option will force all clients to continue the download using unicast.
- Deactivating it. This allows ongoing clients to complete their download using multicast but does not allow new connections.
- Viewing the properties. This option allows you to view the properties of the transmission, but you cannot edit the transmission after it has been created. To do that, you would need to delete it and re-create it.

## Multicast Server Properties

In addition to managing the transmissions, we can configure other multicast settings such as bandwidth, IP address range, and so on. To configure those settings, right-click the server and select Properties, then go to the Network Settings tab, shown in Figure 1.18. In the Network Profile section, select the network speed. Each profile is tuned to match each specified network speed (such as window size, cache size, block size, and so on). You can create a custom profile or view the default profile settings in the Registry at `HKLM\SYSTEM\CurrentControlSet\Services\WDS\Server\Providers\WDSMC\Profiles`.



If you have multiple WDS servers on your network running multicast transmissions, make sure to give them different IP address ranges. Don't forget to restart WDS after changing the IP address range.

## Beyond the Basics: Care and Feeding of WDS and Your Images

By this point, you've got the WDS basics down. You can create your own new Windows Vista clients and you can zap your existing Windows XP machines up into .WIM files and into the WDS server—and you can zap them down, too. But there are several ways to go beyond the basics of WDS. That's what this section is about.

One of the key features of WDS and the WIM format (and the resulting .WIM files) is that you can maintain an image—even after it's captured. The result is that if you have a mere driver update or a hotfix to add, it's a piece of cake to add those to an existing image.

**FIGURE 1.18** Multicast server properties

The original idea with Offline Servicing was that you also should be able to apply service packs this way; however, that is not the case, at least not with Windows Vista SP1.

This idea of maintaining an image once it's captured is really only for Windows Vista and Windows Server 2008 WIMs, even though we've explored how to also capture Windows XP as a WIM. We'll spend most of our time here talking about how to perform these maintenance steps for Windows Vista WIMs.

Additionally, we'll cover how to associate an answer file with a WIM image, how to set up a larger WDS infrastructure, and finally, how to speed up WDS (under some specific circumstances).

## Understanding Image Groups

As we've already discussed, WDS maintains WIM images in image groups. Image groups serve two functions: performing single instancing and maintaining security.



The WIM file itself is a unit of single instancing. A single WIM file can contain multiple images. However, putting multiple images inside a WIM file isn't recommended if you use WDS, because WDS image groups already perform the function of performing single instancing. So, even though you can put multiple images inside a single WIM file, why bother? WDS's image groups go the extra mile and do the work for you.

### Single Instance Storing

Windows Vista is about 1.7GB compressed on the DVD, so you might think that if you had seven versions of Windows Vista in an image group, that image group would be as large as 12GB! But it's not.

Again, that's because WDS will single-instance store all the images. Think about it: the differences between each Windows Vista version is ever-so-slight, and those differences are all that's captured between the various images.

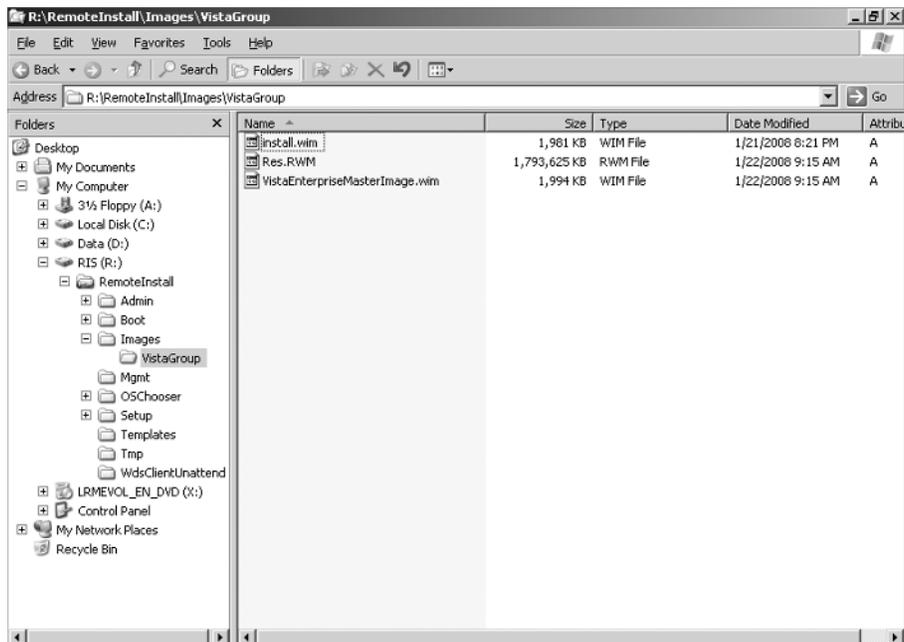
If you dive into the actual image store via command line or Explorer, you'll see the images as various .WIM files with a big ol' Res.RWM file alongside them. Ninety-eight percent of all the images are contained inside that big file. The rest of the files are stored as the differences inside the remaining .WIM files.



The Res.RWM file is really a WIM file, too. Technically, however, it's called a *resource WIM* and so is distinguished by a different filename.

You can see a list of my available images inside my VistaGroup image group and the corresponding underlying files in Figure 1.19.

**FIGURE 1.19** WDS will perform single instancing inside an image group.



## Setting Security for an Image Group

In addition to single instancing, the benefit of image groups is the ability to set security on them. By default, anyone who provides credentials when booting from a boot WIM can see the available images for download, format their machine, and get a fresh image.

I'll let that last sentence sink in for a while.

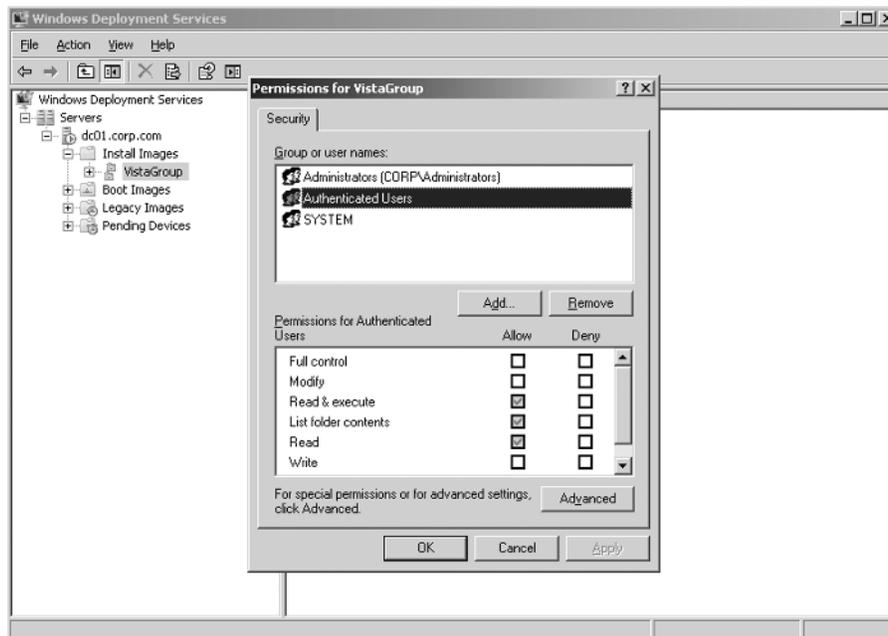
Go ahead and reread that if you need to.

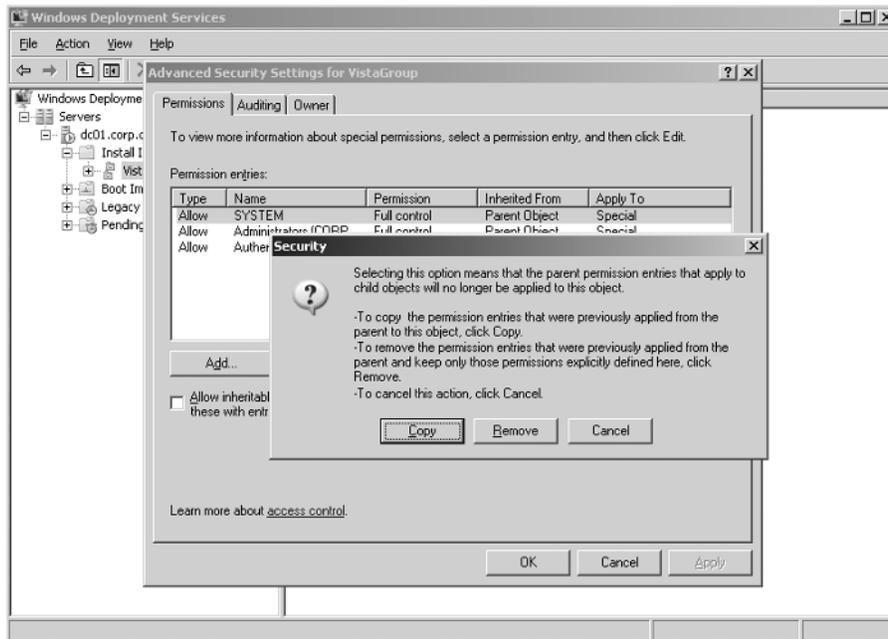
The key word is *anyone*; specifically, Authenticated Users. This means that if you want to ensure that only the right people have access to WDS install images, you need to set permissions on an image group.

To do this, right-click the image group, say, VistaGroup, and select Security. Remove Authenticated Users' ability to "Read & execute," "List folder contents," and "Read." Add a group of people you trust, like DesktopAdmins or something similar, and give them those rights. You can see the default image group rights in Figure 1.20.

You'll need to get rid of Authenticated Users and add the group you want. However, Authenticated Users is being inherited at a higher level, so you'll also need to click the Advanced security button and uncheck the "Allow inheritable permissions from the parent to propagate to this object and all child objects" check box, as partially seen in Figure 1.21 (the Security pop-up that displays when you check this box is covering it). On the Security pop-up, choose Copy, which will copy the permissions from the parent, also seen in Figure 1.21.

**FIGURE 1.20** By default, users can read and leverage the list of images inside an image group.



**FIGURE 1.21** Remove the inheritable permissions and copy them from the parent.

You'll be able to remove Authenticated Users and select just the users and groups you want. When a user logs on using their credentials, and the WDS client will present only the images you've enabled them to see.

## Servicing a WDS WIM Image

One really nice feature of Windows Vista is the ability to offline service drivers, language packs, and hotfixes. So when new hotfixes or security updates arrive, you'll want to just jam those hotfixes or that service pack into your existing image.

When you're done, you can either save the image out to another image (preserving the original one), or overwrite the original one. Here's the short version of how to do this:

1. Use the WDS MMC snap-in to mark the image as disabled (just right-click the image name and select Disabled).
2. Right-click the disabled image and select Export Image to export the image to a whole .WIM file. This can take a while.
3. Mount the image using the `imagex.exe` utility. Usually, the command will be `imagex /mount /mount:rw <image.wim> 1 <mountpoint>`, where `<mountpoint>` is a directory somewhere you can write to.
4. Service the image with the hotfix or driver. An example of how to add a driver can be found in the section, "Adding Drivers to an Existing Image."

5. Import the image back into the image group. If you want to completely replace the image, right-click the existing image and select **Replace Image**. If you want to add it as a new image while preserving the existing one, right-click the image group and select **Add Install Image**.

## Adding Drivers to an Existing Image

With WDS you can easily add in new network, audio, video, and modem drivers by configuring the Vista setup to get the drivers from a folder during setup, or you can inject them offline into the image (offline servicing). Please note that only Windows Vista and Windows Server 2008 support offline servicing.

### Option 1: Add Drivers Automatically During Windows Vista Setup

You can configure WDS to automatically inject drivers during setup. To do this, you create subdirectories with specific names within your image group's directory structure. In my examples, all the WDS stuff is being stored in `R:\RemoteInstall` in the `\images` directory. My first (and only) image group is named `VistaGroup`. To add drivers to your image, follow these steps.

**Step 1: Prepare the Folder Structure and Add the Drivers** First, you create the folder structure needed for adding drivers to the Windows Vista Enterprise setup:

1. Create a subdirectory with the same name as the `.WIM` file, say, `Install`.
2. Then, create a `$OEM$` subdirectory and within that subdirectory create `\$1`, as seen here:

```
\Images
  \VistaGroup
    RES.RWM
    Install.WIM
    \Install
      \OEM$
        \$1
          \Drivers
            \NIC
            \Audio
            \SCSI
```

3. Copy the drivers to the corresponding folders.



The `$OEM$` structure is also used for `sysprep.inf` when working with down-level operating systems like Windows XP or Windows Server 2003.

**Step 2: Create the Answer File** Next, you use Windows System Image Manager to create an answer file containing information about which driver path the Windows Vista setup will

**36** Chapter 1 • Deploying Windows with Style

inject drivers from. Remember from the Windows deployment terms section that WSIM is an authoring tool for editing `unattend.xml` files, the replacement for Setup Manager used with legacy operating systems.

1. Using WSIM, in the Windows Image pane, right-click the “Select a Windows Image or catalog file” node and click Select Windows Image.
2. In the “File name” text box, type `C:\6000\CustomVistaEnterprise.wim` and click Open. In the Windows System Image Manager dialog box, click Yes to create a catalog for the image.
3. From the File menu, select New Answer File.
4. In the Windows Image pane, expand Windows Vista ENTERPRISE, expand Components, expand `x86_Microsoft_Windows_PnpCustomizationsNonWinPE_6.0.6000.16386_neutral`, and expand `DriverPaths`.
5. Right-click `PathAndCredentials` and select “Add Settings to Pass 2 offlineServicing.”
6. In the PathAndCredentials Properties pane, in the Key field, type `Drivers`, and in the Path field, type `C:\Drivers`.
7. From the Tools menu, select Validate Answer File.
8. From the File menu, select Save Answer File.
9. In the “File name” text box, type `R:\ImageDrivers.xml` and click Save.

**Step 3: Associate the Answer File with the Image** Next, for Windows Vista setup to pick up and use your previously created answer file, you need to associate it with your Windows Vista Enterprise Image in the WDS Console:

1. Using the WDS Console, expand `Install Images`, and select the `VistaGroup` image group.
2. Right-click your Windows Vista Enterprise image and select Properties.
3. On the General tab, select the “Allow image to install in unattended mode” check box. Click Select File, browse to `R:\ImageDrivers.xml`, and click OK.

Once you add the answer file, it will be renamed `ImageUnattend.xml` and copied to the `Unattend` folder inside the image folder, for example, `R:\RemoteInstall\Images\VistaGroup\Install\Unattend`.

When starting the deployment, the `Drivers` folder will be copied from the `$OEM$\$1` folder on the server to `C:` on the client after the image is applied to disk but before the first reboot.

**Option 2: Inject the Drivers into the Actual Image**

Injecting drivers into the actual image involves exporting the WDS resource image to a standard WIM image, mounting it using ImageX, creating an answer file with WSIM, using PKGMGR to inject the driver, and then reimporting the WIM to WDS.

Follow these steps to inject the VMware NIC driver into the Windows Vista Enterprise install image. In this example, I have already created a `C:\6000` folder to store the WIM image

and a `C:\6000\Mount` subfolder that will be used to mount the WIM image. I have also copied the VMware NIC drivers to `C:\6000\vmxnet`.

**Step 1: Export the WIM Image** Before you can mount the image, you need to export to a standard WIM image:

1. Using the WDS Console, expand `Install Images`, and select the `VistaGroup` image group.
2. Right-click the `Windows Vista ENTERPRISE` image and select `Export`. In the “File name” textbox, type `C:\6000\CustomVistaEnterprise.wim` and click `Save`.

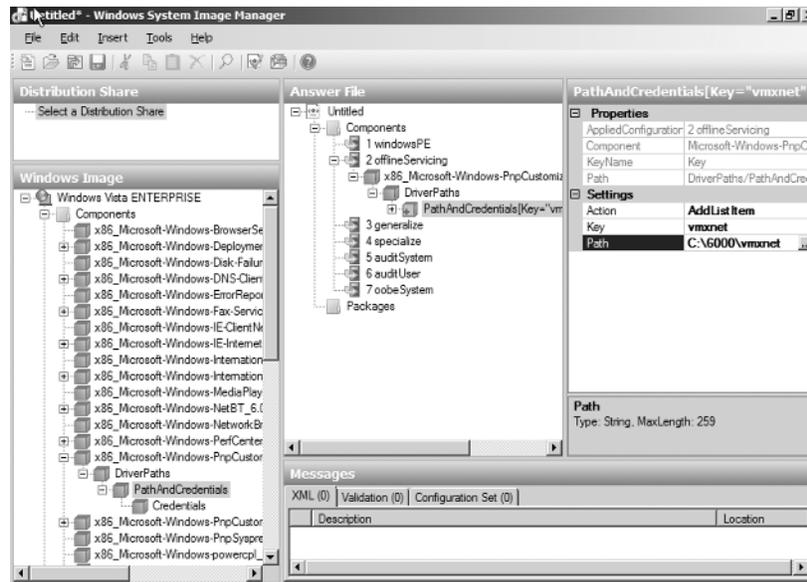
**Step 2: Create an Answer File for Package Manager** When injecting drivers into an install image, we need to create an answer file with the path to the driver(s). You will then use this answer file, together with Package Manager, to do the actual driver injection.

1. Using Windows System Image Manager, in the Windows Image pane, right-click the “Select a Windows Image or catalog file” node and click `Select Windows Image`.
2. In the “File name” text box, type `C:\6000\CustomVistaEnterprise.wim` and click `Open`. In the Windows System Image Manager dialog box, click `Yes` to create a catalog for the image.
3. From the `File` menu, select `New Answer File`.
4. In the Windows Image pane, expand `Windows Vista ENTERPRISE`, expand `Components`, expand `x86_Microsoft_Windows_PnpCustomizationsNonWinPE_6.0.6000.16386_neutral`, and expand `DriverPaths`.
5. Right-click `PathAndCredentials` and select “Add Settings to Pass 2 offlineServicing.”
6. In the `PathAndCredentials Properties` pane, in the `Key` field, type `vmxnet`, and in the `Path` field, type `C:\6000\vmxnet`, as shown in Figure 1.22.
7. From the `File` menu, select `Save Answer File`. In the “File name” textbox, type `C:\6000\Drivers.xml` and click `Save`.

**Step 3: Using Pkgmgr to Inject the Driver(s)** Now you are ready for the heavy lifting; that is, mounting the image. You’ll use Package Manager to inject the driver, and then unmount and save the changes.

1. Start a Windows PE tools command prompt. Type the following command to mount the image:  
`ImageX /mount /w C:\6000\CustomVistaEnterprise.wim 1 C:\6000\Mount`
2. Go the right location in the file system for driver injection (where the servicing stack is) by typing the following command:  
`cd /d "C:\Program Files\Windows AIK\Tools\Servicing"`
3. Inject the drivers using Package Manager by typing:  
`Start /w pkgmgr /o:C:\6000\Mount\Windows /n:C:\6000\Drivers.xml /l:C:\6000\DriverInjection`

FIGURE 1.22 Using WSIM to create answer files



4. Verify the injection by opening the `C:\6000\DriverInjection.txt` log file in Notepad. A return code of `0x0` is considered a good thing.
5. Commit the changes by typing  
`ImageX /Unmount /Commit C:\6000\Mount`

**Step 4: Add the Updated Image to WDS** Now, you need only to add the updated image back to WDS. You can choose between adding a new image or replacing the existing image. Remember, if you add the image as a new image to the same image group it will only use slightly more disk space due to the single instancing capabilities in WDS. This example, you will add a new image:

1. Using WDS Console, expand `Install Images`, right-click the `VistaGroup` image group and select `Add Install Image`. This will start the `Windows Deployment Services—Add Image Wizard`.
2. On the `Image File` page, in the “File name” text box, type `C:\6000\CustomVistaEnterprise.wim` and click `Next`.
3. On the `List of Available Images` page, clear the “Use default name and description for each of the selected image” check box, and click `Next`.
4. On the `Image Metadata` page, in the `Image Name` text box, type `Custom Windows Vista Enterprise`, and click `Next`.
5. On the `Summary` page, click `Next`.
6. On the `Task Progress` page, click `Finish`.



The WAIK for Windows Vista and Windows Server 2008 has the rest of the story about customizing and servicing images. Be sure to read the documentation, specifically the section on `DriverPaths` and unattend settings to see how to best make use of this technique.

## Creating an Answer File Using WSIM

As you know by now, we can configure the Vista setup by using an `unattend.xml` answer file. In fact, we can also configure the WDS client by using a second answer file. This means that when automating WDS deployments, we are using a two-stage approach with two different answer files.

First, the WDS client unattend file, `WDSClientUnattend.xml`, allows you to automate processes like image selection, logon credentials, and disk partitioning.

Second, the image unattend file, either `ImageUnattend.xml` or `sysprep.inf`, depending on the operating system, allows you to automate the remaining steps of the Windows setup.



The WDS console can assign unattend files only for Windows Vista and Windows Server 2008 images. For down-level images, you must use your existing `sysprep.inf` files and manually create an `$OEM$` directory structure. See the online documentation for more information: <http://tinyurl.com/2kj1ps>.

The next sections will show you how to create those answer files and to associate them with the WDS server and a Windows Vista image. The end result will be a fully automated deployment solution.

## Creating an Answer File for WDS Client

Here are the steps to create an answer file for the WDS Client. You will configure it to wipe the disk and create a 40GB partition, select UI language, log on automatically, and select an image automatically.

1. Using WSIM, in the Windows Image pane, right-click the “Select a Windows Image or catalog file” node and click **Select Windows Image**.
2. In the “File name” text box, type `C:\6000\CustomVistaEnterprise.wim` and click **Open**. In the Windows System Image Manager dialog box, click **Yes** to create a catalog for the image.
3. From the File menu, select **New Answer File**.
4. In the Windows Image pane, expand **Windows Vista ENTERPRISE**, expand **Components**, expand `x86_Microsoft-Windows-Setup_6.0.6000.16386_neutral`, and expand **DiskConfiguration**.
5. Right-click **Disk** and select “Add Setting to Pass 1 windowsPE.”
6. In the Answer File pane, select **Disk**, then in the Disk Properties pane, set the following values:  
DiskID: 0  
WillWipeDisk: true

**40** Chapter 1 • Deploying Windows with Style

7. In the Answer File pane, expand `Disk(DiskID="0")`, and then right-click `CreatePartitions` and select `Insert New CreatePartition`.
8. In the `CreatePartition Properties` pane, set the following values:
  - Order: 1
  - Size: 40000
  - Type: Primary
9. In the Answer File pane, right-click `ModifyPartitions` and select `Insert New ModifyPartitions`.
10. In the `ModifyPartition Properties` pane set the following values:
  - Active: true
  - Extend: false
  - Format: NTFS
  - Label: OSDisk
  - Letter: C
  - Order: 1
  - PartitionID: 1
11. In the Windows Image pane, expand `WindowsDeploymentServices`, and expand `ImageSelection`.
12. Right-click `InstallImage` and select “Add Setting to Pass 1 windowsPE.”
13. In the `InstallImage Properties` pane, set the following values:
  - Filename: `Install.wim`
  - ImageGroup: `VistaGroup`
  - ImageName: `Windows Vista ENTERPRISE`
14. In the Windows Image pane, right-click `InstallTo` and select “Add Setting to Pass 1 windowsPE.”
15. In the `InstallTo Properties` pane, set the following values:
  - DiskID: 0
  - PartitionID: 1
16. In the Windows Image pane, expand `Login`, right-click `Credentials`, and select “Add Setting to Pass 1 windowsPE.”
17. In the `Credentials Properties` pane, set the following values:
  - Domain: `corp.com`
  - Password: `<blank>`
  - Username: `BuildAccount`



For security reasons, you may not want to add all the credentials information, but to speed up the process for lab purposes, you can, of course, do it. If you add all the credentials, the process will be fully automated, and once a user presses F12, the old machine is gone.

18. In the Windows Image pane, right-click `x86_Microsoft-Windows-International-Core-WinPE`, and select “Add Setting to Pass 1 windowsPE.”
19. In the Microsoft-Windows-International-Core-WinPE Properties pane, set the `UILanguage` value to `en-US`.
20. In the Answer File pane, expand `x86_Microsoft-Windows-International-Core-WinPE` and select `SetupUILanguage`.
21. In the `SetupUILanguage` Properties pane, set the `UILanguage` value to `en-US`.
22. From the Tools menu, select `Validate Answer File`.
23. From the File menu, select `Save Answer File`.
24. In the “File name” text box, type `R:\WDSClientUnattend.xml`, and click `Save`.

### Answer File for the Vista Setup

Now the automation for the WDS Client is done. It is time to create the answer file for automating the Windows Vista Setup:

1. Using WSIM, in the Windows Image pane, right-click the “Select a Windows Image or catalog file” node and click `Select Windows Image`.
2. In the “File name” text box, type `C:\6000\CustomVistaEnterprise.wim` and click `Open`. In the Windows System Image Manager dialog box, click `Yes` to create a catalog for the image.
3. From the File menu, select `New Answer File`.
4. In the Windows Image pane, expand `Components`, expand `x86_Microsoft-Windows-Setup_6.0.6000.16386_neutral`, expand `ImageInstall`, and expand `OSImage`.
5. Right-click `InstallTo` and select “Add Setting to Pass 1 windowsPE.”
6. In the `InstallTo` Properties pane, set the following values:
  - DiskID: 0
  - PartitionID: 1
7. In the Windows Image pane, right-click `x86_Microsoft-Windows-International-Core-WinPE`, and select “Add Setting to Pass 1 windowsPE.”
8. In the Microsoft-Windows-International-Core-WinPE Properties pane, set the following values:
  - InputLocale: 0409:00000409
  - SystemLocale: en-US
  - UILanguage: en-US
  - UserLocale: en-US

## 42 Chapter 1 • Deploying Windows with Style

9. In the Answer File pane, expand `x86_Microsoft-Windows-International-Core-WinPE` and select `SetupUILanguage`.
10. In the `SetupUILanguage Properties` pane, set the following values:
  - UILanguage: en-US
  - WillShowUI: OnError
11. In the Windows Image pane, expand `x86_Microsoft-Windows-Setup_6.0.6000.16386_neutral`, right-click `UserData` and select “Add Setting to Pass 1 windowsPE.”
12. In the `UserData Properties` pane, set the following values:
  - AcceptEula: true
  - FullName: IT
  - Organization: Corp
13. In the Windows Image pane, right-click `x86_Microsoft-Windows-International-Core_6.0.6000.16386_neutral` and select “Add Setting to Pass 4 specialize.”
14. In the `Microsoft-Windows-International-Core Properties` pane, set the following values:
  - InputLocale: 0409:00000409
  - SystemLocale: en-US
  - UILanguage: en-US
  - UserLocale: en-US
15. In the Windows Image pane, right-click `x86_Microsoft-Windows-International-Core_6.0.6000.16386_neutral` and select “Add Setting to Pass 7 oobeSystem.”
16. In the `Microsoft-Windows-International-Core Properties` pane, set the following values:
  - InputLocale: 0409:00000409
  - SystemLocale: en-US
  - UILanguage: en-US
  - UserLocale: en-US
17. From the Tools menu, select `Validate Answer File`.
18. From the File menu, select `Save Answer File`.
19. In the “File name” text box, type `R:\ImageUnattend.xml` and click `Save`.

### Associating Answer Files with a WDS

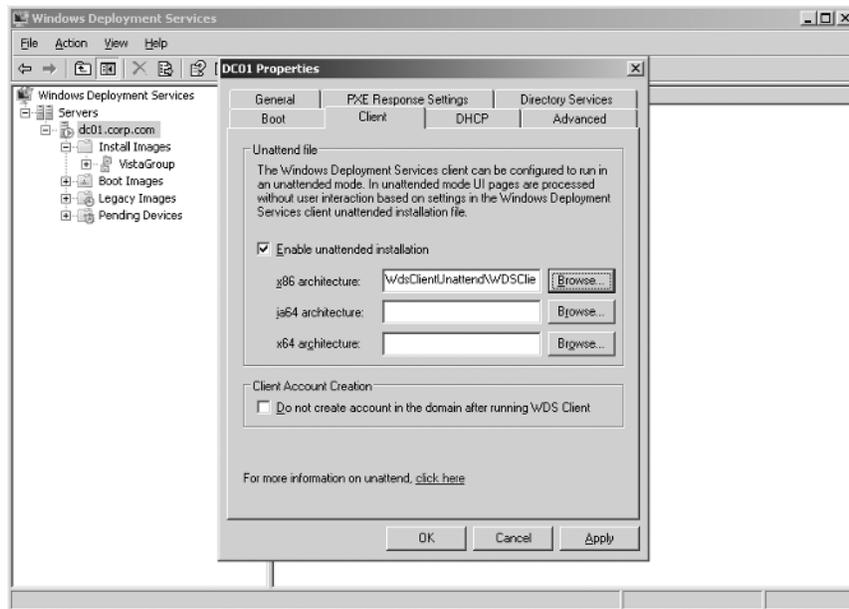
Associating answer files with WDS is easy. The hard part is creating the answer files first, as you just did.

Again, in WDS you have two types of answer files, one type for the WDS client that is associated on the Server, and one type for the images that is associated on the image. Let’s start with the WDS Client answer file:

1. Copy the `R:\WDSClientUnattend.xml` to `R:\RemoteInstall\WdsClientUnattend`.
2. Using the WDS Console, right-click the WDS Server and select `Properties`.

3. On the Client tab, browse and select `WDSClientUnattend.xml` and click OK.  
The process can be seen in Figure 1.23.

**FIGURE 1.23** Associate an answer file for the WDS client directly with the server.



Next, associate the image answer file (I named mine `ImageUnattend.xml`).

1. Using the WDS Console, expand `Install Images`, and select the `VistaGroup` image group.
2. Right-click your Windows Vista Enterprise image and select `Properties`.
3. On the Client tab, browse to `R:\ImageUnattend.xml` and click OK.

The process can be seen in Figure 1.24.

## Installing Your First Fully Automated Client

Once you're sitting down at your target machine, you're ready to install your first fully automated client.

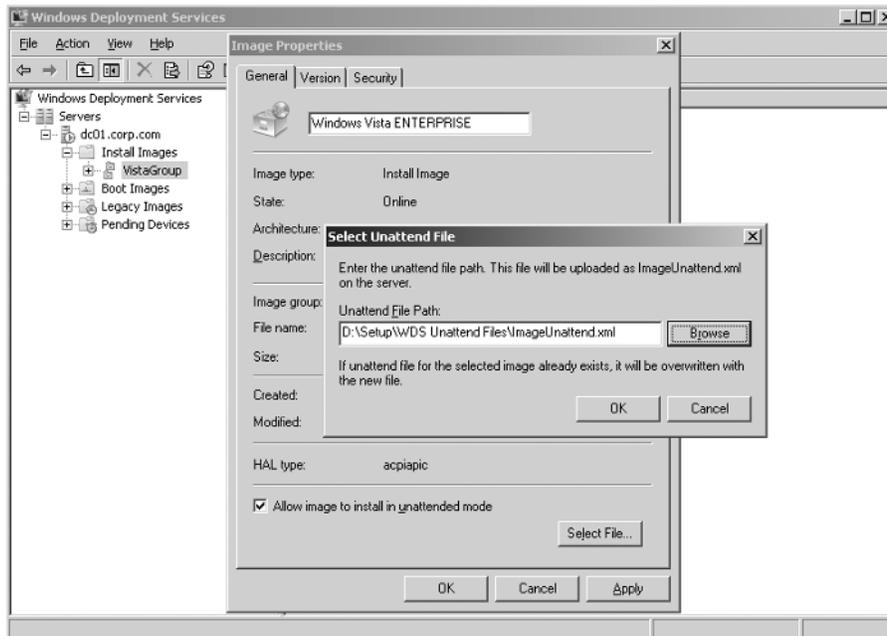


Running WDS on your workstations completely formats the first hard drive on your system.

To use WDS to install a client, follow these steps:

1. Turn on the target system.
2. When prompted, immediately press F12.

**FIGURE 1.24** Associate an answer file for the Windows Vista Setup directly with the image.



If you have done everything right, the setup process should prompt you for a password and then finish the process, fully unattended. Again, the password prompt can be automated too, but it may be a security risk since any user with access to the computer would then be able start the deployment.

### Use at Your Own Risk: Speeding Up the Download Time of a *boot.wim* Image

For me, the only big drawback in WDS is the slow boot time of the *boot.wim* file. That's because it's about 126MB, which is, well, pretty big, even on most networks. Actually, the problem really isn't that the file itself is that big, it's the underlying protocol used to transfer the *boot.wim* from the server to your Windows Vista machine-to-be: TFTP (Trivial FTP) protocol, which really stinks. The major limitation is the TFTP block size used when downloading an image, which is around 1000 bytes; small enough to still fit in an Ethernet frame without causing packet fragmentation.

I need to stress this here: What I'm about to show you works only if your network cards and the switches and routers support it. If they do, you can bump up the frame size WDS uses when sending the *boot.wim* file via TFTP.

First, make sure you're network doesn't have too much network latency; TFTP transfers just hate that.

On gigabit connections you can increase the speed up to four times by increasing the frame size (measured on 50 and 75 clients), for example:

- Fifty boot images on gigabit connections typically take about 5 minutes to start with the default TFTP block size, about 3 minutes with a 4KB block size, and about 1.5 minutes with a 16KB block size.
- Seventy-five boot images on gigabit connection typically take about 8 minutes with the default TFTP block size, about 6 minutes with 4KB block size, and about 2 minutes with 16KB block size.

Now, let's say you have two kinds of network cards, three kinds of network switches, and four kinds of routers. All of the equipment (from the server's network card through the switch through the router to the target machine's network card) has to be able to support the increased block size trick I'm about to show you, which is how tell the WDS server to increase the frame size from 1000 to another number of bytes per block.



## Real World Scenario

### WDS Best Practices

This sidebar could also be named "Common Issues" because one way of looking at best practices really could be about avoiding common issues.

Since WDS relies heavily on network configuration, most of the common issues relate to just that, but it doesn't stop there. Here are some tips to get the best environment for WDS possible.

- IPv6 is not supported by WDS, so if you are not using it for other services, unbind it from the network card.
- Always use the latest boot image version, which means use the one from Windows Server 2008 (required for multicast support).
- Don't use more than 13 boot images; the boot menu cannot list more than 13 boot images.
- When using Microsoft DHCP and WDS on the same server, configure WDS not to listen on port 67 and allow it to configure the 60 PXE option.
- When using a non-Microsoft DHCP and WDS on the same server, configure WDS not to listen on port 67 and use the non-Microsoft DHCP Server Configuration tool to configure the 60 PXE option.
- When using a DHCP server located in a different subnet, you will need to do one of the following:
  - Configure your IP helper tables (recommended).
  - Add DHCP options 66 and 67.
- If you have multiple domains, make sure they are in the right search order in your network card DNS settings.

## 46 Chapter 1 • Deploying Windows with Style

The program you need to run, `bcdedit .exe`, is available only on a Windows Vista or Windows Server 2008 machine. So, to do this trick, you'll need to start on a Windows Vista machine and then map a drive to the `REMINST\Boot\x86` on the server. You will then use the following command to edit the default Boot Configuration Data store, named `default.bcd`.

```
Bcdedit -store default.bcd -set  
{68d9e51c-a129-4ee1-9725-2ab00a957daf} ramdisktftpblocksize  
<yourblock size here>
```

Just go up in even multiples: 4096, 8192, 16384. I wouldn't set it any higher than 16384. I've tried a lot of values. The two values that worked best were 4096 and 8192.

Finally, run this command when you're back on the server:

```
Sc control wdsserver 129
```

The next time you boot a client, it will use the new TFTP block size to download the Windows PE image.

Again, the maximum value isn't a function of your network bandwidth; it's a function of the maximum buffer size your PXE ROM allocates for UDP packets.

## Troubleshooting WDS

When you need to troubleshoot the WDS, you will find that most of the time the log files and a good network sniffer are your very best friends.

### PXE Client

One of the most common issues is that PXE client cannot locate the WDS server or can't find a network boot file to load, as shown in Figure 1.25.

**FIGURE 1.25** PXE-boot failure due to wrongly configured DHCP or IP helpers

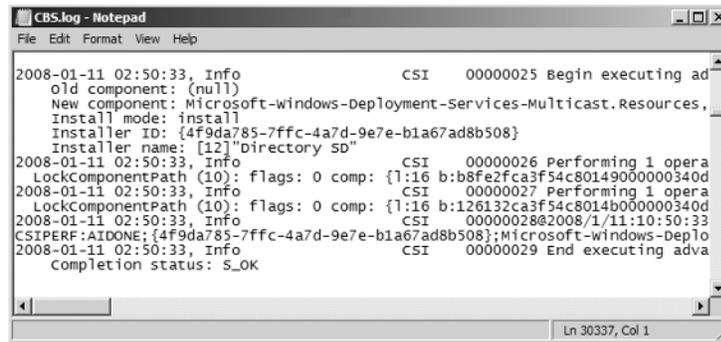
```
Network boot from AMD Am79C970A  
Copyright (C) 2003-2005 VMware, Inc.  
Copyright (C) 1997-2000 Intel Corporation  
  
CLIENT MAC ADDR: 00 0C 29 F8 AA 39 GUID: 564D199F-9CF0-F97D-DE3F-CEAECDF0AA39  
PXE-E53: No boot filename received  
  
PXE-M0F: Exiting Intel PXE ROM.  
Operating System not found  
-
```

To troubleshoot this, first go back and verify your DHCP configuration according to the previous section. If that doesn't help, install a network sniffer and follow the traffic between the PXE client and DHCP/WDS server.

## Setup

If you get any errors during setup, you can find additional information for the WDS setup for Windows Server 2008 in C:\Windows\logs\cbs\cbs.log. This CBS (component-based services) log, shown in Figure 1.26, is enabled by default and is used not only for WDS, but for all components (packages) in Windows Server 2008.

**FIGURE 1.26** CBS log file entries when the WDS multicast components are installed



```

CBS.log - Notepad
File Edit Format View Help
2008-01-11 02:50:33, Info                CSI    00000025 Begin executing ad
old component: (null)
New component: Microsoft-windows-Deployment-Services-Multicast.Resources,
Install mode: install
Installer ID: {4f9da785-7ffc-4a7d-9e7e-b1a67ad8b508}
Installer name: [12]"Directory sd"
2008-01-11 02:50:33, Info                CSI    00000026 Performing 1 opera
LockComponentPath (10): flags: 0 comp: {1:16 b:b8fe2fca3f54c8014900000340d
2008-01-11 02:50:33, Info                CSI    00000027 Performing 1 opera
LockComponentPath (10): flags: 0 comp: {1:16 b:126132ca3f54c8014b00000340d
2008-01-11 02:50:33, Info                CSI    00000028@2008/1/11:10:50:33
CSI PERF:AIDONE;{4f9da785-7ffc-4a7d-9e7e-b1a67ad8b508};Microsoft-windows-Deplo
2008-01-11 02:50:33, Info                CSI    00000029 End executing adva
Completion status: S_OK
Ln 30337, Col 1

```

The component-based services will also raise events to the event log, but it is in the log file you will get all details. For example, an error like “cannot connect” in the UI might reveal itself as error code 5 in the log file, which is really an access denied error. Or, you may find that general “Setup failed” messages really mean something like a corrupt setup file is found, or that you have insufficient disk space to complete the operation.

## Tracing and Logging

You can enable logging and tracing for all WDS components. This can be very useful when troubleshooting PXE client timeout issues due to high latency networks, or if the WDS server has problems contacting Active Directory.

**Enabling Logging for the WDS Client** The WDS client supports four logging levels:

- NONE: No logging (default)
- ERRORS: Errors only
- WARNINGS: Warnings and errors
- INFO: Errors, warnings, and informational events

To configure WDS client logging, set the following Registry keys:

```
HKLM\System\CurrentControlSet\Services\WDS\Server\Providers\WdsImgSrv\
ClientLogging\Enabled
```

- Type: REG\_DWORD
- Value: not set or 0 means not enabled (default), and 1 means enabled

**48** Chapter 1 • Deploying Windows with Style

HKLM\System\CurrentControlSet\Services\WDS\Server\Providers\WdsImgSrv\  
ClientLogging\LogLevel

- Type: REG\_DWORD
- Value: not set or 0 means OFF, 1 means ERRORS, 2 means WARNINGS, 3 means INFO

**Enable Tracing for WDS Server Component** Set the following registry key to 1:

HKLM\Software\Microsoft\Tracing\WDS\Server\EnableFileTracing

Log file: C:\Windows\tracing\wds\server.log

**Enable Tracing for Management Components** Set the following Registry key to 1:

HKLM\Software\Microsoft\Tracing\WDS\MGMT\EnableFileTracing

Log files: C:\Windows\tracing\wds\mgmt.log and  
C:\Windows\tracing\wds\mmc.log.

**Enable Tracing for WinPE Client (WDS Capture) Components** After booting the image, when the capture wizard starts, press Shift+F10 to get a command prompt and set the following Registry key to 1:

HKLM\Software\Microsoft\Tracing\WDS\Capture\EnableFileTracing

Log file: X:\Windows\Tracing\WDS\Capture.log.



### Real World Scenario

#### Mere-Mortals Can Add Only Ten Workstations

In Windows NT, only administrators can add computer accounts to the domain. Under Active Directory, the Authenticated Users group can add computer accounts to the domain via the "Add Workstation to Domain" user right. But there's a catch. Each authenticated user can add only ten new computer accounts. On the eleventh try, the user is presented with the error message: "The machine account for this computer either does not exist or is unavailable." This is a little-known problem that has three little-known solutions:

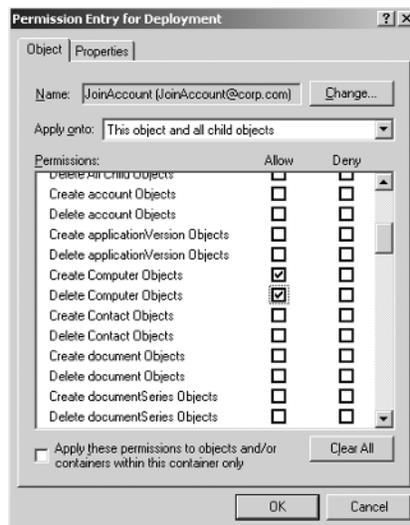
#### Pre-creating Computer Accounts

Administrators can pre-create the computer accounts, as many accounts as they like. They are exempt from the "10 strikes and you're out" rule.

### Granting Create Computer Objects and Delete Computer Objects Rights

You can grant the Create Computer Objects (and if desired) the Delete Computer Objects rights to the Computers folder in Active Directory. These rights are different from the “Add Workstation to Domain user” right that all Authenticated Users are given. To make this change, follow these steps:

1. Choose Start > Programs > AdministrativeTools > Active Directory Users and Computers.
2. Choose View > Advanced to enable the Advanced view.
3. Right-click the Computers folder, and choose Properties from the shortcut menu to open the Properties dialog box.
4. Click the Security tab, and then click the Advanced button to open the “Advanced Settings for Computers” properties.
5. On the Permissions tab, click Authenticated Users, and then click the Edit button to open the Permissions Entry for Authenticated Users box.
6. Before proceeding, make sure the “This Object and All Child Objects” option is displayed in the Apply Onto box.
7. In the Permissions list, click the Allow check box for Create Computer Objects and, optionally, Delete Computer Objects, as seen here:



Use ADSI edit to manipulate the ms-DS-MachineAccountQuota to increase (or decrease) the value to the desired number of times a user can create a computer account. However, for security reasons, I prefer delegating the correct permissions rather than allowing any user to join computers to the domain.

## Microsoft Deployment Toolkit 2008 (MDT), Formerly Known as BDD

Microsoft has a gift they want to give you: free Windows deployment tools and guidance. The best part: the guidance isn't just for Windows clients, but Windows servers, as well. The gift's name is Microsoft Deployment Toolkit 2008, or MDT for short, and it's the successor to BDD 2007, the Business Desktop Deployment Solution Accelerator.

If you are currently using BDD 2007, I strongly recommend that you look into Microsoft Deployment Toolkit 2008. You will find many goodies in this update, and the transition isn't terribly complicated. Microsoft Deployment Toolkit 2008 offers upgrade paths from existing BDD 2007 setups so you can take easily advantage of the new features.

The Microsoft Deployment Toolkit 2008 solution is called, in Microsoft-speak, a *solution accelerator*, but it's really tools and guidance wrapped up in a nice little gift pack. Microsoft Deployment Toolkit 2008 spans most aspects of a deployment project: planning, developing, piloting, executing, and more. In this section we will focus on the image engineering components.

By the time you're done with this section, you'll be able to do the following:

- Deploy Windows operating systems in a very rich and dynamic way
- Add drivers and applications to the deployment process
- Automate the Windows Desktop rollout experience

I know that you'll want to do even more, and that's great. To discover more about what Microsoft Deployment Toolkit 2008 can do, be sure to go to [www.microsoft.com/deployment](http://www.microsoft.com/deployment) so you can find out about things like infrastructure remediation, application management, security, operation readiness, office deployments, deployment process, and testing processes.

### Understanding Microsoft Deployment Toolkit 2008

Microsoft Deployment Toolkit 2008 has two core ways to be utilized. One is called *Lite Touch* and does not require much except a decent Windows server. In fact, it can even be run on a Windows workstation, but please don't go that route; you really want the power and performance of a real server.

The other way to use Microsoft Deployment Toolkit 2008 is *Zero Touch*, but it requires a lot more management infrastructure, like SMS 2003 or SCCM 2007. In this chapter we'll focus on the Lite Touch components so you can start working with it immediately.

You won't find it in Microsoft documentation, but we'll call Microsoft Deployment Toolkit 2008 Light Touch *MDLT* for short.



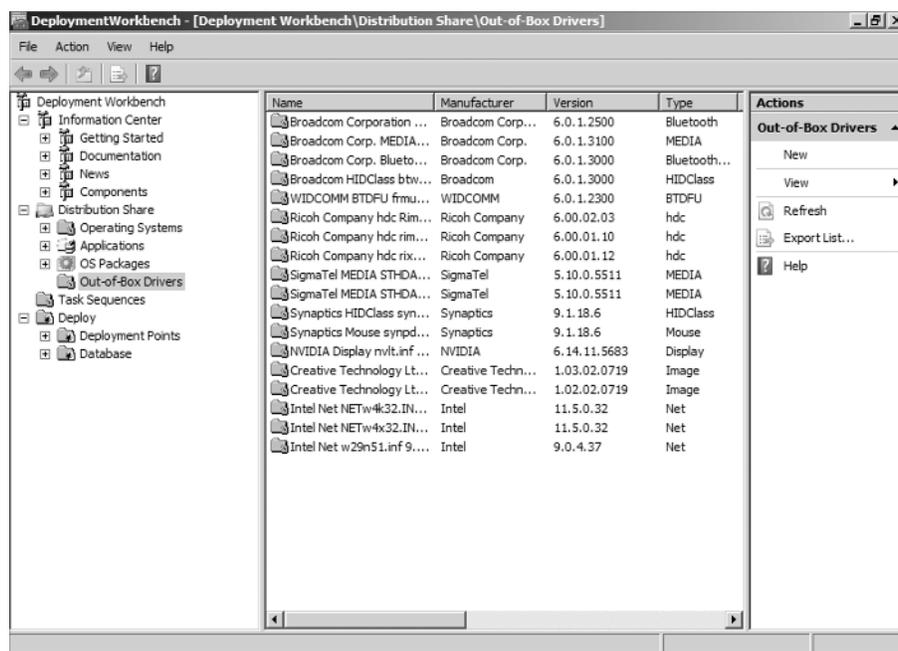
MDLT shouldn't to be confused with the McDonald's McDLT, a not so wonderful hamburger from the 1980s. If you want to walk down nostalgia lane, go to <http://tinyurl.com/29yvh8> for a McDLT commercial.

The MDLT components have the following core features:

**Wide Operating System Support** MDTL supports deploying Windows XP, Windows Vista, Windows Server 2003, and Windows Server 2008.

**Deployment Workbench** This is an MMC 3.0 front-end for configuring most aspects of the deployment environment, as shown in Figure 1.27 (I'll show you how to get here in the next section).

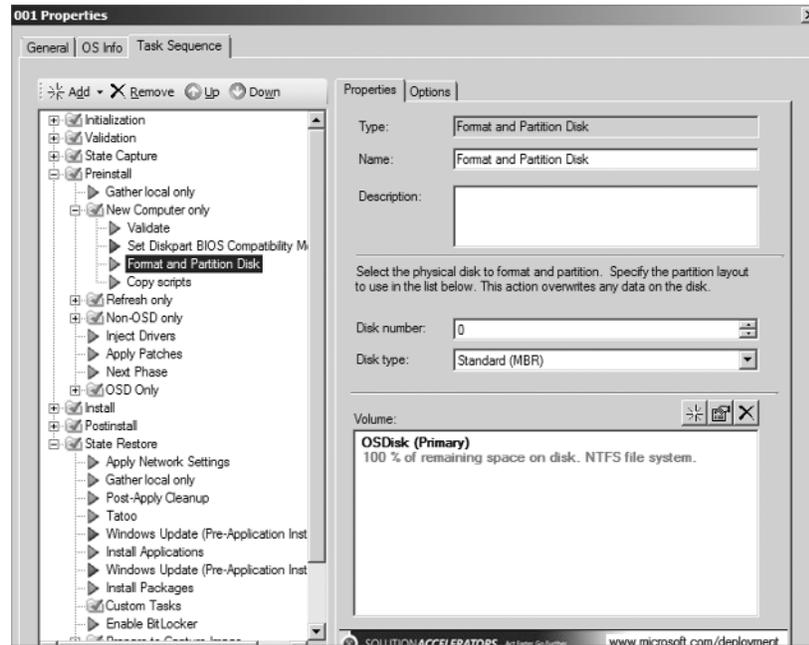
**FIGURE 1.27** The Deployment Workbench, viewing the driver repository



**Task Sequencer** This uses technology from SCCM but does not require SCCM (or its high price). This is the heart and soul of Lite Touch. The task sequencer controls all steps of the deployment, as shown in Figure 1.28.

**Driver Management** This is the central repository for drivers and tools for automatic driver injection during deployment. When deploying a client using Microsoft Deployment Toolkit 2008 the client will do a plug-and-play ID scan of the hardware and automatically download and install the needed driver from the driver store.

**Patching** Microsoft Deployment Toolkit 2008 can add patches, language packages, and so on, both offline and online. There are also built-in functions for forcing Windows Update as part of the deployment process.

**FIGURE 1.28** A sample task sequencer

**Advanced Disk and Network Configurations** You can configure multiple disks, partitions, and multiple NICs (although these are mostly used for server deployments).

**Server-side Rules** Microsoft Deployment Toolkit 2008 uses central server-side rules to configure Windows deployment settings like regional settings, screen resolution, and so on.

**User State Migration Support** Microsoft Deployment Toolkit 2008 can back up and restore user state and data as part of the deployment process.

**Backup Capability** Microsoft Deployment Toolkit 2008 can take an optional full backup (image) of the old computer when deploying a new image.

**Deployment Database** Microsoft Deployment Toolkit 2008 also supports storing the configuration in a database. For example, you can create a role that will install five applications and then associate that role to a computer account (based on MAC address, serial number, asset tag, or UUID). If you start deploying that machine, it will query the database and install the five applications, as shown in Figure 1.29.

**Server Roles** You can select which server roles should be installed on an Windows Server 2008 operating system, as shown in Figure 1.30.

Configuration of some of the server roles is supported, as well, as shown in Figure 1.31.

FIGURE 1.29 The Applications pane of a role that can be assigned to a computer

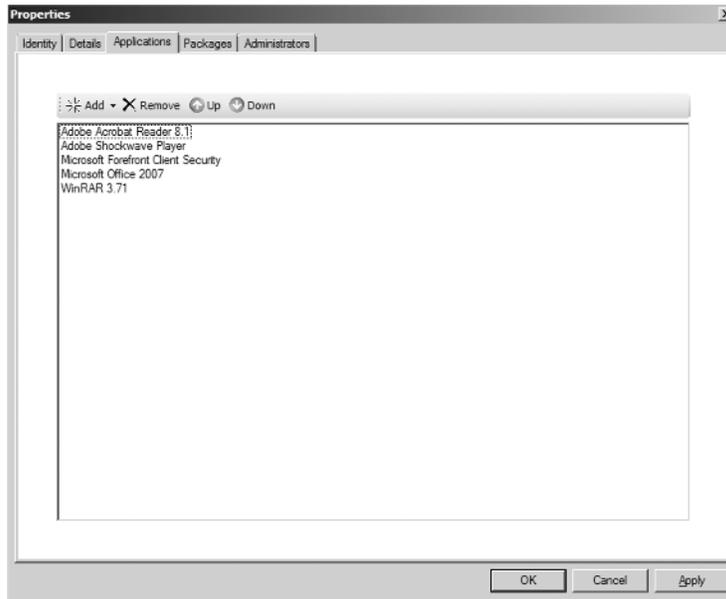
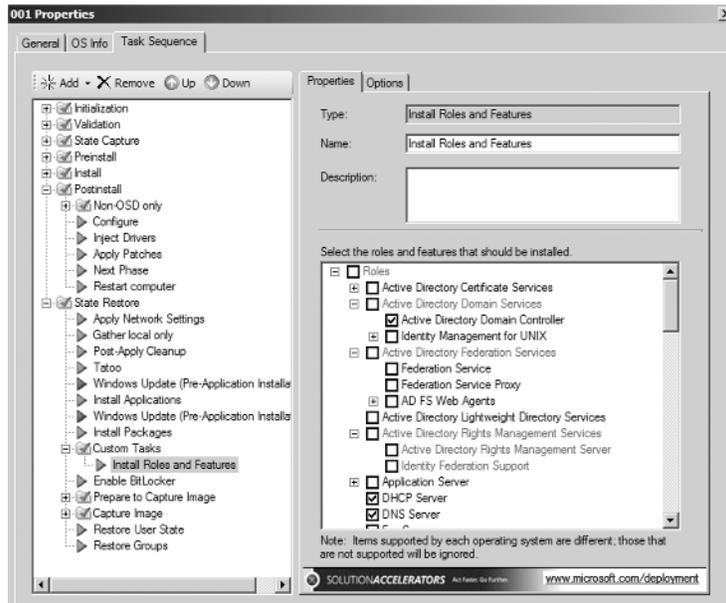
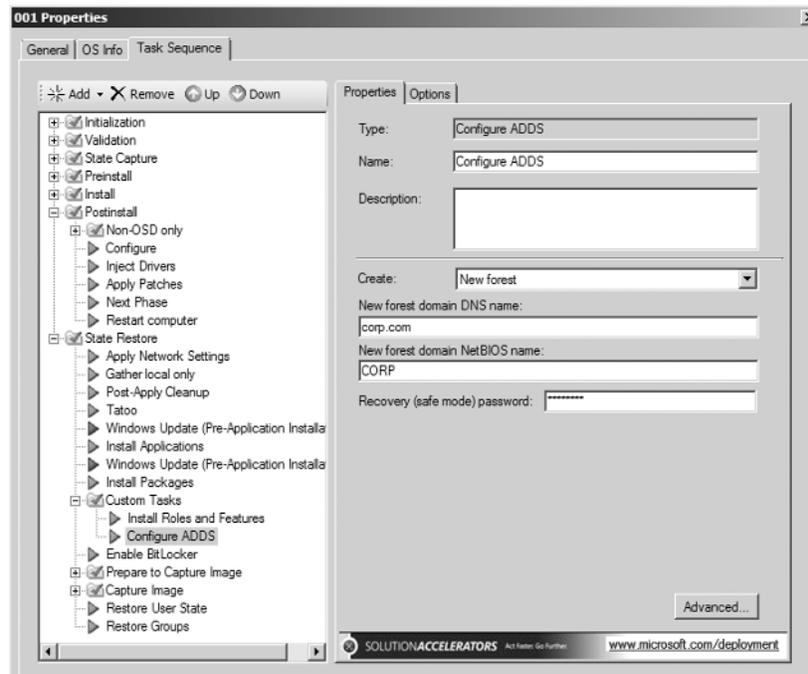


FIGURE 1.30 Install roles and features.



**FIGURE 1.31** Configure Active Directory Domain Services.

## WDS vs. Microsoft Deployment Toolkit 2008 (Better Together?)

Before continuing, we need to clarify what roles WDS and Microsoft Deployment Toolkit 2008 play in your Windows deployment strategy. Will you use them together? Separately? What are the advantages of one versus the other?

Organizations tend to either just use WDS, or use WDS and Microsoft Deployment Toolkit 2008 together. The major difference between the two is that Microsoft Deployment Toolkit 2008 leverages a task sequencer. This task sequencer controls the deployment flow using static and dynamic server side rules and has many built in routines for things like driver injection, user state migration, monitoring, and so on.



If you want to configure more things than you can with a standard answer file, you should add the free MDLT components to your WDS solution. For example, with the built-in database support, you can assign role-based settings and applications to your deployments.

Let's review how WDS works, and I'll explain how the Microsoft Deployment Toolkit 2008 adds value.

**Scenario 1: Image Engineering Using WDS** From an image-engineering point of view, WDS works like this:

1. A boot image (Windows PE) is downloaded over the network.
2. A client (the WDS client) connects to a server share containing the operating system(s).
3. Deployment settings (like regional settings and so on) are either prompted for or configured using answer files.
4. A Windows image is selected and applied to disk.

That's about it—a basic, straightforward deployment solution.

**Scenario 2: Image Engineering Using WDS and Microsoft Deployment Toolkit 2008** From an image-engineering point of view, WDS and Microsoft Deployment Toolkit 2008 work like this:

1. A custom boot image (Windows PE, created by the Deployment Workbench) is downloaded over the network.
2. A client (an HTA-driven wizard) connects to a server share containing operating systems, updates, drivers, and applications and reads the deployment configuration.
3. Deployment settings (like regional settings, applications to install, and so on) are either prompted for or configured from the server-side rules.
4. The Task Sequencer starts and, as one of many steps, a Windows image is applied to disk.

As you can see, the scenarios are quite alike from a high-level overview. The major overt difference when leveraging Microsoft Deployment Toolkit 2008 is that you can take advantage of the Task Sequencer controlling the deployment.

However, underneath the surface there's quite a difference. Microsoft Deployment Toolkit 2008 starts where WDS functionality ends and will ease a lot of deployment pains for you. It will give you all the Microsoft Deployment Toolkit 2008 features like automatic driver injection and assigning applications to the deployment process. It will give you a whole rule-based driven central framework for doing all sorts of configurations to the operating system during deployment. And, since WDS can network boot the Microsoft Deployment Toolkit 2008 boot image, you are just an F12 keypress away from starting the deployment.

## Inside Microsoft Deployment Toolkit 2008

Let's define a few terms first. That way, we'll share a common vocabulary as we work through the issues with imaging.

**Deployment Workbench** This is the premium management tool, which also acts as a front-end for many of the WAIK tools and infrastructure support. Remember, the WAIK adds support to APIs like WIMGAPI, which allows you to manage WIM images; tools like PEIMG for driver injection; and OSCDIMG for creating ISO (CD/DVD) images. WAIK also contains Windows PE 2.0, our deployment platform. It is used as boot image for both WDS and Microsoft Deployment Toolkit 2008.

**Task Sequencer** This controls all aspects of the setup, allowing you to select unique configurations and operating systems during deployment. This is the framework that drives the deployment process.

**Distribution Share** This is a folder containing all setup files, updates, drivers, applications, deployment scripts, and so on. It is maintained by the Deployment Workbench.

**Deployment Point** Here is where the deployment settings, the server side rules, are configured. This is also where you create the boot images used to start the deployment process. The boot image will connect to the deployment point and find out what deployment settings to use.

**Boot Images** As WDS does, Microsoft Deployment Toolkit 2008 uses boot images to start the deployment process. Guess what service I recommend you use to boot these images over the network? Yep, the Windows Deployment Service!

## Setting Up Microsoft Deployment Toolkit 2008

Implementing Microsoft Deployment Toolkit 2008 may seem hard, complicated, and unwieldy at first. But once you get into it, you'll see that it's not too bad. In this section, I'll take you from start to finish implementing MDLT components.

If you follow the steps, you will build a great deployment solution in just a few hours.

### Microsoft Deployment Toolkit 2008 Prerequisites

Before installing Microsoft Deployment Toolkit 2008 there are some system requirements that need to be met:

- Microsoft Core Extensible Markup Language (MSXML) Services 6.0, found at <http://tinyurl.com/2h396u>.
- Microsoft .NET Framework 2.0. The Windows AIK 1.0 media includes the .NET Framework 2.0 installation file. Download WAIK (and .NET Framework 2.0) from <http://tinyurl.com/37b42g>.
- Microsoft .NET Framework 2.0 requires Microsoft Windows Installer 3.1, but you likely already have this. (If you are running SP2 on your Windows Server 2003 machine then you have it.) If not, check out <http://tinyurl.com/6camd>.
- Microsoft Management Console (MMC) 3.0 (part of Windows Server 2003 SP2) found at <http://tinyurl.com/2ytam3>.
- For database support (optional, but recommended) you need Microsoft SQL Server Express Edition SP2. Download it at <http://tinyurl.com/2ax2am>.
- Managing the database is a lot easier with Microsoft SQL Server Management Studio Express, which you can find at <http://tinyurl.com/mkvgw>.



The first four components are dead easy to install and the default configuration works fine. The database components are another story, and that's why you'll find a step-by-step guide for configuring those components in the "Beyond the Microsoft Deployment Toolkit 2008 Basics" section later in this chapter.

## Microsoft Deployment Toolkit 2008 Core Configuration

Setting up Microsoft Deployment Toolkit 2008 is a quite straightforward process. It involves the following steps, which are each covered in the next sections:

1. Installing the prerequisites software (listed in the previous section) and of course Microsoft Deployment Toolkit 2008 itself.
2. Creating a distribution share and importing setup files like operating system images, applications, updates, and drivers.
3. Creating one or more task sequences.
4. Creating a deployment point.
5. Adding database support (optional; this is covered in the section “Beyond the Microsoft Deployment Toolkit 2008 Basics”).

Let’s explore each of these tasks right now.

### Downloading and Installing Microsoft Deployment Toolkit 2008 Plus Some Updates

Installing Microsoft Deployment Toolkit 2008 is quite easy; The Microsoft Deployment Toolkit 2008 Setup files can be found at: <http://tinyurl.com/2x1p2y>.

Once you download these files, you are ready to install Microsoft Deployment Toolkit 2008. Run `MicrosoftDeploymentToolkit_x86.msi` and run through the Setup Wizard with the default settings.

After installing Microsoft Deployment Toolkit 2008 you will find the documentation in `C:\Program Files\Microsoft Deployment Toolkit\Documentation`. There is also a compiled help file of all the documents in the `C:\Program Files\Microsoft Deployment Toolkit\bin` folder. When working with the image engineering components, `Getting_Started_Guide.doc`, `Release_Notes.chm`, and `Image_Engineering_Feature_Team_Guide.doc` are excellent reading materials to start with.

### Creating the Server Structure

While you can use the administrator account for all credentials, this isn’t recommended. Microsoft Deployment Toolkit 2008 is really designed for using role-based security. So let’s create a few service accounts and an organizational unit and then assign the permissions as needed.

**Creating an OU to Store Workstations** Using Active Directory Users and Computers, create an OU (for storing your computer accounts). I recommend naming it Workstations, and for our example, it can be right off the domain.

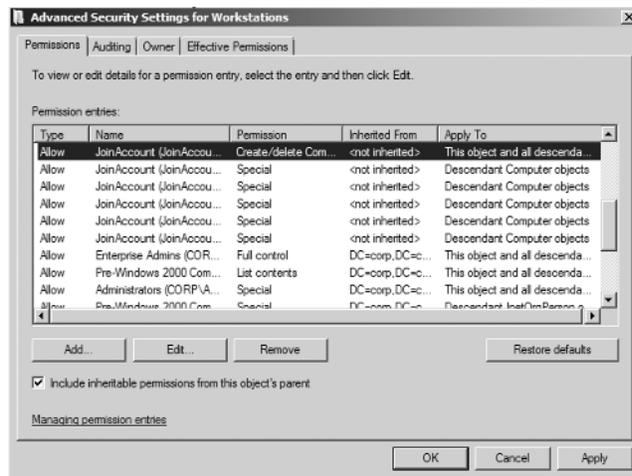
**Creating Two User Accounts for Building and Joining Computers** Using Active Directory Users and Computers, create the following user accounts:

- BuildAccount, to be used to access the deployment server and start the deployment process when booting from our boot image.
- JoinAccount, to be used by Windows Setup to join the machine to the domain.

## 58 Chapter 1 • Deploying Windows with Style

Using Active Directory Users and Computers, allow JoinAccount permissions to manage computer accounts in the Workstations OU by using the following lists of permissions. (The first set of permissions is for adding and removing computer accounts, and the second set is to update existing computer accounts). The end result should look something like Figure 1.32.

**FIGURE 1.32** Active Directory permissions for JoinAccount



1. Start Active Directory Users and Computers. From the View menu, select Advanced Features. This will enable the Security tab on Active Directory objects.
2. Right-click the Workstations OU and select Properties.
3. On the Workstation Properties screen, in the Security tab, click the Advanced button.



If you added the JoinAccount user directly on the first screen, the user would be given high permissions. For security reasons, assign only the minimum permissions needed.

4. On the Advanced Security Settings screen for Workstations, click Add.
5. In the “Select User, Computer, or Group” field, type **JoinAccount** and click OK.
6. On the Permission Entry for Workstations in the Permissions list, select Allow for the following permissions and then click OK:
  - Create Computer Objects
  - Delete Computer Objects
7. On the Advanced Security Settings for Workstations windows, click Add.
8. In the “Select User, Computer, or Group” screen, type **JoinAccount** and click OK.

9. On the Permission Entry for Workstations, in the “Apply to” drop-down list, select “Descendant Computer objects” (or “Computer objects” if doing this on a Windows Server 2003 server).
10. In the Permissions list, select Allow for the following permissions and then click OK.
  - Read All Properties
  - Write All Properties
  - Read Permissions
  - Modify Permissions
  - Change Password
  - Reset Password
  - Validated write to DNS host name
  - Validated write to service principal name
11. On the Advanced Security Settings for Workstations window, click OK.
12. On the Workstation Properties screen, click OK.

Next, create a folder, like D:\LOGS. Share it as **Logs\$** and configure the necessary permissions following these steps.

1. Using Windows Explorer, right-click the D:\Logs folder and select Properties.
2. In the Logs Properties window, on the Sharing tab, click Advanced Sharing. Select the “Share folder” check box, and in the “Share name” text box, type **Logs\$**.
3. Still in the Advanced Sharing window, click Permissions. In the “Permissions for everyone” list, select Allow for Change permissions and click OK twice.
4. In the Logs Properties window, on the Security tab, click Edit.
5. In the “Permissions for Logs” window, click Add, type **BuildAccount**, and click OK.
6. In the “Permissions for Logs” window, select BuildAccount, and in the Permissions for BuildAccount, select Allow for Modify Permissions. Click OK and then click Close.

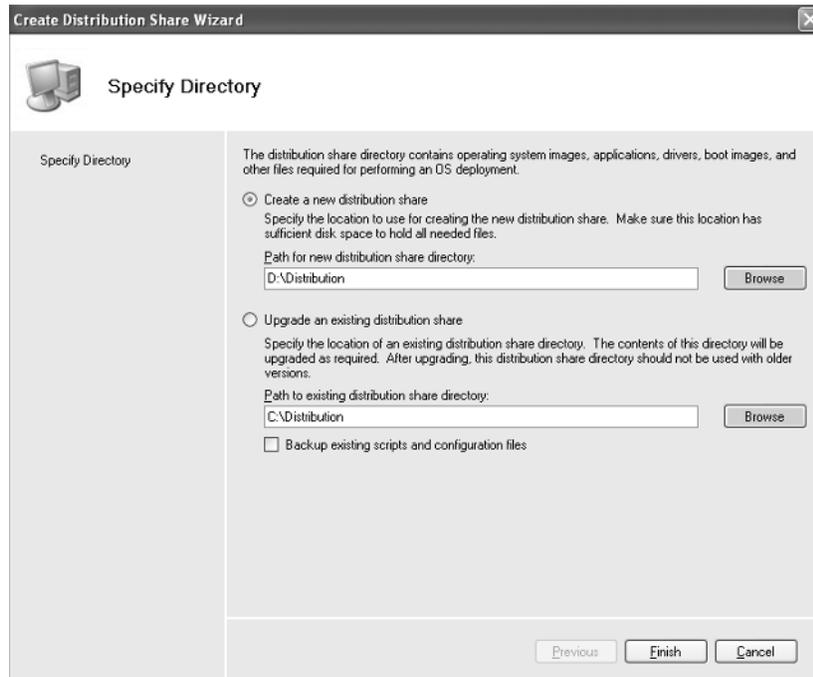


In Windows Server 2008 the procedure has changed a bit for creating shares, but I’m sure you get the idea.

### Creating the Distribution Share

The Distribution share will hold all your setup files, drivers, applications, scripts and so on, so let’s go ahead and create it.

1. Using the Deployment Workbench, right-click the Distribution Share node and select “Create distribution share directory.” The Create Distribution Share Wizard will start, as shown in Figure 1.33.
2. On the Specify Directory screen, in the “Path for new distribution share directory” field, type **D:\Distribution** and click Finish.

**FIGURE 1.33** Creating Distribution Share Wizard

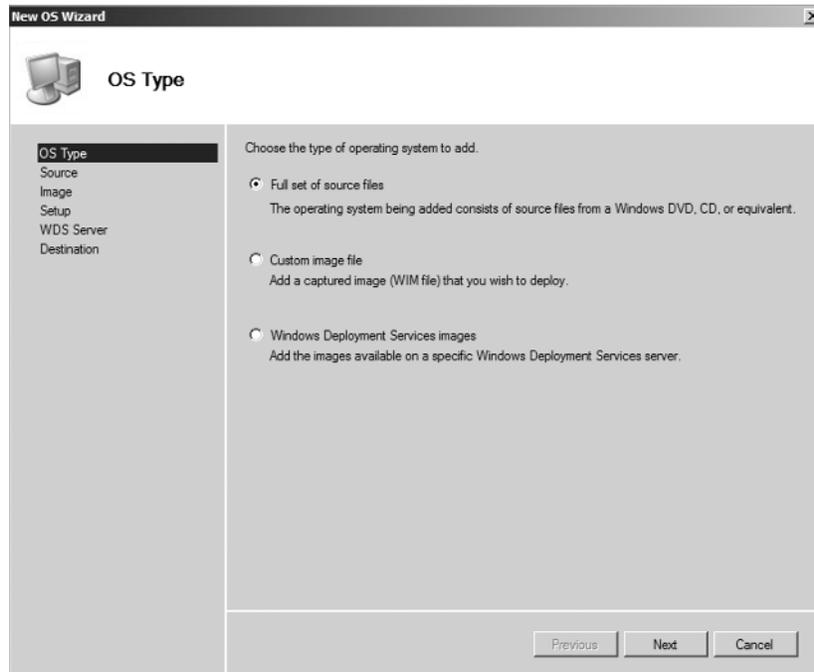
### Adding Windows OS and Other Setup Files

To add Windows Vista Enterprise installation files to the Deployment Workbench follow these steps:

1. Using the Deployment Workbench, expand the Distribution Share node, right-click the Operating Systems node, and select New. The New OS Wizard starts, as shown in Figure 1.34.
2. On the OS Type page, select “Full set of source files” and click Next.
3. On the Source page, in the “Source directory” text box, type the path to Windows Vista Enterprise media and click Next.
4. On the Destination page, in the “Destination directory name” text box, type **Windows Vista Enterprise Eng x86** and click Finish.

### Adding an Application to the Deployment Workbench

In this example you will be adding the Word Viewer 2003 application to the Deployment Workbench. You can download Word Viewer 2003 from this URL: <http://tinyurl.com/3q1b4>.

**FIGURE 1.34** The New OS Wizard in Deployment Workbench

When downloading files that you add to Deployment Workbench, make sure to unblock them first. (Right-click the file, select Properties, and click Unblock). If you don't, the deployment process will halt and ask you for permissions to execute the file, and this is not what you want during an automated deployment. The reason for this halt is that since Windows XP SP2 there is an Attachment Manager in Windows that will block certain file extensions from executing. For more information, see a description of how the Attachment Manager works in Windows XP Service Pack at <http://support.microsoft.com/kb/883260>.

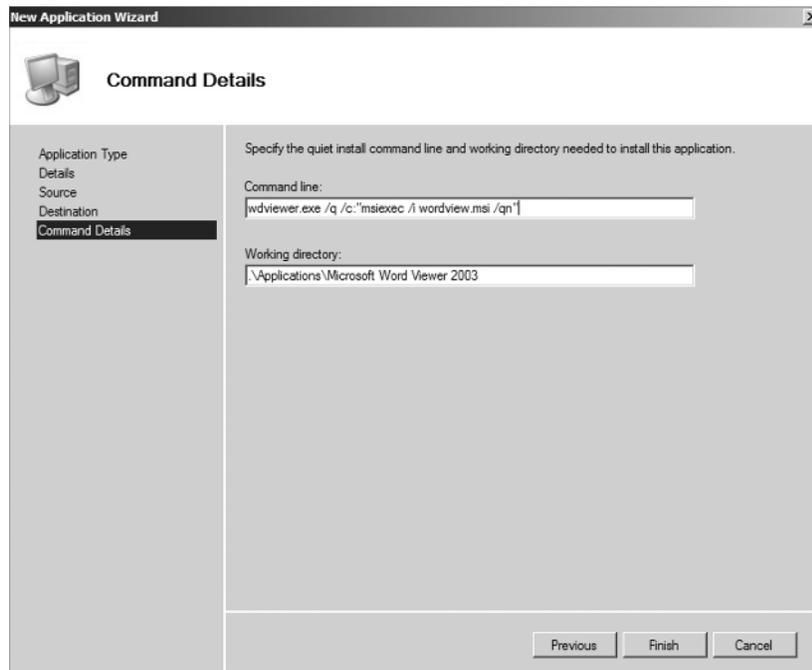
For this example, I downloaded Word Viewer to a folder named D:\Setup\Word Viewer 2003 (and unblocked `wdviewer.exe`).

1. Using Deployment Workbench, expand the Distribution node, right-click the Applications node, and select New. The New Application Wizard starts.
2. On the Application Type page, select “Application with Source files” and click Next.

## 62 Chapter 1 • Deploying Windows with Style

3. On the Details page, add the following information and click Next:  
 Publisher: (Optional): Microsoft  
 Application Name: Word Viewer  
 Version: (Optional): 2003
4. On the Source page, in the Source Directory text box, type **D:\Setup\Word Viewer 2003** and click Next.
5. On the Destination page, in the “Specify the name of the directory that should be created” text box, type **Microsoft Word Viewer 2003** and click Next.
6. On the Command Details page, in the “Command line” text box, type **wdviewer.exe /q /c:"msiexec /i wordview.msi /qn"** as shown in Figure 1.35. Click Finish.

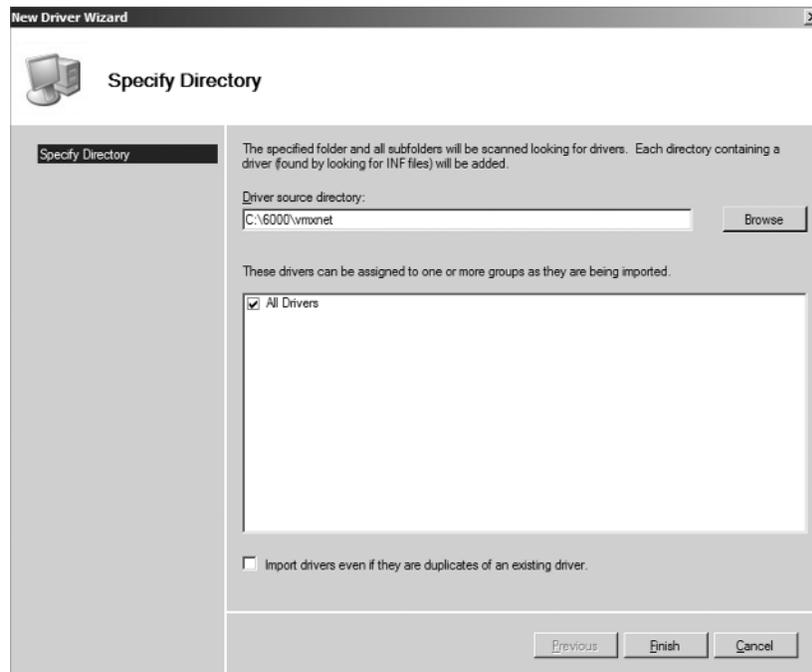
**FIGURE 1.35** The New Application Wizard in Deployment Workbench



### Adding an Out-of-Box Driver to the Deployment Workbench

In this example, you will be adding the VMware NIC drivers to the driver repository.

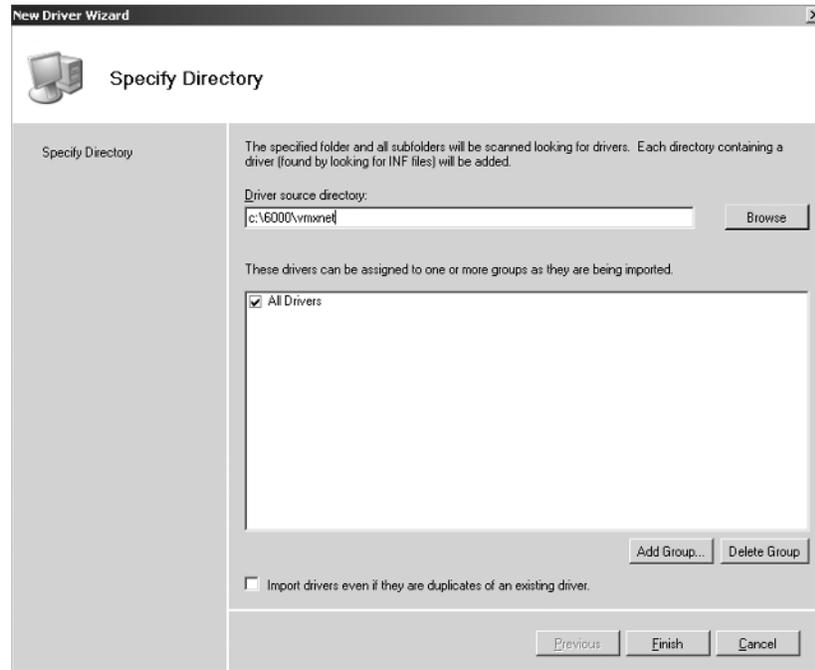
1. Using Deployment Workbench, expand the Distribution node, right-click the Out-of-Box Drivers node, and select New. The New Driver Wizard starts, as shown in Figure 1.36.
2. On the Specify Directory page, type the path to your drivers (in previous examples I stored mine in **C:\6000\vmxnet**) and click Finish.

**FIGURE 1.36** The New Driver Wizard in Deployment Workbench

### Creating a Task Sequence

Task sequences are core to the deployment process; it will control most aspects of the deployment. Here are the steps to create a task sequence.

1. Using the Deployment Workbench, right-click the Task Sequences node, and select New. The New Task Sequence Wizard starts.
2. On the General Settings page, add the following settings, as shown in Figure 1.37. Click Next.
  - Task Sequencer ID: 001
  - Task Sequence name: Windows Vista Enterprise Eng x86
  - Task Sequence comment: Company Master Image
3. On the Select Template page, select the Standard Client Task Sequence and click Next.
4. On the Select OS page, select the Windows Vista Enterprise Eng x86 image and click Next. However, there's something important you need to know about. If the "Select OS" list box is empty it's either because you did not add an operating system or because of a well known bug in Microsoft Deployment Toolkit 2008. If you add an operating system without selecting (left-clicking) the operating system node at least once, the `operatingsystems.xml` file is not generated, and the Select OS list will be empty. The workaround is, of course, to cancel the guide, select the Operating System node once (so the operating systems list is generated), and start the guide again.

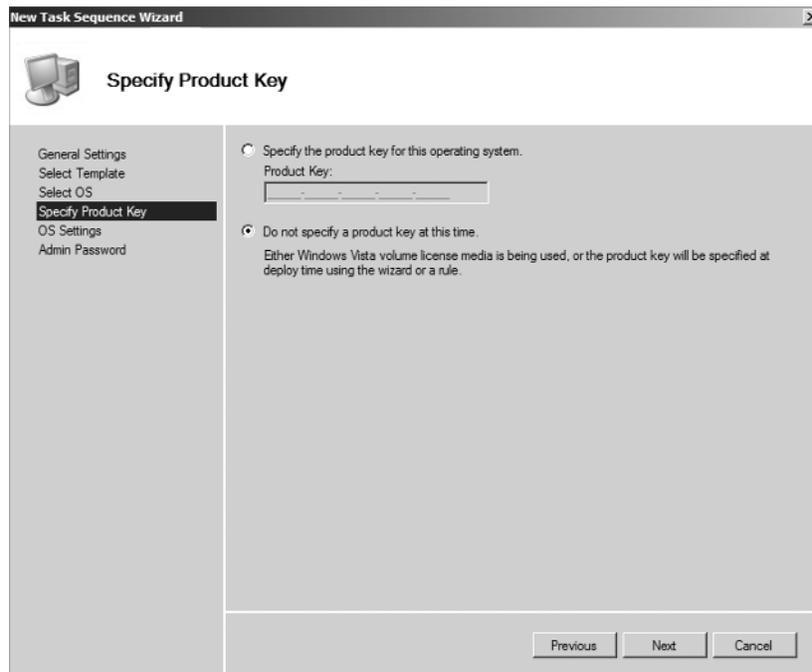
**FIGURE 1.37** The New Task Sequence Wizard in Deployment Workbench

5. On the Specify Product Key page, select “Do not specify a product key at this time,” as shown in Figure 1.38. Click Next. Since we are deploying Windows Vista Enterprise (Volume license) you should not specify the product key, it will be handled by the Key Management Service (KMS). In case you’re doubting me, let me put it this way, if you do add the product key, you will break KMS activation.
6. On the OS Settings page, add the following settings:
  - FullName: IT
  - Organization: CORP
  - Internet Explorer home page: <http://www.gpanswers.com>
 Click Next to continue.
7. On the Admin Password page, in the Administrator Password and Confirm Password text boxes, type the standard password we’ll use for the rest of the book, **P@ssw0rd** (that’s an “at sign” and a zero) and click Finish.

### Creating and Configuring the Deployment Point

Creating a deployment point is basically sharing the D:\Distribution folder, configuring the rules, and then creating the boot image you will be using to start the deployment. Here are the steps for creating and configuring the Lite Touch deployment point.

**FIGURE 1.38** The New Task Sequence Wizard Specify Product Key screen in Deployment Workbench



**Creating the Deployment Point** Microsoft Deployment Toolkit 2008 supports several types of deployment points, but the one we will be creating here is called LAB and is the type we use for Lite Touch Deployments.

1. Using the Deployment Workbench, expand the Deploy node, right-click Deployment Points and select New. The New Deployment Point Wizard starts.
2. On the Choose Type page, select “Lab or single-server deployment” and click Next
3. On the Specify Deployment Point Name page, in the “Deployment point name” text box, type **LiteTouch** and click Next.
4. We’re not making any changes here nor on the remaining screens in this wizard because we will override them when configuring the deployment point in the next section So, on the Application List page, just click Next.
5. On the Allow Image Capture page, click Next.
6. On the Allow Admin Password page, click Next.
7. On the “Allow Product key” page, click Next.
8. On the Network Share page, click Next.
9. On the Configure User State page, click Finish.

**Configuring the Deployment Point** Now that you've created the deployment point in Microsoft Deployment Toolkit 2008, you need to know a thing or two about it before going on to configure it.

When starting, the deployment client will connect back to the deployment point on the server and read its settings. These settings are called rules, and they will control the behavior of the Deployment Wizard (that is, what you see when you start the actual deployment) and configure Windows settings like keyboard, time zone, and so on. For example, the following rules will skip prompting for computer name during deployment and set the keyboard and time zone to Swedish for Windows Vista (I happen to be from Sweden, but you can set the time zone however you wish):

- SkipComputerName=YES
- KeyboardLocale=041d:0000041d
- TimeZoneName=W. Europe Standard Time

By adding more rules, you can fully automate the entire deployment if you like.

With that knowledge, you can go ahead and configure your deployment point:

1. Using the Deployment Workbench, right-click the Lite Touch Deployment Point and select Properties. If you are installing Microsoft Deployment Toolkit 2008 on a Windows Server 2008 with WDS installed, you can take advantage of the new multicast capabilities by checking the "Enable multicast for this deployment point (requires Windows Server 2008 WDS)" check box.
2. Remember all those rules I told you about? Well, now's the time to configure them. In the Rules tab, add the following rules for automating most of the Deployment Wizard screens:



For a great primer on Microsoft Deployment Toolkit 2008 rules, check out Ben Hunter's rule processing article on <http://tinyurl.com/2zuw8t>.

```
[Settings]
Priority=Default
[Default]
_SMSTSORGNAME=CORP
OSInstall=Y
DoCapture=NO
SkipCapture=YES
SkipAppsOnUpgrade=NO
SkipAdminPassword=YES
SkipProductKey=YES
SkipBDDWelcome=YES
SkipComputerName=YES
```

```
SkipDomainMembership=YES
SkipUserData=YES
SkipLocaleSelection=YES
SkipBuild=NO
SkipTimeZone=YES
SkipApplications=NO
SkipBitLocker=YES
SkipSummary=YES
SkipFinalSummary=NO
SkipBDDWelcome=YES
SLShare=\\DC01\Logs$
JoinDomain=CORP
DomainAdmin=CORP\JoinAccount
DomainAdminPassword=P@ssw0rd
MachineObjectOU=ou=workstations,dc=corp, dc=com
TimeZoneName= Pacific Standard Time
```

3. Click the `bootstrap.ini` and configure the following rules:

```
[Settings]
Priority=Default
[Default]
DeployRoot=\\DC01\Distribution$
userID=BuildAccount
UserDomain=CORP
UserPassword=
SkipBDDWelcome=YES
```



For security reasons, I don't recommend configuring the password in `bootstrap.ini`, but if you want to speed up the process for lab purposes, you can do it.

4. Save the file and click OK to close the Lite Touch Properties window.
5. Right-click the Lite Touch deployment point and select Update. The update process will take five to ten minutes.

### Adding the Lite Touch Deploy Boot Image to WDS

Now it is time to add the Lite Touch boot image to WDS so you can boot it over the network.

1. In the WDS Console, right-click Boot Images and select Add Boot Image. The Windows Deployment Services—Add Image Wizard starts.

## 68 Chapter 1 • Deploying Windows with Style

2. On the Image File page, in the File location text box, type **D:\Distribution\Boot\LiteTouchPE\_x86.wim** and click Next.
3. On the Image Metadata page, in the Image Name and Image Description text boxes, type **Lite Touch Windows PE (x86)** and click Next.
4. On the Summary page, click Next.
5. On the Task Progress page, click Finish.

### Verifying Your Microsoft Deployment Toolkit 2008 Lite Touch Configuration

That's it! You have now configured all the components you need in Microsoft Deployment Toolkit 2008, in Lite Touch style, to start deploying Windows Vista to your client.

Before you start installing your first client using Lite Touch, however, let's review what you have done so far:

- You installed Microsoft Deployment Toolkit 2008 and its prerequisite software.
- Using Deployment Workbench, you created a distribution share (D:\Distribution).
- You added an operating system (Windows Vista Enterprise), some drivers (VMware NIC drivers), and an application (Word Viewer).
- You created a task sequence, which will drive the deployment process.
- You created the deployment point, and in that process you also configured the deployment rules and created the WinPE boot image that starts the deployment.
- You added the WinPE boot image to WDS so that you can easily start it by pressing F12 on the client.

This sounds easy enough, doesn't it? It *is* easy. Now it is time to put it to the test and install your first client.

### Installing Your First Client Using Lite Touch

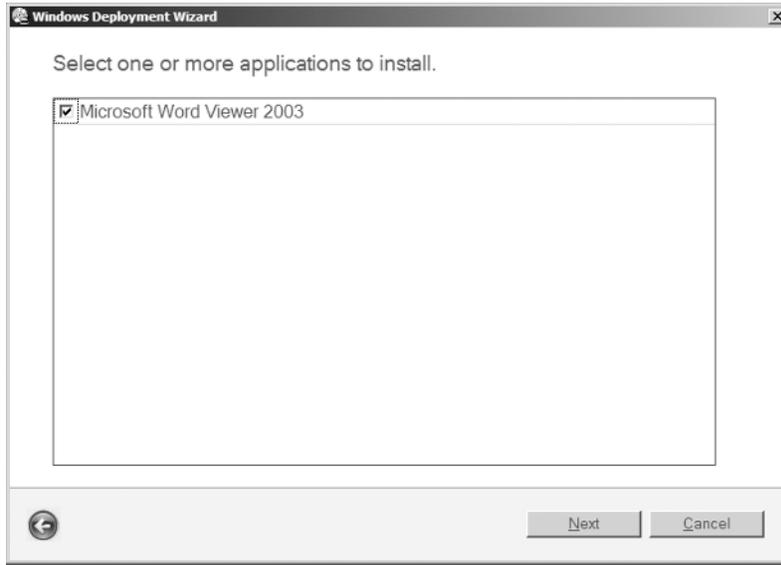
It's time to install your first client using MDLT. When booting the Lite Touch boot image, it will connect to the distribution share, read the rules, and start the deployment.

1. PXE-boot your computer, and select the Lite Touch boot image. After a while the Deployment Wizard starts.
2. On the "Specify credentials for connecting to network shares" page, enter the password and click OK.
3. On the "Select a task sequence to execute on this computer" page, select the "Windows Vista Enterprise Eng x86" image and click Next.
4. On the "Select one or more applications to install" page, select the Microsoft Word Viewer 2003 application, as shown in Figure 1.39. Click Next.

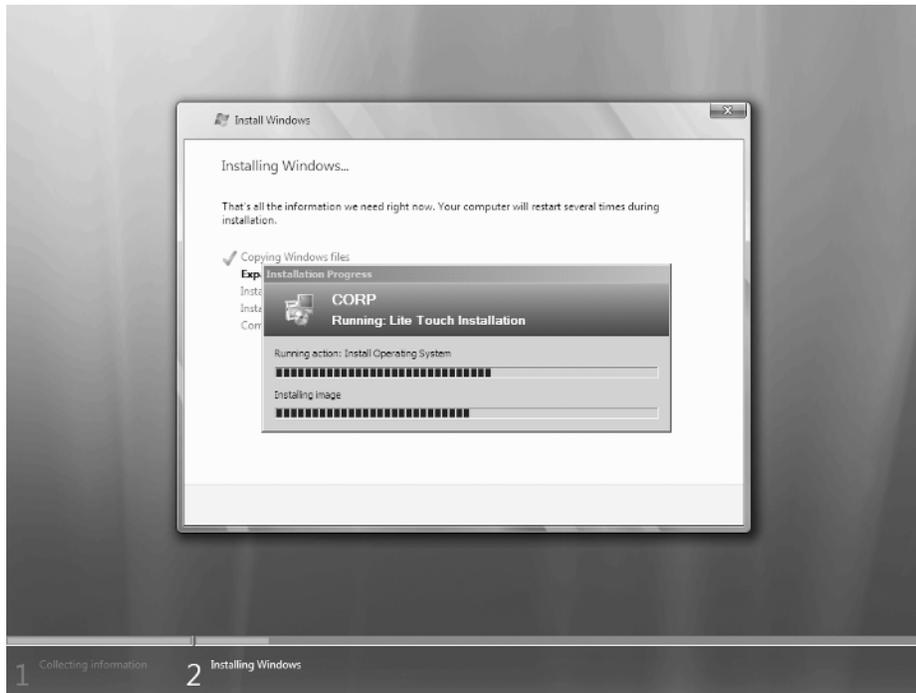
If you have done everything right, Windows Vista Enterprise will now install fully unattended with the settings you have configured. Figure 1.40 shows a screenshot of the process.

You should also end up with two log files from the setup in the D:\Logs folder on DC01.

**FIGURE 1.39** Selecting application(s) during setup



**FIGURE 1.40** The Lite Touch setup running



## Beyond the Microsoft Deployment Toolkit 2008 Basics

In this section, we're going to go beyond the regular setup of Microsoft Deployment Toolkit 2008. Heck, there's a lot you can do with Microsoft Deployment Toolkit 2008, but here are two of my favorites.

First, I'll show you how to save user state as you upgrade from Windows XP to Windows Vista. That's right, you can preserve a lot of stuff from your users' Desktops and more using the User State Migration Tool.

I'll also show you how to add a deployment database (SQL database), which is nice if you have lots of systems and want to dig in and get more out of your data. This database is really useful even for smaller networks since it allows you to configure the client settings before you actually deploy the client. You can configure settings like the computer name, what applications it should have, what drivers it should install and so on.

### Saving User State During an XP-to-Windows-Vista Upgrade

As I mentioned, Microsoft Deployment Toolkit 2008 supports migrating user settings and data when reinstalling a machine. This scenario is called Refresh and does not involve PXE booting or booting from a WinPE CD.

We will not go into great detail about this in here but you will find much more detail in the Microsoft Deployment Toolkit 2008 documentation. In fact, there is a whole document dedicated to user-state migration: `User_State_Migration_Guide.doc`. Reading up on the Refresh scenario in the `Deployment_Feature_Team_Guide.doc` is recommended, too.

#### Refresh Scenario

Say you want to upgrade a Windows XP machine to Windows Vista. You don't want to do an in-place upgrade, but you do want to keep the user data and settings. This is where the Lite Touch Refresh scenario comes in handy.

You can start the process by going to your XP Workstation, logging in as a local administrator and then running the `LiteTouch.vbs` script from the DC01 Server (for example, `\\DC01\Distribution$\Scripts\LiteTouch.vbs`).

The Windows Deployment Wizard will start and you will have to complete the wizard, but after that the process is fully automated. If you choose to do a refresh during this wizard, the following will happen:

- The User State Migration Tool (USMT) will back up your settings to the local `MININT` folder.
- Windows PE will be applied to the disk, the boot loader will be changed, and the computer will be rebooted.
- Windows PE will start and apply the Windows Vista Enterprise operating system to the disk.
- After Windows Vista is installed, the backed up user settings and data will be restored.



The User State Migration Tool (USMT) version 3.0.1 is not included by default in Microsoft Deployment Toolkit 2008. You will have to download it and copy it to the `Distribution\Tools\x86` folder. The URL is <http://tinyurl.com/2jsb6h>.

## Adding Database Support

In order to get a more dynamic, database-driven deployment solution, you can configure Microsoft Deployment Toolkit 2008 to store some of its rules in a SQL Server Express Database (of course, a full-blown SQL Server 2005 will also work, but we'll use SQL Server Express because it's free).



Adding a database may sound like a bit too much if you are a smaller organization, but trust me: it is really worth going that extra mile.

The benefits of using the database are that you get a central portal from where you can control all your client (and server) deployments in great detail. You even get a nice GUI to manage this information directly from the Deployment Workbench.

You can, for example, prestage all your computers into the database, complete with computer name, settings, applications and so on. So when you press F12 to deploy a client, it already knows how it should be configured.

### Installing SQL Server Express Edition SP2

Install SQL Server Express Edition SP2 by following these steps:

1. Run the `SQLSRV32.exe` file. The Microsoft SQL Server 2005 Setup Wizard starts.
2. On the End User License Agreement page, accept the agreement and click Next.
3. On the Installing Prerequisites page, click Install, and then click Next.
4. On the “Welcome to the Microsoft SQL Server installation wizard” page, click Next.
5. On the System Configuration Check page, click Next.
6. On the Registration Information page, clear the “Hide advanced configuration options” check box and click Next.
7. On the Feature Selection page, expand `Database Services` and configure Data Files to be stored on `D:\`. Then click Next.
8. On the Windows Authentication Mode page, click Next.
9. On the Configuration Options page, clear the Enable User Instances check box, and click Next.
10. On the Error and Usage Report Settings page, click Next.
11. On the Ready to Install page, click Install.
12. On the Setup Progress page, click Next.
13. On the Completing Microsoft SQL Server 2005 Setup page, click Finish.

### Installing SQL Server Express Edition Management Tools

The SQL Server Express Edition management tools are useful for configuring the database. Install SQL Server Express Edition management tools by following these steps:

1. Run the `SQLServer2005_SSMSEE.msi` file. The Microsoft SQL Server Management Studio Express Wizard starts.

## 72 Chapter 1 • Deploying Windows with Style

2. On the Welcome to the Install Wizard for Microsoft SQL Server Management Studio Express page, click Next.
3. On the License Agreement page, accept the agreement and click Next.
4. On the Registration Information page, click Next.
5. On the Feature Selection page, click Next.
6. On the “Ready to Install the Program” page, click Install.
7. On the “Completing the Microsoft SQL Server Management Studio Express Setup” page, click Finish.

### Configuring SQL Server Express Edition

To make SQL Server Express Edition work with Microsoft Deployment Toolkit 2008, you need to enable named pipes and start the SQL Server Browser service. Follow these steps:

1. Start SQL Server Configuration Manager and select the SQL Server 2005 Services node.
2. Right-click SQL Server Browser and select Properties. In the Service tab, change Start Mode to Automatic and click OK.
3. Right-click SQL Server Browser and select Start.
4. Expand the SQL Server 2005 Network Configuration node and select “Protocols for SQLEXPRESS.”
5. Right-click Named Pipes and select Enable. In the Warning dialog box, click OK.
6. Select the SQL Server 2005 Services node, right-click SQL Server (SQLEXPRESS) and select Restart.

### Creating the Deployment Database

Now you are ready to create the Microsoft Deployment Toolkit 2008 database. The database is created from the deployment database by following these steps:

1. Using Deployment Workbench, expand the Deploy node and select the Database node.
2. Right-click the Database node and select New. The New DB Wizard starts.
3. On the SQL Server Details page in the SQL Server name text box, enter **DC01**. In the Instance text box, enter **SQLEXPRESS**, and then click Next.
4. On the Database page in the “Create a new database—Database” text box, enter **MD** and click Next.
5. On the SQL Share page, in the SQL Share text box, enter **Logs\$** and click Finish.

### Configuring Microsoft Deployment Toolkit 2008 to Use the Deployment Database

In order for Microsoft Deployment Toolkit 2008 to use the database, we need to configure the rules on the deployment point. This process is a guide where you simply check the settings that you want to query the database for.

1. Using Deployment Workbench, expand the Deploy node and then expand the Database node.
2. Right-click Computers and select New. The Configure DB Wizard starts.



As a general rule, start small. Checking every check box will cause a lot of database queries during deployment. This will increase the deployment time, especially if contacting the database over a WAN link.

3. On the Computer Options page, select the first two check boxes, clear the others, and click Next.
4. On the Location Options page, select the first two check boxes, clear the others, and click Next.
5. On the Make/Model Options page, select the first and the third option, clear the others, and click Next.
6. On the Role Options page, select the first two options, clear the others, and click Finish.

### Adding Entries to the Database

Now you are ready to go ahead and create computer accounts, role-based settings, and so on.

1. Using Deployment Workbench, expand the Deploy node, and then expand the Database node.
2. Right-click Roles and select New. The Properties Windows is displayed.
3. On the Properties window in the Identity tab in the “Role name” text box, type **Standard PC**.
4. In the Details tab, in the “Full name” text box, type **IT**. In the Orgname text box, type **Corp**, and then click OK. You have created a role named Standard PC with some settings.
5. Right-click Computers and select New. The Properties window is displayed.
6. On the Properties window in the Identity tab, complete the Description and Identify fields for one of your target computer. I use the MAC address of one of my virtual machines in this example, as shown in Figure 1.41.

**FIGURE 1.41** Adding a computer account

The screenshot shows a 'Properties' dialog box with the following fields and values:

Field	Value
ID	1
Description	PC00042
Asset tag	
UUID	
Serial number	
MAC address	00:0C:29:84:00:74

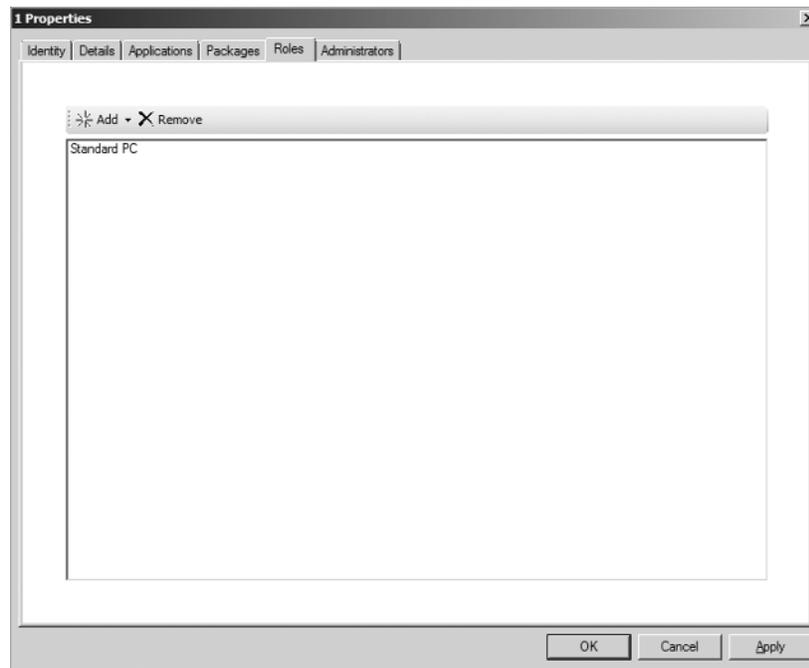
Buttons: OK, Cancel, Apply



The database input fields in Microsoft Deployment Toolkit 2008 are a bit weird to edit in; you hardly ever see the cursor. I recommend editing in Notepad and then copying and pasting into the window.

7. In the Details tab, enter the computer name for the new computer. I use the name PC00042 in this example.
8. In the Roles tab, add the Standard PC role and click OK. Now you have associated the Standard PC role (and its settings) with PC00042, as shown in Figure 1.42.

**FIGURE 1.42** Assign roles to a computer account.



## Troubleshooting Microsoft Deployment Toolkit 2008

When things go wrong (and things will go wrong), you need to find out why. Each Microsoft Deployment Toolkit 2008 script automatically creates a log file when it runs. Log files are stored in the C:\MININT\SMSOSD\OSDLOGS folder during the deployment process and are moved to C:\Windows\Temp when the deployment is completed.



## Real World Scenario

### Microsoft Deployment Toolkit 2008 Best Practices

You will quickly discover that Microsoft Deployment Toolkit 2008 is a very open platform, mostly driven by VBScripts, and you can tweak it to do most things with regards to deployment. With regard to tweaking, the first tip I want to give you is this: don't do that right away. Start small, get the basic built-in features working, and learn how the system works and how to troubleshoot it. Then take it from there.

Here, I summarize some tips and tricks that will help you steer clear of the most common pitfalls.

#### Get Rid of the Junk

The most common issue with Microsoft Deployment Toolkit 2008 is when leftover junk from a previous attempt of deploying an operating system prevents a new attempt. It could be a missing network driver; it could be a rule or setting you made on the deployment point that is causing it to fail in the first place.

If you try several deployments, make sure to either wipe the disk, or remove the MININT and \_SMSTaskSequence folders before trying again. (If you want to go real hardcore, check out my blog entry on how to configure the boot image to always remove the MININT and \_SMSTaskSequence. You can find that at <http://tinyurl.com/2x6ohq>.)

#### Meet the System Requirements

Verify that the clients have at least 512MB RAM (and remember 512MB RAM is not always 512MB RAM, as sometimes shared video memory can be stealing "real" memory).

#### Group Boot Image Drivers

The boot image should have only Ethernet NIC and storage drivers. Adding wireless drivers to WinPE might even cause it to fail when loading other drivers. Group Ethernet NIC and storage drivers to a WinPE drivers group and configure the deployment point to use only that driver group.

#### Dealing with Two-Tier Drivers for WinPE

If you have two-tier network drivers, make sure to download the monolithic version and add it to the WinPE drivers group. The monolithic drivers are sometimes disguised under names like RIS, ADS, or WinPE driver.

#### Do a Final Reboot

Don't leave the system logged in as an admin; make sure it reboots after completing the setup.

See this URL for more information: <http://tinyurl.com/yrhepn>.

**If You Don't Prestage, Then Generate Computer Names**

Make sure you name the computers something useful, or else have the system generate a useful name. Here is information on how to configure a stored procedure for doing that: <http://tinyurl.com/277bd3>.

**If Deploying XP or Windows Server 2003**

Make sure to enable the Set Diskpart BIOS Compatibility Mode action in the Task Sequence. Address a disk partitioning bug in WinPE 2.0, also known as the WinPE 2.0 uberbug. See <http://support.microsoft.com/?id=931760> and <http://support.microsoft.com/?id=931761> for more information about the issue.

Make sure to add a storage driver to `sysprep.inf` before sealing the image (and install the necessary hotfixes). Install at least KB888111 and KB883667, which add support for HD audio and fix an issue with certain video cards freezing the Sysprep mini-setup.

Don't add every storage driver on the planet; add only those you do have. This helps solve issues with Sysprep and the magic CloneTag it adds to the Registry.



The log files have been formatted to be read with Trace32 from SMS 2003 Toolkit 2, not with Notepad. You can download the Trace32 toolkit from <http://tinyurl.com/2zy7no>.

**Creating a Test Environment**

To speed up troubleshooting, you can configure a test environment where you simulate a deployment without actually performing a deployment. This is very valuable when testing new rules, scripts, database queries, and so on. Here are the steps:

1. On a client PC, create a folder named `C:\ZTI`.
2. Copy the `Customsettings.ini` file from `\\DC01\Distribution$\Control` to `C:\ZTI`.
3. Copy the following files from `\\DC01\Distribution$\Scripts` to `C:\ZTI`:
  - `ZTIGather.wsf`
  - `ZTIGather.xml`
  - `ZTIUtility.vbs`
4. Create a batch file (named `LTITest.cmd` in this example) with the following code:

```
Cls
if exist c:\minint\nul rd c:\minint /s /q
cscript.exe ZTIGather.wsf /debug:true
```
5. Run the script from a command prompt, and review the output as well as the log files in `C:\MININT\SMSOSD\OSDLOGS`.

## Interpreting the *ZTIGather.log* File

After running the script, examine the *ZTIGather.log* file. Remember, during deployment the WinPE boot image reads the settings from the deployment point. We are simulating that.

When you review the log file, you will find out how the client read the current deployment point rules (*customsettings.ini*). In the log file you will see, for example, the queries to the database and, if they succeeded, what result the client got back.

Following are some sample lines from a *ZTIGather.log* file and what they really mean.

This entry means the MAC address is 00:0C:29:84:00:74:

```
MAC address = 00:0C:29:84:00:74
```

The next two entries show that the client has read the [Default] section in the rules and found out where to store the log files:

```
Processing the [DEFAULT] section  
Property SLSHARE is now = \\DC01\Log$
```

The next entries show that the client has connected to the deployment database, queried for settings related to its MAC address, and got one record back. In the record there was information on which role this PC had in the database, the Standard PC Role.

```
OPENING TRUSTED SQL CONNECTION to server DC01.
```

```
Connecting to SQL Server using connect string: Provider=SQLOLEDB;OLE DB  
Services=0;Data Source=DC01\SQLEXPRESS;Initial Catalog=MD;Network  
Library=DBNMPNTW;Integrated Security=SSPI
```

```
Successfully opened connection to database.
```

```
About to issue SQL statement: SELECT * FROM ComputerRoles WHERE UUID =  
'125B4D56-5CC1-2697-D688-A2755C840074' OR ASSETTAG = 'No Asset Tag' OR  
SERIALNUMBER = 'VMware-56 4d 5b 12 c1 5c 97 26-d6 88 a2 75 5c 84 00 74' OR  
MACADDRESS IN ('00:0C:29:84:00:74')
```

```
Successfully queried the database.
```

```
Records returned from SQL = 1
```

```
Property ROLE001 is now = Standard PC
```

The next entries show that the client has connected to the deployment database, queried for settings related to its role (Standard PC), and got five records back. These records had information about what five applications the client should install (retrieved from the role).

```
About to issue SQL statement: SELECT * FROM RoleApplications WHERE ROLE IN  
( 'Standard PC' ) ORDER BY Sequence
```

```
Successfully queried the database.
```

```
Records returned from SQL = 5
```

```
The Above three lines
```

```
Property APPLICATIONS001 is now = {87099ba0-77df-4986-9eca-f0bc65c7b202}
```

```
Property APPLICATIONS002 is now = {53f4552a-6125-4b20-914e-5f9ffe5009cc}
```

```
Property APPLICATIONS003 is now = {8fece820-dc80-4cbe-a576-4841dd1a6e84}
```

```
Property APPLICATIONS004 is now = {0980e3d7-f9da-4006-a2e5-566429d28190}
```

```
Property APPLICATIONS005 is now = {26fb69a7-3f9b-4064-8f8a-efe739e6639f}
```

## Final Thoughts

WDS and Microsoft Deployment Toolkit 2008 are big places. If I had to leave you with three things to remember as you work through the exercises in this chapter, they would be:

- When testing, remember that the client needs to have 512MB of RAM, even if you deploy down-level operating systems like Windows XP.
- To maintain a high-level overview, read through the exercises once before actually doing them.
- WDS and Microsoft Deployment Toolkit 2008 are best when used together. These components combined will form a very powerful deployment platform. Did I mention that they are free?

The following links will help you work with WDS and Microsoft Deployment Toolkit 2008:

- <http://www.truesec.com>: where I work as a deployment consultant, and where you can find training classes on WDS and Microsoft Deployment Toolkit 2008.
- <http://www.deployvista.com>: my blog.
- <http://blogs.technet.com/richardsmith>: a Microsoft Services Consultant from the U.K.
- <http://blogs.technet.com/benhunter>: a Microsoft Services Consultant from New Zealand.
- <http://blogs.technet.com/deploymentguys>: a team effort from some of your favorite deployment guys including Ben Hunter, Richard Smith, Adam Shepherd, Daniel Oxley, and more.
- <http://blogs.technet.com/mniehaus>: the Lead Developer for Microsoft Deployment Toolkit 2008.
- <http://blogs.technet.com/msdeployment>: the Deployment team blog.
- <http://www.myitforum.com/absolutevc/avc-view.aspx?v=735>: TechNet Webcast: Automated Windows Server 2008 Imaging and Deployment Using the Microsoft Deployment Accelerator (Level 300).