Microsoft Press

Microsoft TechNet

Deploying Windows® 7

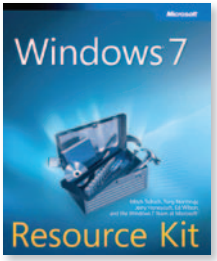
Essential Guidance from the *Windows 7 Resource Kit* and *TechNet Magazine*

Windows 7

Microsoft® Press

FROM THE

Windows® 7 Resource Kit



Mitch Tulloch, Tony Northrup,
Jerry Honeycutt, Ed Wilson,
and the Windows 7 Team at Microsoft

CHAPTER 3

Deployment Platform

Building on technology that the Windows Vista operating system introduced, Windows 7 deployment technology has evolved significantly since Windows XP Professional. For example, it supports file-based disk imaging to make high-volume deployments quicker, more efficient, and more cost effective. The Windows 7 operating system also provides more robust deployment tools through the Windows Automated Installation Kit 2.0 (Windows AIK 2.0), including Windows System Image Manager (Windows SIM) and Windows Preinstallation Environment (Windows PE).

This chapter helps you get started with the Windows 7 deployment platform. It introduces these tools, describing how they relate to each other and providing you with a basic understanding of why and when to use each tool. The remaining chapters in Part II, "Deployment," describe in detail the tools introduced in this chapter. The *Windows Automated Installation Kit User's Guide* in the Windows AIK 2.0 also details each tool described in this chapter.

Tools Introduction

Compared to Windows XP, Windows 7 introduces numerous changes to the technology you use for deployment. Additionally, Windows 7 improves and consolidates many of the tools you used for Windows Vista deployment. The Windows AIK 2.0 includes most of

these tools. Others are built into the operating system. The Windows AIK 2.0 fully documents all of the tools this chapter describes, including command-line options for using them, how they work on a detailed level, and so on.

> ***NOTE*** **The Windows AIK 2.0 is not included in the Windows 7 media. (By comparison, Windows XP has a file called Deploy.cab that includes its deployment tools.) Instead, the Windows AIK 2.0 is a free download from the Microsoft Download Center at *http://www.microsoft.com/downloads*.**

The following features are new for Windows 7 deployment:

- **Windows System Image Manager** Windows System Image Manager (Windows SIM) is a tool for creating distribution shares and editing answer files (Unattend.xml). It exposes all configurable settings in Windows 7; you use it to save customizations in Unattend.xml. The Windows AIK 2.0 includes the Windows SIM.
- **Windows Setup** Setup for Windows 7 installs the Windows image (.wim) file and uses the new Unattend.xml answer file to automate installation. Unattend.xml replaces the set of answer files used in earlier versions of Windows (Unattend.txt, Sysprep.inf, and so on). Because image-based setup (IBS) is faster, you can use it in high-volume deployments and for automating image maintenance. Microsoft made numerous improvements to Windows Setup (now called Setup.exe instead of Winnt.exe or Winnt32.exe), such as a completely graphical user interface, use of a single answer file (Unattend.xml) for configuration, and support for configuration passes (phases).
- **Sysprep** The System Preparation (Sysprep) tool prepares an installation of Windows 7 for imaging, auditing, and deployment. You use imaging to capture a customized Windows 7 image that you can deploy throughout your organization. You use audit mode to add additional device drivers and applications to a Windows 7 installation and test the integrity of the installation before handing off the computer to the end user. You can also use Sysprep to prepare an image for deployment. When the end user starts Windows 7, Windows Welcome starts. Unlike earlier versions of Windows, Windows 7 includes Sysprep natively—you no longer have to download the current version.
- **Windows Preinstallation Environment** Windows Preinstallation Environment 3.0 (Windows PE 3.0) provides operating system features for installing, troubleshooting, and recovering Windows 7. Windows PE 3.0 is the latest release of Windows PE based on Windows 7. With Windows PE, you can start a computer from a network or removable media. Windows PE provides the network and other resources necessary to install and troubleshoot Windows 7. Windows Setup, Windows Deployment Services, Microsoft System Center Configuration Manager 2007 R2, and Microsoft Deployment Toolkit 2010 (MDT 2010) all use Windows PE to start computers. The Windows AIK 2.0 includes Windows PE 3.0.

- **Deployment Image Servicing and Management** Deployment Image Servicing and Management (DISM) is a new command-line tool that you can use to service a Windows 7 image or prepare a Windows PE image. DISM consolidates the functionality of the Package Manager (Pkgmgr.exe), PEImg, and Intlcfg tools from Windows Vista. You can use DISM to service packages, device drivers, Windows 7 features, and international settings in Windows 7 images. Additionally, DISM provides rich enumeration features that you can use to determine the contents of Windows 7 images.
- **ImageX** ImageX is a command-line tool that you can use to capture, modify, and apply file-based images for deployment. Windows Setup, Windows Deployment Services, System Center Configuration Manager 2007, and MDT 2010 all use ImageX to capture, edit, and deploy Windows 7 images. Windows 7 improves ImageX over Windows Vista by enabling it to mount multiple images simultaneously and support interim saves (you must still service each mounted image individually by using DISM). Additionally, the Windows 7 version of ImageX has a new architecture for mounting and servicing images that is more robust than in Windows Vista. The Windows AIK 2.0 includes ImageX. You can also mount images in Windows PE, and Windows 7 includes the device driver inbox.
- **Windows Imaging** Microsoft delivers Windows 7 on product media as a highly compressed Windows Imaging (.wim) file. You can install Windows 7 directly from the Windows 7 media or customize the image for deployment. Windows 7 images are file based, allowing you to edit them nondestructively. You can also store multiple operating system images in a single .wim file.
- **DiskPart** Using DiskPart, you can mount a virtual hard disk (.vhd) file offline and service it just like a Windows image file.
- **User State Migration Tool** You can use the User State Migration Tool 4.0 (USMT 4.0) to migrate user settings from the previous operating system to Windows 7. Preserving user settings helps ensure that users can get back to work quickly after deployment. USMT 4.0 provides new features that improve its flexibility and performance over USMT 3.0. Hard-link migration improves performance in refresh scenarios, offline migration enables you to capture user state from within Windows PE, and the document finder reduces the need for you to create custom migration Extensible Markup Language (XML) files when capturing all user documents. The Windows AIK 2.0 includes USMT 4.0.

Windows 7 Deployment Terminology

The following terms are unique to Windows 7 deployment and MDT 2010. Understanding this terminology will help you better understand the deployment content in this book and the resources it refers to:

- **Answer file** An XML file that scripts the setup experience and installation settings for Windows 7. The answer file for Windows Setup is usually Unattend.xml or

Autounattend.xml. You can use Windows SIM to create and modify this answer file. MDT 2010 builds answer files automatically, which you can customize if necessary.

- **Catalog file** A binary file that contains the state of all the settings and packages in a Windows 7 image. When you use Windows SIM to create a catalog file, it enumerates the Windows 7 image for a list of all settings in that image as well as the current list of features and their current states. Because the contents of a Windows 7 image can change over time, it is important that you re-create the catalog file whenever you update an image.
- **Feature** A part of the Windows 7 operating system that specifies the files, resources, and settings for a specific Windows 7 feature or part of a Windows 7 feature. Some features include unattended installation settings, which you can customize by using Windows SIM.
- **Configuration pass** A phase of Windows 7 installation. Windows Setup installs and configures different parts of the operating system in different configuration passes. You can apply Windows 7 unattended installation settings in one or more configuration passes. For more information about configuration passes, see the *Windows Automated Installation Kit User's Guide* in the Windows AIK 2.0.
- **Configuration set** A file and folder structure that contains files that control the preinstallation process and define customizations for the Windows 7 installation.
- **Destination computer** The computer on which you install Windows 7 during deployment. You can either run Windows Setup on the destination computer or copy a master installation onto a destination computer. The term *target computer* is also commonly used to refer to this.
- **Deployment share** A folder that contains the source files for Windows products that you install. It may also contain additional device drivers and application files. You can create this folder manually or by using Windows SIM. In MDT 2010, the *deployment share*, called a distribution share in previous versions of MDT, contains operating system, device driver, application, and other source files that you configure with task sequences.
- **Image-based setup** A setup process based on applying an image of an operating system to the computer.
- **Master computer** A fully assembled computer containing a master installation of Windows 7 that you capture to a master image and deploy to destination computers. The term *source computer* is also commonly used to refer to this.
- **Master image** A collection of files and folders (usually compressed into one file) captured from a master installation. This image contains the base operating system as well as additional applications, configurations, and files.
- **Master installation** A Windows 7 installation on a master computer that you can capture as a master image. You can create the master installation using automation to ensure a consistent and repeatable configuration each time.

- **Package** A group of files that Microsoft provides to modify Windows 7 features. Package types include service packs, security updates, language packs, and hotfixes.
- **Task sequence** A sequence of tasks that runs on a destination computer to install Windows 7 and applications and then configures the destination computer. In MDT 2010, task sequences drive the installation routine.
- **Task Sequencer** The MDT 2010 component that runs the task sequence when installing a build.
- **Technician computer** The computer on which you install and use MDT 2010 or Windows AIK 2.0. This computer is typically located in a lab environment, separate from the production network. It can be a workstation- or a server-class computer.
- **Unattend.xml** The generic name for the Windows 7 answer file. Unattend.xml replaces all the answer files in earlier versions of Windows, including Unattend.txt, Winbom.ini, and others.
- **.wim** A file name extension that identifies Windows image files created by ImageX.
- **Windows 7 feature** An optional feature of Windows 7 that you can enable or disable by using Unattend.xml or DISM.
- **Windows image file** A single compressed file containing a collection of files and folders that duplicate a Windows installation on a disk volume. Windows image files have the .wim file extension.

Platform Components

Understanding the new deployment tools and how they interconnect is the first step in beginning a Windows 7 deployment project. Figure 3-1 illustrates the Windows 7 deployment platform. At the lowest tier are Windows Imaging (.wim) files, which are highly compressed, file-based operating system images.

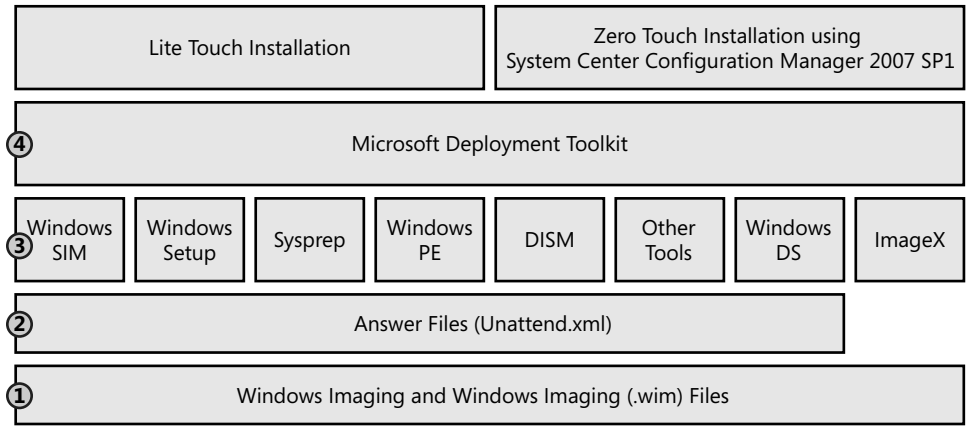


FIGURE 3-1 Windows 7 deployment platform components

At the second tier are answer files. Versions of Windows earlier than Windows Vista had numerous answer files, including Unattend.txt and Sysprep.inf, to drive the deployment process. Windows 7 uses a single XML-based answer file, Unattend.xml, to drive all its *configuration passes*. (A configuration pass is an installation phase.) This improvement makes configuration more consistent and simplifies engineering.

At the third tier are the various deployment tools for Windows 7. The Windows 7 distribution media includes some of these tools, including Sysprep, DISM, and other command-line tools—they aren't on the media in a separate file such as Deploy.cab. The Windows AIK 2.0 includes the bigger tools, such as Windows SIM, Windows PE, and ImageX. These are the basic tools necessary to create, customize, and deploy Windows 7 images. They are standalone tools that don't provide a deployment framework or add business intelligence and best practice to the process.

The fourth tier, MDT 2010, provides the framework, business intelligence, and best practices. MDT 2010 is a process and technology framework that uses all the tools in the third tier, potentially saving your organization hundreds of hours of planning, developing, testing, and deployment. MDT 2010 is based on best practices developed by Microsoft, its customers, and its partners. It includes time-proven management and technology guidance as well as thousands of lines of thoroughly tested script code that you can use as is or customize to suit your organization's requirements.

Using MDT 2010, you can perform both Lite Touch Installation (LTI) and Zero Touch Installation (ZTI) deployment. LTI requires very little infrastructure and is suitable for most small and medium businesses. ZTI requires a System Center Configuration Manager 2007 R2 infrastructure and is suitable for organizations that already have the infrastructure in place.

The following sections provide more information about the components shown in Figure 3-1. For more information about the deployment process using the components in the first three tiers, see the section titled "Basic Deployment Process" later in this chapter. For more information about the deployment process using MDT 2010, see the section titled "Microsoft Deployment Toolkit Process" later in this chapter.

Windows Imaging

Windows 7 is distributed in .wim files, which use the Windows Imaging file format. This format has the following advantages:

- Windows Imaging files are a file-based image format that lets you store multiple images in one file. You can perform partial volume captures by excluding files, such as paging files, that you don't want to deploy using the image.
- This format reduces file sizes significantly by using a compressed file format and single-instance storage techniques: The image file contains one physical copy of a file for each instance of it in the image file, which significantly reduces the size of image files that contain multiple images.

- You can service the image contained in the .wim file—adding and deleting packages, software updates, and device drivers, for example—without re-creating a new image by applying it, customizing it again, and recapturing it.
- You can mount .wim files as folders, making it easier to update files in images they contain.
- Windows Imaging files allow you to apply an image nondestructively to the destination computer's hard disk. You can also apply an image to different-sized destination drives because .wim files don't require the destination hard disk to be the same size or larger than the source hard disk.
- Windows Imaging files can span media so that you can use CD-ROMs to distribute large .wim files.
- Windows PE .wim files are bootable. For example, you can start Windows PE from a .wim file. In fact, Windows Setup and Windows Deployment Services start Windows PE from the .wim file Boot.wim, which you can customize by adding items such as device drivers and scripts.

NOTE **ImageX is the tool you use to manage .wim files. For more information about ImageX, see the section titled "ImageX" later in this chapter, and Chapter 6, "Developing Disk Images."**

Answer Files

An answer file is an XML-based file that contains settings to use during a Windows 7 installation. An answer file can fully automate all or part of the installation process. In an answer file, you provide settings such as how to partition disks, the location of the Windows 7 image to install, and the product key to apply. You can also customize the Windows 7 installation, including adding user accounts, changing display settings, and updating Windows Internet Explorer favorites. Windows 7 answer files are commonly called Unattend.xml.

You use Windows SIM (see the section titled "Windows SIM" later in this chapter) to create an answer file and associate it with a particular Windows 7 image. This association allows you to validate the settings in the answer file against the settings available in the Windows 7 image. However, because you can use any answer file to install any Windows 7 image, Windows Setup ignores settings in the answer file for features that do not exist in the Windows image.

The features section of an answer file contains all the feature settings that Windows Setup applies. Answer files organize features into different configuration passes: windowsPE, offlineServicing, generalize, specialize, auditSystem, auditUser, and oobeSystem. (See the sidebar titled "How It Works: Configuration Passes" later in this chapter.) Each configuration pass represents a different installation phase, and not all passes run during the normal Windows 7 setup process. You can apply settings during one or more passes. If a setting is available in more than one configuration pass, you can choose the pass in which to apply the setting.

MORE INFO **The *Windows Automated Installation Kit User's Guide* in the Windows AIK 2.0 fully documents the features you can configure using Windows SIM and the settings available for each feature.**

Microsoft uses packages to distribute software updates, service packs, and language packs. Packages can also contain Windows features. By using Windows SIM, you can add packages to a Windows 7 image, remove them from a Windows 7 image, or change the settings for features within a package.

The Windows Foundation Package, included in all Windows 7 images, includes all core Windows 7 features such as Media Player, Games, and Windows Backup. Features are either enabled or disabled in Windows 7. If a Windows 7 feature is enabled, the resources, executable files, and settings for that feature are available to users on the system. If a Windows 7 feature is disabled, the package resources are not available, but the resources are not removed from the system.

Windows SIM

Windows SIM is the tool you use to create and configure Windows 7 answer files. You can configure features, packages, and answer file settings. Windows Setup uses Unattend.xml to configure and customize the default Windows 7 installation for all configuration passes. For instance, you can customize Internet Explorer, configure Windows Firewall, and specify the hard drive configuration. You can use Windows SIM to customize Windows 7 in many ways, including the following:

- Install third-party applications during installation.
- Customize Windows 7 by creating answer files (Unattend.xml).
- Apply language packs, service packs, and updates to an image during installation.
- Add device drivers to an image during installation.

With versions of Windows earlier than Windows Vista, you had to edit answer file settings manually using a text editor, even after initially creating an answer file by using Windows Setup Manager. The Windows 7 answer file (Unattend.xml) is based on XML and is far too complex to edit manually, however. So you must use Windows SIM to edit Windows 7 answer files. Figure 3-2 shows Windows SIM.

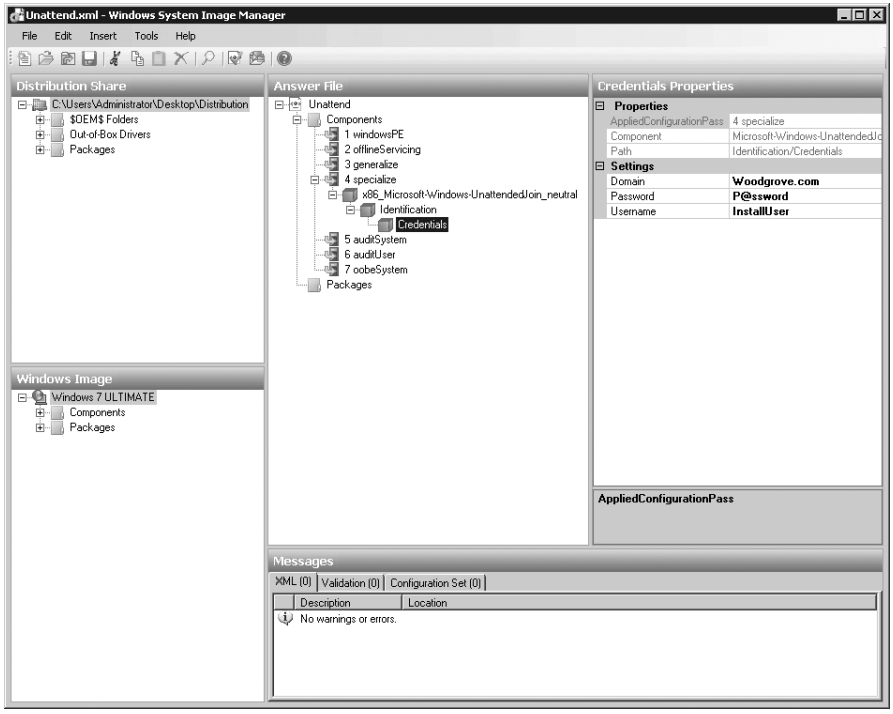


FIGURE 3-2 Windows SIM

Windows Setup

Windows Setup (Setup.exe) is the program that installs Windows 7. It uses image-based setup (IBS) to provide a single, unified process with which all customers can install Windows. IBS performs clean installations and upgrades of Windows. Windows Setup and IBS allow you to deploy Windows 7 in your organization easily and cost effectively.

Windows Setup includes several new features that facilitate installations that are faster and more consistent than Windows XP, including the following:

- **Improved image management** Windows 7 images are stored in a single .wim file. A .wim file can store multiple instances of the operating system in a single, highly compressed file. The install file, Install.wim, is located in the Sources folder on the Windows 7 media.
- **Streamlined installation** Windows Setup is optimized to enable the deployment scenarios used by most organizations. Installation takes less time and provides a more consistent configuration and deployment process, resulting in lower deployment costs.

- **Faster installations and upgrades** Because Windows Setup is now image based, installing and upgrading Windows 7 is faster and easier. You can perform clean installations of Windows 7 by deploying the Windows image to destination computers; you perform upgrades by installing a new image onto an existing installation of Windows. Windows Setup protects the previous Windows settings during the installation.

Windows Setup improves the installation experience over Windows Vista. For example, Windows Setup moves the license key to the Windows Welcome page, allowing users to type a product key after completing installation. Windows Setup also automatically creates a small, hidden partition for BitLocker Drive Encryption. This makes it easier to enable BitLocker Drive Encryption later, because users don't have to prepare the drive. Additionally, the last phase of Windows Setup, Windows Welcome, is faster and gives more feedback on the progress of the setup process.

Sysprep

You use Sysprep to prepare a master installation for imaging and deployment. Sysprep does the following:

- **Removes computer-specific and operating system–specific installation data from Windows 7** Sysprep can remove all computer-specific information from an installed Windows 7 image, including the computer security identifier (SID). You can then capture and install the Windows installation throughout your organization.
- **Configures Windows 7 to boot in audit mode** You can use audit mode to install third-party applications and device drivers, as well as to test the functionality of the computer, before delivering the computer to the user.
- **Configures Windows 7 to boot to Windows Welcome** Sysprep configures a Windows 7 installation to boot to Windows Welcome the next time the computer starts. Generally, you configure a system to boot to Windows Welcome as a final step before delivering the computer to the user.
- **Resets Windows Product Activation** Sysprep can rearm (reset) Windows Product Activation up to three times.

Sysprep.exe is located in the %WinDir%\System32\Sysprep directory on all Windows 7 installations. (You do not have to install Sysprep separately, as in earlier versions of Windows, because it's a native part of the installation.) You must always run Sysprep from the %WinDir%\System32\Sysprep directory on the version of Windows 7 with which it was installed. For more information about Sysprep, see the *Windows Automated Installation Kit User's Guide* in the Windows AIK 2.0.

Windows PE

Prior to Windows PE, organizations often had to use MS-DOS boot floppies to start destination computers and then start Windows Setup from a network share or other distribution media. MS-DOS boot floppies had numerous limitations, however, including that they offered no support for the NTFS file system and no native networking support. In addition, they needed to locate 16-bit device drivers that worked in MS-DOS.

Now Windows PE 3.0 provides a minimal Win32 or Win64 operating system with limited services—built on the Windows 7 kernel—that you use to prepare a computer for Windows 7 installation, copy images to and from a network file server, and start Windows Setup. Windows PE 3.0 is a stand-alone preinstallation environment and an integral component of other setup and recovery technologies, such as Windows Setup, Windows Deployment Services, System Center Configuration Manager 2007 R2, and MDT 2010. Unlike earlier versions of Windows PE, which were available only as a Software Assurance (SA) benefit, Windows PE 3.0 is now publicly available in the Windows AIK 2.0.

Windows PE provides the following features and capabilities:

- Native support for NTFS 5.x file system, including dynamic volume creation and management
- Native support for Transmission Control Protocol/Internet Protocol (TCP/IP) networking and file sharing (client only)
- Native support for 32-bit (or 64-bit) Windows device drivers
- Native support for a subset of the Win32 Application Programming Interface (API); optional support for Windows Management Instrumentation (WMI) and Windows Script Host (WSH)
- Can be started from multiple media, including CD, DVD, USB Flash drive (UFD), and Windows Deployment Services

Windows PE runs every time you install Windows 7, whether you install the operating system by booting the computer with the Windows 7 DVD or deploying Windows 7 from Windows Deployment Services. The graphical tools that collect configuration information during the windowsPE configuration pass run within Windows PE. In addition, you can customize and extend Windows PE to meet specific deployment needs. For example, MDT 2010 customizes Windows PE for LTI by adding device drivers, deployment scripts, and so on.

For Windows 7, Windows PE 3.0 includes improvements that make it easier to customize. First, the functionality of PEImg is now included in DISM, providing a single tool you can use to service images whether they're Windows 7 images or Windows PE images. Second, Windows PE 3.0 uses a new package model. Instead of the base image including all the feature packages from which you remove the disabled features, the base image doesn't include these feature packages, and you add the features that you want to include in the image. For more information about Windows PE, see Chapter 9, "Preparing Windows PE."

NOTE **Because Windows PE is only a subset of Windows 7, it has limitations. For example, Windows PE automatically stops running the shell and reboots after 72 hours of continuous use to prevent piracy. You cannot configure Windows PE as a file server, terminal server, or embedded operating system. Moreover, mapped driver letters and changes to the registry are not persistent between sessions. For more information about the limitations of Windows PE, see the *Windows Preinstallation Environment User's Guide* in the Windows AIK 2.0.**

Deployment Image Servicing and Management

Deployment Image Servicing and Management (DISM) is a new command-line tool that you can use to service Windows 7 images offline before deployment. With DISM, you can install, remove, configure, and update Windows features, packages, device drivers, and international settings. You can use some DISM commands to service online Windows 7 images.

You can use DISM to:

- Add, remove, and enumerate packages.
- Add, remove, and enumerate drivers.
- Enable or disable Windows features.
- Apply changes based on the offlineServicing section of an Unattend.xml answer file.
- Configure international settings.
- Upgrade a Windows image to a different edition.
- Prepare a Windows PE image.
- Take advantage of better logging.
- Service earlier versions of Windows.
- Service all platforms (32-bit, 64-bit, and Itanium).
- Service a 32-bit image from a 64-bit host, and vice versa.
- Use old Package Manager scripts.

DISM consolidates the functionality of the Package Manager (Pkgmgr.exe), PEImg, and Intlcfg tools from Windows Vista. It provides one tool to use for servicing Windows 7 and Windows PE images.

Other Tools

Windows 7 and the Windows AIK 2.0 also provide various command-line tools that are useful during deployment:

- **BCDboot** BCDboot can set up a system partition or repair the boot environment on a system partition quickly. It copies a small set of boot environment files from the installed Windows 7 image to the system partition. It also creates a boot configuration data (BCD) store on the system partition, which includes a new boot entry that enables the Windows image to boot.

- **Bootsect** Bootsect.exe updates the master boot code for hard-disk partitions to switch between BOOTMGR and NTLDR. You can use this tool to restore the boot sector on your computer. This tool replaces FixFAT and FixNTFS.
- **DiskPart** DiskPart is a text-mode command interpreter in Windows 7. You can use DiskPart to manage disks, partitions, or volumes by using scripts or direct input at a command prompt. In Windows 7, DiskPart can also mount .vhd files. Mounting a .vhd file allows you to service it or make other offline changes.
- **Drvload** The Drvload tool adds out-of-box drivers to a booted Windows PE image. It takes one or more driver .inf files as inputs. To add a driver to an offline Windows PE image, use the DISM tool. If the driver .inf file requires a reboot, Windows PE will ignore the request. If the driver .sys file requires a reboot, you cannot add the driver with Drvload.
- **Expand** The Expand tool expands one or more compressed update files. Expand.exe supports opening updates for Windows 7 as well as previous versions of Windows. By using Expand, you can open and examine updates for Windows 7 on a Windows XP or Microsoft Windows Server 2003 operating system.
- **Lpksetup** You can use Lpksetup to perform unattended or silent-mode language pack operations. Lpksetup runs only on an online Windows 7 operating system.
- **Oscdimg** Oscdimg is a command-line tool for creating an image (.iso) file of a customized 32-bit or 64-bit version of Windows PE. You can then burn an .iso file to a CD-ROM or DVD-ROM or copy its contents to a bootable UFD.
- **Powercfg** You can use the Powercfg tool to control power settings and configure computers to default to Hibernate or Standby modes. In Windows 7, Powercfg provides troubleshooting help for diagnosing energy consumption problems.
- **Winpeshl** Winpeshl.ini controls whether a custom shell is loaded in Windows PE instead of the default Command Prompt window.
- **Wpeinit** Wpeinit is a command-line tool that initializes Windows PE each time it boots. When Windows PE starts, Winpeshl.exe executes Startnet.cmd, which starts Wpeinit.exe. Wpeinit.exe specifically installs Plug and Play (PnP) devices, processes Unattend.xml settings, and loads network resources. Wpeinit replaces the initialization function previously supported using the Factory.exe –*winpe* command. Wpeinit outputs log messages to C:\Windows\System32\Wpeinit.log.
- **Wpeutil** The Windows PE utility (Wpeutil) is a command-line tool that you can use to run various commands in a Windows PE session. For example, you can shut down or reboot Windows PE, enable or disable Windows Firewall, set language settings, and initialize a network.

Windows Deployment Services

Windows Deployment Services is the updated and redesigned version of Remote Installation Services (RIS) in Windows Server 2008. Windows Deployment Services helps organizations rapidly deploy Windows operating systems, particularly Windows 7. Using Windows Deployment Services, you can deploy Windows operating systems over a network without having to be physically present at the destination computer and without using the media.

Windows Deployment Services delivers a better in-box deployment solution than RIS. It provides platform components that enable you to use custom solutions, including remote boot capabilities; a plug-in model for Pre-Boot Execution Environment (PXE) server extensibility; and a client-server communication protocol for diagnostics, logging, and image enumeration. Also, Windows Deployment Services unifies on a single image format (.wim) and provides a greatly improved management experience through the Microsoft Management Console (MMC) and scriptable command-line tools.

Windows Deployment Services uses the Trivial File Transfer Protocol (TFTP) to download network boot programs and images. TFTP uses a configurable windowing mechanism that reduces the number of packets network boot clients send, improving performance. Also, Windows Deployment Services now logs detailed information about clients to the Windows Server 2008 logging feature. You can export and process these logs by using Microsoft Office InfoPath or other data mining tools. The most significant new feature, and possibly the most anticipated, is multicast. Multicast deployment allows you to deploy Windows 7 to many computers simultaneously, conserving network bandwidth.

In Windows 7 and Windows Server 2008 R2, Windows Deployment Services includes the following new features:

- **Multicast Multiple Stream Transfer** Allows you to set performance thresholds on multicast clients. Slower clients can move to slower streams so that they don't slow down faster clients, which was a limitation in the original multicast feature.
- **Dynamic Driver Provisioning** Allows Windows Setup to choose device drivers stored on Windows Deployment Services servers dynamically during deployment. This makes updating Windows images with new device drivers less important because you can just add them to the driver store, reducing image size and maintenance costs. You can also insert device drivers into Windows PE images directly from the driver store.

For more information about Windows Deployment Services, see Chapter 10, "Configuring Windows Deployment Services."

ImageX

ImageX is the Windows 7 tool that you use to work with .wim image files. ImageX is an easy-to-use command-line utility. You use ImageX to create and manage .wim image files. With ImageX, you can capture images and apply them to destination computers' hard drives. You can mount .wim image files as folders and thereby edit images offline. ImageX addresses the challenges that organizations faced when using sector-based imaging formats or the MS-DOS

XCopy command to copy an installation of Windows onto new hardware. For example, sector-based imaging:

- Destroys the existing contents of the destination computer's hard drive, complicating migration scenarios.
- Duplicates the hard drive exactly; therefore, the image can deploy only to partitions that are the same type and at least as large as the source partition on the master computer.
- Does not allow for direct modification of image file contents.

The limitations of sector-based imaging led Microsoft to develop ImageX and the accompanying .wim image file format. You can use ImageX to create an image, modify the image without going through the extraction and re-creation process, and deploy the image to your environment—all using the same tool.

Because ImageX works at the file level, it provides numerous benefits. It provides more flexibility and control over your images. For example, you can mount an image onto a folder and then add files to, copy files from, and delete files from the image using a file-management tool such as Windows Explorer. ImageX allows for quicker deployment of images and more rapid installations. With the file-based image format, you can also deploy images nondestructively so that ImageX does not erase the destination computer's hard drive.

ImageX also supports highly compressed images. First, .wim files support single instancing: File data is stored separately from path information so .wim files can store duplicate files that exist in multiple paths at one time. Second, .wim files support two compression algorithms—fast and maximum—that give you control over the size of your images and the time required to capture and deploy them.

Deployment Scenarios

In general, you will perform automated Windows 7 deployments in four scenarios: Upgrade Computer (in-place upgrade), New Computer (wipe-and-load), Refresh Computer, and Replace Computer. The following sections provide an overview of each scenario.

Upgrade Computer Scenario

You can upgrade from Windows Vista Service Pack 1 (SP1) to Windows 7 in place, which means you can install Windows 7 and retain your applications, files, and settings as they were in your previous version of Windows Vista. If you want to upgrade from Windows XP to Windows 7, however, you will need to use the Refresh Computer scenario, which preserves files and settings but not applications. For more information, see the section titled "Refresh Computer Scenario" later in this chapter.

Although upgrading might be the simplest way to deploy Windows 7, you run the risk of preserving misconfigurations and unauthorized software or settings. In many cases, the existing system configuration is difficult to assess and change-control processes are more difficult

to implement. Upgrading from Windows Vista with Service Pack 1 computers in an unknown state to Windows 7 does not change the computer's status—its state is still unknown. A better scenario for managed environments is to use the New Computer scenario with user state migration to preserve settings selectively (that is, the Refresh Computer scenario).

New Computer Scenario

In the New Computer scenario, you install a clean copy of Windows 7 on a clean (freshly partitioned and formatted) hard drive. This scenario has the most consistent results, creating a configuration in a known state. Installing a known configuration on a clean computer is the foundation of good configuration management. You can use established change-control processes to manage any subsequent changes closely.

Refresh Computer Scenario

The Refresh Computer scenario is similar to the New Computer scenario. The differences are that the destination computer contains a Windows operating system, and this scenario preserves users' existing files and settings, as shown in Figure 3-3.

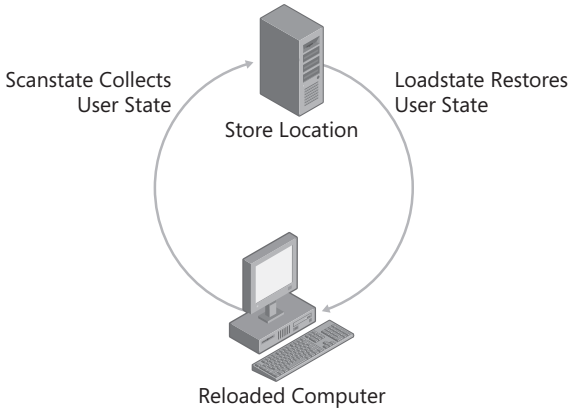


FIGURE 3-3 Preserving user state during migration

You can use migration technologies, such as USMT 4.0, to migrate users' files and settings from the previous version of Windows to Windows 7. This helps ensure that no data is lost while still establishing the best possible system configuration. For more information about using USMT 4.0, see Chapter 7, "Migrating User State Data." You can think of the Refresh Computer scenario as combining the benefits of a new installation that the New Computer scenario provides with the benefits of preserving files and settings.

Replace Computer Scenario

Windows migration technologies such as the Windows Easy Transfer tool and USMT 4.0 allow side-by-side data migration between an old computer running Windows XP or Windows Vista and a new computer running Windows 7. This scenario, which is called Replace Computer, allows you to perform a clean installation on the new computer and simply migrate files and settings from the old one. Figure 3-4 shows an overview of this scenario.

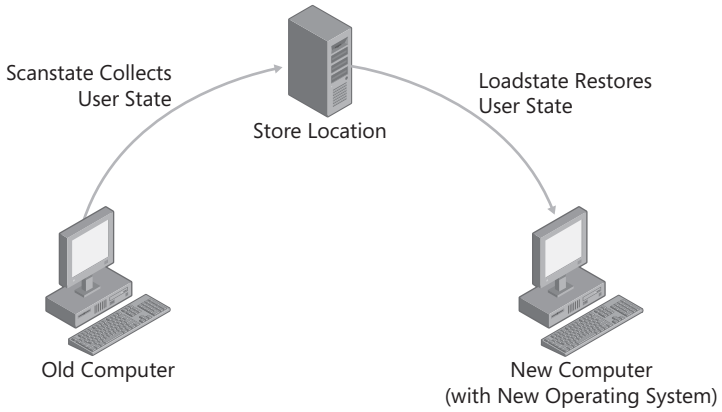


FIGURE 3-4 Side-by-side upgrades begin with a clean, new system.

Understanding Setup

To automate Windows Setup, you must first understand the installation process. Knowing the underlying process will help you understand the decisions you must make when developing Windows 7 for deployment.

The Windows 7 installation process is simple. All editions of Windows 7 use the same installation image (Install.wim in the Sources folder of the installation media), but Microsoft is shipping edition-specific media. As a result, you can install only one edition of Windows 7 through the user interface, but you can use an Unattend.xml file to install a different edition. The installation process is divided into three phases: Windows PE, Online Configuration, and Windows Welcome.

Windows Setup runs in *phases*, which the following sections describe. These phases—Pre-installation Phase, Online Configuration Phase, and Windows Welcome Phase—occur in order and simply designate a point in the installation process. Windows Setup also has configuration *passes*. Each configuration pass performs a specific function and applies related settings from the Unattend.xml answer file.

ON THE COMPANION MEDIA **The *Windows Automated Installation Kit User's Guide* (Waik.chm), which is in the Windows AIK 2.0, fully describes the command-line options for running Windows Setup (Setup.exe).**

You can customize the setup process at many phases through the use of answer files. The following list describes the answer files you use to customize the Windows 7 installation experience:

- **Unattend.xml** The generic name given to an answer file that controls most unattended installation actions and settings for most phases. When named Autounattend.xml and placed in the appropriate folder, such as the root of a UFD, this file can fully automate installations from the original Windows 7 media.
- **Oobe.xml** Oobe.xml is a content file you use to customize the out-of-box experience: Windows Welcome and Welcome Center.

Preinstallation Phase

During the Preinstallation phase, Windows Setup loads and prepares the target system for installation. Figure 3-5 illustrates where this phase fits in the installation process.

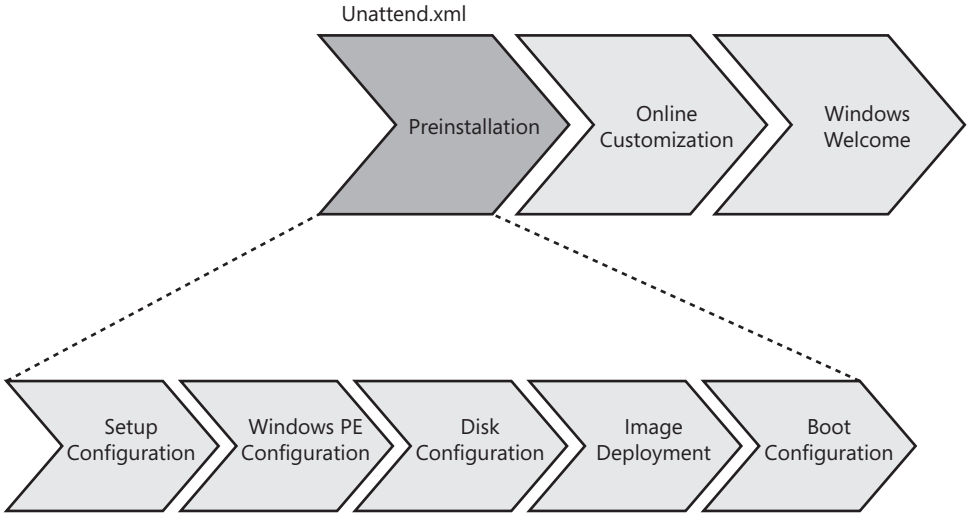


FIGURE 3-5 Preinstallation phase

Tasks performed during the Preinstallation phase include:

- **Windows Setup configuration** Windows Setup is configured by using either the Windows Setup dialog boxes (interactive) or an answer file (unattended). Windows Setup configurations include configuring a disk or language settings.

- **Windows PE configuration** Answer file settings are applied during the Windows PE configuration pass.
- **Disk configuration** The hard disk is prepared for image deployment. This might include partitioning and formatting the disk.
- **Windows image file copy** The Windows 7 image is copied to the disk from the distribution media or a network share. By default, the image is contained in Sources \Install.wim on the product media or distribution share.
- **Prepare boot information** The Windows 7 boot configuration is finalized. This includes configuring single- or multiboot configuration settings.
- **Process answer file settings in the offlineServicing configuration pass** Updates and packages are applied to the Windows 7 image, including software fixes, language packs, and other security updates.

MORE INFO **Windows Setup produces numerous log files that are useful for troubleshooting installation. For more information about these log files, see "Windows Setup Log File Locations," at *http://support.microsoft.com/default.aspx/kb/927521*.**

Online Configuration Phase

During the Online Configuration phase, Windows 7 performs customization tasks related to the computer's identity. Figure 3-6 shows where this phase fits into the overall process.

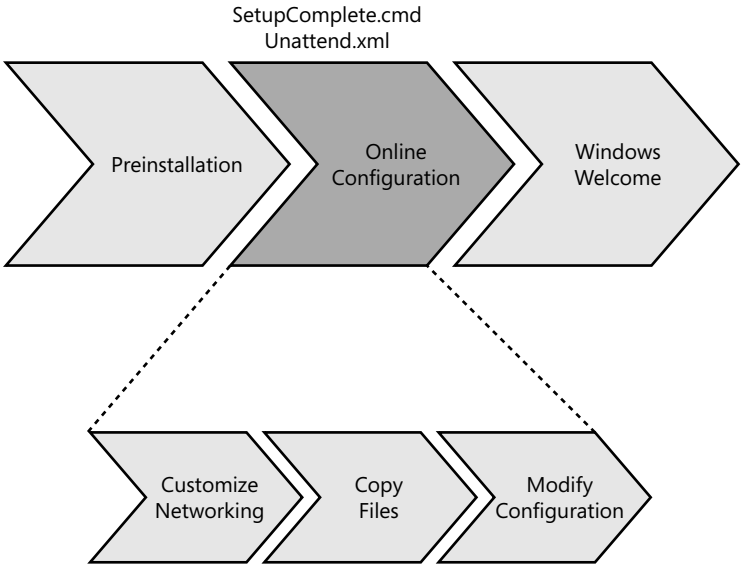


FIGURE 3-6 Online Configuration phase

The Specialize pass, which runs during this phase, creates and applies computer-specific information. For example, you can use an unattended setup answer file (Unattend.xml) to configure network settings, international settings, and domain information, as well as run installation programs.

During the Online Configuration phase, you can use scripts to configure the destination computer. However, a task sequencer, which enables you to filter tasks based on conditions, such as whether a particular device is installed, is better suited to this purpose. A task sequencer also provides advanced features such as the ability to wait until a certain condition arises before continuing, and grouping tasks into folders and then filtering the entire group.

ON THE COMPANION MEDIA **The companion media includes a script-based task sequencer, Taskseq.wsf, that provides all of these advanced features, among others. It reads tasks sequences from .xml files and then executes them. The file Sample_Task_Sequences.zip includes sample task sequences that demonstrate how to build .xml files for Taskseq.wsf. Do not run these sample task sequences on production computers. Read the documentation included in the source code for more information about using Taskseq.wsf.**

Windows Welcome Phase

In the Windows Welcome phase, shown in Figure 3-7, the installation is finalized, and any first-use customizations you want to apply are presented. Additionally, Windows 7 prompts for the product key during this phase. You can customize the Windows Welcome screens and messages and store these customizations in an Oobe.xml file.

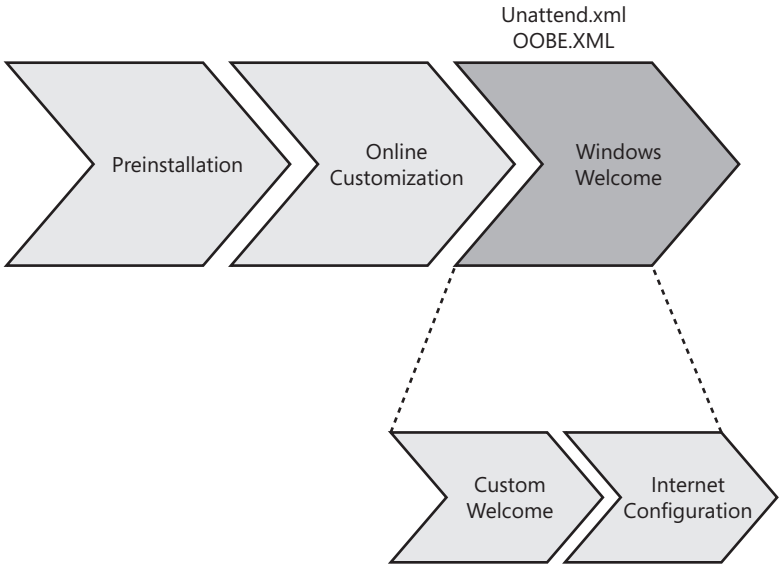


FIGURE 3-7 Windows Welcome phase

DIRECT FROM THE SOURCE

Text-Mode Setup Is Gone

Michael Niehaus, Systems Design Engineer
Microsoft Deployment Toolkit

The basic process used to install Windows XP has been unchanged since the earliest days of Microsoft Windows NT. This time-consuming procedure involved an initial text-mode installation step in which every operating system file was decompressed and installed, all registry entries were created, and all security was applied. Beginning with Windows Vista, this text-mode installation phase is completely gone. Instead, a new setup program performs the installation, applying a Windows image to a computer.

After this image is applied, it needs to be customized for the computer. This customization takes the place of what was called mini-setup in Windows XP and Microsoft Windows 2000. The purpose is the same: the operating system picks the necessary settings and configuration for the specific computer to which it was deployed.

The image preparation process has also changed. With Windows XP, you would Sysprep a computer to prepare the reference operating system for deployment. Beginning with Windows Vista, you'll still run Sysprep.exe, but it's installed by default in C:\Windows\System32\Sysprep.

Beginning with Windows Vista, the Windows operating system is provided on the DVD as an already-installed, generalized (Sysprepped) image, ready to deploy to any computer. Some customers may choose to deploy this image as is (possibly injecting fixes or drivers using the servicing capabilities provided by the deployment tools).

Basic Deployment Process

Figure 3-8 illustrates the basic deployment process using only the Windows 7 deployment tools to build images for high-volume deployments. Although this is useful background information, direct use of these tools isn't recommended. Using a framework like MDT 2010 is the best way to deploy Windows 7.

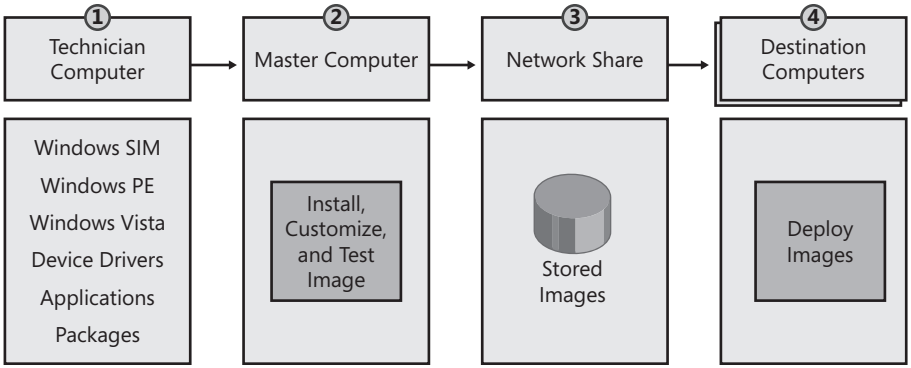


FIGURE 3-8 Basic deployment process

The following list describes the steps in Figure 3-8:

- **Technician Computer** You build a distribution share on a technician computer. The distribution share includes the Windows 7 source files, applications, device drivers, and packages. You use Windows SIM to configure the distribution share by adding source files to it. You also use Windows SIM to create and customize the Windows 7 answer file to use for installation.
- **Master Computer** On a master computer, you create a master installation by running Windows Setup from the distribution share, using an answer file you created with Windows SIM. The installation should be automated fully to ensure a consistent, repeatable process from one build to the next. After creating the master installation, run Sysprep to prepare it for duplication. In low-volume deployments, you can skip this step and deploy to desktop computers directly from the volume license or retail Windows 7 media that Microsoft provides and then customize the installation during deployment.
- **Network Share** You use ImageX to capture an image of the master installation from the master computer. Then you store the image on a network share accessible to the destination computers to which you're deploying the image. Alternatives to deploying from a network share include deploying the image from a DVD, a UFD, or Windows Deployment Services.
- **Destination Computers** On the destination computers, run Windows Setup to install Windows 7. Windows Setup accepts the image file and answer file to use as command-line options. Using Windows Setup to apply an image to destination computers is preferable to using ImageX to apply the image. Windows Setup includes logic that ImageX does not include, such as properly preparing the BCD.

HOW IT WORKS

Configuration Passes

Windows Setup uses configuration passes to configure systems. The following list describes each configuration pass that Windows Setup runs:

- **windowsPE** Configures Windows PE options as well as basic Windows Setup options. These options can include configuring a disk or language settings.
- **offlineServicing** Applies updates to a Windows 7 image. Also applies packages, including software fixes, language packs, and other security updates.
- **generalize** The generalize pass runs only if you run sysprep /generalize. In this pass, you can minimally configure Windows 7 as well as configure other settings that must persist on your master image. The sysprep /generalize command removes system-specific information. For example, the unique SID and other hardware-specific settings are removed from the image.
- **specialize** Creates and applies system-specific information. For example, you can configure network settings, international settings, and domain information.
- **auditSystem** Processes unattended Setup settings while Windows 7 is running in system context, before a user logs on to the computer in audit mode. The auditSystem pass runs only if you boot in audit mode.
- **auditUser** Processes unattended Setup settings after a user logs on to the computer in audit mode. The auditUser pass runs only if you boot in audit mode.
- **oobeSystem** Applies settings to Windows 7 before Windows Welcome starts.

Microsoft Deployment Toolkit Process

Microsoft Deployment Toolkit 2010 (MDT 2010) is a holistic approach to desktop deployment, bringing together the people, processes, and technology required to perform highly successful, repeatable, and consistent deployment projects. Because of its strong focus on methodology and best practices, MDT 2010 is much more valuable than the sum of its parts. Not only does it have the benefit of decreasing the time required to develop a desktop-deployment project, but it also reduces errors and helps you create a higher-quality desktop-deployment project.

Microsoft recommends that you use MDT 2010 to deploy Windows 7 instead of using the basic deployment tools directly. All the deployment tools in Windows 7 and the Windows AIK 2.0 are huge improvements over the deployment tools for earlier versions of Windows. However, they are simply tools without a framework, without any business logic. They have no "glue" to bind them into an end-to-end process. MDT 2010 provides this glue in the form of a complete technology framework. Internally, MDT 2010 is an extremely complex solution. It provides solutions for the problems facing most customers during deployment, including pre-

installation phases (disk partitioning, formatting, and so on), installation (disk imaging), and postinstallation phases (user state migration, application installation, customization, and so on). Even though MDT 2010 is complex internally, the solution makes building, customizing, and deploying Windows 7 images easier by masking the details.

DIRECT FROM THE SOURCE

Microsoft Deployment Toolkit

Manu Namboodiri
Windows Product Management

Microsoft has invested a lot to provide innovative technologies that help customers deploy desktops effectively, especially the new capabilities around file-based imaging, feature-based architectures, hardware independence, and so on. These have significant benefits in reducing image count, costs, and complexity.

However, where we have heard a lot of feedback from our customers and partners is regarding the best practices and methodology to use these tools most effectively. We also hear from industry analysts that most of the migration challenges that customers face center around building teams, schedules, project plans, business cases, and the right set of images as well as process and methodology. Technology, in itself, plays a smaller role than we would think in successful deployments.

The challenges our customers face are the following:

- **No standard set of deployment guidelines, which results in widely varying results and costs for desktop deployments**
- **More focus on technology and less on methodology, which has caused varying types of solutions and, therefore, varying results**
- **Customer perception of cost/complexity because of the lack of repeatable and consistent processes around the technology**
- **Unclear guidance about which of our many new tools to use and when**

Discovering these concerns has made us extremely focused on enhancing our guidance around deployments. The result is the significantly improved MDT methodology for desktop deployment. We are working with industry experts, system integrators, and deployment/management software providers to enhance this guidance so that it captures best practices from throughout the industry.

Figure 3-9 describes the typical process for using MDT 2010 to deploy Windows 7. The process is the same whether you're capturing an image in the lab or deploying images in a production environment. Additionally, MDT 2010 provides a user interface to configure all of its processes. Behind the scenes, thousands of lines of code work to implement your choices during deployment.

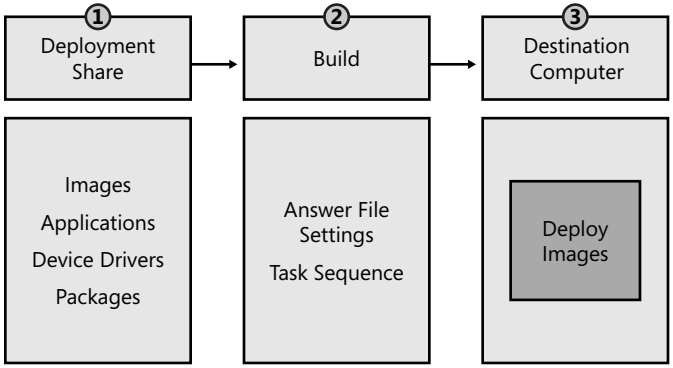
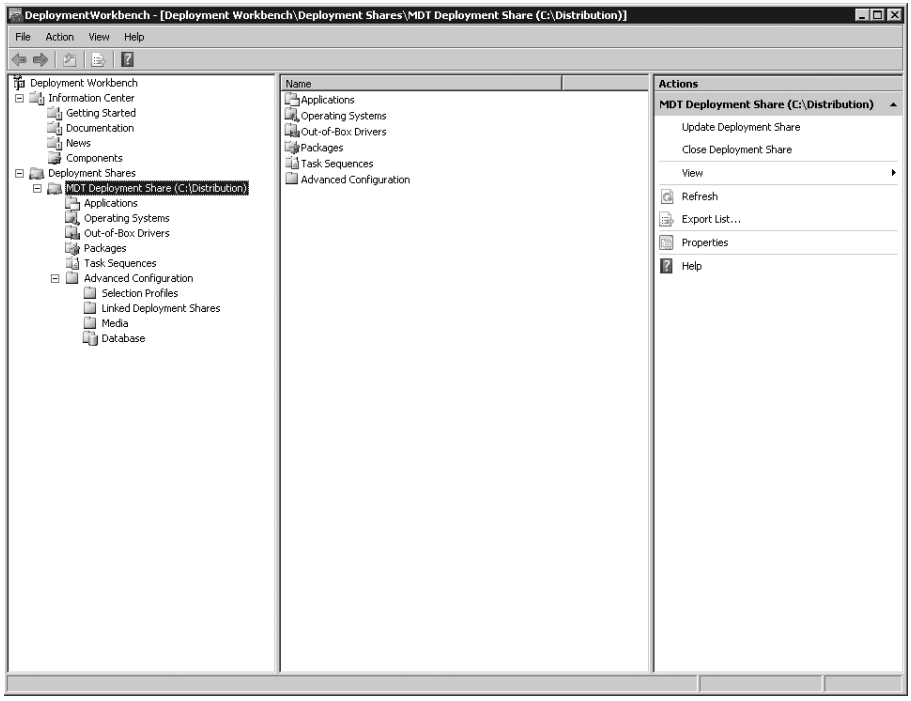


FIGURE 3-9 Microsoft Deployment Toolkit process

The following list describes each part of the MDT 2010 process. (See Chapter 6 and Chapter 12, "Deploying with Microsoft Deployment Toolkit," for more information.)

- **Deployment share** After installing MDT 2010 on a build server in a lab environment, you use the Deployment Workbench to stock the deployment share with source files. Source files include Windows 7 images, applications, device drivers, and packages. The Deployment Workbench provides a user interface for adding all source files to the deployment share. The user interface also provides intelligence, such as error checking and building a device driver database for device driver injection during deployment.



- **Task sequence** After the deployment share is fully stocked, you use the Deployment Workbench to create a task sequence. A task sequence associates source files from the deployment share with a list of steps to take during installation. The task sequence specifies when to take each step and when to skip it (filtering). The task sequence supports reboots during installation, and data collected during the task sequence persists between reboots. The task sequence represents one of the primary customization points for MDT 2010.
- **Destination computer** With a fully stocked deployment share and a task sequence, you can use MDT 2010 to deploy Windows 7 to destination computers. You can use LTI to deploy Windows 7. To use LTI, you start the destination computer using the deployment share's Windows PE boot image. You can put the boot image on removable media (DVD, UFD, and so on) or add it to a Windows Deployment Services server. Either way, you start the destination computer using the Windows PE boot image provided by the deployment share to begin the Windows Deployment Wizard. The wizard displays several pages to collect data from you (computer name, domain membership, applications to install, and so on), and then installs the operating system without any further interaction.

You can also use ZTI to deploy Windows 7. MDT 2010 integrates directly in System Center Configuration Manager 2007. For more information about using ZTI, see the MDT 2010 documentation.

Note that Figure 3-9 makes no reference to creating a master installation and capturing an image. In MDT 2010, creating and capturing an image is an LTI process. You can configure any deployment share to capture an image of an installation and store the image in the deployment share automatically. After you make this choice, the imaging process is fully automated. You don't have to run Sysprep or ImageX—the Windows Deployment Wizard automatically runs Sysprep and then runs ImageX to capture the image and store it in the deployment share. Then you can simply add the image to the deployment share using Deployment Workbench.

NOTE **You can download MDT 2010 from *http://technet.microsoft.com/en-us/desktopdeployment/default.aspx*.**

Summary

The Windows 7 deployment platform and tools will make deploying the operating system in your organization easier than deploying earlier versions of Windows. The .wim file format makes it possible to deploy highly compressed image files. Windows 7 helps reduce image count by removing hardware dependencies from the image. Modularization in Windows 7 makes servicing images easier than legacy methods so that you no longer have to apply, customize, and recapture an image to update it. The answer file format, Unattend.xml, provides a

more flexible and consistent configuration. Finally, the deployment tools, DISM, ImageX, and Windows SIM, provide a robust way to create, customize, and manage Windows 7 images.

Although the Windows AIK 2.0 provides the basic tools for customizing and deploying Windows 7, MDT 2010 provides a more flexible framework for deploying Windows 7 in organizations. With MDT 2010, you can create and customize multiple image builds. The framework includes automation common to most organizations and is highly extensible to suit any requirements.

Additional Resources

These resources contain additional information and tools related to this chapter.

Related Information

- *Windows Automated Installation Kit User's Guide* in the Windows AIK 2.0 includes detailed information about each of the tools described in this chapter.
- Chapter 6, "Developing Disk Images," includes more information about using MDT 2010 to create deployment shares, create builds, and capture images.
- Chapter 9, "Preparing Windows PE," includes more information about customizing Windows PE for Windows 7 deployment.
- Chapter 10, "Configuring Windows Deployment Services," includes more information about installing, configuring, and using Windows Deployment Services to deploy Windows 7.
- Chapter 12, "Deploying with Microsoft Deployment Toolkit," includes more information about using the Microsoft Deployment Toolkit to deploy Windows 7 images.
- *http://technet.microsoft.com/en-us/desktopdeployment/default.aspx* contains the latest information about using the Microsoft Deployment Toolkit to deploy Windows 7.
- Deployment Forum at *http://www.deploymentforum.com/* is a member-driven community for IT professionals deploying Windows 7.
- Download the Windows Automated Installation Kit 2.0 (Windows AIK 2.0) and the Microsoft Deployment Toolkit 2010 (MDT 2010) from the Microsoft Download Center at *http://www.microsoft.com/downloads*.

On the Companion Media

- Taskseq.wsf
- Sample_Task_Sequences.zip

CHAPTER 4

Planning Deployment

This chapter helps you plan the deployment of the Windows 7 operating system in your organization. Deploying an operating system requires careful planning. Application compatibility, user state migration, automation, and other issues complicate the process—making deployment more than just installing a new operating system on a handful of desktop computers. This chapter helps you use the best planning tools available and discover issues that require planning so that you can make informed decisions early in the process.

Using the Microsoft Deployment Toolkit

Microsoft Development Toolkit 2010 (MDT 2010) is Microsoft's best solution for high-volume Windows 7 deployment projects. It reduces complexity and increases standardization by allowing you to deploy a hardware and software baseline to all users and computers. With standard baselines, you can manage the computing environment more easily, spend less time managing and deploying computers, and spend more time on mission-critical tasks.

MDT 2010 provides automation tools and guidance that help reduce labor and increase reliability by producing standardized configurations. It provides fully developed processes for you to do the following:

- Determine which applications can be redeployed on new systems and start a process for packaging or scripting those applications so that you can reinstall them quickly and consistently without user intervention.
- Create an imaging process to produce a standard enterprise image of Windows 7 to aid in configuration management and to speed deployments.
- Establish a process for capturing user state from existing computers and for restoring user state on the newly deployed computers.
- Provide a method for backing up the current computer before deploying Windows 7.
- Provide an end-to-end process for the actual deployment of the new computers. The guidance includes Lite Touch and Zero Touch Installations.

Although you can certainly undertake a high-volume deployment project without MDT 2010, that approach is discouraged. This is because without MDT 2010, you must develop your own development and deployment processes. You also must define your own best practices and develop your own automation. By using MDT 2010 as your deployment framework, you save potentially hundreds of hours that you would otherwise spend writing scripts, writing answer files, developing images, and so on. MDT 2010 handles most scenarios intrinsically, and you can easily extend MDT 2010 for additional scenarios. You can even use MDT 2010 with most third-party deployment technologies. This chapter assumes you'll be using MDT 2010.

MDT 2010 has two major components: the documentation and the solution framework. The following sections describe these components in more detail. Earlier versions of MDT provided detailed planning guidance and job aids. However, due to the overwhelming size of the documentation in MDT, Microsoft has reduced the documentation in MDT to essential technical guidance only. Additionally, MDT now includes quick-start guides that provide end-to-end instructions for Lite Touch Installation (LTI) and Zero Touch Installation (ZTI) deployment. The section titled "Planning High-Volume Deployment" later in this chapter describes how to plan high-volume deployment projects in lieu of the MDT planning documentation.

NOTE **Install MDT 2010 to view its documentation as compiled help (.chm) files. After installing MDT 2010, click Start, point to All Programs, select Microsoft Deployment Toolkit, and then click Microsoft Deployment Help. To learn how to install MDT 2010, see the section titled "Installing the Microsoft Deployment Toolkit," later in this chapter. You can download printer-ready documentation from the Microsoft Download Center at *http://www.microsoft.com/downloads*.**

Documentation

MDT 2010 includes three types of documentation. The technical guides provide detailed information about specific technical areas, such as application packaging or image engineering. The reference guides contain content formatted as lists and tables so readers can find information quickly and easily. For example, MDT 2010 provides a reference for the properties that it supports. Finally, the quick-start guides provide end-to-end instructions for scenarios such as LTI and ZTI deployment. The following sections describe these guides.

Technical Guides

The following list describes the technical guides in MDT 2010:

- **Application Packaging Guide** Provides guidance for repackaging applications.
- **Deployment Customization Guide** Describes how to customize LTI and ZTI deployments.
- **Microsoft Deployment Toolkit 2010 Samples Guide** Identifies deployment scenarios and corresponding configuration settings when deploying target computers using LTI and ZTI deployment. You can use the sample configuration files in this guide as a starting point for your own project.
- **Microsoft Deployment Toolkit 2010 Management Pack** Describes the installation and configuration of the management pack. The MDT 2010 Management Pack can provide detailed information about the MDT 2010 deployment process to IT professionals involved in the deployment and operations processes.
- **Image Customization Guide** Describes how to customize reference images by setting the task sequence, developing custom scripts, revising existing MDT 2010 scripts, and so on. It includes information about customizing actions, such as disk, network, and role configuration.
- **Preparing for LTI Tools** Describes how to create a default installation of MDT 2010 for LTI deployment.
- **Preparing for Microsoft Systems Center Configuration Manager 2007** Describes how to create a default installation of MDT 2010 for ZTI deployment by using Microsoft Systems Center Configuration Manager.
- **Microsoft System Center Configuration Manager 2007 Imaging Guide** Describes how to use the Configuration Manager to prepare for image creation and deployment.
- **User State Migration Guide** Describes key concepts and decisions regarding the use of the User State Migration Tool (USMT) to migrate user state data from the previous configuration to the new configuration.
- **Workbench Imaging Guide** Describes how to use Deployment Workbench to prepare for image creation and deployment.

Reference Guides

The following list describes the reference guides in MDT 2010:

- **Toolkit Reference** Describes all customizable task sequence steps; properties that you can configure, use in scripts, or use in the Task Sequencer; each script contained in the task sequence; and customization points.
- **Troubleshooting Reference** Describes common error codes and failures. Where available, it provides resolutions for certain issues.

Quick Start Guides

The following list describes the quick-start guides in MDT 2010:

- **Quick-Start Guide for Lite Touch Installation** Helps you evaluate MDT 2010 quickly by providing condensed, step-by-step instructions for using MDT 2010 to install Windows operating systems by using LTI
- **Quick-Start Guide for Microsoft Systems Center Configuration Manager 2007** Helps you evaluate MDT 2010 quickly by providing condensed, step-by-step instructions for using MDT 2010 to install Windows operating systems by using Configuration Manager

Solution Framework

You use the solution framework (technology files) to set up the imaging and deployment servers. This framework helps you create standard desktop configurations. It includes tools to build and deploy custom Windows 7 images with a variety of special needs, such as backing up the destination computer prior to deployment, capturing and restoring user state, enabling Windows BitLocker Drive Encryption, and so on. By using the solution framework as your starting point, you can take advantage of the deployment best practices that Microsoft and its customers have developed over several years, most of which are manifested in the framework's script code.

> ***NOTE*** **The solution framework does not contain copies of Windows 7 or the 2007 Microsoft Office system. To use MDT 2010, you must acquire licensed copies of this software and other hardware-specific software such as DVD-player software and CD-creation software. Each technical guide in MDT 2010 describes requirements for using the guidance as well as the tools.**

Planning High-Volume Deployment

MDT 2008 and earlier versions included detailed planning guidance and job aids that helped you set up project teams, synchronize their work, and manage milestones. However, Microsoft condensed the documentation in MDT 2008 Update 1 and MDT 2010 to eliminate much of the

planning guidance. This move helped reduce an overwhelming amount of documentation, making it easier for people who just want technical guidance to use it.

Microsoft still provides excellent planning and management guidance for high-volume deployment projects, however. In fact, it's better: The Microsoft Operations Framework (MOF) 4.0, available at *http://technet.microsoft.com/en-us/library/cc506049.aspx*, provides project management guidance and job aids based on the original *Planning Guide* in MDT 2008. In MOF 4.0, the Deliver phase uses familiar terminology, such as *envisioning, planning, building, stabilizing,* and *deploying.* [These were phases in previous MDT documentation but are Service Management Functions (SMFs) in MOF 4.0 guidance.] This guidance maps out a workflow, including inputs, responsibilities, activities, deliverables, and reviews for each step. Figure 4-1 is an example of the types of workflow that MOF 4.0 provides for high-volume deployment.

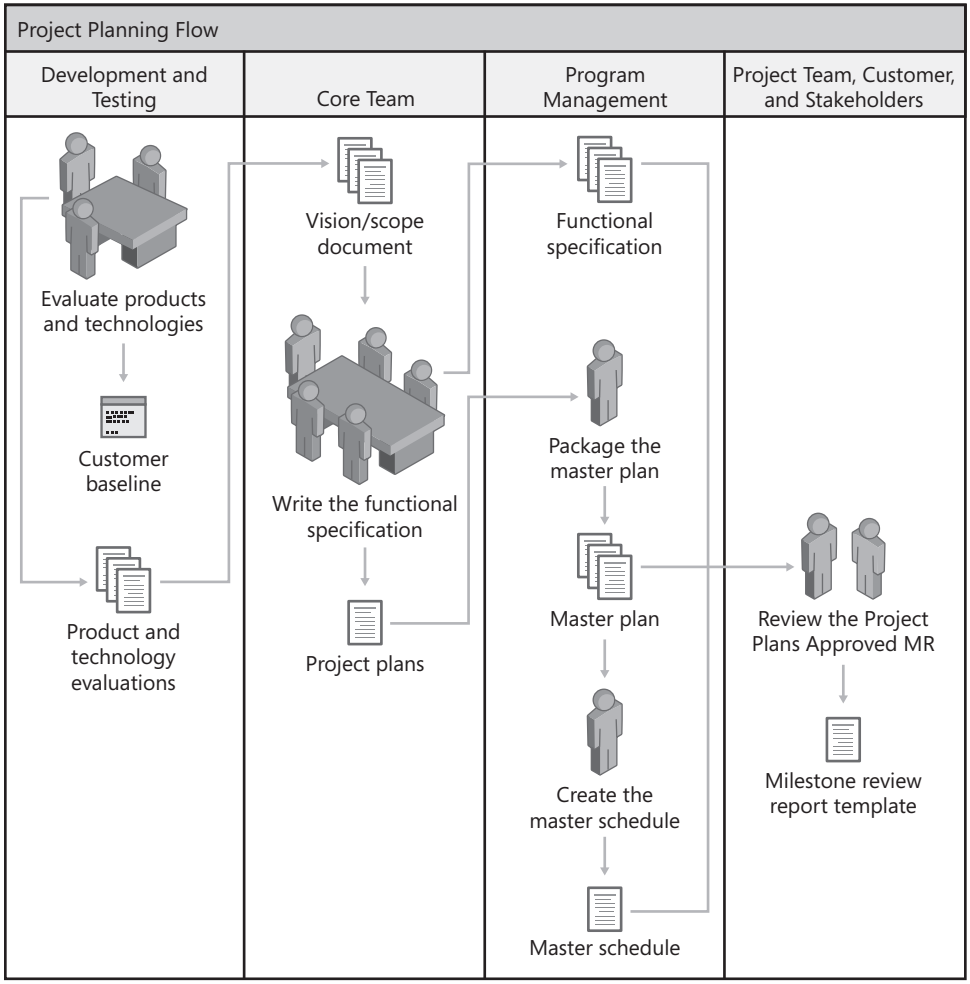


FIGURE 4-1 MOF 4.0 Project Planning SMF

This guidance helps you to do the following:

- Capture the business needs and requirements prior to planning a solution
- Prepare a functional specification and solution design
- Develop work plans, cost estimates, and schedules for the deliverables
- Build the solution to the customer's specification so that all features are complete and the solution is ready for external testing and stabilization
- Release the highest-quality solution by performing thorough testing and release-candidate piloting
- Deploy a stable solution to the production environment and stabilize the solution in production
- Prepare the operations and support teams to manage and provide customer service for the solution

NOTE **MDT 2010 no longer includes job aids for writing vision documents, functional specifications, and so on. MOF 4.0 now includes these job aids. You can download the job aids from the Microsoft Download Center at *http://go.microsoft.com/fwlink/?LinkId=116390*.**

The following sections describe each MOF 4.0 Deliver SMF. Because MOF 4.0 is generic, they relate each SMF specifically to performing a high-volume deployment by using MDT 2010.

Envision

The Envision SMF involves envisioning the deployment project and determining goals and expected outcomes. The Envision SMF is largely a management exercise; you don't assemble full project teams until this phase is complete. The Envision SMF includes the following key steps:

- **Set up core teams** The initial task is to define the teams that will plan, design, and perform the deployment.
- **Perform a current assessment** This step includes identifying existing systems and applications, determining existing operating systems, and identifying deficiencies in the current environment that the Windows 7 deployment will address.
- **Define business goals** Concrete, quantifiable business goals should drive your need for the deployment. Rather than simply planning to deploy the latest technology for technology's sake, identify key deficiencies in the existing system that Windows 7 will address, as well as process and productivity gains that the deployment will make possible.
- **Create a vision statement and define the scope** Create a vision statement that defines how planned technology changes (including the Windows 7 deployment) will meet the defined business goals. The scope determines the extent of the vision that can be accomplished through the deployment of Windows 7.

- **Create user profiles** Develop an accurate and complete picture of users' functions, needs, and wants. Refine these into user profiles that accurately identify the types of users in the organization. Understanding users and what they need is the first step in determining how to structure the deployment to benefit the most users.
- **Develop a solution concept** Create a high-level document to define how the team will meet the requirements of the project.
- **Create risk-assessment documents** In this step, evaluate the overall deployment with the intent to anticipate, address, mitigate, and prevent risks associated with the deployment. Documentation of risk assessment is an ongoing task throughout the project.
- **Write a project structure** This document describes how the team manages and supports the project and describes the administrative structure for the project team. This document should define standards that the team will use, including methods of communication, documentation standards, and change-control standards.
- **Approve milestones** When you complete the initial planning and documentation, identify and schedule key milestones for the deployment.

NOTE **MOF 4.0 provides job aids to help complete many of these envisioning steps.**

Project Planning

The Envision SMF creates the framework for the Windows 7 deployment. The Project Planning SMF serves as a transition between vision and implementation, laying the groundwork for the actual deployment. The Project Planning SMF uses the documents and processes created in the Envision SMF to add structure and content to the deployment plan. Key steps in this phase include the following tasks:

- **Create the development and testing environment** Build a testing lab that adequately embodies the target deployment environment, using virtualization to reduce the cost of creating labs. In addition to resources such as servers and sample target systems used to develop and test the deployment, the labs should also include the resources that the project team will use to prepare and accomplish the final deployment.
- **Develop the solution design** The solution design builds on the solution concept, project structure, and other documents created by the Envision SMF to define the conceptual, logical, and physical solution designs for the planned deployment. This document serves as a road map for the project team to begin building the deployment.
- **Create the functional specification** The functional specification defines the requirements of all stakeholders targeted by the deployment and serves as a contract between the customer and the project team. It should clearly define the goals, scope, and outcomes of the deployment.

- **Develop the project plan** The project plan is actually a collection of plans that address the tasks the project team will perform to carry out the project, as defined by the functional specification. Each plan in this document covers a particular area, such as facilities and hardware, testing, training, and communication.
- **Create the project schedule** The project schedule compiles individual schedules created by team members for the purpose of planning deployment activities.
- **Complete a computer inventory** In the Project Planning SMF, a complete computer inventory must be made to identify existing systems and applications that the deployment will affect. In addition, the server resources to be used for deployment must also be identified and evaluated for suitability.
- **Perform a network analysis** Diagram network topology and identify and inventory network devices.

NOTE **MOF 4.0 includes job aids for many of these planning tasks.**

Build

The Build SMF is the period during which the team builds and unit-tests the solution. The Build SMF includes six key tasks:

- **Start the development cycle** In this initial step, the team creates a lab server for development work and begins the process of creating images, installation scripts, and application packages. The team should also create an issue-tracking system so that team members can communicate about and coordinate solutions to issues.
- **Prepare the computing environment** In this key task, the teams build a deployment environment with facilities such as servers, networking, system backup, and data repositories (such as Microsoft Visual SourceSafe) with separate workspaces (that is, computers and network shares) for each team role. This environment provides the infrastructure for teams to work both independently and jointly as necessary to complete their development tasks.
- **Develop the solution scripts** In this step, the teams begin the process of packaging applications, creating computer images, and developing remediation steps for application-compatibility issues. The teams also plan how and what user data will be retained and migrated during the deployment and validate that network infrastructure (that is, shares, credentials, and other components) is in place and functioning properly prior to deployment.
- **Develop deployment procedures** Using the documents, processes, and other resources created up to this point, begin creating the documents that the teams will use to accomplish the deployment and post-deployment tasks. These documents include training materials for users, administrators, and others who will maintain systems and applications after deployment; a plan for communicating with users about the upcoming

changes; and site-deployment procedures to simplify and standardize the deployment of solutions across sites.

- **Develop operations procedures** Create a document that describes the operations procedures to support, maintain, and carry out the solution following deployment. Key processes to describe include maintenance, disaster recovery, new-site installation, performance and fault monitoring, and support and troubleshooting.
- **Test the solution** Perform test deployments and remedy any issues that arise using the issue-tracking framework created by the Project Planning SMF to monitor and address these issues.

Stabilize

The Stabilize SMF addresses the testing of a solution that is feature-complete. This phase usually occurs when pilots are conducted, with an emphasis on real-world testing and with the goal of identifying, prioritizing, and fixing bugs. Key tasks in this phase include the following:

- **Conducting the pilot** At this stage, the teams use a small pilot deployment to test the deployment and identify any remaining issues. Procedures, resources, and personnel should be in place to assist in addressing any user issues that arise during the pilot deployment. This key task should also include obtaining user feedback as well as review and remediation of issues identified during the pilot.
- **Operational-readiness review** All teams at this stage perform a complete operational-readiness review to determine that the deployment plan is ready to move forward to full-scale deployment. The solution is frozen at this stage, and any remaining issues are addressed.
- **Final release** This task incorporates all fixes and issue resolutions to create the final release of the solution, which should now be ready for full deployment.

Deploy

In the Deploy SMF, the team implements the solution and ensures that it is stable and usable. The key tasks involved in the Deploy SMF include the following:

- **Deploying core technology** Based on the plans and procedures developed in the Project Planning SMF, install, configure, and test deployment servers at each site. Also train administration staff in preparation for deployment.
- **Deploying sites** Teams perform the deployment of Windows 7 at each site using the procedures and resources developed by the Project Planning and Build SMFs. Team members remain on site to stabilize each site deployment, ensuring that users can move forward with reliable systems and applications and that the goals of the deployment plan for the site have been met.
- **Stabilizing the deployment** In this key step, the project team ensures stabilization across all sites and addresses any remaining deployment issues.

- **Completing the deployment** This step marks the transition from deployment to operations and support. Ongoing operations are transferred from the project team to permanent staff. Reporting systems are activated, and support processes are fully operational.

Planning Low-Volume Deployment

In low-volume deployment projects, such as in a small or medium-sized business, the planning guidance in MOF 4.0 can be overwhelming. Regardless, the MDT 2010 technology framework is well suited to low-volume deployment projects. In fact, a small business can prepare MDT 2010 to deploy Windows 7 in as little as a few hours, and a medium-sized business can accomplish it in a few days. This section describes some of the planning steps you should take in this scaled-down scenario. (Even though you can use the MDT 2010 technology framework without using the planning guidance available in MOF 4.0, you should still put some effort into planning your deployment, along the lines of what is outlined here.)

The first step in the deployment process is to assess your business needs so that you can define the project scope and objectives. Next, decide how best to use Windows 7 to meet those needs. Then assess your current network and desktop configurations, determine whether you need to upgrade your hardware or software, and choose the tools for your deployment. Having made these decisions, you are ready to plan your deployment. An effective plan typically includes the following:

- **A schedule for the deployment** Build a simple schedule by using Microsoft Office Excel 2007, or use a more formal tool like Microsoft Office Project 2007.
- **All the details for customizing Windows 7 to suit your requirements** Document the applications, device drivers, updates, and settings that you want to customize.
- **An assessment of your current configuration, including information about users, organizational structure, network infrastructure, and hardware and software** Create a test environment in which you can deploy Windows 7 by using the features and options in your plan. Have your test environment mirror your production network as closely as possible, including hardware, network architecture, and business applications.
- **Test and pilot plans** When you're satisfied with the results in your test environment, roll out your deployment to a specific group of users to test the results in a controlled production environment. This is your pilot test.
- **A rollout plan** Finally, roll out Windows 7 to your entire organization.

Creating the deployment plan is a cyclical process. As you move through each phase, modify the plan based on your experiences.

NOTE **Even if you choose not to use the deployment guidance in MOF 4.0, you can still use the job aids it includes, which provide templates for planning a deployment project more quickly and more thoroughly.**

Scope and Objectives

The scope is the baseline for creating a specification for your deployment project. The scope of your deployment project is defined largely by your answers to the following questions:

- What business needs do you want to address with Windows 7?
- What are the long-term goals for the deployment project?
- How will your Windows 7 client computers interact with your IT infrastructure?
- What parts of your IT infrastructure will the project touch, and how will this happen?

The scope is simply a statement of what you are trying to accomplish and how you plan to accomplish it. Your statement of scope need only be a few paragraphs long and should not be longer than a page.

Current Environment

Document your existing computing environment, looking at your organization's structure and how it supports users. Use this assessment to determine your readiness for desktop deployment of Windows 7. The three major areas of your computing environment to assess include your hardware, software, and network.

- **Hardware** Do your desktop and laptop computers meet the minimum hardware requirements for Windows 7? In addition to meeting these requirements, all hardware must be compatible with Windows 7. For more information, see Chapter 1, "Overview of Windows 7 Improvements."
- **Software** Are your applications compatible with Windows 7? Make sure that all of your applications, including line-of-business (LOB) applications, work with computers running Windows 7. For more information about application compatibility, see Chapter 8, "Deploying Applications."
- **Network** Document your network architecture, including topology, size, and traffic patterns. Also, determine which users need access to various applications and data, and describe how they obtain access.

NOTE **Where appropriate, create diagrams to include in your project plan. Diagrams convey more information than words alone. A good tool for creating these diagrams is Microsoft Office Visio 2007. See *http://www.microsoft.com/office* for more information.**

Configuration Plan

Determine which features to include in your configuration and how to implement these features to simplify the management of users and computers in your organization. An important means of simplification is standardization. Standardizing desktop configurations makes it easier to install, update, manage, support, and replace computers that run Windows 7. Standardizing users' configuration settings, software, hardware, and preferences simplifies deploying operating system and application upgrades, and configuration changes can be guaranteed to work on all computers.

When users install their own operating system upgrades, applications, device drivers, settings, preferences, and hardware devices, a simple problem can become complex. Establishing standards for desktop configurations prevents many problems and makes it easier for you to identify and resolve problems. Having a standard configuration that you can install on any computer minimizes downtime by ensuring that user settings, applications, drivers, and preferences are the same as before the problem occurred. The following list provides an overview of some of the features that you must plan to use:

- **Management** Desktop management features allow you to reduce the total cost of ownership in your organization by making it easier to install, configure, and manage clients. For more information about Windows 7 management features, see Part III of this book, "Desktop Management."
- **Networking** You can configure computers that run Windows 7 to participate in a variety of network environments. For more information about Windows 7 networking features, see Part V of this book, "Networking."
- **Security** Windows 7 includes features to help you secure your network and computers by controlling authentication and access to resources and by encrypting data stored on computers. These features include BitLocker Drive Encryption, Windows Firewall with Advanced Security, and so on. For more information about Windows 7 security features, see Chapter 2, "Security in Windows 7."

Testing and Piloting

Before rolling out your deployment project, you need to test it for functionality in a controlled environment. Before you begin testing your deployment project, create a test plan that describes the tests you will run, who will run each test, a schedule for performing tests, and the expected results. The test plan must specify the criteria and priority for each test. Prioritizing your tests can help you avoid slowing down your deployment because of minor failures that you can easily correct later; it can also help you identify larger problems that might require redesigning your plan.

The testing phase is essential because a single error can be replicated to all computers in your environment if it is not corrected before you deploy the image. Create a test lab that is not connected to your network but that mirrors your organization's network and hardware configurations as closely as possible. Set up your hardware, software, and network services as

they are in your production environment. Perform comprehensive testing on each hardware platform, testing both application installation and operation. These steps can greatly increase the confidence of the project teams and the business-decision makers, resulting in a higher-quality deployment.

Microsoft recommends that you pilot the project (that is, roll out the deployment) to a small group of users after you test the project. Piloting the installation allows you to assess the success of the deployment project in a production environment before rolling it out to all users. The primary purpose of pilot projects is not to test Windows 7, but to get user feedback. This feedback will help to determine the features that you must enable or disable in Windows 7. For pilots, you might choose a user population that represents a cross-section of your business in terms of job function and computer proficiency. Install pilot systems by using the same method that you plan to use for the final rollout.

The pilot process provides a small-scale test of the eventual full-scale rollout: You can use the results of the pilot, including any problems encountered, to create your final rollout plan. Compile the pilot results and use the data to estimate upgrade times, the number of concurrent upgrades you can sustain, and peak loads on the user-support functions.

Rolling Out

After you thoroughly test your deployment plan, pilot the deployment to smaller groups of users, and are satisfied with the results, begin rolling out Windows 7 to the rest of your organization. To create your final rollout plan, you need to determine the following:

- The number of computers to include in each phase of the rollout
- The time needed to upgrade or perform a clean installation for each computer that you include
- The personnel and other resources needed to complete the rollout
- The time frame during which you plan to roll out the installations to different groups
- The training needed for users throughout the organization

Throughout the rollout, gather feedback from users and modify the deployment plan as appropriate.

Windows 7 Requirements

To plan deployment, you must understand the deployment requirements for Windows 7. The following sections describe the minimum hardware requirements and the migration paths for Windows 7. For more information about Windows 7 hardware requirements and editions, see Chapter 1.

Hardware Requirements

Table 4-1 describes the minimum hardware requirements for installing Windows 7. Part of the Project Planning SMF is collecting a hardware inventory. Compare the hardware requirements in Table 4-1 to your hardware inventory to identify any computers that require upgrades or replacements.

TABLE 4-1 Minimum Hardware Requirements for Windows 7 Computers

HARDWARE	MINIMUM REQUIREMENT
Processor	1 GHz or faster 32-bit or 64-bit processor
Memory	1 GB for 32-bit computers, or 2 GB for 64-bit computers
Graphics Processor	DirectX 9 graphics processor with Windows Display Driver Model (WDDM) 1.0 or later driver
Free Hard Disk Drive Space	16 GB

NOTE **The minimum requirements for Windows 7 are the same across all editions.**

Upgrade Paths

Table 4-2 describes the Windows 7 upgrade and migration paths. As shown in the table, performing an in-place upgrade from Windows Vista with Service Pack 1 (SP1) or later to Windows 7 is supported. This means you can install Windows 7 on a computer running Windows Vista with SP1 and retain your applications, files, and settings. Using Windows Easy Transfer or the USMT to migrate user states from Windows XP to Windows 7 is also supported.

TABLE 4-2 Windows 7 Migration Paths

FROM	UPGRADE TO WINDOWS 7?	MIGRATE TO WINDOWS 7 USING WINDOWS EASY TRANSFER?	MIGRATE TO WINDOWS 7 USING THE USMT?
Windows XP with SP2 or later	No	Yes	Yes
Windows Vista with SP1 or later	Yes	Yes	Yes
Windows 7	Yes (higher SKU)	Yes	Yes

NOTE **To assess the readiness of client computers for Windows 7, you can use the Microsoft Assessment and Planning Solution Accelerator, a centralized and agentless tool that can remotely inventory computers, identify their supported Windows 7 experience, and recommend specific hardware upgrades where appropriate. For more information on this tool, see "Microsoft Assessment and Planning (MAP) Toolkit" found at *http://technet.microsoft.com/en-us/solutionaccelerators/dd537566.aspx*.**

Preparing for Development

Whether your organization uses MDT 2010 or not, it will likely require multiple teams to develop high-volume deployment projects. Most teams need a lab environment. Although each team can construct a separate lab, most organizations create a single lab that shares facilities such as servers, networks, system backup, and source control with separate workspaces (computers and network shares) for each team. Within this environment, teams can work separately when necessary and jointly when appropriate. It also helps minimize the number of computers and servers required.

The remaining chapters in Part II of this book, "Deployment," describe specific lab requirements for each team working on a high-volume deployment project. The Project Planning SMF is the best time to begin preparing the development environment, however. The process includes installing MDT 2010, stocking the lab environment with the files necessary to perform each team's job, locating application media, and so on. The following sections describe steps to complete during the Project Planning SMF to expedite the development process.

Application Management

Application management is the process of repackaging applications or automating their installation and configuration. Organizations can have hundreds or thousands of applications. Often, users install each application differently on each computer, leading to inconsistency across computers and resulting in support and management issues.

Repackaging or automating an application's installation has many benefits. Most obviously, it allows applications to install without user intervention, which is especially desirable when deploying applications as part of a disk image or during disk image deployment. In addition, repackaging or automating leads to consistency that lowers deployment and ownership costs by reducing support issues and enhancing management. Chapter 8 describes a process for repackaging and automating application installations.

Before migrating from your current version of Windows to Windows 7, the project team must also test applications to ensure that they are compatible with Windows 7. You might have several thousand applications installed across distributed networks. Compatibility problems with one or many of these applications can disrupt productivity and damage the user

experience and satisfaction with the project. Testing applications and solving compatibility problems saves time and money for the organization. It also prevents roadblocks to future deployment projects based on perceived pain.

Although most applications developed for earlier versions of Windows will probably perform well on Windows 7, some applications might behave differently because of new technologies in it. Test the following applications to ensure compatibility:

- Custom tools, such as logon scripts
- Core applications that are part of the standard desktop configurations, such as office productivity suites
- LOB applications, such as Enterprise Resource Planning (ERP) suites
- Administrative tools, such as antivirus, compression, backup, and remote-control applications

Chapter 5, "Testing Application Compatibility," describes how to build an application compatibility test lab and how application compatibility integrates into the overall deployment process. The following list describes steps that you can take in the Project Planning SMF to begin building the lab environment for application packaging and compatibility testing:

- **Installation media** For each application you test, repackage, and automate, you must have a copy of the application's installation media, any configuration documentation, and product keys. If your IT department doesn't have the media or other information, check with the subject matter expert (SME) for each application.
- **Destination computers** Within the lab, the project team requires destination computers that resemble computers found in the production environment. Each destination computer should have Windows 7 installed on it to test applications' compatibility with the operating system and application installation.
- **Application Compatibility Toolkit** For more information on the Application Compatibility Toolkit (ACT), see Chapter 5.
- **SQL Server** Install Microsoft SQL Server in the lab environment. The ACT stores the application inventory using SQL Server, which is available on volume-licensed media.
- **Host computer with network shares** You must have a computer on which to host the application installations. Shares on this computer hold the original installation sources and completed packages. You can install the ACT and SQL Server on the host computer.
- **Application packaging software** The project team needs software with which to repackage applications. Chapter 8 describes this software. The application packaging software will be installed on each team member's development computer.
- **Deployment mechanism** The project team requires a mechanism for deploying ACT and application packages. This can be through logon scripts, a local Web site, or another deployment mechanism.

Image Engineering

You're probably already familiar with disk imaging tools such as Symantec Ghost or ImageX (which was introduced by Windows Vista). Using imaging tools effectively is a significant challenge, however, and this challenge is the reason that Microsoft developed MDT 2010. With MDT 2010, you do not have to engineer the entire imaging process; the framework provides most of the code for you already. All you need to do is customize it to suit your organization's requirements. Using MDT 2010 to build Windows 7 images involves the following steps:

- **Create a build server** The build server is the host for MDT 2010 and its distribution share.
- **Configure a deployment share** The deployment share contains the source files (Windows 7, applications, device drivers, and so on) from which you build operating system images.
- **Create and customize task sequences** After stocking the deployment share, you create task sequences. Task sequences associate source files from the deployment share with the steps necessary to install and configure them. For more information about answer files and task sequences, refer to Chapter 3, "Deployment Platform."
- **Build initial operating system images** With MDT 2010, building a custom Windows 7 image is as simple as installing the operating system from a deployment share by using the Windows Deployment Wizard. This is an LTI Installation process that requires minimal user interaction; it automatically captures a Windows 7 image and stores it back in the deployment share.

Chapter 6, "Developing Disk Images," describes how to use MDT 2010 to build custom Windows 7 images. In preparation for the development process, you can begin building the lab during the Project Planning SMF, using the items in the following list:

- **Windows 7 media** You will need media and volume license keys for Windows 7.
- **Destination computers** You will need computers on which to create, install, and test Windows 7 images.
- **A build computer for MDT 2010** You must have a computer on which to host MDT 2010 and the deployment share. The build computer should have a DVD-RW drive and should be networked with the destination computers. You can install MDT 2010 on a desktop or server computer.
- **Windows Deployment Services** The lab environment should contain a server running Windows Deployment Services. Using Windows Deployment Services to boot destination computers is much faster than burning DVDs and starting computers with them.
- **Additional source files** Early in the Project Planning SMF, the project team can begin assembling the source files required to stock the distribution share. Source files include device drivers and hardware-specific applications for each computer in the production environment. Additionally, the team should begin assembling any security updates and operating system packages that must be added to the distribution share.

NOTE **The Project Planning SMF is the best time to install MDT 2010 in the lab environment and begin familiarizing yourself with it. The section titled "Installing the Microsoft Deployment Toolkit" later in this chapter describes the requirements for installing MDT 2010 and how to install MDT 2010 in the lab environment.**

Deployment

Deployment is an intense, time-consuming process during any high-volume deployment. MDT 2010 provides technical guidance and tools that help streamline the following processes:

- Choosing server placement
- Evaluating server and network capacity
- Installing the distribution shares and tools
- Deploying the client computers

Chapter 12, "Deploying with Microsoft Deployment Toolkit," describes how to use MDT 2010 to deploy Windows 7 using the LTI process. During the Project Planning SMF, the project team should begin preparing the lab environment using the items in the following list:

- **Production replica** The project team needs a replica of the production environment to unit-test the combined efforts of all the other teams. Destination computers should be running the versions of Windows found in the production environment with user data loaded. These computers are used for the unit-test deployment, including user state migration.
- **Network shares on a host computer** Two types of network shares are required: one for the MDT 2010 deployment share and a second for the data server share. These shares could be all on the same physical server or on separate servers. Also, it's useful to store images of the production computers on the host computer to restore them quickly and easily after each test pass.
- **Windows Deployment Services** The lab environment should contain a server running Windows Deployment Services. Using Windows Deployment Services to boot destination computers is much faster than burning DVDs and starting computers with them. Team members can use the same Windows Deployment Services server for image engineering and deployment testing.

Infrastructure Remediation

Understanding the network environment is critical with any project that introduces changes. To plan and prepare to incorporate these changes, first understand the current status of the organization's environment, identify other sources of change that may affect this project, perform a risk-mitigation approach to the changes, and then incorporate the proposed changes. Organizations can solve and possibly avoid most networking problems by creating

and maintaining adequate network documentation. Using a networking tool, the team can do the following:

- Gather information necessary to help understand a network as it exists today.
- Plan for growth.
- Diagnose problems when they occur.
- Update the information with network changes.
- Work with the information to manage network assets. (Often, an apparently simple configuration change can result in an unexpected outage.)
- Present information visually so that the network structure appears in as much detail as necessary for each task.

The project team must have access to SQL Server. The team uses SQL Server to create hardware inventory reports against the application compatibility database. This could be the same installation that the team uses for application management. The team must also have access to current network topology diagrams and network device inventory information.

Operations Readiness

The project team is responsible for a smooth and successful handoff of the deployed solution to the operations staff. This aspect of the overall project is important, because the success of the handoff directly reflects the success of the deployment project. To ensure success, the activities of the team must be integrated with the ongoing management and operating functions of the operations staff. The project team can facilitate deployment by completing the following tasks:

- Confirm that the workstation roles identified in the functional specification are valid.
- Analyze and evaluate the management tools currently in use.
- Assess the maturity of the operations environment in key operational areas.
- Establish effective management processes and tools in deficient key areas.
- Develop a training program for operations and support staff.
- Prepare the operations staff for the pilot.

The project team does not initially have any additional lab requirements for operations readiness.

Security

Security is important to the overall success of the deployment project. Security is a primary concern in all organizations, and a goal of the project team is to secure the organization's data. Inadequate security in an organization can result in lost or corrupted data, network downtime, lost productivity, frustrated employees, overworked IT employees, and possibly stolen proprietary information that results in lost revenue. Additionally, many organizations

are subject to compliance regulations that carry significant penalties for security breaches, such as exposing customer data. To ensure that adequate security measures are in place, the project team should do the following:

- Analyze and determine the existing security level of the organization.
- Identify vulnerabilities caused by software upgrades and update network security specifications accordingly.
- Ensure that security measures are current.

The project team does not initially have any additional lab requirements for security. For more information about Windows 7 security features, see Chapter 2.

Migration

One of the most tedious and time-consuming tasks during deployment is identifying data files and settings on users' current computers (known as the *user state*), saving them, and then restoring them. Users spend significant time restoring items such as wallpaper, screen savers, and other customizable features. And most users don't remember how to restore these settings. Migrating user state can increase user productivity and satisfaction. The project team can perform the following steps to plan for user state migration:

- Inventory existing production client computer applications.
- Identify applications with data or preference migration requirements.
- Prioritize the application list to be addressed.
- Identify SMEs for each application.
- Identify data file requirements for each application.

The following list describes actions that you can take in the Project Planning SMF to begin building the lab environment for migration. Team members working on migration can share resources with team members working on application management.

- **Installation media** For each application containing settings to migrate, you must have a copy of the application's installation media, any configuration documentation, and product keys. If your IT department doesn't have the media or other information, check with the SME for each application.
- **Destination computers** The project team requires computers in the lab on which to test user state migration solutions. Destination computers should be running the versions of Windows found in the production environment with applications and user data loaded. These computers are used for the unit-test user state migration.
- **Host computer** You must have a computer on which to host application source files and migration solutions. It's useful to store images of the destination computers on the host computer to restore them quickly and easily after each test pass.

- **Data store** The data store is a network share on which you can put user state data during testing. You can create the data store on the host computer, or you can optionally create the data store on each destination computer.
- **USMT** MDT 2010 uses the USMT to migrate the user state. The functionality is already built into the MDT 2010 framework. The team must download and prepare the distribution share with the USMT executables, however. Chapter 7, "Migrating User State Data," describes where to place these files.

Installing the Microsoft Deployment Toolkit

MDT 2010 requires Windows PowerShell 2.0. If you're installing MDT 2010 on Windows Server 2008 or Windows Server 2008 R2, you must also add the Windows PowerShell feature by using the Add Features Wizard.

If you're installing MDT 2010 on Microsoft Windows Server 2003 SP1, you must install additional prerequisite software. Windows Server 2008, Windows Server 2008 R2, and Windows 7 already contain this software. The following list describes software that you must install before installing and using MDT 2010 on Windows Server 2003 SP1 (Windows Server 2003 SP2 requires only Microsoft .NET Framework 2.0):

- **Windows PowerShell** Download Windows PowerShell from the Microsoft Download Center at *http://www.microsoft.com/downloads*.
- **Microsoft .NET Framework 2.0** The Windows AIK distribution media includes the .NET Framework 2.0 installation file. Alternatively, download .NET Framework 2.0 from the following addresses:
  - **x86** *http://www.microsoft.com/downloads/details.aspx?FamilyID=0856eacb-4362-4b0d-8edd-aab15c5e04f5&DisplayLang=en*
  - **x64** *http://www.microsoft.com/downloads/details.aspx?FamilyID=b44a0000-acf8-4fa1-affb-40e78d788b00&DisplayLang=en*
- **Microsoft Management Console (MMC) 3.0** Download MMC 3.0 from the following addresses:
  - **x86** *http://www.microsoft.com/downloads/details.aspx?FamilyID=4c84f80b-908d-4b5d-8aa8-27b962566d9f&DisplayLang=en*
  - **x64** *http://www.microsoft.com/downloads/details.aspx?FamilyID=b65b9b17-5c6d-427c-90aa-7f814e48373b&DisplayLang=en*

NOTE **If you choose to install only the MDT 2010 documentation, the only software requirements are .NET Framework 2.0 and MMC 3.0. The remaining software in the previous list is not required to view the documentation.**

To install MDT 2010, perform the following steps:

1. Right-click MicrosoftDeploymentToolkit_*platform*.msi, where *platform* is either x86 or x64, and then click Install.
2. Click Next to skip the Welcome page.
3. On the End-User License Agreement page, review the license agreement, click I Accept The Terms In The License Agreement, and then click Next.
4. On the Custom Setup page, choose the features to install and then click Next. To change a feature's state, click the feature and then choose a state (Will Be Installed On Local Hard Drive and Entire Feature Will Be Unavailable). The following list describes each feature:
   - **Documents** This feature installs the solution's guidance and job aids. By default, this feature is installed in C:\Program Files\Microsoft Deployment Toolkit \Documentation.
   - **Tools and templates** This feature installs the solution's wizards and template deployment files, such as Unattend.xml. By default, this feature is installed in C:\Program Files\Microsoft Deployment Toolkit.
5. Click Install to install the solution.
6. Click Finish to complete the installation and close the installer.

NOTE **Versions of MDT earlier than 2008, including Microsoft Solution Accelerator for Business Desktop Deployment 2007, included a feature to create a distribution share. MDT 2010 does not create a distribution share during installation, however. Instead, you create a distribution share or upgrade an existing distribution share by using Deployment Workbench.**

The following list describes the subfolders in the MDT 2010 program folder (C:\Program Files \Microsoft Deployment Toolkit) after installing the solution:

- **Bin** Contains the MMC Deployment Workbench add-in and supporting files
- **Documentation** Contains the MDT 2010 documentation
- **Downloads** Provides storage for components that MDT 2010 downloads
- **ManagementPack** Contains the MDT 2010 management pack files
- **Samples** Contains sample task sequence scripts
- **SCCM** Contains files that support Microsoft System Center Configuration Manager (SCCM) 2007 integration
- **Templates** Contains template files that the Deployment Workbench uses

NOTE **The hard drive containing the program folders must have at least 1 gigabyte (GB) of free space available. MDT 2010 downloads components, including the Windows AIK, to the Downloads folder.**

Starting Deployment Workbench

Deployment Workbench is the MDT 2010 tool that you use to stock deployment shares, create task sequences, and so on. See Chapter 6 for more information about using Deployment Workbench to stock a distribution share and create custom Windows 7 images. To start Deployment Workbench, click Start, point to All Programs, select Microsoft Deployment Toolkit, and then click Deployment Workbench. The console tree shows the following items:

- **Information Center** This item provides access to the documentation, breaking news about MDT 2010, and the components required for using Deployment Workbench. The Documentation item helps you quickly navigate the solution's guidance. Click a link to open that guide as a compiled help (.chm) file.
- **Deployment Shares** Under Deployment Shares, you see an item for each deployment share that you create. Each deployment share contains applications, operating systems, out-of-box device drivers, packages, and task sequences. Additionally, you can create boot media, link deployment shares, and connect the deployment share to a deployment database.

NOTE **For the Deployment Workbench MMC, the default view includes the Action pane. To remove the Action pane, open the management console in author mode. To open the console in author mode, run C:\Program Files\Microsoft Deployment Toolkit\Bin \DeploymentWorkbench.msc /a. Click View, click Customize, clear the Action Pane check box, and then click OK. To save changes, from the File menu, select Save. When prompted to choose whether to display a single window interface, click Yes.**

Updating Microsoft Deployment Toolkit Components

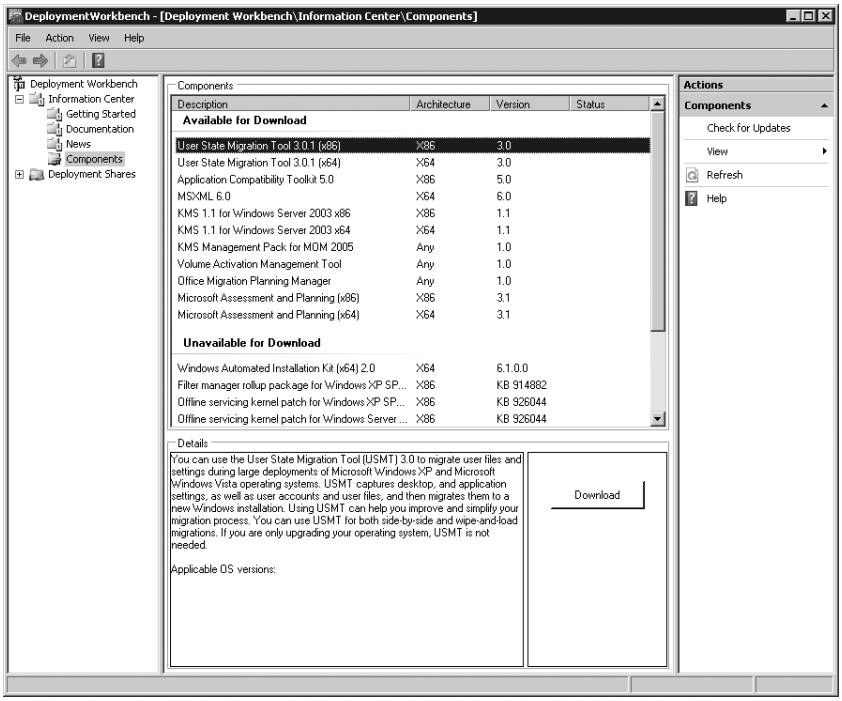
After installing MDT 2010 and becoming familiar with Deployment Workbench, download and install the additional components that MDT 2010 requires. The following components are mandatory in MDT 2010:

- **Windows AIK 2.0** You can install Windows AIK 2.0 manually by downloading it from the Microsoft Download Center at *http://www.microsoft.com/downloads* or use Deployment Workbench to download and install it automatically.

- **MSXML Services 6.0** You can preinstall MSXML Services 6.0 or use Deployment Workbench to download and install it. The Windows AIK distribution media includes the MSXML Services 6.0 SP1 installation file. You can also download MSXML Services 6.0 SP1 from *http://www.microsoft.com/downloads/details.aspx?FamilyID=d21c292c-368b-4ce1-9dab-3e9827b70604&displaylang=en*. Download both x86 and x64 versions at this location. Windows Vista with SP1, Windows 7, Windows Server 2008, and Windows Server 2008 R2 already include this feature.

To download components using Deployment Workbench, perform the following steps:

1. In Deployment Workbench, under Information Center, click Components.



2. In the Available For Download section of the Components list, click a component. In the bottom pane, click Download. Deployment Workbench displays the download status in the Components list. When it finishes downloading the component, it moves the component to the Downloaded section in the right pane.
3. In the Downloaded section of the Components list, click a downloaded component. In the bottom pane, click Install to install the component or Browse to open the folder containing the component in Windows Explorer. MDT 2010 cannot install some components automatically. To install them, click Browse to open the folder containing the component and then manually install the component.

NOTE **Check the Internet for updated components frequently. On the Deployment Workbench main menu, from the Action menu, click Check For Updates. On the Check For Updates page of the Check For Updates Wizard, select the Check The Internet check box and then click Check.**

Summary

Early planning is important to a successful Windows 7 deployment project. Although MDT 2010 provides the technology framework for deploying Windows 7, MOF 4.0 contains the most comprehensive deployment planning guidance from Microsoft. It includes proven best practices from Microsoft's own experience and that of its partners and its customers.

Planning to deploy Windows 7 is a far larger initiative than just figuring out how to install the operating system. Key planning topics include compatibility testing, application packaging, image engineering, user state migration, security, and deployment. This Resource Kit, MDT 2010, and MOF 4.0 help you plan and develop solutions for each of these topics.

Additional Resources

These resources contain additional information and tools related to this chapter.

- The *Getting Started Guide* in MDT 2010 contains essential information for installing and configuring MDT 2010 as well as preparing your infrastructure for using it.
- Chapter 6, "Developing Disk Images," explains how to plan and engineer Windows 7 disk images by using MDT 2010.
- Chapter 7, "Migrating User State Data," explains how to plan and design a user state migration solution by using USMT.
- Chapter 8, "Deploying Applications," includes more information about packaging and automating application installations. This chapter also discusses application-compatibility remediation.
- Chapter 11, "Using Volume Activation," explains how to account for Windows 7 volume activation in large-scale deployment projects.
- Chapter 12, "Deploying with Microsoft Deployment Toolkit," explains how to use MDT 2010 to deploy Windows 7 by using the LTI process.
- MOF 4.0, at *http://technet.microsoft.com/en-us/library/cc506049.aspx,* provides project management guidance and job aids.

CHAPTER 5

Testing Application Compatibility

Application compatibility is often a deployment-blocking issue. It's also the issue that most deployment projects focus on the least—until things begin to fall apart. By focusing on application compatibility early, you can better ensure a successful deployment project.

Three common reasons that application compatibility blocks operating-system deployment are fear, uncertainty, and doubt. Companies simply don't know what applications are in their environments, whether the applications are compatible with the Windows 7 operating system, and what risks each application poses if it fails after deployment.

To help overcome these issues, this chapter describes the Microsoft tools that are available for discovering the applications in your environment, evaluating their compatibility with Windows 7, and then developing fixes for any issues. The primary tool in Windows 7 is the Microsoft Application Compatibility Toolkit (ACT) 5.5.

Understanding Compatibility

Since the arrival of Microsoft Windows as a ubiquitous application platform, Independent Software Vendors (ISVs) and internal developers have created thousands of applications for it. Many are mission-critical applications—some of which aren't compatible with the latest version of Windows. Types of applications that might not be compatible include the following:

- Line-of-business (LOB) applications, such as enterprise resource-planning suites
- Core applications that are part of standard desktop configurations
- Administrative tools, such as antivirus, compression, and remote-control applications
- Custom tools, such as logon scripts

What Compatibility Means

Applications designed for earlier versions of Windows have been carried forward for a number of reasons. Maybe the application is a necessary tool that is used daily to accomplish some otherwise tedious task. Maybe users have learned the application and are reticent to move to another, similar application. Maybe the application has no replacement because the original creator either is no longer in business or has left the company. All these issues make application compatibility a critical issue that you must consider when deploying a new operating system such as Windows 7. In this chapter, you learn the many issues that affect application compatibility, how to discover the applications on which the organization depends, and what you can do to ensure that mission-critical applications work with Windows 7.

An application is compatible with Windows 7 if it runs as designed in Windows 7—that is, the application should install and uninstall correctly. Users should be able to create, delete, open, and save any data files that are native to the application. Common operations such as printing should work as expected. A compatible application runs on Windows 7 out of the box, without any special assistance. If an application is not compatible, you might find that a newer, compatible version of the application is available or that using one of the tools that Microsoft provides to remediate the compatibility problem is all you need. You might also find that an application will require a combination of fixes to run properly. This chapter discusses all these scenarios.

Why Applications Fail

The following list describes common compatibility issues for Windows 7, particularly when using an application originally designed for Windows XP:

- **User Account Control** In Windows 7, by default, all interactive users, including members of the Administrators group, run as standard users. User Account Control (UAC) is the mechanism through which users can elevate applications to full administrator privileges. Because of UAC, applications that require administrator rights or check for administrator privileges behave differently in Windows 7, even when run by a user as administrator.

- **Windows Resource Protection** Windows Resource Protection (WRP) is designed to protect the system in a read-only state to increase system stability, predictability, and reliability. This will affect specific files, folders, and registry keys. Updates to protected resources are restricted to the operating-system trusted installers (TrustedInstaller group), such as Windows Servicing. This helps to protect features and applications that ship with the operating system from any impact of other applications and administrators. This impact can be an issue for custom installations not detected as set up by Windows 7 when applications try to replace WRP files and registry settings and check for specific versions and values.
- **Internet Explorer Protected Mode** In Windows 7, Windows Internet Explorer 8 processes run in IEPM with greatly restricted privileges to help protect users from attack. Internet Explorer Protected Mode (IEPM) significantly reduces the ability of an attack to write, alter, or destroy data on the user's computer, or to install malicious code. This could affect ActiveX controls and other script code that try to modify higher-integrity-level objects.
- **Operating system and Internet Explorer versioning** Many applications check the version of the operating system and behave differently or fail to run when an unexpected version number is detected. You can resolve this issue by setting appropriate compatibility modes or applying versioning shims (application-compatibility fixes).
- **New folder locations** User folders, My Documents folders, and folders with localization have changed since Windows XP. Applications with hard-coded paths may fail. You can mitigate this by using directory junctions or by replacing hard-coded paths with appropriate API calls to get folder locations.
- **Session 0 isolation** Running services and user applications together in Session 0 poses a security risk because these services run at an elevated privilege level and therefore are targets for malicious agents looking for a means to elevate their own privilege level. In earlier versions of the Windows operating system, services and applications run in the same session as the first user who logs on to the console (Session 0). To help protect against malicious agents in Windows 7, Session 0 has been isolated from other sessions. This could impact services that communicate with applications using window messages.

Choosing the Best Tool

You can use five primary tools to mitigate application-compatibility issues: Program Compatibility Assistant, Program Compatibility Wizard, ACT, Windows XP Mode, and application virtualization. The following sections describe each tool and when it is appropriate to use each tool versus using other tools or technologies.

The first two tools provide approaches for users and one-off support issues but are not for use in a large-scale deployment. The remainder of this chapter focuses on using the ACT to inventory, analyze, and mitigate compatibility issues, because this is the tool that organizations primarily use in large-scale deployment.

NOTE **The Microsoft Assessment and Planning (MAP) Toolkit 4.0 is a free tool from Microsoft that you can use to plan your Windows 7 migration project. It can help you assess your environment's readiness for Windows 7. For more information about MAP, see *http://technet.microsoft.com/en-us/solutionaccelerators/dd537566.aspx*.**

Program Compatibility Assistant

If you have an individual application that needs Windows 7 compatibility remediation, one mitigation tool that might be effective is the built-in Program Compatibility Assistant. To run the Program Compatibility Assistant, right-click the application's .exe file, click Properties, and then click the Compatibility tab to view the application's compatibility settings, as shown in Figure 5-1. If the program is used by multiple users on the same computer, click Change Settings For All Users to change the settings selected here to affect all users.

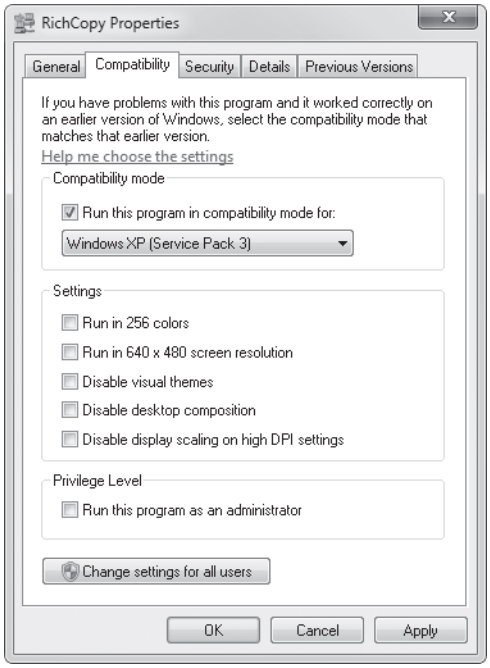


FIGURE 5-1 Compatibility settings

Program Compatibility Troubleshooter

The Program Compatibility troubleshooter can help resolve many application issues. Using the troubleshooter, you can test various compatibility options on various programs to find the setting that allows the programs to run under Windows 7. To start the Program Compat-

ibility troubleshooter, click Start, Control Panel, Programs, and then Run Programs Made For Previous Versions Of Windows. The Program Compatibility troubleshooter starts as shown in Figure 5-2. To begin the application compatibility diagnostic process, click Next.

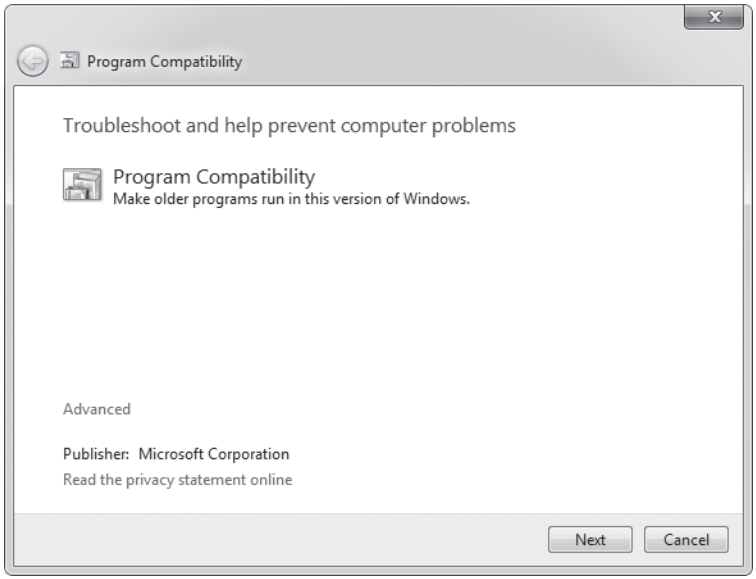


FIGURE 5-2 Program Compatibility troubleshooter

NOTE **To get the most accurate results with this troubleshooter, log on to the computer with a user account that has standard user rights, not administrator rights.**

Application Compatibility Toolkit

You can use the ACT for anything more than a few one-off, simple mitigations. It helps you create an inventory of the applications in your organization. It also helps identify which applications are compatible with Windows 7 and which applications require further testing. The following are some of the major components of the ACT solution:

- **Application Compatibility Manager** A tool that enables you to collect and analyze your data so that you can identify any issues prior to deploying a new operating system or a Windows update in your organization. You use this program heavily during the initial phases of an application migration project. You should consider this tool as the primary user interface for the ACT.
- **Application Compatibility Toolkit Data Collector** The Application Compatibility Toolkit Data Collector is distributed to each computer and scans by using compatibility evaluators. Data is collected and stored in the central compatibility database.

- **Setup Analysis Tool (SAT)** Automates the running of application installations while monitoring the actions taken by each application's installer.
- **Standard User Analyzer (SUA)** Determines the possible issues for applications running as a standard user in Windows 7.

The ACT is indispensable for testing a wide variety of applications across a variety of computers and operating systems within your organization, using solutions based on a common pool of application fixes provided by vendors and users. You can use the ACT tools separately or together, based on your organization's needs. Some tools, such as the SAT and SUA, are intended for developers to enable application remediation and are not necessarily used in the scanning and mitigation process.

Windows XP Mode

In some instances, such as when standard application mitigation strategies fail, virtualization technologies might be appropriate. For example, you can use Windows XP Mode as a safety net for applications that aren't compatible with Windows 7. The Windows XP Mode environment is available for Windows 7 Enterprise, Professional, and Ultimate Edition operating systems.

Using Windows XP Mode, users can run a Windows XP virtual machine on a computer that is running Windows 7. This way, you can proceed with your Windows 7 deployment rather than delay because of application incompatibility. Your organization can take full advantage of the new features and capabilities in Windows 7 and still provide user access to earlier versions of mission-critical applications. In addition, the organization can realize a return on the investment of upgrading to Windows 7 faster than it would by implementing other short-term application compatibility solutions.

Windows XP Mode requires Windows Virtual PC, which is an update that you apply to Windows 7 Enterprise, Professional, and Ultimate Editions. Windows Virtual PC provides a time-saving and cost-saving solution anywhere users must run multiple operating systems (x86 operating systems only) and is an excellent short-term solution for mitigating application compatibility issues; it allows you to continue with Windows 7 deployment. However, you should consider selecting a longer-term solution. Note that Windows Virtual PC requires a CPU with the Intel Virtualization Technology or AMD-V feature turned on. This feature must be enabled in the system BIOS.

Installing Windows XP Mode is easy. You must first install Windows Virtual PC and then install Windows XP Mode. You can perform both tasks from the Windows Virtual PC Web site at *http://www.microsoft.com/windows/virtual-pc/download.aspx*.

MORE INFO **For step-by-step instructions on using Windows XP Mode, including installing and using applications, see *http://www.microsoft.com/windows/virtual-pc/support/default.aspx*.**

Application Virtualization

Very often, even if an application does work with Windows 7, it will still create conflicts with other applications running in the same environment because they are competing for system resources. Application virtualization plays a key role in mitigating those concerns.

Microsoft Application Virtualization (App-V) transforms applications into virtualized, network-available services, resulting in dynamic delivery of software that is never installed, never conflicts, and minimizes costly application-to-application regression testing. By using this technology, users and their application environments are no longer computer specific, and the computers themselves are no longer user specific. Although App-V typically provisions applications to run independently of each other in isolated environments, App-V does permit some application interaction. You should carefully examine any dependencies that applications might have on one another and sequence applications together if they rely on interacting with each other.

This allows IT administrators to be flexible and responsive to business needs and significantly reduces the cost of computer management, including enabling application and operating-system migrations, by separating the application deployment from the core operating-system image. App-V is an integral tool in the Microsoft Desktop Optimization Pack for Software Assurance solution, a dynamic desktop solution available to Software Assurance customers that helps reduce application deployment costs, enable delivery of applications as services, and better manage and control enterprise desktop environments. For more information, see *http://www.microsoft.com/windows/enterprise/default.aspx*.

Understanding the ACT

Figure 5-3 illustrates the architecture of the ACT. The following list describes each component of this architecture:

- **Application Compatibility Manager** A tool that enables you to configure, collect, and analyze your data so that you can triage and prioritize any issues prior to deploying a new operating system, updating your version of Internet Explorer, or deploying a Windows update in your organization.
- **Data Collection Package** An .msi file created by the Application Compatibility Manager (ACM) for deploying to each of your client computers. Each Data Collection Package (DCP) can include one or more compatibility evaluators, depending on what you are trying to evaluate.
- **ACT Log Processing Service** A service used to process the ACT log files uploaded from your client computers. It adds the information to your ACT database.
- **ACT Log Processing Share** A file share, accessed by the ACT Log Processing Service, to store the log files that will be processed and added to the ACT database.
- **ACT Database** A Microsoft SQL Server database that stores the collected application, computer, device, and compatibility data. You can view the information stored in the ACT database as reports from the ACM.

- **Microsoft Compatibility Exchange** A Web service that propagates application compatibility issues from the server to the client and enables the client computers to connect to Microsoft via the Internet to check for updated compatibility information. This service does not automatically fix compatibility issues, as it is only an information sharing system.

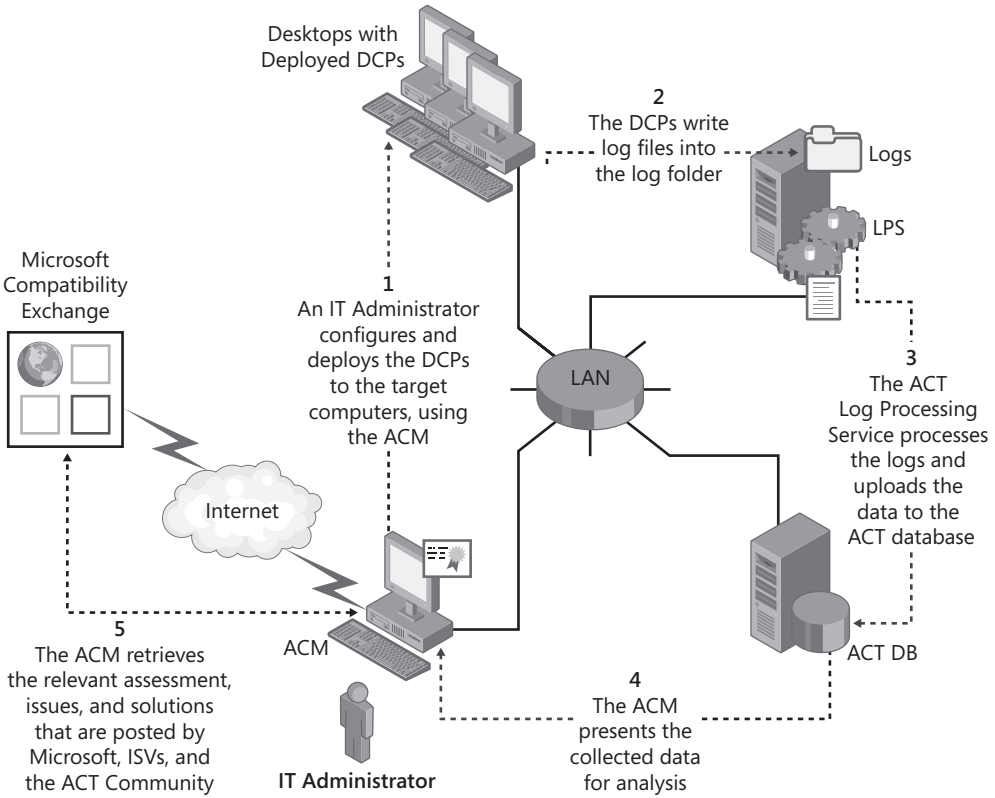


FIGURE 5-3 ACT architecture

Support Topologies

Figure 5-4 shows the topologies that the ACT supports in the order that Microsoft recommends them. For example, Microsoft most highly recommends using the distributed ACT Log Processing Service, ACT Log Processing share, and ACT Database topology, and least recommends using a consolidated server. If you choose to employ a topology based on distributed logging with a rollup to your central share, you must move the files to the ACT Log Processing share before actual processing can occur. You can move the files manually or use a technology such as Distributed File System Replication (DFSR) or any other similar technology already employed in your organization.

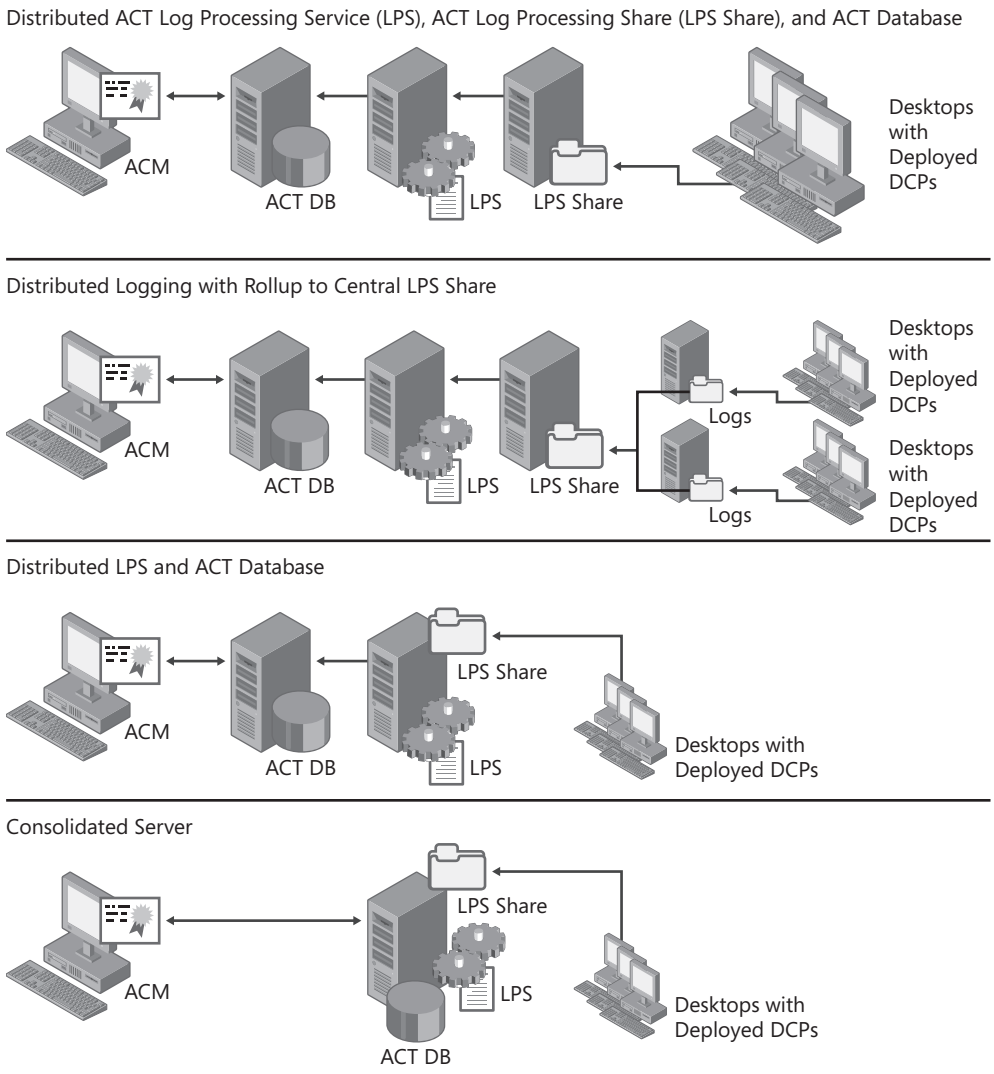


FIGURE 5-4 Supported topologies

Compatibility Evaluators

In addition to collecting application and hardware inventory, the ACT includes compatibility evaluators. Compatibility evaluators are run-time detection tools specifically designed to log behaviors as they occur on the user's computer and locate potential compatibility issues. You should use the compatibility evaluators prior to deploying Windows 7, as they cannot identify problems with an application if the application cannot run. The ACT includes the following compatibility evaluators:

- **Inventory Collector** Examines each of your organization's computers, identifying the installed applications and system information.
- **User Account Control Compatibility Evaluator** Identifies potential compatibility issues because of an application running under a protected administrator (PA) or standard user account on the Windows 7 operating system. When running, the User Account Control Compatibility Evaluator (UACCE) monitors your running applications to verify interactions with the operating system and identify potentially incompatible activities.
- **Update Compatibility Evaluator (UCE)** Identifies the potential impact of a new Windows update. Using the collected update impact data, you can prioritize your testing and reduce the uncertainty in deploying updates.
- **Windows Compatibility Evaluator** Identifies potential compatibility issues resulting from deprecated features in the new operating system, Graphical Identification and Authentication (GINA) dynamic-link libraries (DLLs), and the isolation required by Session 0 applications. Do not run the Windows Compatibility Evaluator (WCE) on Windows 7. If you've already upgraded to Windows 7, you already know about any problems that the WCE has reported.

NOTE **Microsoft removed the Internet Explorer Compatibility Evaluator (IECE) from ACT 5.5. You can detect compatibility issues with Internet Explorer 8 by using the Internet Explorer Compatibility Test Tool (IECTT). Microsoft removed the IECE because using the IECTT provides a better experience when testing for Internet Explorer compatibility.**

A DCP can include one or more compatibility evaluators, depending on what you are trying to evaluate. The ACM groups the evaluators based on tasks as described in the following sections.

Planning for the ACT

The ACT provides a way for you to create an inventory for your organization, including your installed applications, computers, and devices. It also enables you to collect compatibility data, to determine the impact of that data in your organization, and, finally, to create mitigation packages to fix the compatibility issues, when possible. The following list describes the three phases for effectively using the ACT in your organization:

- **Collecting data** Before you can analyze your potential compatibility issues, you must first collect your organization's inventory and the associated compatibility issues.
- **Analyzing issues** After collecting your inventory and associated compatibility data, you can organize and analyze your issues. This includes categorizing, prioritizing, setting your deployment status, and setting your application assessment to create customized reports.

- **Testing and mitigating issues** After analyzing your compatibility issue reports, you can test your applications to determine whether the specified compatibility issues are actually problems within your organization. If you determine that the issues are valid, you can create mitigation packages to fix the issues by using the Compatibility Administrator. You can also use the other tools provided with the ACT—including the IECTT, the SAT, and the SUA tool—to determine additional issues and possible mitigation strategies.

Targeting Deployment

For greater control over your collected data, you should deploy DCPs to a small subset of computers based on specific groupings, such as location and department—for example, a DCP for users in the Human Resources department. This enables better categorization and analysis of an application throughout the organization.

If your organization already has a hardware asset inventory list, it is recommended that you sample each unique hardware configuration so that you can synchronize with the Microsoft Compatibility Exchange and obtain the relevant driver compatibility issues. If you do not have a comprehensive inventory, Microsoft recommends that you distribute the DCPs based on the factors described in Table 5-1.

TABLE 5-1 DCP Deployment Considerations

CONSIDERATION	DESCRIPTION
Do you have a managed, unmanaged, or mixed environment?	You categorize your organization as a managed environment, an unmanaged environment, or a mixed management environment through the following criteria: ■ **Managed environment** IT administrators strictly control and manage the application installation and usage based on need and the various divisions in the organization. In this situation, an IT administrator can deploy a DCP on a limited subset of computers for each department, based on known needs and requirements. ■ **Unmanaged environment** Users typically have administrator privileges on their computers and can install applications at their own discretion. Because users in an unmanaged environment can install any software they choose, you need to deploy your DCPs to more computers than you would if you were in a managed environment. ■ **Mixed environment** Your organization uses both managed and unmanaged environments, depending on an individual group's needs and administrative privileges.

CONSIDERATION	DESCRIPTION
How do you use specific applications in your organization?	It is very important that you provide coverage for all applications required by users in your organization, but it's even more important that you provide coverage for your LOB applications. For the most complete coverage of application usage, you must do the following: ■ Consult with your local administrators, support engineers, and department heads to ensure that all applications are in use during the data collection process. ■ Ensure that "seasonal" applications are covered. For example, fiscal year accounting applications might be used only once a year. ■ Attempt to perform the data collection when few employee vacations are scheduled or at the beginning of the week to avoid weekends. Otherwise, you might have limited or incomplete results because of the decreased application usage. In all cases, recruit willing participants that will be responsible for their team and applications and will report everything they find. User acceptance testing is critical.
Do you use role-based applications?	Your organization may use role-based applications, which are applications that relate to job function and the role that a user performs within your organization. A common example is accountants (a financial role) and their finance-related applications. Reviewing application usage in conjunction with job function and roles enables better application coverage in your organization.
How do you distribute your applications in your organization?	You can distribute applications in many ways within an organization—for example, by using Group Policy, IntelliMirror, Microsoft System Center Configuration Manager 2007, or a custom distribution method. Reviewing your software distribution system policies in conjunction with your application inventory enables better application coverage and narrows the deployment of your DCPs.
What is the geographic breakdown of your organization?	You must consider the geographic distribution of your organization when planning for your DCP deployment (for example, if you have branches in North America, Asia, and Europe). You must then consider the application usage patterns across each geographic region. You must account for divisional applications, localized versions of applications, and applications specific to the geographic location and export restrictions. We recommend that you consult with technical and business leaders from each region to understand these differences.

CONSIDERATION	DESCRIPTION
What types of computers do you have in your organization and how are they used?	Computer types and usage patterns can play an important role in your DCP deployment. The following sections describe some of the most common computer types and usage patterns: ■ **Mobile and laptop computers** Mobile users frequently work offline, occasionally synchronizing with the corporate network through either a LAN or virtual private network (VPN) connection. Because of the high possibility of a user going offline for long periods of time, you must consider the odds of the user being online for the DCP to be downloaded and installed, and then online again for the logged data to be uploaded. ■ **Multiuser computers** Multiuser computers are typically located in university computer labs, libraries, and organizations that enable job sharing. These computers are highly secure and include a core set of applications that are always available, as well as many applications that can be installed and removed as necessary. Because these computers typically have a basic set of applications assigned to users or computers, you can narrow the application coverage and usage to identify only a subset of client computers to receive the DCP. ■ **AppStations/TaskStations** AppStations running vertical applications are typically used for marketing, claims and loan processing, and customer service. TaskStations are typically dedicated to running a single application, such as on a manufacturing floor as an entry terminal or in a call center. Because both of these types of computers do not commonly allow users to add or to remove applications and might be designated for specific users and job roles, the application coverage and usage can be narrowed to identify a subset of client computers to receive the DCP. ■ **Kiosks** Kiosks are generally in public areas. These computers run unattended and are highly secure, generally running a single program by using a single-use account and automatic logon. Because these computers typically run a single application, the application coverage and usage can be narrowed to identify a subset of computers to receive the DCP.

Choosing a Deployment Method

Microsoft recommends that you base your method for deploying the DCP on your existing infrastructure. You can choose one of several ways to distribute a DCP to your identified client computers, including the following (listed in order of preference):

- **System Center Configuration Manager 2007** After performing an inventory of your applications, you can use the software deployment feature in System Center Configuration Manager 2007 to deploy the DCPs to the client computers. Additionally, the inventory information that it contains is a valuable aid.
- **Group Policy Software Installation** Create an .msi package for each DCP, and then use the Group Policy Software Installation feature of Active Directory Domain Services (AD DS) in Windows Server 2008 and Windows Server 2008 R2 for deployment. All client computers to which you will deploy the DCP must be part of the AD DS forest.
- **Logon scripts** While logged on to a domain from the client computers, you can initiate the installation of DCPs using logon scripts in Windows Server 2008 and Windows Server 2008 R2.
- **Non-Microsoft deployment software** If your organization has a non-Microsoft software deployment infrastructure, use that method to deploy the DCPs. For information about the requirements of the non-Microsoft deployment software, consult the software vendor.
- **Manual distribution** For computers that are not connected to the network or that have slow connections, such as small branch offices, manual distribution methods are available. These methods include distributing the collection packages through e-mail or on physical media such as a USB Flash drive (UFD) or CD.

Choosing a Log File Location

When you are creating a DCP in the ACM, you can select an output location for your log files. The following configuration options are available:

- **Select a default ACT Log Processing share location** If you use this option, the DCP automatically writes the log files to the ACT Log Processing share. If the ACT Log Processing share is unavailable when the specified upload time interval is reached, the DCP will make two more attempts. If the problem persists, the DCP will store the log file in the location defined in the next option. All files are then retried during the next upload interval.
- **Select the Local (%ACTAppData%\DataCollector\Output) location** If you use this option, the DCP creates the log files on the local system and the computer Administrator must manually copy the files to the ACT Log Processing share location. This is a good option for mobile users that are not always connected to the network. If you select this option, Microsoft recommends that you either notify your users to copy the collected data to the ACT Log Processing share or employ an alternate method to collect the data from the client computers and copy the information into the ACT Log Processing share.

- **Type an alternate network share location** If you use this option, you must verify that the DCP service can write to the location. This is a good option for companies that are geographically diverse (for example, if you have branches in North America and Europe). An IT administrator can create DCPs and file shares individually for North America and Europe, which further enables administrators at a central location to roll up all the collection log files to a central location. These log files are then mapped to the ACT Log Processing share for final processing and entry into the ACT database.

Preparing for the ACT

Before configuring and running the ACT, you must verify that you are using supported software, that you meet the minimum hardware requirements, and that you have configured the required permissions and infrastructure. Table 5-2 lists the software required by the ACT. Table 5-3 lists the hardware requirements for using the ACT.

You must provide special system requirements before you can successfully use the Update Compatibility Evaluator (UCE), the SAT, or the Compatibility Administrator. For more information, see the ACT 5.5 documentation. The UCE is not compatible with any 64-bit version of Windows.

TABLE 5-2 Software Requirements for the ACT

SOFTWARE	SUPPORTED VERSIONS
Operating systems	■ Windows 7 ■ Windows Vista ■ Windows Vista SP1 ■ Windows Vista SP2 ■ Windows XP SP2 ■ Windows XP SP3 ■ Windows Server 2008 R2 ■ Windows Server 2008 ■ Windows Server 2003 SP2
Proxy server	The ACT supports only the Microsoft Internet Security and Acceleration (ISA) Server proxy server.
Database	After the ACT is installed, it requires one of the following database components: SQL Server 2005, SQL Server 2005 Express, SQL Server 2008, or SQL Server 2008 Express. *Note:* The ACT does not support the Microsoft Database Engine (MSDE) or Microsoft SQL Server 2000.
.NET Framework	The ACT requires Microsoft .NET Framework 2.0 or later.

TABLE 5-3 Hardware Requirements for the ACT

ACT COMPONENT	MINIMUM REQUIREMENT	RECOMMENDED REQUIREMENT
ACM client and ACT Log Processing Service servers	550-megahertz (MHz) processor with 256 megabytes (MB) of RAM	2.8-gigahertz (GHz) processor with 2 gigabytes (GB) of RAM
ACT client databases	1-GHz processor with 512 MB of RAM	2.8-GHz processor with 2 GB of RAM

Sharing the Log Processing Folder

If your DCPs write to a network ACT Log Processing share, you must verify that you have the correct permissions at both the share and the folder levels, as follows:

- **Share-Level Permissions** Verify that the Everyone group has Change and Read permissions for the ACT Log Processing share folder.
- **Folder-Level Permissions (NTFS Only)** Verify that the Everyone group has Write access and that the ACT Log Processing Service account has List Folder Contents, Read, and Write permissions. If the ACT Log Processing Service is running as Local System, this must be the *domain**computer*$ account. If the ACT Log Processing Service is running with a user account, this is the user account information.

Preparing for Microsoft Compatibility Exchange

Configure your organization's infrastructure to support the Microsoft Compatibility Exchange while also protecting your intranet security and stability. The recommended method of configuration requires you to allow the appropriate users, on designated computers, to access the Microsoft Compatibility Exchange through your security and network infrastructure. To configure the infrastructure to support the Microsoft Compatibility Exchange, follow these steps:

1. Configure your firewalls and Uniform Resource Locator (URL) scanners to allow access to the Microsoft Compatibility Exchange by setting the following conditions:
   - Allow outbound access for the standard Secure Sockets Layer (SSL) TCP port 443 on any computer running the ACM.
   - Restrict outbound access to the Microsoft Compatibility Exchange, allowing access only from designated computers and designated users within your organizations.
   - Enable access to the Microsoft Compatibility Exchange (*https://appinfo.microsoft.com/AppProfile50/ActWebService.asmx*), which is necessary only if passing through a firewall.
2. Grant the *db_datareader, db_datawriter,* and *db_owner* database roles to any user account that will log on to the computer running the ACT Log Processing Service.
3. Grant the *db_datareader* and *db_datawriter* database roles to any user account that will log on to the computer running the ACM.

Installing the ACT 5.5

You can download the ACT 5.5 from the Microsoft Download Center at *http://www.microsoft.com/downloads*. Before you install the ACT, ensure that the computer on which you're installing it meets the requirements described in the section titled "Preparing for the ACT" earlier in this chapter.

To install the ACT, perform the following steps:

1. Right-click Application Compatibility Toolkit.msi and then click Install.
2. Click Next.
3. On the License Agreement page, click I Accept The Terms In The License Agreement and then click Next.
4. If you want to install the ACT 5.5 in a different location than the default folder, on the Installation Folder page, click Change to change the installation folder and then click Next.
5. Click Install.
6. Click Finish.

Configuring the ACM

Before you can use the ACM to collect and analyze your compatibility data, you must configure the tool. This includes configuring the following: your SQL Server instance and database, your ACT Log Processing Service account, and your ACT Log Processing share.

The ACT Configuration Wizard enables you to configure the ACT database, the ACT Log Processing share, and the ACT Log Processing Service account. Before running the wizard, you must verify the following:

- You are an administrator on the computer, and you have Read and Write permissions to the database.
- Your domain computer has Write permissions to the ACT Log Processing Service share.
- The ACT Log Processing Service account has Read and Write permissions to the ACT database for the *domain**computer*$ account.
- The ACT client is installed on any computer that acts as an ACT Log Processing Server.

To configure the ACM, perform the following steps:

1. Click Start, point to All Programs, Microsoft Application Compatibility Toolkit 5.5, and then select Application Compatibility Manager to start the ACT Configuration Wizard.
2. Review the information on the page and then click Next.
3. On the Select The Configuration Option page, click Enterprise Configuration and then click Next.
4. On the Configure Your ACT Database Settings page, type the name of the SQL Server instance that will contain the ACT database in the SQL Server box and then click Connect.

In the Database box, type a unique name for your new database, such as **ACT_Database**, and then click Create. Click Next.

5. On the Configure Your Log File Location page, type the path of the folder in which to store the ACT log files in the Path box or click Browse to choose an existing folder or create a new folder. In the ShareAs box, type a name for the share and then click Next.
6. On the Configure Your ACT Log Processing Service Account page, click Local System to use your local system account credentials to start the ACT Log Processing Service and then click Next. You also have the option to click User Account. If you choose this option, the ACT will use the local computer user account to start the ACT Log Processing Service. Additionally, for this option, you must enter your user name, password, and domain, and provide Log On As A Service user rights.
7. Click Finish.

You have the option to change any of your ACT configuration settings after completing the configuration wizard. On the Tools menu, select Settings and then make your changes in the Settings dialog box (Figure 5-5).

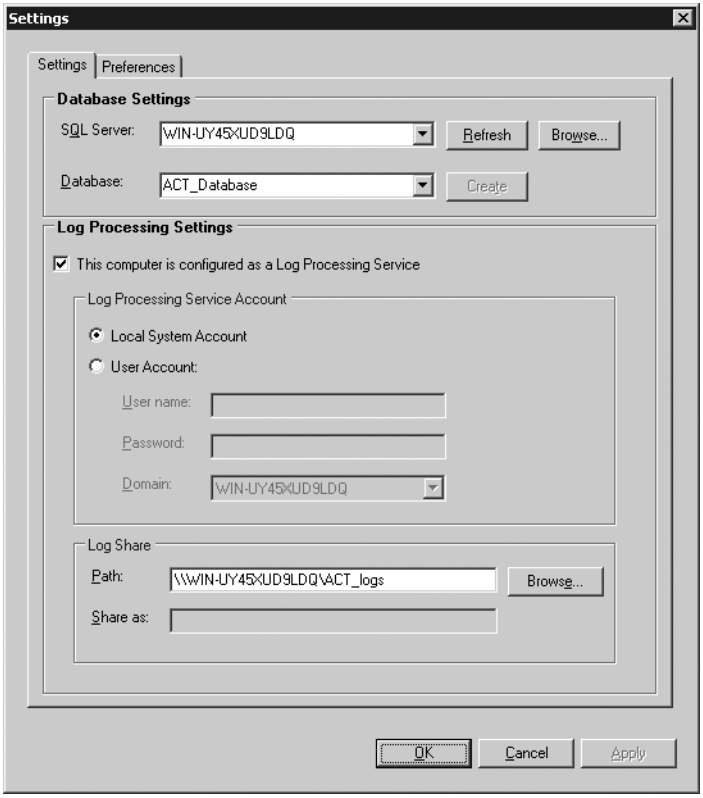


FIGURE 5-5 ACT settings

Collecting Compatibility Data

The ACT enables you to collect an inventory of all installed software, hardware, and devices within your organization. Additionally, the ACT provides compatibility evaluators, which you will use in your DCPs for deployment to your client computers. Compatibility evaluators are run-time detection tools designed to log behaviors as they occur on the user's computer and locate potential compatibility issues.

The ACT collects data according to the following workflow:

1. You create a new DCP by using the ACM. Each DCP can contain one or more compatibility evaluators, including the Inventory Collector.
2. You deploy the DCPs to your identified subset of client computers using System Center Configuration Manager 2007, Group Policy, or any other software distribution technology. The evaluators run for the length of time that you specified when creating the DCP and then the data (.cab) file is uploaded to your ACT Log Processing share.
3. The ACT Log Processing Service, running on a server, accesses the data from the ACT Log Processing share, processes the data, and then uploads the information to your ACT database.
4. The ACM reads the data from your ACT database to determine how many computers have uploaded data and the status of the collection process. The ACM also uses the data from the ACT database to enable reporting and viewing of the collected data.

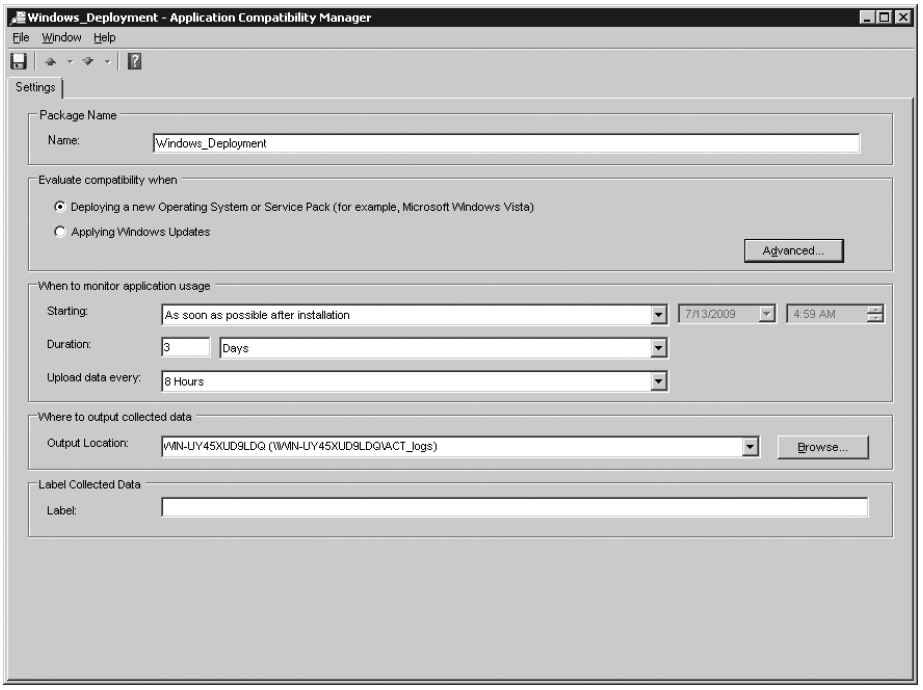
By using the ACM, you can create DCPs to gather your installed software, hardware, and device information, in addition to determining any associated compatibility issues based on applications, Web sites, or Windows updates for your selected client computers. The ACT includes the compatibility evaluators described in the section titled "Compatibility Evaluators" earlier in this chapter.

After creating a DCP, deploy it using the method chosen from the list in the section titled "Choosing a Deployment Method" earlier in this chapter. Because a DCP is an .msi file that installs silently, deploying it is just like deploying any other application. For more information about deploying applications, see Chapter 8, "Deploying Applications."

To create a DCP for deploying Windows 7, perform the following steps:

1. In the ACM, click File and then click New.
2. The New_Package dialog box appears. In the Package Name box, type a unique name for your DCP, such as **Windows_Deployment**.
3. In the Evaluate Compatibility When area, click Deploying A New Operating System Or Service Pack. This evaluator option includes the Inventory Collector, the UACCE, and the WCE by default. If you want, you can click Advanced to choose the specific evaluators to include in the package.
4. In the When To Monitor Application Usage area, configure the starting time, duration, and upload interval.

5. In the Output Location box, shown here, keep your default value, previously specified in the Configuration Wizard.



6. On the File menu, click Save And Create Package, saving the compiled DCP as an .msi file in an accessible location, such as a network share.

To view the status of a DCP, perform the following steps:

1. In the left pane of the ACM, click Collect.
2. Click By Status in the Current View section of the Collect screen. The Collect screen changes to show you the deployed DCPs and their status, including whether they are in progress or complete.

Analyzing Compatibility Data

The ACT enables you to organize and to analyze your data by using categorization, prioritization, organizational assessments, issue and solution management, report management, and filtering. You can access and view all your compatibility data by using the Quick Reports area of the ACM, shown in Figure 5-6.

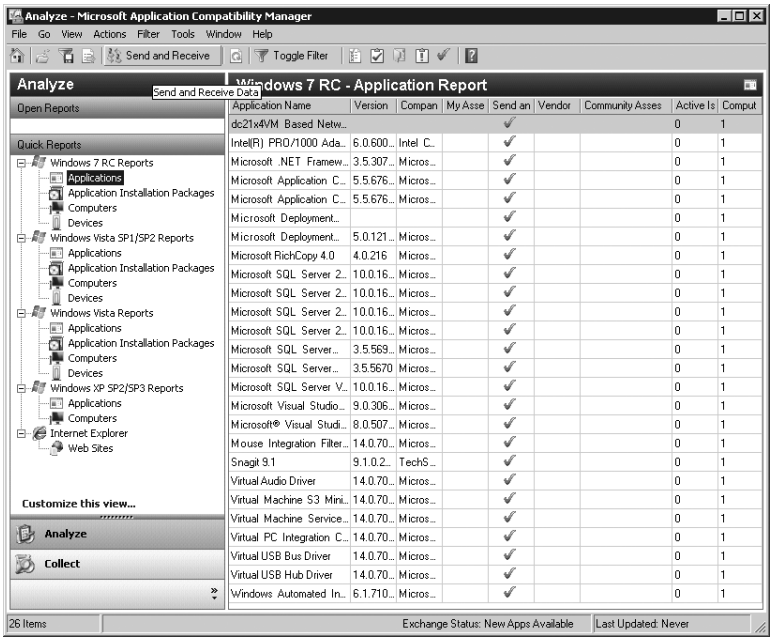


FIGURE 5-6 Quick Reports in the ACM

Creating and Assigning Categories

You can create, modify, and assign categories to all your applications, computers, devices, Web sites, and updates for a more customized ACT compatibility report and for filtering purposes. After assigning the priority categories, the second most commonly used analysis tool is assigning arbitrary categories to each piece of software:

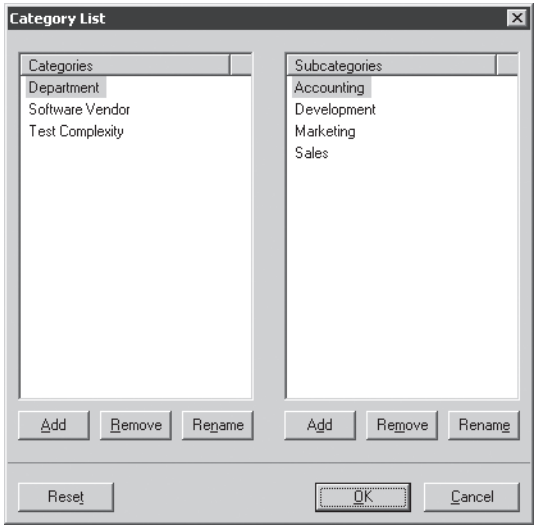
- Software Vendor can be a useful category because you might have varying relationships with each of your vendors. Generating reports and groupings by software vendor can be useful when you have discussions with that vendor and evaluate the vendor's performance with regard to your compatibility needs.
- Test Complexity can be useful for planning and assigning resources. Applications with higher complexity might require additional resources or help to make support decisions. For example, you might assign additional resources to a Business Critical application with an elevated test complexity but remove a Nice To Have application with an elevated test complexity from the supported software list.
- Unit of Deployment is another commonly used set of categories, such as Division and Region. Your organization might choose a different naming convention for this information, but typically, this category enables you to track the software needs of one unit of deployment so that as the necessary software is tested and approved, that deployment unit can proceed.

Because the category option is a completely extensible multiple-selection string value, you can potentially use it for just about anything. Some creative uses include creating a category for signoff from multiple owners so that the software can be authorized only when all categories have been selected (indicating that each group has signed off). You can brainstorm other ideas about how to use categories and how your group perceives the organization of its software ecosystem.

NOTE **By default, the Master Category List dialog box has two categories: Software Vendor and Test Complexity. These are the only default subcategories. For more information about creating and assigning categories and subcategories, see "Categorizing Your Data" in the ACT documentation.**

To create new categories and subcategories, perform the following steps:

1. In the ACM, click Analyze.
2. In the Analyze screen, in the Quick Reports pane, click Applications in the Windows 7 Reports section.
3. On the Actions menu, click Assign Categories.
4. In the Assign Categories dialog box, click Category List.
5. In the Categories area of the Category List dialog box, click Add, type the name of the new category, and then press Enter.
6. In the Subcategories area of the Category List dialog box, shown here, click Add, type the name of a new subcategory, and then press Enter. Repeat this step for each subcategory that you want to add to the category.

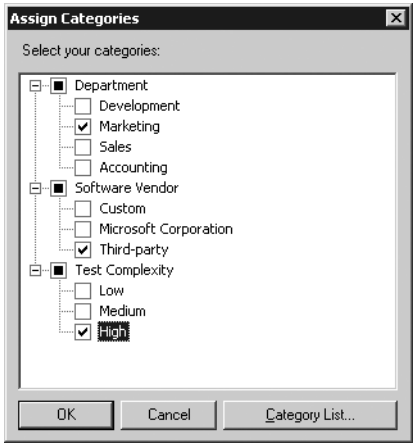


7. Click OK to close the Category List dialog box.

8. Click OK to close the Assign Categories dialog box.

To assign a category or subcategory, perform the following steps:

1. In the ACM, click Analyze.
2. In the Analyze screen, in the Quick Reports pane, click Applications in the Windows 7 Reports section.
3. In the Windows 7 - Application Report, right-click an application and then click Assign Categories.
4. In the Assign Categories dialog box, shown here, select the check box next to each category and subcategory to which you want to assign the application.



5. Click OK to close the Assign Categories dialog box.

Prioritizing Compatibility Data

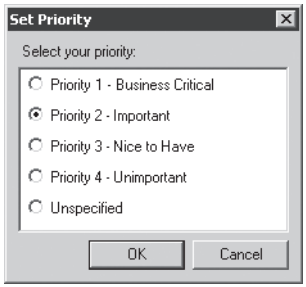
You can prioritize any of your collected compatibility data, except for your setup installation packages, based on your organization's requirements. Prioritizing your data enables you to organize your data better, for both a more customized ACT compatibility report and filtering purposes. The following priority levels are available:

- **Priority 1 – Business Critical** Includes any item that is so important to your organization that, unless you can certify it, you will not continue with your deployment.
- **Priority 2 – Important** Includes any item that your organization regularly uses but can continue to function without. It is your choice whether to continue your deployment without certification.
- **Priority 3 – Nice To Have** Includes any item that does not fall into the previous two categories, but that should appear in your ACT compatibility reports. These items will not prevent you from continuing with your deployment.

- **Priority 4 – Unimportant** Includes any item that is irrelevant to your organization's daily operations. You can use this priority level to filter the unimportant items from your reports.
- **Unspecified** The default priority level, which is automatically assigned to any item. Your organization can use this priority level to denote applications that have not yet been reviewed.

To prioritize your compatibility data, perform the following steps:

1. In the left pane of the ACM, click Analyze.
2. In the Quick Reports pane, click Applications in the Windows 7 Reports section.
3. Right-click an application in the Windows 7 - Application Report and then click Set Priority.
4. In the Set Priority dialog box, shown here, click a priority and then click OK.



Assessing Application Compatibility

You can set your organization's assessment rating for each application, application installation report, and Web site. Setting your assessment rating enables you to specify which applications might be problematic while going through your organization's testing process. Additionally, setting your assessment enables you to organize your data better, for both a more customized ACT compatibility report and for filtering purposes.

NOTE **Microsoft, the application vendor, and the ACT Community also can add assessment ratings. You can view high-level assessment summaries and specific application assessment details in the applicable report screen or report detail screen. For more information about how to view the assessment details, see the ACT documentation.**

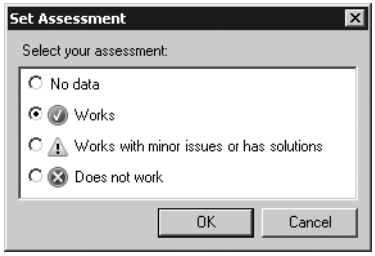
Your assessment choices include the following:

- **Works** Indicates that during your organization's testing process, you did not experience any issues.

- **Works With Minor Issues Or Has Solutions** Indicates that during your organization's testing process, you experienced minor issues (severity 3), such as showing a typographical error, or an issue that already had a known solution.
- **Does Not Work** Indicates that during your organization's testing process, you experienced a severity 1 or severity 2 issue.
- **No Data** Neither your organization, Microsoft Corporation, the vendor of the application or Web site, nor the ACT Community has provided any data.

To assess your compatibility data, perform the following steps:

1. In the left pane of the ACM, click Analyze.
2. In the Quick Reports pane, click Applications in the Windows 7 Reports section.
3. Right-click an application in the Windows 7 - Application Report and then click Set Assessment.
4. In the Set Assessment dialog box, shown here, click an assessment and then click OK.



Setting the Deployment Status

You can set your organization's deployment status for each application, application installation report, Web site, and Windows update. Setting your deployment status enables you to determine where each item is in your testing process. Additionally, setting your deployment status enables you to organize your data better, for both a more customized ACT compatibility report and for filtering purposes. Your deployment status choices include the following:

- **Not Reviewed** Your organization has not yet reviewed this item to determine its impact, testing requirements, or deployment options.
- **Testing** Your organization is in the process of locating compatibility issues.
- **Mitigating** Your organization is in the process of creating and applying solutions for your compatibility issues.
- **Ready To Deploy** Your organization has completed its testing and mitigation processes and has determined that you can deploy the item in your organization.
- **Will Not Deploy** Your organization has decided that you will not deploy the item in your organization.

To assess your deployment status, perform the following steps:

1. In the left pane of the ACM, click Analyze.
2. In the Quick Reports pane, click Applications in the Windows 7 Reports section.
3. Right-click an application in the Windows 7 - Application Report and then click Set Deployment Status.
4. In the Set Deployment Status dialog box, shown here, click a deployment status and then click OK.

Managing Compatibility Issues

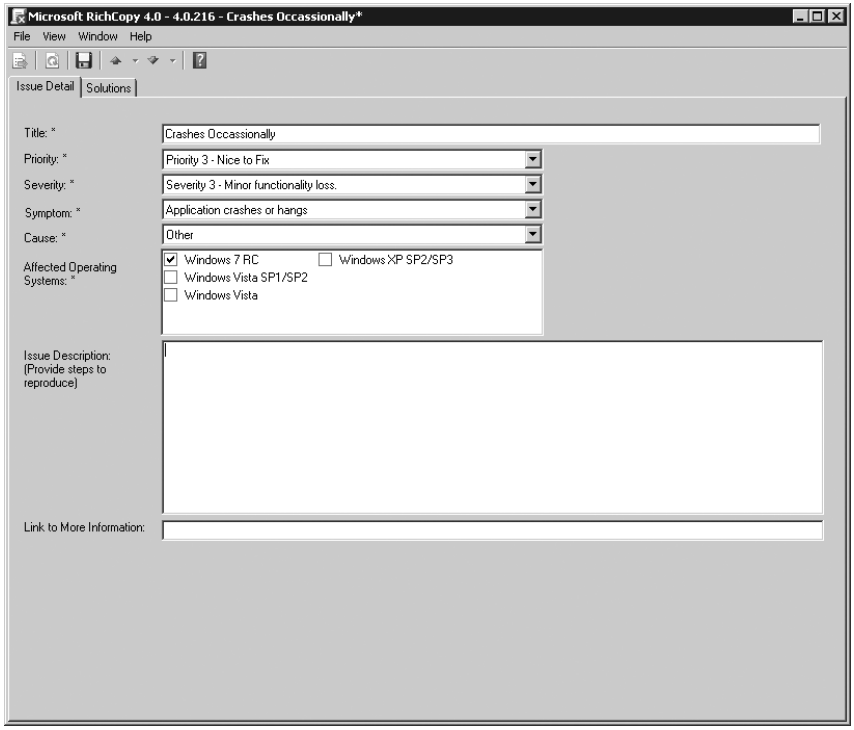
Although the compatibility evaluators, the Microsoft Compatibility Exchange, and the ACT Community all provide information about application compatibility issues, you might still uncover an undocumented issue. After adding your compatibility issue, you can use the Microsoft Compatibility Exchange to upload and to share your issue information with both Microsoft and the ACT Community, if you are a member. You can also add compatibility solutions to any compatibility issue in your ACT database, regardless of whether you entered the issue.

You also can resolve any active compatibility issue in your ACT database, regardless of whether you entered the issue. Resolving an issue means that you are satisfied with the state of the issue and are closing it from further edits. However, you can still add solutions or reactivate the issue if you discover that you resolved it in error. Marking an issue as resolved also changes the issue status from a red X to a green check mark in your compatibility reports, report detail screens, and for the overall group score in the ACT Community data.

To add a compatibility issue, perform the following steps:

1. In the left pane of the ACM, click Analyze.
2. In the Quick Reports pane, click Applications in the Windows 7 Reports section.
3. In the Windows 7 - Application Report, right-click an application and then click Open.
4. On the Actions menu, click Add Issue to open the New Issue dialog box.
5. In the Title box, type a title for the issue.
6. In the Priority list, click a priority.
7. In the Severity list, click a severity level.

8. In the Symptom list, click a symptom.
9. In the Cause list, click a cause for the issue.
10. In the Affected Operating Systems dialog box, shown here, select the check boxes next to each operating system on which this issue appears.



11. In the Issue Description box, type a description of the issue.
12. On the File menu, click Save.

To add a compatibility solution, perform the following steps:

1. In the left pane of the ACM, click Analyze.
2. In the Quick Reports pane, click Applications in the Windows 7 Reports section.
3. In the Windows 7 - Application Report, right-click an application and then click Open.
4. On the Issues tab, double-click the issue for which you want to add a solution.
5. Click the issue's Solutions tab.
6. On the Actions menu, click Add Solution.
7. In the Title box, type a title for the solution.
8. In the Solution Type box, click a solution type.
9. In the Solution Details box, type a description of the solution.
10. Click Save.

To resolve a compatibility issue, perform the following steps:

1. In the left pane of the ACM, click Analyze.
2. In the Quick Reports pane, click Applications in the Windows 7 Reports section.
3. In the Windows 7 - Application Report, right-click an application and then click Open.
4. On the Issues tab, double-click the issue that you want to resolve.
5. On the Actions menu, click Resolve. A note appears in the Issues tab that says the issue is resolved and a green check mark appears in the Status column of the Issues tab.

Filtering Compatibility Data

You can filter your organization's compatibility issue data by selecting specific restriction criteria in context, based on the report that you are viewing. For example, you can filter your applications by category, your Web sites by priority, or a Windows update by deployment status.

To create a filter, perform the following steps:

1. In the ACM, click Analyze.
2. In the Analyze screen, in the Quick Reports pane, click Applications in the Windows 7 Reports section.
3. On the Filter menu, select Toggle Filter to turn on the filter.
4. In the Filter pane, choose a field, an operator, and a value on which to filter. For example, to display only applications with a company name containing *Microsoft*, click Company in the Field column, click Contains in the Operator column, and type **Microsoft** in the Value column. After adding a clause (row), the ACM automatically adds a new, empty clause.
5. Add additional clauses as necessary. You can specify whether all clauses must be true or whether any one of the clauses must be true by choosing And or Or in the And/Or column for each individual clause.
6. Select Refresh from the View menu to display the compatibility database based on your filter.

You can further edit your filter by clicking the Filter menu and then selecting Cut, Copy, Paste, Insert Clause, Delete Clause, or Clear.

To save a filter, perform the following steps:

1. On the File menu, select Save As.
2. In the Save As dialog box, type the path and file name of the ACM Report File (.adq) to save and then click Save.

To export a report, perform the following steps:

1. On the File menu, select Export Report.
2. In the Export Report Data dialog box, choose from one of the following report types in the Save As Type list:

- Microsoft Excel Files (*.xls)
- SV (Comma Delimited) (*.csv)
- XML Document (*.xml)

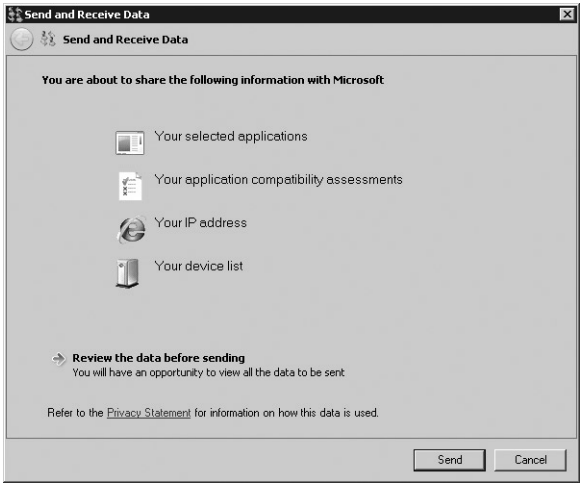
3. In the File Name box, type the path and file name of the report and then click Save.

Synchronizing with the Compatibility Exchange Service

The ACT enables you to synchronize your ACT database with Microsoft and the ACT Community through the Microsoft Compatibility Exchange Web service. This Web service downloads new information from authoritative sources, such as Microsoft and ISVs, and it uploads your compatibility issues to Microsoft. The ACT only displays applications that your environment has in common with the service.

To synchronize with the Microsoft Compatibility Exchange, perform the following steps:

1. In the ACM, click Actions and then click Send And Receive.
2. If you want, in the Send And Receive Data dialog box, click Review The Data Before Sending to view a list of the applications for which you are sending your compatibility data. You can choose the applications that you will share. You can also click Review All Data to save a list of the data that you're sending in an audit log, as shown here.



3. Click Send.
4. Review the updated issue data for your applications in the ACM.

Rationalizing an Application Inventory

After you have finished organizing and analyzing your data, Microsoft recommends that you create an application portfolio for your organization. The application portfolio is a list of all the applications in your organization, including their specific details and compatibility status.

To create an application portfolio, perform the following steps:

1. Collect your application inventory and compatibility data by using the ACT.
2. Organize your data based on your organization's requirements and then analyze the information.
3. Identify any applications that are missing from the inventory.
4. Select specific versions of your inventoried applications to be included in your deployment.

Identifying the Missing Applications

You must identify any applications that were not located during the automated inventory collection process. These applications might be located on portable computers or high-security systems that cannot be accessed for inventory. In these situations, you must document the application manually.

To identify missing applications, perform the following steps:

1. Distribute the application portfolio in your organization; specifically, distribute it to those who have knowledge of the required applications currently in use.
2. Request that the group specified in step 1 review the portfolio for errors.
3. Review the feedback provided from step 2 to analyze the errors in the existing portfolio.
4. Make the appropriate changes to the portfolio based on the review.
5. Publish the revised application portfolio and obtain stakeholder approval of the list and application compatibility status.

Selecting Specific Application Versions

To help reduce the long-term total cost of ownership (TCO), you must reduce the number of supported applications in your organization. For each supported application, you must allocate time, training, tools, and resources to plan, deploy, and support the application. Standardizing your list of supported applications can help to reduce the amount of effort required to support your deployed computer configurations.

If you determine that multiple applications are performing the same task in your organization, Microsoft recommends that you select a single application and include it in your standard portfolio, with an emphasis on the following criteria:

- The application is part of a suite of applications. Applications that are part of a suite (for example, Microsoft Office Word 2007) are more difficult to eliminate from your portfolio because you typically must eliminate the entire suite.
- The vendor supports the application on the new operating system. Identifying support options early can reduce your costs later.
- The application adheres to the Designed for Windows logo program. Applications that display the current compatibility logo have met stringent guidelines for compatibility with the current version of Windows.

- The application provides an .msi package for deployment. If the application provides an .msi package, you will spend less time preparing the application for deployment.
- The application is AD DS–aware. You can manage AD DS–aware applications through Group Policy.
- The application is the latest version available in your inventory. Deploying a later version helps ensure the long-term support of the application because of obsolescence policies.
- The application provides multilingual support. Multilingual support within the application, when coupled with multilingual support in the operating system (such as the multilingual support in Windows 7), enables your organization to eliminate localized versions of the application.
- The application provides a greater number of features. Applications that support a greater number of features are more likely to address the business needs of a larger number of your users.

To select the appropriate version of an application, perform the following steps:

1. Identify the latest version of the application currently installed in your organization.
2. Determine whether a later version of the application is currently available. If so, Microsoft recommends that you include the later version of the application in your analysis.
3. Verify that you have vendor support for each version of the application.
4. Identify the license availability and cost for each application and version.
5. From all the versions available, select one version that is supported on all your client computers.
6. Validate the selected version in your test environment, verifying that it is compatible with your new operating system, Windows update, or Internet Explorer version.

Testing and Mitigating Issues

After you analyze your issues in the ACM, you can continue to explore your compatibility issues by using several development tools provided with the ACT. The development tools enable you to test for a variety of compatibility issues, including Web site and Web application issues, issues related to running as a standard user in Windows 7, and issues that might arise because of actions taken by an application's installer program. Additionally, the ACT provides a tool that can help you resolve many of your compatibility issues: the Compatibility Administrator. To resolve your compatibility problems, you must follow these steps:

1. Identify your most critical applications. Create an inventory of your organization's applications and then verify certification status of the included applications to see whether they require testing.
2. Identify any application compatibility problems. Test each application, determining any compatibility issues if necessary.

3. Resolve any application compatibility issues. Identify and create application compatibility solutions by using the ACT tools, which include the IECTT, either the stand-alone version or the virtual version of the SAT, the SUA, and the Compatibility Administrator.
4. Deploy or distribute your test and certified applications and solutions. Use a deployment and distribution tool, such as System Center Configuration Manager 2007, to deploy your certified applications and compatibility issue solution packages to your client desktops.

When testing an application in a new operating system, Microsoft recommends that you retain the default security feature selections. Microsoft also recommends that you thoroughly test the applications, replicating as many of the usage scenarios from within your organization as possible. Finally, Microsoft recommends that you enter your issues and solutions into the ACM so that you can track the data from a central location.

When testing a Web site or a Web application, Microsoft recommends that you include both intranet and extranet sites, prioritizing the list based on how critical the site or the application is to your organization. Microsoft also recommends that you thoroughly test the Web sites and Web applications, replicating as many of the usage scenarios from within your organization as possible. Finally, Microsoft recommends that you enter your issues into the ACM so that you can share that data with both Microsoft and the ACT Community to receive potential solutions for your issues.

Building a Test Lab

Your test environment should be a long-term investment in the overall deployment process. Retain the test environment after the deployment to assist in future deployment projects. To create the test environment, you must determine how to model the production environment in the test environment and configure the test environment to support automated testing of the mitigation strategies.

Microsoft recommends that you establish a dedicated and isolated lab environment for use in developing and testing the application compatibility mitigation. The lab should mirror your production environment as closely as possible. In some cases, you might find that it is better to open up the test network to existing production services, instead of replicating your production environment in detail. For example, you might want to permit your Dynamic Host Configuration Protocol (DHCP) packets to pass through routers into the test network. Some operations can be safely conducted in the production environment, such as the application inventory collection process. At a minimum, your lab environment should include:

- DHCP services
- Domain Name System (DNS) services
- SQL Server 2005 or SQL Server 2005 Express
- Lab test user accounts, with both normal user and administrative privileges
- Network hardware to provide Internet access (for downloading updates, files, and so on)

- Test computers that accurately reflect production computers in both software and hardware configuration
- A software library representing all the applications to be tested
- Windows Server 2008 R2 with Hyper-V
- Windows Internet Naming Service (WINS) services (optional)

In most instances, you must test the mitigation strategies more than once and must be able to revert reliably to a previous test state. Automating your testing process enables you to ensure reproducibility and consistency in your testing process. Using test automation tools enables you to run your test cases in a standardized, reproducible manner. Using disk-imaging software for physical images of the servers and using software virtualization features for reversing changes to virtualized hard disks enables you to restore your test environment back to a previous state.

Modeling the Production Environment

The goal of the test environment is to model your production environment. The more accurate the production environment, the greater the validity of the testing performed in that test environment. Microsoft recommends the following best practices in creating your test environment:

- Use virtual or physical images of production computers to create their test environment counterparts. Virtual or physical images can help ensure that the test environment configuration accurately reflects the production environment. In addition, the images contain live information (such as users, user profiles, and file permissions) to use in testing.
- Separate your test environment physically from your production environment. A physically separate test environment enables you to use an identical IP configuration and helps ensure that tests conducted in the test environment do not affect the production environment. Using the identical IP address, subnets, and other network configuration information helps to ensure the fidelity of the test environment. However, duplicating IP addresses might not always be the best option when applications do not rely on a hard-coded IP address. You might also pass some network traffic through the router from the production environment to reduce the need for replicating network services. For example, opening the ports for DHCP to pass through eliminates the need for a separate DHCP server in the test lab.
- Ensure that your test environment is at the same service pack and update level as your production environment. Before performing application mitigation testing, update your lab environment by applying service packs and updates or by refreshing the virtual or physical images of your production counterparts. Consider adding the test environment to the change-management process to simplify tracking the updates.
- Ensure that you perform all your application mitigation tests by using accounts that have similar permissions as the accounts in your production environment. For example,

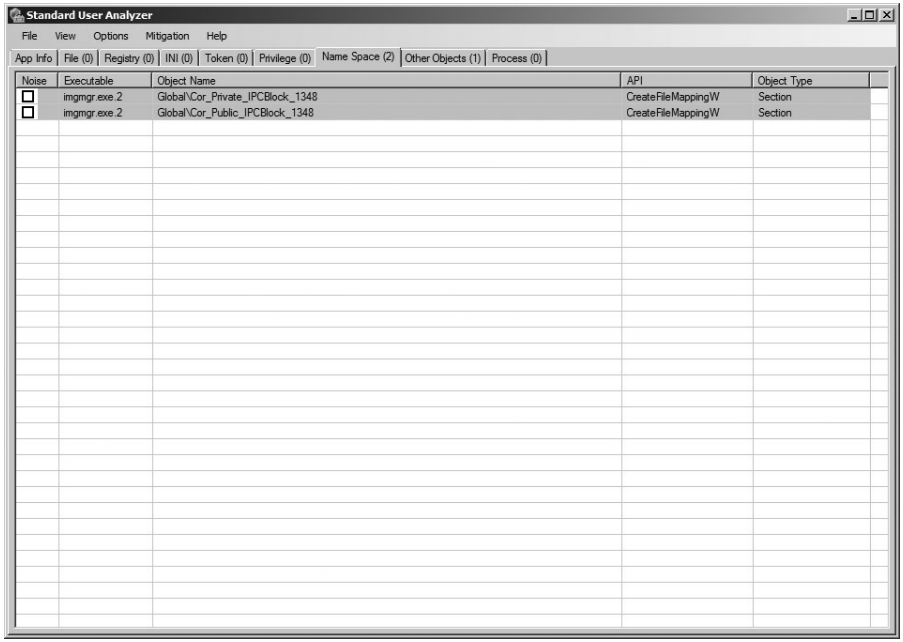
if your organization does not allow users to run as administrators on their local computers, ensure that similar permissions are granted to users in the test environment. This process ensures that you can determine potential security issues.

Using the Standard User Analyzer

The SUA tool enables you to test your applications and monitor API calls to detect potential compatibility issues resulting from the User Account Control (UAC) feature in Windows 7. UAC requires that all users (including members of the Administrator group) run as standard users until the application is deliberately elevated. However, not all applications can run properly as a standard user because of access violations. For more information about SUA, see the Standard User Analyzer Usage document (SUAnalyzer.rtf) in the \Microsoft Application Compatibility Toolkit 5\Standard User Analyzer folder, where Microsoft Application Compatibility Toolkit 5 is the folder in which you installed the toolkit.

To test an application using SUA, perform the following steps:

1. Click Start, point to All Programs, select Microsoft Application Compatibility Toolkit 5.5, choose Developer And Tester Tools, and then click Standard User Analyzer.
2. In the Target Application box, type the path and file name of the application that you want to test by using the SUA.
3. In the Parameters box, type any command-line options for the application.
4. Click Launch. Exercise each of the application's features and then close the application.
5. Click through each of the SUA tabs, reviewing the detected issues, as shown here.



Using the Compatibility Administrator

The Compatibility Administrator tool can help you to resolve many of your compatibility issues by enabling the creation and the installation of application mitigation packages (shims), which can include individual compatibility fixes, compatibility modes, and AppHelp messages. The flowchart in Figure 5-7 illustrates the steps required while using the Compatibility Administrator to create your compatibility fixes, compatibility modes, and AppHelp messages.

FIGURE 5-7 Using the Compatibility Administrator

The following terminology is used throughout the Compatibility Administrator:

- **Application fix** A small piece of code that intercepts API calls from applications, transforming them so that Windows 7 will provide the same product support for the application as previous versions of the operating system. This can mean anything from disabling a new feature in Windows 7 to emulating a particular behavior of a previous version of the Win32 API set.
- **Compatibility mode** A group of compatibility fixes that work together and are saved and deployed as a single unit.
- **AppHelp message** A blocking or non-blocking message that appears when a user starts an application that you know has major functionality issues with Windows 7.
- **Application mitigation package** The custom database (.sdb) file, which includes any compatibility fixes, compatibility modes, and AppHelp messages that you plan on deploying together as a group.

The Compatibility Administrator is the primary tool that most IT professionals will use when testing and with mitigation application compatibility issues. To start the Compatibility Administrator, click Start, point to All Programs, select Microsoft Application Compatibility Toolkit 5.5, and then choose Compatibility Administrator.

Creating a Custom Compatibility Database

You must apply compatibility fixes, compatibility modes, and AppHelp messages to an application and then store them in a custom database. After creating and applying the fixes, you can deploy the custom databases to your local computers to fix the known issues.

To create a custom database, perform the following steps:

1. On the Compatibility Administrator toolbar, click New.
2. The New Database(*n*) [Untitled_*n*] entry appears under the Custom Databases item in the left pane.

To save a custom database, perform the following steps:

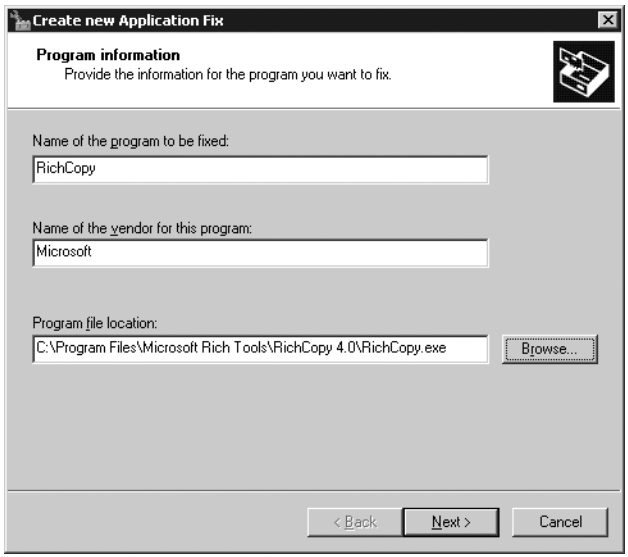
1. On the Compatibility Administrator toolbar, select Save from the File menu.
2. In the Database Name dialog box, type a name for the compatibility database and then click OK.
3. In the Save Database dialog box, type the path and file name of the new compatibility database and then click Save.

Creating a Compatibility Fix

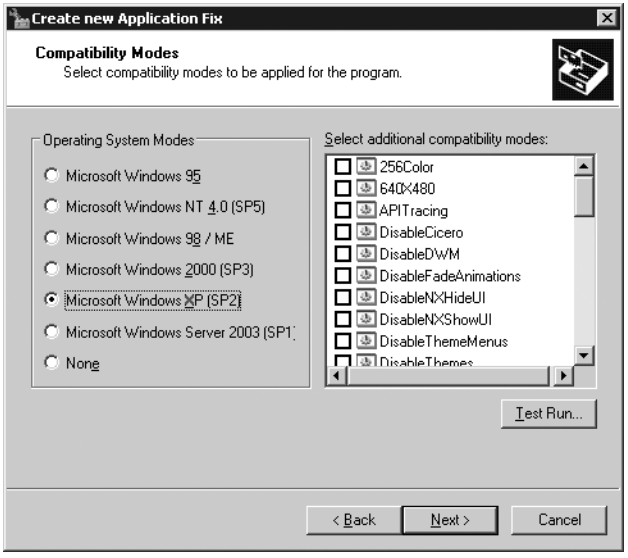
The Compatibility Administrator provides several compatibility fixes found to resolve many common application compatibility issues. You might find that a compatibility fix is not properly associated with an application because it was not found during previous testing by Microsoft or the ISV. If this is the case, you can use the Compatibility Administrator to associate the compatibility fix with the application. Compatibility fixes apply to a single application only. Therefore, you must create multiple fixes if you need to fix the same issue in multiple applications.

To create a new compatibility fix, perform the following steps:

1. In the left pane of the Compatibility Administrator, click the custom database to which you will apply the compatibility fix.
2. From the Database menu, select Create New and then select Application Fix.
3. Type the name of the application to which this compatibility fix applies, type the name of the application vendor, browse to the location of the application file (.exe) on your computer, as shown here, and then click Next.



4. Select an operating system to emulate, click any applicable compatibility modes to apply to your compatibility fix, and then click Next. If you know that an application worked properly with a previous operating system version, such as Window XP, you can apply the existing compatibility mode and then test the application to ensure that it works on Windows 7, as shown here.



5. Select any additional compatibility fixes to apply to your compatibility fix. Click Test Run to verify that your choices enable the application to work properly. When you are satisfied that the application works, click Next.
6. Click Auto-Generate to automatically select the files that the Compatibility Administrator recommends to represent your application and then click Finish. The Compatibility Administrator adds your compatibility modes, fixes, and matching information to your custom database, and the information appears in the right pane.

Creating a Compatibility Mode

The Compatibility Administrator provides several compatibility modes, which are groups of compatibility fixes found to resolve many common application compatibility issues. You can create custom compatibility modes that contain multiple fixes and then apply these compatibility modes to applications.

To create a compatibility mode, perform the following steps:

1. In the left pane of the Compatibility Administrator, click the custom database to which you will apply the compatibility mode.
2. From the Database menu, select Create New and then select Compatibility Mode.
3. Type the name of your custom compatibility mode in the Name Of The Compatibility Mode text box.

4. Select each of the available compatibility fixes to include in your custom compatibility mode and then click >. If you are unsure which compatibility modes to add, you can click Copy Mode to copy an existing compatibility mode.
5. Click OK after adding all of the applicable compatibility modes.

Creating AppHelp Messages

The Compatibility Administrator enables you to create the following blocking or non-blocking AppHelp messages, which appear when a user starts an application that you know has functionality issues with Windows 7:

- **Blocking AppHelp message (also called a HARDBLOCK)** Prevents the application from starting. Instead, it provides an error message dialog box that explains why the application did not start. In this situation, you can also define a specific URL where the user can download an updated driver or other fix to resolve the issue. When using a blocking AppHelp message, you must also define the file-matching information to identify the problematic version of the application and allow the corrected version to continue.
- **Non-blocking AppHelp message (also called a NOBLOCK)** Allows the application to start but also provides an error message dialog box to the user. The dialog box includes information about security issues, updates to the application, or changes to the location of network resources.

To create an AppHelp message, perform the following steps:

1. In the left pane of the Compatibility Administrator, click the custom database to which you will apply the AppHelp message.
2. On the Database menu, click Create New and then click AppHelp Message.
3. Type the name of the application to which this AppHelp message applies, type the name of the application vendor, browse to the location of the application file (.exe) on your computer, and then click Next.
4. Click Auto-Generate to automatically select the files the Compatibility Administrator recommends to represent your application and then click Next.
5. Select one of the following options for your AppHelp message:
   - Non-blocking Display a message and allow this program to run.
   - Blocking Display a message and do not allow this program to run.
6. Click Next. Type the URL and message text to appear when the user starts the application and then click Finish.

Deploying Application Mitigation Packages

Distribution of the custom compatibility databases (.sdb files) can be facilitated using a variety of methods such as logon scripts, System Center Configuration Manager 2007, injection into disk images, and so on. After the file is on the target system, the actual installation of the custom databases is done using a tool that ships with the operating system called Sdbinst.exe. After the file exists on the target computer, the custom database file must be installed (registered) before the operating system will identify the fixes present when starting the affected applications. (For example, the command line might be *sdbinst* C:\Windows\AppPatch\Myapp.sdb.) After the database file is registered on a computer, the compatibility information will be used any time the application is started. Table 5-4 describes the command-line options for Sdbinst.exe, which has the following syntax:

```
sdbinst [-?] [-q] filename.sdb [-u] [-g {guid}] [-n name]
```

TABLE 5-4 Sdbinst.exe Command-Line Options

OPTION	DESCRIPTION
-?	Displays Help text
-q	Runs quietly with no message boxes
filename.sdb	Specifies the file name of the database to install
-u	Uninstalls the database
-g {guid}	Specifies the globally unique identifier (GUID) of the database to uninstall
-n name	Specifies the name of the database to uninstall

The Sdbinst.exe command can be written into a machine logon script to automatically install the custom database from a share network location when the users log on to their computers. This process could even be accomplished as part of a custom job to be pushed out to the desktops via System Center Configuration Manager 2007 or another third-party management application. One of the best methods of distribution of these custom databases is to include them in your disk image. Installing them as part of the original image before adding the application that needs the fixes ensures that the application will run the first time the user needs it.

Summary

For many companies, issues with application compatibility prevent them from fully taking advantage of the technology that they are already paying for, such as Windows 7. Many of the issues are related to fear, uncertainty, and doubt as to whether the applications in their environment are compatible with Windows 7. You can help overcome these concerns by

creating an application inventory and then rationalizing it. In the case of application compatibility, knowledge helps companies overcome challenges.

This chapter described the primary tool that Microsoft provides for gaining this understanding and then putting it to use by creating a rationalized application portfolio as well as testing and mitigating compatibility issues. That tool is the ACT, and it is available as a free download from the Microsoft Download Center.

Additional Resources

These resources contain additional information and tools related to this chapter.

- Chapter 7, "Migrating User State Data," describes how to migrate users' documents and settings as part of a Windows 7 deployment.
- Chapter 8, "Deploying Applications," describes how to deploy applications as part of a Windows 7 deployment.
- "Application Compatibility" in the Windows Client TechCenter on Microsoft TechNet at *http://technet.microsoft.com/en-us/windows/aa905066.aspx*.
- "Microsoft Assessment and Planning (MAP) Toolkit 4.0" at *http://technet.microsoft.com/en-us/solutionaccelerators/dd537566.aspx*.
- "Microsoft Application Compatibility Toolkit (ACT) Version 5.5" at *http://technet.microsoft.com/en-us/library/cc722055.aspx* describes the ACT.
- "Microsoft Application Compatibility Toolkit 5.5" at *http://www.microsoft.com/downloads/details.aspx?FamilyId=24DA89E9-B581-47B0-B45E-492DD6DA2971&displaylang=en* contains the ACT download.
- Chris Jackson's blog at *http://blogs.msdn.com/cjacks/default.aspx*.
- Aaron Margosis's blog at *http://blogs.msdn.com/aaron_margosis/default.aspx*.

CHAPTER 6

Developing Disk Images

Beginning with Windows Vista and continuing with the Windows 7 operating system, the Windows operating system natively supports image-based deployment. In fact, with Windows Vista and later versions, only image-based deployment is supported—even when performing an unattended installation.

Image-based deployment is the most efficient method in high-volume deployment projects. Two factors make image-based deployment superior to other methods: time and cost. Creating a single image that you deploy to each computer is significantly faster than installing the operating system on each computer manually or using unattended installation. Image-based deployment significantly reduces costs by allowing you to better manage the computing environment: You're starting each computer with a known, standardized configuration. It also reduces deployment errors and support costs by using a standardized, stable, and repeatable process to develop and deploy operating systems.

Although the process of building and deploying images is not new, features first introduced with Windows Vista specifically address the challenges of the process. First, servicing images (adding device drivers, security updates, and so on) is easier because you don't have to rebuild and recapture an image every time you need to service it. Second, you can build hardware- and language-independent images that are not

dependent on the Hardware Abstraction Layer (HAL), which means you can build and maintain fewer images (and ideally, only one).

The Windows Automated Installation Kit (Windows AIK) 2.0 provides essential tools for building, servicing, and deploying Windows images. These tools include the Windows System Image Manager (Windows SIM) for creating Extensible Markup Language (XML) answer files for unattended installation; the Windows Preinstallation Environment (Windows PE) 3.0 for starting bare-metal destination computers; the new Deployment Image Servicing and Management (DISM) command-line tool for servicing images by adding drivers and packages; and the ImageX command-line tool for capturing images. The Windows AIK also includes extensive documentation about using these tools. You can download the Windows AIK 2.0, including the *Windows Automated Installation Kit User's Guide* and other documentation, from *http://www.microsoft.com/downloads*.

Although the Windows AIK 2.0 provides essential imaging tools, the Microsoft Deployment Toolkit (MDT) 2010 is a complete deployment framework that provides end-to-end guidance for planning, building, and deploying Windows 7 images. MDT 2010 takes full advantage of the Windows AIK 2.0 as well as other tools, such as the User State Migration Tool (USMT) 4.0 (which is now included in the Windows AIK 2.0), the Application Compatibility Toolkit (ACT) 5.5, Microsoft System Center Configuration Manager 2007 Service Pack 2 (SP2), and so on. Microsoft recommends that you use MDT 2010 to develop and deploy Windows 7 images, so this chapter focuses primarily on MDT 2010. For readers who prefer to use the Windows AIK directly, the *Windows Automated Installation Kit User's Guide* provides complete information about using the Windows AIK tools.

Getting Started

A typical difficulty with deployment efforts is the number of images that you must manage. In heterogeneous environments with diverse requirements, many organizations build numerous images. Adding new hardware, language packs, security updates, and drivers usually requires re-creating each disk image. Updating multiple images with a critical security update and testing each of them requires a lot of effort from you. Therefore, a major Microsoft design goal, beginning with Windows Vista and continuing in Windows 7, is to significantly reduce the number of images you must maintain and help you maintain those images more easily.

A key way that Windows 7 helps you reduce the number of images that you must build and maintain is by reducing dependencies on features that typically differ from one image to the next. These include languages, HALs, and device drivers. For example, unlike Windows XP and earlier versions of Windows, Windows Vista and later images are no longer tied to a HAL type. (Windows Vista and later versions support only Advanced Configuration and Power Interface (ACPI)–based computers.) The operating system can redetect the HAL when you apply it to each destination computer. Windows Vista and later versions are also language neutral, which means that all languages are operating system features, and adding or removing language packages is very easy. In addition to reducing dependencies, Microsoft modularized

Windows Vista and later versions to make customization and deployment easier, based the installation of Windows Vista and later versions on the file-based disk imaging format called Windows Imaging (WIM), and made other significant deployment features to the core operating system. (For more information, see Chapter 3, "Deployment Platform.")

MDT 2010 is a framework for these tools and features. Rather than using each tool and feature individually and using scripts to cobble them together, this chapter recommends that you develop and deploy Windows 7 images by using MDT 2010. To learn how to install MDT 2010, see Chapter 4, "Planning Deployment."

Prerequisite Skills

To build Windows 7 images—with or without MDT 2010—you should be familiar with the following tools and concepts:

- Unattended setup answer files (Unattend.xml)
- Windows AIK 2.0, including the following tools:
  - Windows SIM
  - DISM
  - ImageX
- Hardware device drivers and hardware-specific applications
- Microsoft Visual Basic Scripting Edition (VBScript)
- Disk imaging technologies and concepts, including Sysprep
- Windows PE 3.0

Lab Requirements

While developing and testing Windows 7 images, you will copy large volumes of files between the build server and destination computers. Because of these high-volume data transfers, you should establish a lab that is physically separate from the production network. Configure the development lab to represent the production environment as much as possible.

Lab Hardware

Ensure that the following hardware is available in the lab environment:

- **Network switches and cabling** 100 megabits per second (Mbps) or faster is recommended to accommodate the high volumes of data.
- **Keyboard Video Mouse (KVM) switches** It's useful to have the client computers connected to a KVM switch to minimize the floor space required to host the computers.
- **CD and DVD burner** A system should be available in the lab for creating CD-ROMs or DVD-ROMs.

- **Client computers** In the lab, duplicate any unique type of computer configuration found in the production environment to allow for testing each hardware configuration.
- **Build server** This computer (running Windows XP SP2, Windows Server 2003 SP1, or a newer version of Windows) can be a client- or server-class computer. The computer should have at least 50 gigabytes (GB) of disk space and backup equipment, such as a tape drive or a storage area network (SAN). Using Windows Server 2008 R2 is recommended because it already includes the MDT 2010 prerequisites.

Network Services

Make sure that the following network services are available in the lab environment:

- **A Windows domain for the computers to join and to host user accounts** This domain could be a Microsoft Windows 2000, Windows Server 2003, or Windows Server 2008 domain.
- **Dynamic Host Configuration Protocol (DHCP) services** DHCP provides Transmission Control Protocol/Internet Protocol (TCP/IP) addresses to client computers.
- **Domain Name System (DNS) services** DNS provides TCP/IP host name resolution to client and server computers.
- **Windows Internet Naming Service (WINS)** WINS provides NetBIOS name resolution to client and server computers. This service is optional but recommended.
- **Windows Deployment Services** Windows Deployment Services delivers Windows PE to computers that do not yet have an operating system. Windows Deployment Services servers require a Windows Server 2003 or later domain. For more information about Windows Deployment Services, see Chapter 10, "Configuring Windows Deployment Services."
- **Internet access** The lab (or a portion of the lab) should have access to the Internet for downloading software updates.

IMPORTANT **Windows protects users against malicious programs by warning them when they try to run a program that they have downloaded from the Internet. Users must acknowledge the warning to continue. This warning, however, prevents MDT 2010 from installing applications automatically during the build process. After verifying that the file is safe, disable the warning by right-clicking the file, clicking Properties, and then clicking Unblock. Windows does not display this warning when files are downloaded from sites listed in the Trusted Sites security zone, and Windows Server 2003 SP1 or later versions do not allow program downloads from untrusted sites.**

Installation Media

The installation media required for your environment include the following:

- Windows media (x86 and x64 editions) and product keys. Windows Vista is available on the volume-licensed media. MDT 2010 also supports retail media.

> ***NOTE*** **Earlier versions of Windows, such as Windows XP, supported slipstreaming. This process allowed you to integrate a service pack into the operating system source files. For example, you can integrate SP1 with the original release of Windows XP to create Windows XP SP1 media. Microsoft does not support slipstreaming service packs into Windows Vista or later versions. Instead, you can download fully integrated media from the volume licensing Web site, TechNet, or MSDN.**

- Any additional application media you plan to include in the images, such as the 2007 Microsoft Office system. The 2007 Office system is available on volume-licensed media; MDT 2010 also supports retail media.
- Any hardware-specific software, such as device drivers, CD-ROM burner software, and DVD-viewing software. Downloading all the known device drivers and hardware-specific applications early in the process saves time when developing and building Windows images.

Capturing Images Using Microsoft Deployment Toolkit

Capturing images using MDT 2010 is essentially a Lite Touch Installation (LTI) process, which ends by capturing a customized image of a master or *reference computer* that contains applications, language packs, and various other customizations needed. The following list of steps outlines the overall process for using MDT 2010 to create and capture operating system images as illustrated in Figure 6-1:

- **Create and configure a deployment share** Create a deployment share and then configure the share by adding operating system source files, applications, out-of-box device drivers, and packages to it as needed. The section titled "Creating and Configuring a Deployment Share" later in this chapter describes this step in detail.
- **Create and configure a task sequence** Create a task sequence that associates an operating system with an unattended setup answer file (Unattend.xml) to define a sequence of tasks to run during installation. The section titled "Creating Task Sequences" later in this chapter describes this step in detail.
- **Configure and update the deployment share** Configure and update the deployment share to update the MDT 2010 configuration files and generate a customized version of Windows PE you can use to boot your reference computer. The section titled "Updating the Deployment Share" later in this chapter describes this step in detail.

- **Run the Windows Deployment Wizard on the reference computer** Start your reference computer using the Windows PE image generated when you update your deployment share. Then run the Windows Deployment Wizard on the reference computer to install Windows from your deployment share. During the phase when gathering information, the Windows Deployment Wizard will prompt you to specify whether you want to capture a custom image of the reference computer that you can later deploy to target computers. The section titled "Capturing a Disk Image for LTI" later in this chapter describes this step in detail.
- **Add the custom image as an operating system source** After capturing the custom image of your reference computer, you add it to the deployment share as an operating system source. You can then deploy your custom image by using LTI as described in Chapter 12, "Deploying with Microsoft Deployment Toolkit."

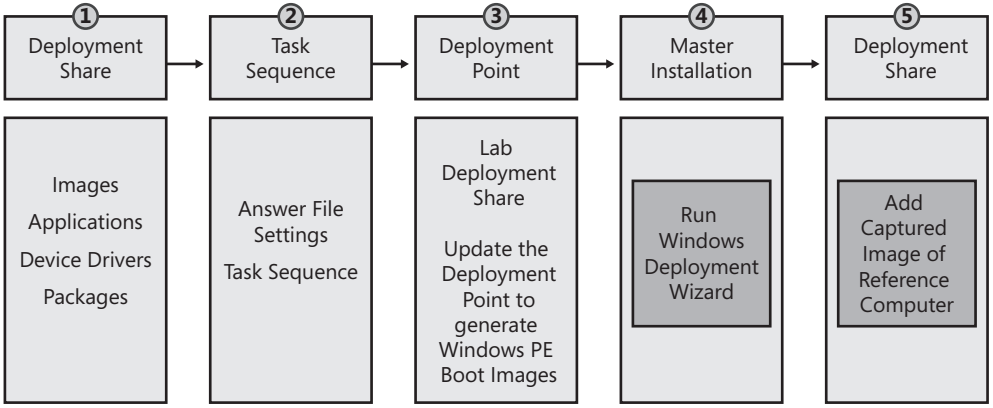


FIGURE 6-1 Image engineering with MDT

Creating and Configuring a Deployment Share

Before you can use MDT 2010 to deploy Windows 7, you must create a deployment share. A *deployment share* is a repository for the operating system images, language packs, applications, device drivers, and other software that will be deployed to your target computers. Deployment shares are new in MDT 2010 and consolidate two separate features found in MDT 2008:

- **Distribution share** Contains operating system source files, application source files, packages, and out-of-box drivers.
- **Deployment point** Contains files needed to connect to the distribution share and install a build from it.

By consolidating these two separate features into a single feature (the deployment share), MDT 2010 simplifies the deployment process. In addition, a deployment share does not have to be located on a specific computer—it can be stored on a local disk volume, a shared folder on the network, or anywhere in a stand-alone Distributed File System (DFS) namespace. (Windows PE cannot access domain-based DFS namespaces.)

NOTE **See the Microsoft Deployment Toolkit 2010 Documentation Library for information on how to upgrade to MDT 2010 from previous versions of MDT or Business Desktop Deployment (BDD). After you upgrade to MDT 2010, you must also upgrade any deployment points created using the previous version of MDT or BDD.**

To create a new deployment share, perform the following steps:

1. In the Deployment Workbench console tree, right-click Deployment Shares and then click New Deployment Share.
2. On the Path page, specify the path to the folder for your deployment share. The default path is <*drive*>*DeploymentShare,* where <*drive*> is the volume with the most available space. For best performance, you should specify a path to a separate physical disk that has sufficient free space to hold the operating system source files, application source files, packages, and out-of-box drivers you use for your deployments.
3. On the Share page, specify the share name for the deployment share. By default, this will be a hidden share named DeploymentShare$.
4. On the Descriptive Name page, specify a descriptive name for the deployment share. By default, this will be MDT Deployment Share.
5. On the Allow Image Capture page, leave the Ask If An Image Should Be Captured option selected so you will be able to capture an image of your reference computer.
6. On the Allow Admin Password page, choose whether the user will be prompted to set the local Administrator password during installation.
7. On the Allow Product Key page, choose whether the user will be prompted to enter a product key during installation.
8. Finish the remaining steps of the wizard.

Once your deployment share has been created, you can view the hierarchy of folders under it in the Deployment Workbench (Figure 6-2).

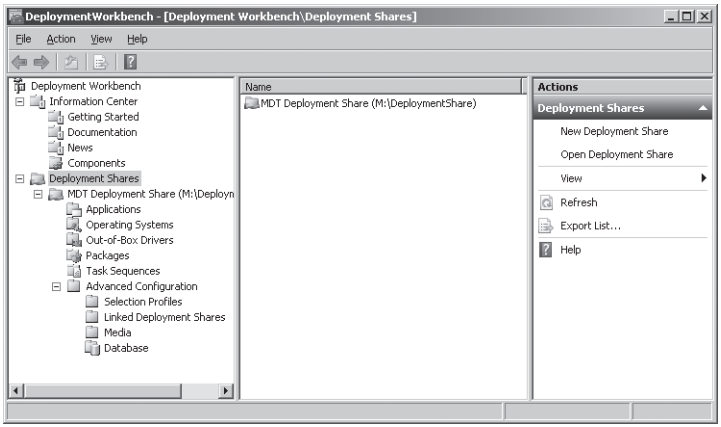


FIGURE 6-2 The folder structure of a deployment share in the Deployment Workbench

> ***NOTE*** **The default view in Deployment Workbench includes the action pane. The action pane often gets in the way of viewing the entire details pane. You can remove the action pane by authoring the management console. To author the console, run C:\Program Files \Microsoft Deployment Toolkit\Bin\DeploymentWorkbench.msc /a. Click View, click Customize, clear the Action Pane check box, and then click OK. Save your changes by clicking File and then clicking Save on the main menu. When prompted whether you want to display a single window interface, click Yes.**

After creating a deployment share, you can configure it in the following ways (at minimum, you must add the Windows 7 source files to deploy Windows 7):

- Add, remove, and configure operating systems.
- Add, remove, and configure applications.
- Add, remove, and configure operating system packages, including updates and language packs.
- Add, remove, and configure out-of-box device drivers.

When you add operating systems, applications, operating system packages, and out-of-box device drivers to a deployment share, Deployment Workbench stores the source files in the deployment share folder specified when you create the deployment share (see Figure 6-3). You will associate these source files and other files with task sequences later in the development process.

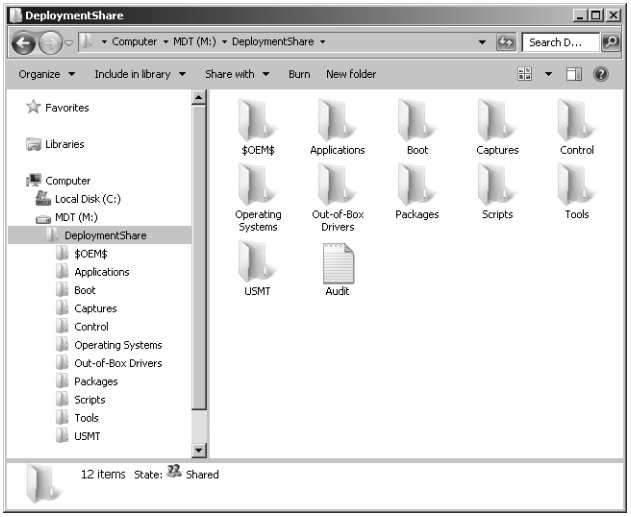


FIGURE 6-3 The folder structure of a deployment share in the file system

In the distribution share's Control folder, Deployment Workbench stores metadata about operating systems, applications, operating system packages, and out-of-box device drivers in the following files:

- **Applications.xml** Contains metadata about applications in the distribution share
- **Drivers.xml** Contains metadata about device drivers in the distribution share
- **OperatingSystems.xml** Contains metadata about operating systems in the distribution share
- **Packages.xml** Contains metadata about operating system packages in the distribution share

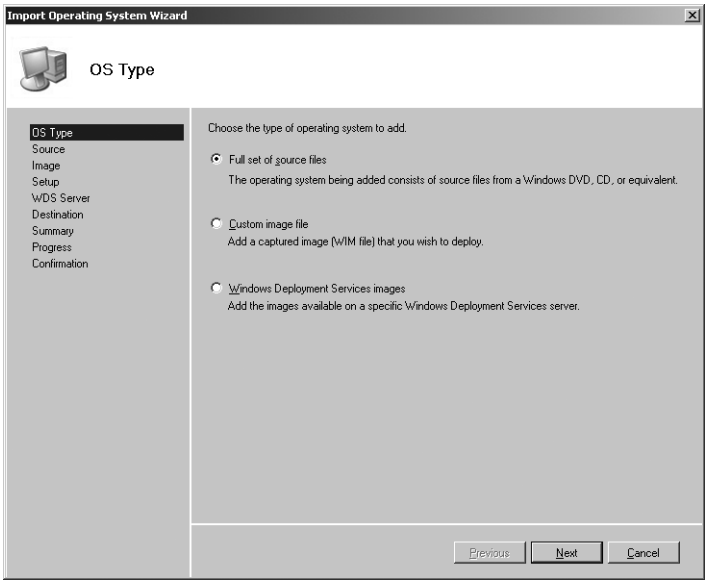
Adding Operating Systems

Windows 7 editions are in a single image file, Install.wim, which is in the \Sources folder on the distribution media. For more information about the Windows 7 distribution media and Install.wim, see the Windows AIK 2.0 documentation. To build images based on Windows 7, you must add the Windows 7 media to the MDT 2010 deployment share. Deployment shares must contain at a minimum the Windows 7 source files.

As well as adding Windows 7 media to the deployment share, you can add Windows 7 images that already exist in Windows Deployment Services. MDT 2010 will not copy these files to the deployment share. Instead, MDT 2010 uses the files from their original location during deployment. There is a requirement for doing this.

To add Windows 7 to a deployment share, perform the following steps:

1. In the Deployment Workbench console tree, right-click the Operating Systems folder (or a subfolder you created under this folder) in your deployment share and select Import Operating System to start the Import Operating System Wizard.
2. On the OS Type page, shown here, select Full Set Of Source Files. This option copies the entire set of operating system source files from the distribution media or folder containing the distribution media. Optionally, you can add operating system images from a specific Windows Deployment Services server by selecting Windows Deployment Services Images. You can also click Custom Image File to add a custom image, created by using the Windows Deployment Wizard. For more information about creating a custom image, see the section titled "Capturing a Disk Image for LTI" later in this chapter.



3. On the Source page, type the path containing the operating system source files you're adding to the deployment share, or click Browse to select the path. If you stage (pre-copy the source files to the local computer) the operating system files on the same partition as the deployment share, you can select Move The Files To The Deployment Share Instead Of Copying Them to speed the process.
4. On the Destination page, type the name of the operating system folder to create in the deployment share. You can accept the default name, which Deployment Workbench derives from the source files, or use a name that describes the operating system version and edition. For example, you can use Windows 7 Enterprise and Windows 7 Professional to distinguish between the different operating system editions of Windows 7. Deployment Workbench uses the default name to create a folder for the operating system in the deployment share's Operating Systems folder.
5. Finish the wizard.

The copy process can take several minutes to complete; the move process is much faster. After you add an operating system to the deployment share, it appears in the details pane when Operating Systems is selected in the console tree. Also, the operating system appears in the deployment share in Operating Systems*subfolder[\subfolder]* (shown in Figure 6-4), where *subfolder[\subfolder]* is the destination specified when adding the operating system.

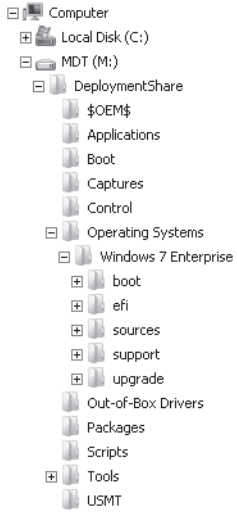


FIGURE 6-4 Operating Systems in the deployment share

To remove Windows 7 from the deployment share, perform the following steps:

1. In the Deployment Workbench console tree, click Operating Systems.
2. In the details pane, right-click the operating system you want to remove and then click Delete.

NOTE **When an operating system is deleted from Deployment Workbench, Deployment Workbench also removes it from the Operating Systems folder in the deployment share. In other words, removing an operating system from Deployment Workbench also removes it from the file system.**

Adding Applications

You must add each application to the deployment share that you intend to deploy by using MDT 2010. Deployment Workbench gives you the option to copy the application source files directly into the deployment share or to just add a reference to the application source files to the deployment share and leave them in their original location. Generally, if the network location containing the application source files will not be available during deployment, you should copy the application source files to the deployment share.

In addition to specifying how to add application source files to the deployment share, you can specify the command line for installing the application, dependencies between applications, and other settings for each application. After adding an application to the deployment share, you can install it at one of two points in the process:

- **During the Windows Deployment Wizard** During the interview, the Windows Deployment Wizard prompts the user with a list of applications that are available for installation. The user can then choose which applications to install. You can configure the applications that the Windows Deployment Wizard installs by using the MDT 2010 database and then skip the application installation pages of the wizard—automating application installation without requiring user intervention. For more information about using the MDT 2010 database, see Chapter 12.
- **During the task sequence** Application installations added to the task sequence—the sequence of tasks that occur during installation to prepare, install, and configure the build on the destination computer—occur when the Windows Deployment Wizard executes the task sequence on the destination computer. This is fully automated.

Chapter 8, "Deploying Applications," describes how to plan for and develop automated application installation. Chapter 8 describes differences between core applications, which are common to every desktop in the organization, and supplemental applications, which are not. You deploy each type of application differently depending on the strategy you choose for application deployment. The strategies are as follows:

- **Thick image** You install applications to the build that you're using to create disk images. You can install applications by using the Windows Deployment Wizard or by adding applications to the task sequence.
- **Thin image** Application deployment usually occurs outside of operating system deployment, typically using a systems management infrastructure such as System Center Configuration Manager 2007 SP2.
- **Hybrid image** You install applications to the build you're deploying to destination computers (most likely a custom image) and possibly install additional applications using a systems management infrastructure. You can install the applications by using the Windows Deployment Wizard or by adding them to the task sequence.

WARNING **Do not allow an application to restart the computer. The Windows Deployment Wizard must control reboots or the task sequence will fail. See the section titled "Installation Reboots" later in this chapter for more information about configuring reboots.**

To add an application to the deployment share, perform the following steps:

1. In the Deployment Workbench console tree, right-click the Applications folder (or a subfolder you created under this folder) in your deployment share and select New Application to start the New Application Wizard.

2. On the Application Type page, do one of the following:
   - Select Application With Source Files to copy the application source files to the deployment share. During deployment, the Windows Deployment Wizard installs the application from the deployment share.
   - Select Application Without Source Files Or Elsewhere On The Network. Choosing this option does not copy the application source files to the deployment share. During deployment, the Windows Deployment Wizard installs the application from another location on the network. You also choose this option to run a command that requires no application source files.
   - Select Application Bundle. Choosing this option does not add an application to the deployment share. Instead, it creates a placeholder to which you can associate dependencies. Then, by installing the placeholder application (the bundle), you also install its dependencies.
3. On the Details page, shown here, provide the information described in Table 6-1.

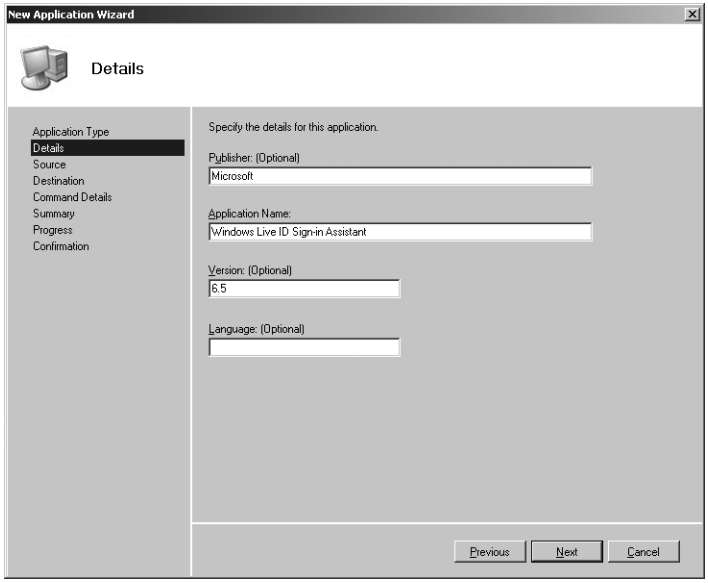


TABLE 6-1 The Specify The Details For This Application Page

IN THIS LOCATION	PROVIDE THIS INFORMATION
Publisher box	Name of the application's publisher
Application Name box	Name of the application
Version box	Version label for the application
Languages box	Languages that the application supports

4. On the Source page, type the path of the folder containing the application to be added or click Browse to open it. If you choose to copy the application source files to the deployment share, Deployment Workbench copies everything in this folder to the deployment share; otherwise, it adds this path to the application's metadata as the application's installation path. If the application source files are staged on the local hard disk, you can select Move The Files To The Distribution Share Instead Of Copying Them to move them quickly to the deployment share instead of copying them.
5. On the Destination page, type the name of the folder to create for the application within the deployment share's Applications folder. The default value is the publisher, application name, and version label concatenated.

> ***WARNING*** **Make sure that the destination specified on the Specify The Destination page is unique. Otherwise, during an LTI deployment, the Windows Deployment Wizard will display multiple applications having the same name but installing different applications. If necessary, change the name on the Destination page to ensure that it is unique.**

6. On the Command Details page, type the command to use to install the application silently. For example, type **msiexec /qb /i *app_name*.msi**. The command is relative to the working directory specified in the Working Directory box. For help finding the appropriate command to automate the installation of various applications, see Chapter 8.
7. Finish the wizard.

After you add an application to the deployment share, it appears in the details pane when the Applications folder (or in a subfolder of this folder) is selected in the console tree. It also appears in the deployment share in Applications*subfolder[\subfolder]*, where *subfolder[\subfolder]* is the destination specified when adding the application.

To edit an application in the deployment share, perform the following steps:

1. In the Deployment Workbench console tree, select the Applications folder (or a subfolder) in deployment share.
2. In the details pane, right-click the application and then click Properties.
3. On the General and Details tabs, edit the application information.

To provide an uninstall registry key name, perform the following steps:

1. In the Deployment Workbench console tree, select the Applications folder (or a subfolder) in your deployment share.
2. In the details pane, right-click the application and then click Properties.
3. On the Details tab, type the uninstall registry key name in the Uninstall Registry Key Name box.

The Windows Deployment Wizard uses the uninstall registry key name to determine whether an application is already installed on the destination computer. This is a subkey of

HKLM\Software\Microsoft\Windows\CurrentVersion\Uninstall. If the Windows Deployment Wizard detects the presence of this key, it assumes that the application is already installed and skips the installation of that application and any dependencies. In the Uninstall Registry Key Name box, type the name of the subkey—not the entire path.

To disable an application, perform the following steps:

1. In the Deployment Workbench console tree, select the Applications folder (or a subfolder) in your deployment share.
2. In the details pane, right-click the application you want to disable and then click Properties.
3. Click the General tab and clear the Enable This Application check box.

If you add an application that you intend to install during the task sequence, disable the application by clearing the Enable This Application check box. The application will still install during the task sequence, but the user will not see it in the applications list.

To remove an application from the deployment share, perform the following steps:

1. In the Deployment Workbench console tree, select the Applications folder (or a subfolder) in your deployment share.
2. In the details pane, right-click the application you want to remove and then click Delete.

When you delete an application from Deployment Workbench, it is also removed from the Applications folder in the deployment share. In other words, removing an application from Deployment Workbench also removes it from the file system.

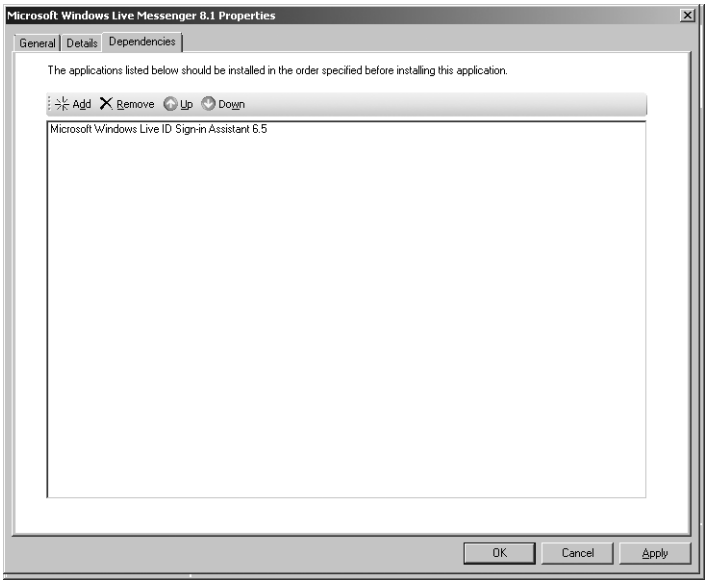
Specifying Application Dependencies

Using Deployment Workbench, you can specify dependencies between applications. For example, if application A is dependent on application B, Deployment Workbench will ensure that application B is installed before installing application A.

To create a dependency between two applications, perform the following steps:

1. In the Deployment Workbench console tree, select the Applications folder (or a subfolder) in your deployment share.
2. In the details pane, right-click the application that has a dependency and then click Properties.
3. Click the Dependencies tab, as shown here, perform any of the following actions, and then click OK:
   - To add an application to the dependencies list, click Add, and then select an application. Deployment Workbench displays only those applications that have already been added to the deployment share.
   - To remove an application from the dependencies list, select an application from the list, and then click Remove.

- To reorder the applications in the dependencies list, select an application in the list and then click Up or Down. The Windows Deployment Wizard installs the dependent applications in the order specified in the dependencies list.



Installation Reboots

Do not allow an application to restart the computer. The Windows Deployment Wizard must control reboots, or the task sequence will fail. For example, you can use REBOOT=REALLYSUPPRESS to prevent some Windows Installer–based applications from restarting. You can cause the Windows Deployment Wizard to restart the computer after installing an application by selecting the Reboot The Computer After Installing This Application check box on the Details tab of the *app_name* Properties, where *app_name* is the name of the application.

To restart the computer after installing an application, perform the following steps:

1. In the Deployment Workbench console tree, select the Applications folder (or a subfolder) in your deployment share.
2. In the details pane, right-click the application for which the Windows Deployment Wizard must restart the computer after installation and then click Properties.
3. Click the Details tab and select the Reboot The Computer After Installing This Application check box. Selecting this check box causes the Windows Deployment Wizard to restart the computer after installing the application and then continue with the next step in the task sequence.

DIRECT FROM THE SOURCE

Reboots in MDT 2010

Michael Niehaus, Lead Developer for Microsoft Deployment Toolkit
Solution Accelerator Team

When a user first logs on to the computer, he can run commands in different ways. One way is to add *RunSynchronous* to the <Microsoft-Windows-Setup> child element *FirstLogonCommands* during the oobeSystem pass.

MDT 2010 doesn't use *RunSynchronous* because it needs to support more complex installation scenarios. For example, an MDT 2010 installation needs to support reboots between application installations, and *RunSynchronous* doesn't support reboot-and-pick-up-where-it-left-off. Instead, MDT 2010 adds a command to *RunSynchronous* to initially start the task sequence. Then, if the task sequence needs to restart the computer, it adds a shortcut to the StartUp group, which continues the task sequence after the computer restarts.

Adding Packages

Packages include operating system updates and language packs. The Windows Deployment Wizard can automatically install operating system updates during deployment. Users can also choose which language packs to install during LTI deployment. The following sections include more information about updates and languages.

To add a package to the deployment share, perform the following steps:

1. In Deployment Workbench, right-click the Packages folder (or a subfolder you created under this folder) in your deployment share and select Import OS Packages to start the Import Package Wizard.
2. On the Specify Directory page, type the path containing the package files you want to add to the deployment share, or click Browse to open it, and then click Finish. Deployment Workbench adds all the packages it finds in the folder and all its subfolders.
3. Finish the wizard. Deployment Workbench adds all the packages it finds in the folder and its subfolders.

After you add a package to the deployment share, it appears in the details pane when the Packages folder (or a subfolder of this folder) is selected in the console tree. It also appears in the deployment share in Packages*subfolder[\subfolder]*, where *subfolder[\subfolder]* is the destination specified when adding the package.

To disable a package and prevent its installation, perform the following steps:

1. In the Deployment Workbench console tree, select the Packages folder (or a subfolder) in your deployment share.
2. In the details pane, right-click the package you want to disable and then click Properties.
3. Click the General tab and clear the Enable (Approve) This Package check box to disable the package.

To remove a package from the deployment share, perform the following steps:

1. In the Deployment Workbench console tree, select the Packages folder (or a subfolder) in your deployment share.
2. In the details pane, right-click the package you want to remove and then click Delete.

When a package is deleted from Deployment Workbench, it is also removed from the Packages folder in the deployment share. In other words, removing a package from Deployment Workbench also removes it from the file system.

Adding Updates

Operating system updates are distributed as Microsoft Standalone Update (.msu) files. For more information about .msu files, see Chapter 23, "Managing Software Updates."

When you are developing an image, take care to ensure that all critical security updates are included in the image so that computers deployed with the image are as up to date as possible. Table 6-2 describes different approaches to performing these updates. (When you use MDT 2010, the first method is recommended.)

TABLE 6-2 Updating Windows 7 Images

METHOD	BENEFITS	DRAWBACKS
Download the security updates from the Microsoft Web site and then install them as part of the image build process. You can search for updates in the Knowledge Base and on the Download Center.	The process is very easy to perform; you can install updates simply by adding them to the deployment share.	The process can be time consuming.
Use Windows Server Update Services (WSUS) or System Center Configuration Manager 2007 SP2 to install the security update post-deployment.	The process is easy to perform and picks up new updates as soon as they are approved.	The image is vulnerable before the updates are installed and the computer is restarted, providing an opportunity for exploitation; the application process can also be time consuming.

METHOD	BENEFITS	DRAWBACKS
		Depending on the System Center Configuration Manager 2007 SP2 server configuration, it may take an hour or more before all updates are applied; having the System Center Configuration Manager 2007 SP2 client included in the image and communicating with a specific site may result in all computers built from the image communicating with only that site.
Download the security updates from the Microsoft Web site and then integrate them into the Windows installation source before beginning the unattended build process.	The image is protected at all times from known security exploits, and the image build process completes faster because all security updates are installed before building the image.	Integrating the security updates takes some effort. It may not be obvious which updates you can integrate; you will need to install some as part of the unattended build process.

NOTE **Download the required Windows security updates from the Microsoft Knowledge Base or Download Center. You can also download updates from the Microsoft Update Catalog at *http://catalog.update.microsoft.com/v7/site/*.**

Adding Language Packs

Language packs make possible a multilingual Windows environment. Windows 7 is language neutral; all language and locale resources are added to Windows 7 through language packs (Lp.cab files). By adding one or more language packs to Windows 7, you can enable those languages when installing the operating system. As a result, you can deploy the same Windows 7 image to regions with different language and locale settings, reducing development and deployment time.

The following resources provide additional information about language packs in Windows Vista:

- Chapter 12 includes instructions on installing language packs during deployment.

- The *Microsoft Deployment Toolkit Reference* in the MDT 2010 documentation lists the properties that you can configure to install language packs automatically.
- The topic, "Understanding Multilingual Deployments," in the *Windows Automated Installation Kit User's Guide for Windows 7* includes more information about Windows Vista language packs.

Adding Out-of-Box Drivers

Depending on the type of computer in the environment and the hardware it contains, you require software from the hardware vendors to make computers in the production environment fully functional. Some of this software may be provided on a CD-ROM or DVD-ROM by the hardware manufacturer; other software must be downloaded from the vendor's Web site.

Deployment Workbench makes adding device drivers to the deployment share an easy process. You simply specify a folder containing one or more device drivers, and Deployment Workbench copies them to the deployment share and organizes them into folders as appropriate. However, you must make sure that you've extracted device drivers from any compressed files containing them. In other words, Deployment Workbench looks for each device driver's .inf file and any related files.

In MDT 2008 you could create driver groups to group together device drivers. You could then associate a driver group with a task sequence. In MDT 2010, you can no longer create driver groups. Instead, you can now create subfolders under the Out-Of-Box Drivers folder in your distribution share. You can import different drivers into different subfolders and then associate each subfolder with a task sequence.

NOTE **Windows Deployment Services in Windows Server 2008 R2 also includes new features that make it simpler to ensure that the appropriate drivers are available during a deployment. You can add driver packages to a Windows Deployment Services server and deploy these driver packages to different client computers based on filtering criteria. You can also add boot-critical driver packages to boot images (supported for Windows 7 and Windows Server 2008 R2 images only). For more information on this topic, see Chapter 10.**

To add device drivers to the deployment share, perform the following steps:

1. In Deployment Workbench, right-click the Out-Of-Box Drivers folder (or a subfolder you created under this folder) in your deployment share and select Import Drivers to start the Import Driver Wizard.
2. On the Specify Directory page, type the path containing the device drivers you want to add to the deployment share or click Browse to open it.
3. If you want, select the Import Drivers Even If They Are Duplicates Of An Existing Driver check box. Choosing this option allows Deployment Workbench to import duplicate drivers, if they exist, but Microsoft recommends against this.

4. Finish the wizard. Deployment Workbench adds all the device drivers it finds in the folder and its subfolders.

After you add a device driver to the deployment share, it appears in the details pane when the Out-Of-Box Drivers folder (or a subfolder of this folder) is selected in the console tree. It also appears in the deployment share in Out-Of-Box Drivers*subfolder[\subfolder]*, where *subfolder[\subfolder]* is the destination specified when adding the driver.

To disable a device driver, perform the following steps:

1. In the Deployment Workbench console tree, click Out-Of-Box Drivers (or a subfolder) in your deployment share.
2. In the details pane, right-click the device driver you want to disable and then click Properties.
3. Click the General tab, clear the Enable This Driver check box, and then click OK.

To remove a device driver from the deployment share, perform the following steps:

1. In the Deployment Workbench console tree, click Out-Of-Box Drivers (or a subfolder) in your deployment share.
2. In the details pane, right-click the device driver you want to remove and then click Delete.

When a device driver is deleted from Deployment Workbench, it is also removed from the Out-Of-Box Drivers folder in the deployment share. In other words, removing a device driver from Deployment Workbench also removes it from the file system.

Creating Task Sequences

A task sequence binds operating system source files with the steps necessary to install them. A task sequence is associated with the following:

- **Operating system** Choose an operating system image to use for the build.
- **Unattended setup answer file (Unattend.xml)** Create an answer file that describes how to install and configure the operating system on the destination computer. For example, the answer file can contain a product key, organization name, and information necessary to join the computer to a domain. Generally, allow MDT 2010 to control the settings in Unattend.xml and use the MDT 2010 database to configure destination computers.

NOTE **This chapter assumes that you are configuring task sequences and deployment points for the purpose of capturing custom images. The settings you configure by using the instructions in this chapter are different than the settings you will configure when deploying images to production computers. For more information about those settings, see Chapter 12.**

To create a task sequence for image capture, perform the following steps:

1. In the Deployment Workbench console tree, right-click the Task Sequences folder (or a subfolder you created under this folder) in your deployment share and select New Task Sequence to start the New Task Sequence Wizard.
2. On the General Settings page, provide the information described in Table 6-3.

TABLE 6-3 The General Settings Page

IN THIS LOCATION	PROVIDE THIS INFORMATION
Task Sequence ID box	Unique ID for the task sequence. You cannot change this ID later, so decide on a naming scheme for task sequence IDs in advance.
Task Sequence Name box	Descriptive name for the task sequence. Users see this name during LTI.
Task Sequence Comments box	Additional information about the task sequence. Users see this description during LTI. Describe the build and what it installs in the image.

3. On the Select Template page, choose a template task sequence to use as a starting point. You can customize the template later. For the purpose of building images, choose the Standard Client Task Sequence template.
4. On the Select OS page, choose an operating system image to install with this task sequence. Only the operating system images previously added to your deployment point are visible.
5. On the Specify Product Key page, select one of the following:
   - Do Not Specify A Product Key At This Time.
   - Specify A Multiple Activation Key (MAK Key) For Activating This Operating System, and then type the product key in the Product Key box.
   - Specify The Product Key For This Operating System, and then type the product key in the Product Key box.

   For more information about volume activation and product keys in MDT 2010, see Chapter 11, "Using Volume Activation." Chapter 11 describes when a product key is necessary. Generally, customers deploying volume-licensed Windows 7 media to 25 or more computers should select the Do Not Use A Product Key When Installing option. Customers deploying volume-licensed Windows 7 media using Windows 7 Multiple Activation Keys (MAKs) should select the Specify A Multiple Activation Key (MAK Key) For Activating This Operating System option and then type a product key in the Product Key box. Customers deploying retail Windows 7 media should select the Specify The Product Key For This Operating System option and then type a product key in the Product Key box.

6. On the OS Settings page, provide the information described in Table 6-4 and then click OK. The values you provide on this page are irrelevant because you are creating a build for image capture, and you will change these values during production deployment.

TABLE 6-4 The OS Settings Page

IN THIS LOCATION	PROVIDE THIS INFORMATION
Full Name box	Owner name
Organization box	Name of the organization
Internet Explorer Home Page box	Uniform Resource Locator (URL) of the default Windows Internet Explorer home page, such as the URL of the organization's intranet home page

7. On the Admin Password page, select Do Not Specify An Administrator Password At This Time. Do not specify a local Administrator password for image task sequences so that you can specialize the password during deployment.
8. Finish the wizard.

After you create a task sequence in your deployment share, it appears in the details pane when the Task Sequences folder (or a subfolder of this folder) is selected in the console tree. It also appears in the deployment share in Task Sequences*subfolder[\subfolder]*, where *subfolder[\subfolder]* is the destination selected when creating the task sequence. Deployment Workbench stores metadata about each build in TaskSequences.xml, which is located in the deployment share's Control folder.

To disable a task sequence, perform the following steps:

1. In the Deployment Workbench console tree, click Task Sequences (or a subfolder) in your deployment share.
2. In the details pane, right-click the task sequence you want to disable and then click Properties.
3. On the General tab, clear the Enable This Task Sequence check box and then click OK. Alternatively, you can hide the task sequence by selecting the Hide This Task Sequence In The Deployment Wizard check box.

NOTE **Disabling a build prevents the Windows Deployment Wizard from displaying it in the list of builds from which a user can choose during an LTI deployment.**

To remove a task sequence, perform the following steps:

1. In the Deployment Workbench console tree, click Task Sequences (or a subfolder) in your deployment share.

2. In the details pane, right-click the task sequence you want to remove and then click Delete.

To edit the task sequence's answer file (Unattend.xml), perform the following steps:

1. In the Deployment Workbench console tree, click Task Sequences (or a subfolder) in your deployment share.
2. In the details pane, right-click the task sequence containing the answer file you want to edit, and then click Properties.
3. On the OS Info tab, click Edit Unattend.xml to open the build's answer file in Windows SIM.

For more information about using Windows SIM to edit Unattend.xml, see the topic "Windows System Image Manager Technical Reference" in the Windows AIK.

DIRECT FROM THE SOURCE

Reducing Image Count

Doug Davis, Lead Architect
Management Operations & Deployment, Microsoft Consulting Services

We put the 2007 Office system and a virus scanner on every image. That way, the customer can be productive regardless of the method we use to deploy other applications. Also, a lot of things just make sense to put in the image so that the user doesn't have to download them later. I can't think of a single customer who doesn't have Adobe Acrobat Reader.

The virtual private network (VPN) and dialer installation programs are in the image, but we don't install them. When we deploy the image, the task sequence checks Windows Management Instrumentation (WMI) to see whether it's a mobile device. If it's a mobile device, we then install the VPN and dialer software; otherwise, we delete the installation programs.

We also never use a product key. Instead, we use the Key Management Service to simplify our images and reduce key loss. Chapter 11 describes the Key Management Service.

Having a single image to deploy is very handy and works well. We encourage people to change an image only when they need new software. Whenever a new update or device driver is required, we just replicate that information and then inject it into the image rather than making a new image every month and replicating the image. If this is the approach you plan to take, image versioning is very important to track.

Editing a Task Sequence

In MDT 2010, the task sequence is a list of tasks to run during deployment. However, it's not a linear list of tasks like a batch script. The task sequence is organized into groups and specifies conditions, or filters, that can prevent tasks and entire groups from running in certain situations.

MDT 2010 uses a *Task Sequencer* to run the task sequence. The Task Sequencer runs the task sequence from top to bottom in the order specified. Each task in the sequence is a step, and steps can be organized into groups and subgroups. When you create a task sequence in Deployment Workbench, you can choose a task sequence template. A key feature of the task sequence is that it stores state data, or variables, on the destination computer. These variables persist, even across reboots. The Task Sequencer can then use these variables to test conditions and possibly filter tasks or groups. The Task Sequencer also can restart the computer and gracefully continue the task sequence where it left off. These are important characteristics when driving a deployment process from beginning to end.

Task sequences contain the following types of items:

- **Steps** Steps are commands that the Task Sequencer runs during the sequence, such as partitioning the disk, capturing user state, and installing the operating system. Within a task sequence, steps do the actual work. In the task sequence templates provided by MDT 2010, most steps are commands that run scripts.
- **Groups** The task sequence steps can be organized into groups, which are folders that can contain subgroups and steps. Groups can be nested as necessary. For example, the default task sequence puts steps in groups by phase and deployment type.

You can filter both steps and groups, including the groups and steps that they contain, based on conditions that you specify. Groups are especially useful for filtering because you can run an entire collection of steps based on a condition, such as the deployment phase or type of deployment.

To edit a task sequence, perform the following steps:

1. In the Deployment Workbench console tree, click Task Sequences (or a subfolder) in your deployment share.
2. In the details pane, right-click the task sequence you want to edit and then click Properties.
3. Click the Task Sequence tab, as shown here, edit the task sequence as described in Table 6-5, and then click OK. For more information about settings on the Properties and Options tabs, see the sections titled "Configuring Group and Task Properties" and "Configuring the Options Tab" later in this chapter.

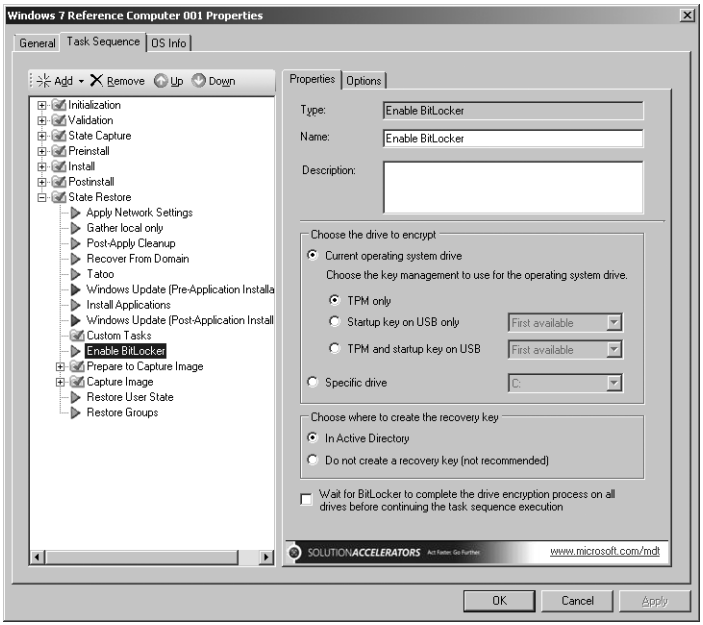


TABLE 6-5 Editing a Task Sequence

TO	USE THESE STEPS
Add a group	In the task sequence, select the item beneath which you want to create a new group, click Add, and then click New Group. Deployment Workbench creates and selects a new group called New Group.
Add a step	In the task sequence, select the item beneath which you want to create a new step and click Add. Then choose the type of step that you want to create by clicking General and then choosing one of the following (MDT 2010 supports more steps than those listed here, but they are already in the task sequence or are primarily for server deployment): ■ Run Command Line ■ Set Task Sequence Variable ■ Run Command Line As Deployment Workbench creates and selects a new step with a name relating to the type of step you're creating.
Add a reboot	In the task sequence, select the item beneath which you want to add a reboot, click Add, click General, and then click Restart Computer. Deployment Workbench creates and selects a new task that restarts the destination computer.

TO	USE THESE STEPS
Add an application	In the task sequence, select the item beneath which you want to add an application installation, click Add, click General, and then click Install Application. Then select the Install Application step you just added, and on the Properties tab, click Install A Single Application. Choose the application you want to install from the Application To Install list.

IMPORTANT **If you install antivirus software as part of the task sequence, be sure to carefully test how the antivirus software interacts with the deployment process before moving to a production environment. Antivirus software can prevent MDT 2010 from successfully deploying Windows 7 and applications. If necessary, you can always disable the antivirus software and then re-enable it at the end of the task sequence.**

To edit an item in a task sequence, select the item you want to work with and then edit the settings in the right pane.

NOTE **MDT 2010 includes a variety of special steps, such as the Enable BitLocker task or Install Operating System step, that you can configure. You change settings for these steps by selecting the step in the left pane and then configuring the step on the Properties tab. In general, the most interesting steps to configure are Validate (under Validation and under Preinstall\New Computer Only), Format and Partition Disk (under Preinstall\New Computer Only), Install Operating System (under Install), Apply Network Settings (under State Restore), and Enable BitLocker (under State Restore).**

To remove an item in a task sequence, select the item you want to work with and then click Remove. If a group is removed, Deployment Workbench removes the group and everything it contains, including subgroups and tasks.

To reorder an item in a task sequence, select the item you want to work with and then click Up or Down to change its position within the task sequence. During deployment, the Windows Deployment Wizard runs the tasks from top to bottom in the order specified.

Configuring Group and Task Properties

In the task sequence, every group and step has a Properties tab. Each group and step has a name and description that you can edit on the Properties tab. The Run Command Line and Run Command Line As steps also have a command line and a starting folder location that you can edit. Other steps have additional properties depending on the type of step. The following list describes what you see on the Properties tab:

- **Type** The Type box indicates the type of step. You cannot change the type.

- **Name** In the Name box, type a short, descriptive name for the group or step. During deployment, this name appears in the status window of the Task Sequencer.
- **Description** In the Description box, type a description of the group or step.
- **Command Line (Run Command Line and Run Command Line As tasks only)** In the Command Line box, type the command to run at this step in the task sequence. Include any command-line arguments. Environment variables are also permitted in command lines.
- **Start In (steps only)** In the Start In box, type the path in which to start the command. This path specifies the current working directory for the command. If you do not provide a path in this box, the paths in the Command Line box must be fully qualified or the command must be in the path.

Configuring the Options Tab

Groups and tasks have the following settings on the Options tab (shown in Figure 6-5):

- **Disable This Step** Select the Disable This Step check box to disable the step or group, including all groups and steps that it contains.
- **Success Codes (steps only)** List the return codes that indicate successful completion. The Windows Deployment Wizard determines whether a step completed successfully by comparing its return code to each code in the Success Codes box. If it finds a match, the step completed successfully. A success code of 0 usually represents successful completion. A success code of 3010 usually represents a successful completion with a reboot required. Thus, most of the steps in the templates that MDT 2010 provides list the success codes as 0 3010.
- **Continue On Error** If an error occurs in the current step, select the Continue On Error check box to continue with the next step in the task sequence. If you clear this check box, the Windows Deployment Wizard stops processing and displays an error message if the step or group does not complete successfully.

Additionally, on the Options tab, you can filter the group or steps based on conditions specified in the Conditions list. If the condition evaluates to true, the group or step runs. If the condition evaluates to false, the group (and all of the groups and steps that group contains) or step does not run. See the following sections for more information about conditions you can add to the Conditions list.

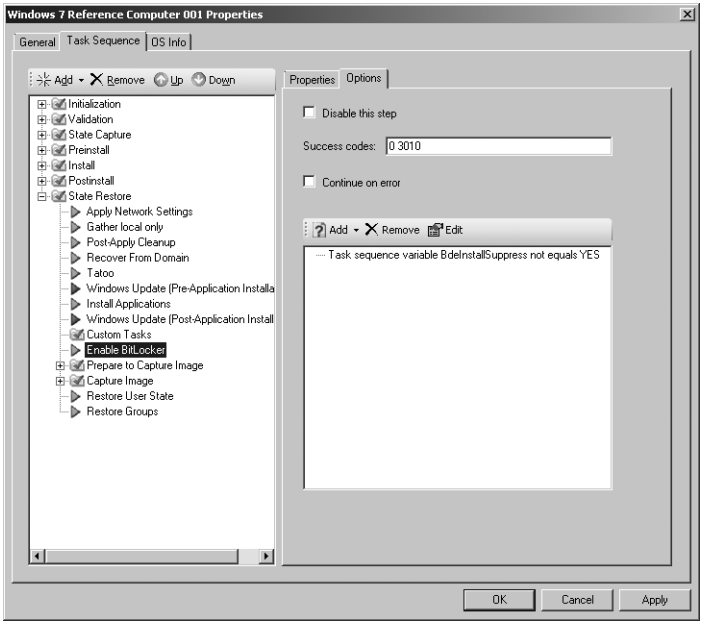


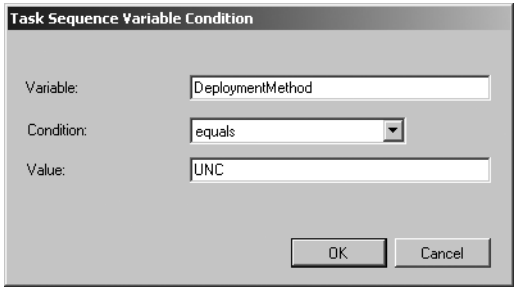
FIGURE 6-5 The Options tab

Task Sequence Variables

Task sequence variables allow you to compare a variable to a static value using a variety of conditions, such as equal, greater than, and less than. The Task Sequencer maintains numerous variables that you can use in these tests. For example, the Task Sequencer defines a variable called *DeploymentMethod* that indicates the method of deployment. One possible value of *DeploymentMethod* is *UNC*. For a complete list of variables that the Task Sequencer maintains, see the *Microsoft Deployment Toolkit Reference* in the MDT 2010 documentation.

To add a variable to an item's Conditions list, perform the following steps:

1. On the Options tab, click Add and then click Task Sequence Variable to display the Task Sequence Variable Condition dialog box, shown here.



2. In the Variable box, type the name of the variable you want to test.

3. From the Conditions list, choose one of the following conditions:
   - Exists
   - Equals
   - Not equals
   - Greater than
   - Greater than or equals
   - Less than
   - Less than or equals
4. In the Value box, type the static value you want to compare to the variable using the condition specified in the previous step.

if Statements

Use *if* statements to combine variables into bigger expressions. For example, create an *if* statement that evaluates to true only if all the conditions it contains are true (the same as a logical *AND*), or create an *if* statement that evaluates to true if any of the conditions it contains are true (the same as a logical *OR*).

To add an *if* statement to an item's Conditions list, perform the following steps:

1. On the Options tab, click Add and then click If Statement to display the If Statement Properties dialog box, shown here.

2. In the If Statement Properties dialog box, choose one of the following options and then click OK:
   - All conditions (*AND*)
   - Any conditions (*OR*)
   - None
3. From the Conditions list, select the *if* statement added in the previous step and then add task sequence variables to it as described in the previous section.

If you choose All Conditions, all variables added must evaluate to true for the group or step to run. If you choose Any Conditions, the group or task will run if any one of the variables added evaluates to true.

NOTE **You can nest *if* statements to create complex logic. If you are familiar with Boolean logic, represent Boolean expressions as *if* statements in the Conditions list.**

Operating System Versions

The Task Sequencer allows you to filter steps and groups based on the computer's current operating system. For example, you can choose to run a preinstallation step only if the destination computer is currently running Windows Vista SP1.

To add an operating system filter to an item's Conditions list, perform the following steps:

1. On the Options tab, click Add, and then click Operating System Version to display the Task Sequence OS Condition dialog box.
2. From the Architecture list, click either X86 or X64.
3. From the Operating System list, choose an operating system version and a service pack level.
4. From the Conditions list, choose one of the following conditions:
   - Equals
   - Not equals
   - Greater than
   - Greater than or equals
   - Less than
   - Less than or equals

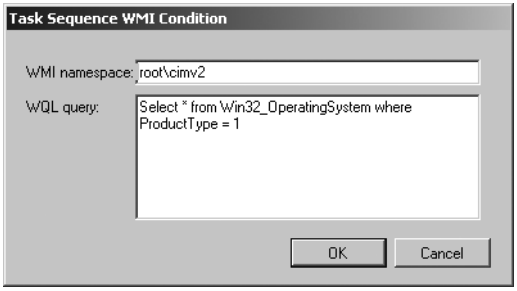
WMI Queries

The Task Sequencer allows you to filter steps and groups based on WMI queries. The WMI query must return a collection. If the collection is empty, the result evaluates to false. If the collection is not empty, the result evaluates to true. The following are some sample WMI queries you could use to filter steps in the task sequence:

- SELECT * FROM Win32_ComputerSystem WHERE Manufacturer = 'Dell Computer Corporation'. This is true only if WMI reports the computer's manufacturer as *Dell Computer Corporation*.
- SELECT * FROM Win32_OperatingSystem WHERE OSLanguage = '1033'. This is true only if WMI reports the operating system language as 1033.
- SELECT * FROM Win32_Service WHERE Name = 'WinMgmt'. This is true only if the WinMgmt service is available.
- SELECT * FROM Win32_Processor WHERE DeviceID = 'CPU0' AND Architecture = '0'. This is true only if the processor architecture is x86.
- SELECT * FROM Win32_Directory WHERE Name = 'D:\Somefolder'. This is true only if D:\Somefolder exists on the computer.

To add a WMI query to an item's Conditions list, perform the following steps:

1. On the Options tab, click Add and then click Query WMI to display the Task Sequence WMI Condition dialog box.
2. In the WMI Namespace box, type the WMI namespace in which to run the query, as shown here. The default namespace is *root\cimv2*.



3. In the WQL Query box, type the WMI query.

Updating the Deployment Share

The Windows AIK 2.0 comes with Windows PE 3.0, so no additional files are necessary to create Windows PE boot images for MDT 2010. When you update your deployment share in the Deployment Workbench, MDT 2010 automatically generates the following custom Windows PE images (here *platform* is x86 or x64):

- Lite Touch Windows PE (*platform*) .wim file
- LiteTouchPE_*platform*.iso

If you want, you can configure the deployment share to also generate the following Windows PE images:

- Generic Windows PE (*platform*)
- Generic_*platform*.iso

You don't need to manually customize Windows PE to add network interface card (NIC) device drivers to it. Deployment Workbench automatically adds the NIC device drivers that you add to the deployment share to the Windows PE boot images. You have the additional option of automatically adding video and system device drivers from the deployment share to the Windows PE boot images. You can also perform additional customizations of your Windows PE images. For example, you can customize the background bitmap, add additional directories, and increase the scratch space size from its default value of 32 megabytes (MB) up to a maximum of 512 MB if needed. To learn more about customizing Windows PE, see the *Windows Preinstallation Environment User's Guide for Windows 7* in the Windows AIK.

Updating a deployment share causes Deployment Workbench to update its configuration files, source files, and Windows PE images. Deployment Workbench updates the deployment share's files and generates the Windows PE boot images when you update the deployment share, not when you create it. Deployment Workbench stores these boot images in the deployment share's \Boot folder. After you have updated the deployment share and generated Windows PE images, you can add the .wim image file to Windows Deployment Services. If you want, you can burn the Windows PE .iso images to CD or DVD media by using third-party CD/DVD-burning software. Windows Deployment Services is the best way to start the Windows PE boot images on lab computers. Updating the boot images is faster than burning media, and booting destination computers is quicker. For more information, see Chapter 10.

NOTE **You must use the same platform edition of Windows PE to start computers for installing each platform edition of Windows. In other words, you must start destination computers using a x86 edition of Windows PE to install a x86 edition of Windows 7. Likewise, you must use a x64 edition of Windows PE to install a x64 edition of Windows 7. If you use mismatched editions, you might see errors indicating that the image is for a different type of computer. Deployment Workbench automatically chooses the correct platform edition of Windows PE to match the operating system you're deploying.**

To configure a deployment share for imaging in the lab, perform the following steps:

1. In the Deployment Workbench console tree, click Deployment Shares.
2. In the details pane, right-click the deployment share you want to configure and then click Properties.
3. Click the General tab and then choose the platforms that the deployment share supports. To indicate that the deployment share supports the x86 platform, select the x86 check box. To indicate that the deployment share supports the x64 platform, select the x64 check box. This option determines the platforms for which Deployment Workbench generates Windows PE boot images.
4. Click the Rules tab and then edit the deployment share's settings. These settings are located in CustomSettings.ini, which is located in the deployment share's Control folder. For more information about the settings that you can configure on this tab, see the *Microsoft Deployment Toolkit Reference* in MDT 2010.
5. Click the Windows PE Settings (*platform*) tab for each platform and edit the settings described in Table 6-6, as shown on the following page. Then, click OK.

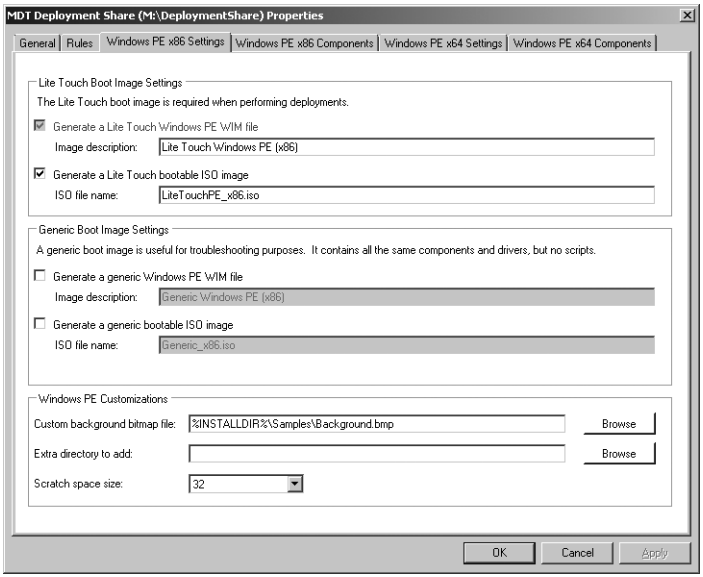


TABLE 6-6 Windows PE Settings Tab

AREA	SETTINGS
Images to Generate	**Generate A Lite Touch Windows PE WIM file** Select this option to generate a customized WIM file that you can use to perform LTI using Windows Deployment Services (this option is selected by default and cannot be cleared). **Generate A Lite Touch Bootable ISO Image** Select this option to generate a bootable customized Windows PE ISO image that you can use to perform LTI by starting your destination computers manually (this option is selected by default). **Generate A Generic Windows PE WIM file** Select this option to generate a generic WIM file that you can use to perform LTI using Windows Deployment Services. **Generate A Generic Bootable ISO Image** Select this option to generate a bootable generic Windows PE ISO image that you can use to perform LTI by starting your destination computers manually.
Windows PE Customizations	**Custom Background Bitmap File** Type the path and file name of a bitmap file to use as the Windows PE background. **Extra Directory To Add** Type the path of a folder containing extra files and subfolders to add to the Windows PE bootable images. **Scratch Space Size** Select the size of the scratch space for your Windows PE image. The available values are 32, 64, 128, 256, and 512 MB, with 32 being the default.

6. Click the Windows PE Components (*platform*) tab for each platform and edit the settings described in Table 6-7, as shown here, and then click OK.

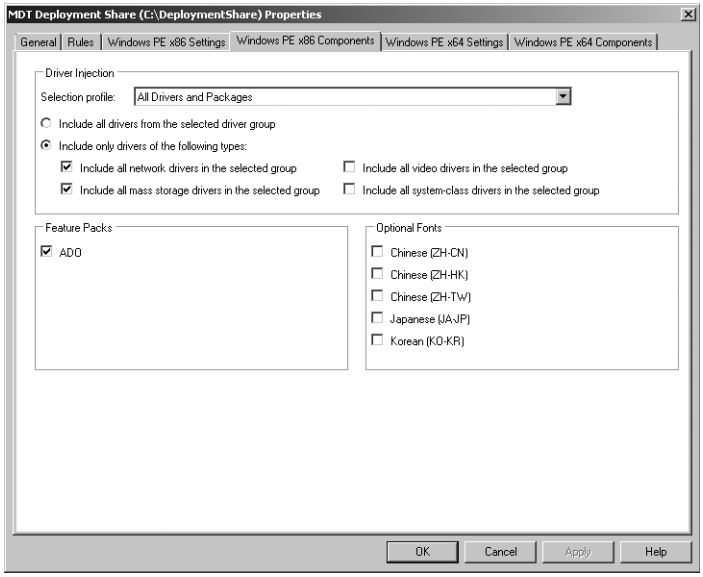


TABLE 6-7 Windows PE Components Tab

AREA	SETTINGS
Feature Packs	**ADO** Select this option to add the Microsoft ActiveX Data Objects (ADO) optional feature to the Windows PE bootable images.
Optional Fonts	Select the font support to add to the Windows PE boot images that Deployment Workbench generates. You must add these fonts when performing an LTI deployment of Windows Vista images when the setup files are Japanese, Korean, or Chinese. The Optional Fonts area provides the following options: ■ Chinese (ZH-CN) ■ Chinese (ZH-HK) ■ Chinese (ZH-TW) ■ Japanese (JA-JP) ■ Korean (KO-KR) Adding additional fonts to Windows PE boot images increases the size of the images. Add additional fonts only if necessary.

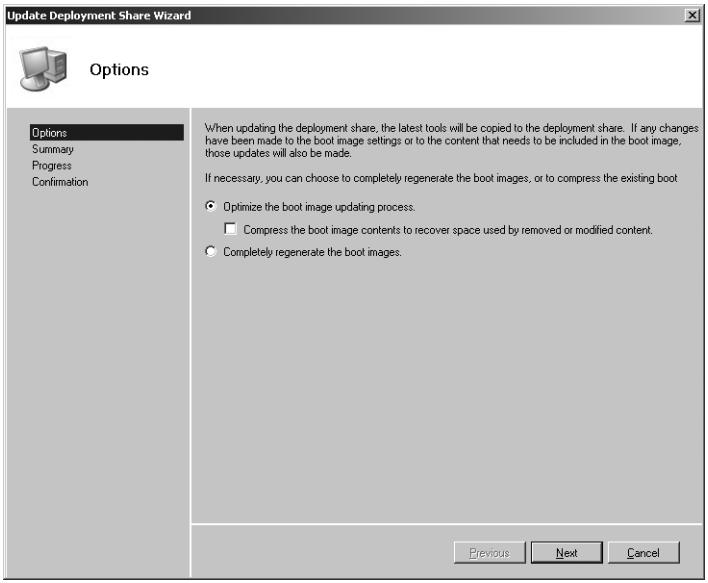
AREA	SETTINGS
Driver Injection	**Selection Profile** Use this list box to choose one of the following selection profiles to include the appropriate device drivers in your Windows PE images: ■ **Everything** Includes all folders from all nodes in Deployment Workbench. This selection profile includes all applications, operating systems, device drivers, operating system packages, and task sequences. ■ **All Drivers** Includes all folders from the Out-Of-Box Drivers node in Deployment Workbench. This selection profile includes all device drivers. ■ **All Drivers And Packages** Includes all folders from the Applications and Out-Of-Box Drivers nodes in Deployment Workbench. This selection profile includes all applications and device drivers. ■ **Nothing** Includes no folders in Deployment Workbench. This selection profile includes no items. ■ **Sample** A sample selection profile that illustrates how to select a subset of the items and include all folders from the Packages and Task Sequences nodes in Deployment Workbench. This selection profile includes all operating system packages and task sequences. ***NOTE*** **If you have created any custom selection profiles, these will also be available for selection here.** Selection profiles are new in MDT 2010 and allow you to select one or more folders in Deployment Workbench that contain one or more items in Deployment Workbench, including applications, device drivers, operating systems, operating system packages, and task sequences. For more information concerning selection profiles, see the topic "Managing Selection Profiles" in the MDT 2010 documentation. **Include All Drivers From The Selected Driver Group** Select this option if you want to include all the device drivers in the selection profile you specified in the Selection Profile list box. **Include Only Drivers Of The Following Types** Select this option to include only specific types of device drivers in the selection profile you specified in the Selection Profile list box. If you select this option, you can select one or more of the following:

AREA	SETTINGS
	■ **Include All Network Drivers In The Selected Group** Select this option to inject all the device drivers in the selection profile specified in the Selection profile list box. ■ **Include All Mass Storage Drivers In The Selected Group** Select this option to inject all mass storage drivers found in the deployment share into the Windows PE boot images. ■ **Include All Video Drivers In The Selected Group** Select this option to inject all video drivers found in the deployment share into the Windows PE boot images. ■ **Include All System-Class Drivers In The Selected Group** Select this option to inject all system drivers (such as motherboard drivers) in the deployment share into the Windows PE boot images.

After creating and configuring a deployment share in Deployment Workbench, you must update it to update the deployment share's configuration files and generate Windows PE boot images in the deployment share's \Boot folder. Deployment Workbench always generates .wim image files, which you can use to start destination computers using Windows Deployment Services. Choose to generate only the Windows PE bootable ISO images that are actually required. If you limit the number of images generated, the updating process is faster.

To update a deployment share, perform the following steps:

1. In the Deployment Workbench console tree, click Deployment Shares.
2. In the details pane, right-click the deployment share you want to configure and then click Update.
3. On the Options page of the Update Deployment Share Wizard, shown on the following page, select one of the following options:
   - **Optimize The Boot Image Updating Process** Select this option to update existing versions of the image files in the deployment share. Choosing this option reduces the amount of time required to update the boot images. If you select this option, you can also select Compress The Boot Image Contents To Recover Space Used By Removed Or Modified Content if desired. Selecting this suboption reduces the size of the boot images but may increase the time needed to generate the images.
   - **Completely Regenerate The Boot Images** Select this option to create a new version of all the image files in the deployment share. You can choose this option when you want to force the creation of new images. Note that this can take some time to complete.



4. Finish the wizard. Depending on how your deployment share is configured and the options you selected in the Update Deployment Share Wizard, generating Windows PE boot images may take some time to complete.

After the deployment share has been updated, Windows PE boot images and other files will be present in the \Boot folder of the deployment share (see Figure 6-6).

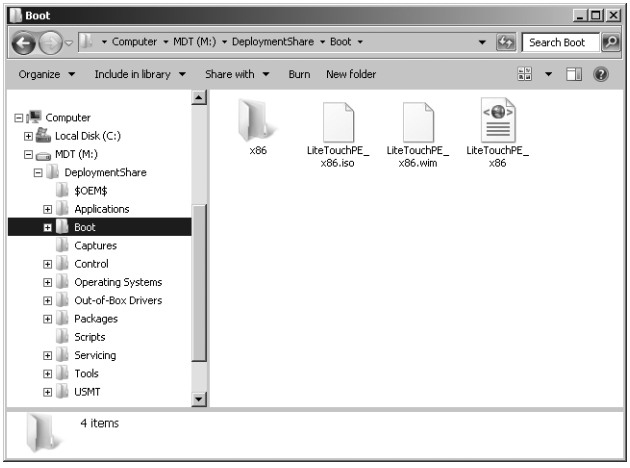


FIGURE 6-6 \Boot folder of updated deployment share showing Windows PE boot images that were generated

IMPORTANT **You *must* update your deployment share if you make changes to any the settings on the properties sheet of your deployment share. The Windows PE boot images will not contain your updated settings until you update your deployment share.**

Capturing a Disk Image for LTI

In MDT 2010, installing a build and capturing an image is essentially an LTI deployment that ends with the Windows Deployment Wizard capturing an image of the destination computer. When you create a deployment share, Deployment Workbench provides the option of prompting to capture an image (the Ask If An Image Should Be Captured check box). You must enable this option, as described in the section titled "Creating and Configuring a Deployment Share" earlier in this chapter.

Then, when you install the build on the destination lab computer, the Windows Deployment Wizard asks whether you want to capture an image after installation is complete. The wizard also allows you to specify a destination for the image. The default destination is the \Captures folder in the deployment share, and the default file name is task sequence.wim, where *task sequence* is the ID of the task sequence you installed.

To capture an image, start a lab computer using the Windows PE boot image generated by updating the deployment share. Start the Windows PE boot image in either of two ways. One way is to burn the .iso images to a DVD. This process is slow and tedious. These ISO image files reside in the \Boot folder of the deployment share. The other way is to add the LiteTouchPE_x86.wim or LiteTouchPE_x64.wim image files to the Boot Images item of a Windows Deployment Services server. The .wim image files are in the \Boot folder of the deployment share. For more information about installing and configuring Windows Deployment Services, see Chapter 10.

To capture an image using the Windows Deployment Wizard, perform the following steps:

1. Start the lab computer using the Windows PE boot image that you created in the section titled "Updating the Deployment Share" earlier in this chapter. You can start this boot image by burning the .iso file to CD or DVD media or by adding the .wim file to Windows Deployment Services. For more information about Windows Deployment Services, see Chapter 10.
2. In the Welcome Windows Deployment dialog box, click Run The Deployment Wizard To Install A New Operating System.
3. In the User Credentials dialog box, type the credentials necessary to connect to the deployment share (user name, domain, and password) and then click OK. The Windows Deployment Wizard starts automatically. To capture an image using the Windows Deployment Wizard, you must use an account that has Read and Write access to the deployment share, such as an account that is a member of the local Administrators group on the computer that contains the deployment share.

4. On the Select A Task Sequence To Execute On This Computer page, choose a task sequence to run from the list of available task sequences and then click Next.
5. On the Configure The Computer Name page, type a computer name or accept the default and then click Next. The default, randomly generated computer name is reasonable because the computer name will change during deployment to the production environment.
6. On the Join The Computer To A Domain Or Workgroup page, click Join A Workgroup. In the Workgroup box, type a workgroup name or accept the default and then click Next. If you join the computer to a domain, the Windows Deployment Wizard does not prompt you to capture an image.
7. On the Specify Whether To Restore User Data page, select Do Not Restore User Data And Settings and then click Next.
8. On the Packages page (if displayed), choose the packages, such as software updates and language packs, that you want to install on the image and then click Next.
9. On the Locale Selection page, choose your locale and keyboard layout and then click Next. Your choice here is irrelevant, because the Windows Deployment Wizard will configure the locale and keyboard layouts during deployment to the production environment.
10. On the Select The Time Zone page, select a time zone and then click Next. Your choice here is irrelevant, because the Windows Deployment Wizard will configure the time zone during deployment to the production environment.
11. On the Select One Or More Applications To Install page (if displayed), select the check box next to each application that you want to install on the image and then click Next.
12. In the Specify Whether To Capture An Image page, select Capture An Image Of This Reference Computer. In the Location box, type the Universal Naming Convention (UNC) path of the folder in which to store the image or accept the default capture location. In the File Name box, type the file name of the image or accept the default file name for the captured image. The default UNC path is the \Captures folder of the deployment share; the default image file name is the ID of the task sequence being installed. Click Next.
12. Click Next, then on the Ready To Begin page, click Begin.

After you click Begin, the Task Sequencer begins running the build's task sequence. By default, it begins by partitioning and formatting the hard disk. Then it installs and configures the operating system, runs Sysprep to prepare the computer for imaging, and restarts the computer in Windows PE to capture the image. The Windows Deployment Wizard stores the captured image in the folder specified on the Specify Whether To Capture An Image page, which is the deployment share's \Captures folder by default. After capturing the image, you can add it to the deployment share as a custom image by using the steps described in the section titled "Creating and Configuring a Deployment Share" earlier in this chapter. For more information about deploying your custom Windows 7 image, see Chapter 12.

Preparing Images Manually

The deployment share tells Windows Setup how to install and configure Windows 7 on the destination computers. It includes the settings (answer file) as well as device drivers and packages that you want to add to the operating system. It might also contain applications that you want to install.

A common way to deliver operating systems to users is to create an image of the desired configuration. This is particularly true when the deployment share includes other files, such as applications. Creating an image that you install on each destination computer is quicker and more efficient than installing the uncustomized Windows 7 image and then installing applications on each destination computer.

Sysprep prepares a Windows 7 installation for imaging or delivery to end users. Sysprep removes all user-specific information from a system and resets any system-specific security identifiers (SIDs) to allow the system to be duplicated. Once duplicated, systems using the duplicated image will register their own SIDs with the domain in which they are deployed. Sysprep has several command-line options to control its behavior, listed in Table 6-8.

TABLE 6-8 Sysprep Command-Line Options

OPTION	DESCRIPTION
/audit	Restarts the computer into audit mode. In audit mode, you can add additional drivers or applications to Windows Vista. You can also test an installation of Windows Vista before it is sent to an end user. If you specify an unattended Windows Vista setup file, the audit mode of Windows Setup runs the auditSystem and auditUser configuration passes.
/generalize	Prepares the Windows installation to be imaged. If you specify this option, all unique system information is removed from the Windows installation. The system's SID is reset, any System Restore points are cleared, and event logs are deleted. The next time the computer starts, the specialize configuration pass runs. A new SID is created, and the clock for Windows activation resets (if the clock has not already been reset three times).
/oobe	Restarts the computer into Windows Welcome mode. Windows Welcome allows end users to customize the Windows operating system, create user accounts, name the computer, and complete other tasks. Any settings in the oobeSystem configuration pass in an answer file are processed immediately before Windows Welcome starts.
/reboot	Restarts the computer. Use this option to audit the computer and to verify that the first-run experience operates correctly.
/shutdown	Shuts down the computer after Sysprep completes.

OPTION	DESCRIPTION
/quiet	Runs Sysprep without displaying on-screen confirmation messages. Use this option if you automate Sysprep.
/quit	Closes Sysprep after running the specified commands.
/unattend: *answerfile*	Applies settings in an answer file to Windows during unattended installation. You can create this answer file in Windows SIM.
answerfile	Specifies the path and file name of the answer file to use.

When you create a Windows 7 installation that you plan to image, you then use Sysprep to generalize the system. The following command generalizes the system and prepares it to run the Windows Welcome Wizard on the next restart.

```
sysprep /oobe /generalize
```

Most organizations use this command. If you are a system builder or an Original Equipment Manufacturer (OEM), however, you can also use Sysprep to create build-to-order systems. The following command lets you place a system into audit mode on the next restart, wherein you can install additional applications and modify configurations.

```
sysprep /audit /generalize /reboot
```

The following command then completes the customization by preparing the system to run the Windows Welcome on the next boot, which is a typical requirement in a retail environment.

```
sysprep /oobe
```

When all system preparations have been made, the system is ready for imaging. You can use the ImageX command with the */FLAGS* parameter to capture an image of the system. You can then burn the image onto a DVD, import it into a deployment share, or leave it on the system for use on the next system start.

Customizing Microsoft Deployment Toolkit

You can brand some features in MDT 2010. You can customize Deployment Workbench and the Windows Deployment Wizard. For example, you can customize Workbench.xml in C:\Program Files\Microsoft Deployment\Bin to change the text displayed in the Deployment Workbench title bar and for each item in the console tree. Although it's generally safe to customize the <*Name*> tag in Workbench.xml, you should avoid changing other tags.

The LTI process is driven by .xml files called *definition files*. You can brand the entire LTI process by customizing the following files, which are found in the \Scripts folder in your deployment share:

- **BDD_Welcome_ENU.xml** Customize this file to change the text displayed on the Windows Deployment Wizard's Welcome page.

- **Credentials_ENU.xml** Customize this file to change the text displayed in the User Credentials dialog box.
- **DeployWiz_Definition_ENU.xml** Customize this file to change the text for each page displayed by the Windows Deployment Wizard.
- **Summary_Definition_ENU.xml** Customize this file to change the text in the Deployment Summary dialog box, which displays at the end of the LTI process.

Summary

The new installation architecture first introduced in Windows Vista and deployment tools included in the Windows AIK make deploying Windows 7 in your organization easier than deploying earlier versions of Windows. The new .wim file format makes it possible to deploy highly compressed image files. Windows 7 helps reduce image count by removing hardware and other dependencies from the image. Modularization in Windows 7 makes servicing images easier than with legacy methods, so you no longer have to apply, customize, and recapture an image to update it. The new answer file format, Unattend.xml, provides a more flexible and consistent configuration. Finally, deployment tools in the Windows AIK 2.0 provide a robust way to create, customize, and manage Windows 7 images.

Although the Windows AIK 2.0 provides the basic tools for customizing and deploying Windows 7, MDT 2010 provides a more flexible framework for deploying Windows 7 in businesses. MDT 2010 enables you to create and customize multiple image builds. The framework includes automation common to most businesses and is highly extensible to suit any requirements. For example, by using MDT 2010 to deploy Windows 7, you can include custom actions such as installing applications, packages, and drivers that are performed during installation.

Additional Resources

These resources contain additional information and tools related to this chapter.

- Chapter 3, "Deployment Platform," includes information about the Windows 7 installation architecture, its key features and technologies, and how the various features interact.
- Chapter 4, "Planning Deployment," includes information about installing and preparing MDT 2010 for use. This chapter also describes how to use the MDT 2010 documentation.
- Chapter 10, "Configuring Windows Deployment Services," explains how to install and configure Windows Deployment Services and how to add images to and deploy images from Windows Deployment Services.
- Chapter 11, "Using Volume Activation," includes more information about Windows 7 product keys and volume activation.

- Chapter 12, "Deploying with Microsoft Deployment Toolkit," includes more information about using MDT 2010 to deploy Windows 7 images in the production environment.
- *Microsoft Deployment Toolkit Reference* in MDT 2010 lists the properties you can configure in a deployment share.
- *Windows Automated Installation Kit User's Guide for Windows 7* contains detailed information about the tools and technologies included in the Windows AIK 2.0. This guide is in the file Waik.chm in the Windows AIK 2.0.

CHAPTER 7

Migrating User State Data

Operating system deployment always involves user state migration—the process of migrating users' documents and settings from one operating system to another. Even when you don't migrate user state during deployment, users will spend countless hours trying to restore their preferences (such as desktop backgrounds, screensavers, and themes). Because this manual process reduces user productivity and usually increases support calls, organizations often choose to migrate some portion of user state to new operating systems as they are deployed.

User satisfaction is another reason to elevate the importance of user state migration in your project. Users are simply more satisfied and feel less overwhelmed when they sit down in front of a new operating system and they don't have to recover their preferences. The fact is that unsatisfied users can lead to poor post-implementation reviews and can have negative consequences for future deployment projects. For example, user dissatisfaction with previous projects can stall a deployment project that you know will benefit the company in the long term. Keep the users happy.

This chapter helps you decide which user state migration tools best suit your environment. It then explores the User State Migration Tool (USMT) 4.0, including customizing and automating the user state migration process. You'll learn how to identify user state data, how to plan the user state migration project, and how to execute the user state migration using tools such as Windows scripting and Microsoft Deployment Toolkit 2010 (MDT 2010).

Evaluating Migration Technologies

Whether you decide to migrate user state individually, as part of a high-volume deployment project, or not at all, you should evaluate the available options to ensure that you make the best choices for your environment. The size and scope of the migration project factor into your choice, as will the type and amount of user state data you choose to migrate.

The following sections describe the different options that Microsoft provides. Several third-party products are also available for migrating user state. If you're using MDT 2010 as your deployment framework, Microsoft recommends that you use USMT to migrate user state to Windows 7. USMT handles most common scenarios out of the box, and exceptional cases are easy to configure. Additionally, MDT 2010 already includes the pre-deployment and post-deployment logic for saving and restoring user state.

Windows Easy Transfer

Windows Easy Transfer is the Windows 7 equivalent of the Windows XP Files And Settings Transfer Wizard. This tool leads the user through a series of pages to determine how much data to migrate and which migration method to use (removable media, universal serial bus (USB) cable connection, or network). Using Windows Easy Transfer is not appropriate in high-volume deployment projects because it is a completely manual process. However, in bench deployments, Windows Easy Transfer can be a viable tool for migrating user state on individual computers.

> ***NOTE*** **Windows Easy Transfer can transfer user state data using a special USB cable available from most cable vendors. The Easy Transfer Cable includes circuitry that links two computers using their USB ports and can transfer data at approximately 20 gigabytes (GB) per hour.**

User State Migration Tool

Use USMT to migrate user state in high-volume deployment projects. It can execute complex, repeatable migrations of user state data between operating systems. You can script USMT; you can execute it as part of an MDT 2010 Lite Touch Installation (LTI) or Zero Touch Installation (ZTI); or you can execute it directly at the command prompt.

In addition to document and settings migration, USMT can migrate application preferences for Microsoft Office applications between versions of Office. For example, USMT can migrate Office XP settings to the Microsoft 2007 Office system.

Version 4.0 is the new version of USMT supporting Windows 7 migrations. It includes numerous changes from USMT 3.0, but the most notable are:

- **Hard-link migration store** For use in Refresh Computer scenarios only, hard-link migration stores are stored locally on the computer that you're refreshing and can migrate user accounts, files, and settings in less time using far less disk space.
- **Support for offline Windows operating systems** You can gather data from an offline Windows operating system using the ScanState command in Windows Preinstallation Environment (Windows PE). In addition, USMT now supports migrations from previous installations of Windows contained in Windows.old directories.
- **Volume Shadow Copy support** With the */vsc* command-line option, the ScanState command can now use the Volume Shadow Copy service to capture files that are locked for editing by other applications.

This chapter mostly describes USMT because of its power and flexibility in large-scale migrations. Later in this chapter, you will learn how to plan, develop, and deploy a custom migration project by using USMT.

Microsoft IntelliMirror

Microsoft introduced IntelliMirror with Microsoft Windows 2000 so that users' data and settings could follow them from computer to computer on the network. For more information about IntelliMirror, see Chapter 15, "Managing Users and User Data." The following two IntelliMirror features in particular minimize the need to migrate user state when deploying Windows 7 because these features store user state on the network.

- **Roaming user profiles** Roaming user profiles ensure that users' data and settings follow them on the network. This feature copies users' data and settings to a network server when they log off their computers and then restores their data and settings when they log on to another computer anywhere on the network. This feature provides a transparent way to back up users' data and settings to a network server.
- **Folder redirection** Folder redirection allows IT professionals to redirect certain folders (My Documents, Application Data, and so on) from the user's computer to a server. This feature protects user data by storing it on a network server, thereby providing centralized storage and administrator-managed backups. When used with roaming user profiles, folder redirection speeds the logon process by removing documents and other large files from the user profile.

NOTE **Windows 7 and Windows Vista store user profiles using a different folder hierarchy than Windows XP. Therefore, carefully review Chapter 15 before you rely on IntelliMirror in any Windows XP migration project.**

Using Windows Easy Transfer

Although USMT will generally be used in most enterprise environments, some businesses may find Windows Easy Transfer a simple and useful alternative to using USMT. This section briefly describes the basic functionality of Windows Easy Transfer, which can be particularly useful in bench deployments. Before you use Windows Easy Transfer, check for the following prerequisites:

- The destination computer must be running Windows 7. Windows 7 can create a Windows Easy Transfer data collection disk to execute the data collection portion of the migration on previous versions of Windows, but the destination computer must be running Windows 7.
- The source computer can be running any of the following operating systems:
  - Windows XP Service Pack 2 (SP2)
  - Windows Vista
  - Windows 7
- You must decide which user state data to migrate. Windows Easy Transfer does not offer the same degree of control as USMT, but you can choose which user accounts to migrate and the types of files and settings to migrate from each.

Windows Easy Transfer, shown in Figure 7-1, steps the user through a series of pages to define and execute the user state migration. Whether you are refreshing computers or replacing computers, Windows Easy Transfer can move user accounts, files and folders, program settings, Internet settings and favorites, and e-mail settings from a computer running earlier versions of Windows to Windows 7.

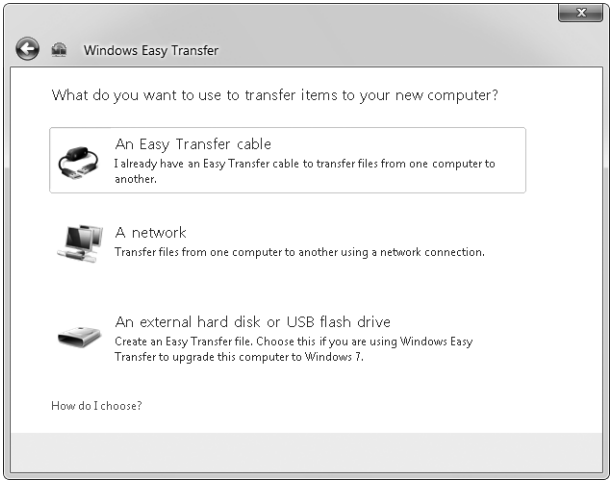


FIGURE 7-1 Windows Easy Transfer

Before using Windows Easy Transfer, though, you must prepare it for use by copying it to media that you can use to run it on earlier versions of Windows. To do this, follow these steps:

1. Close all running programs.
2. Start Windows Easy Transfer by clicking Start, pointing to All Programs, selecting Accessories, selecting System Tools, and then selecting Windows Easy Transfer.
3. On the Welcome To Windows Easy Transfer screen, click Next to continue.
4. Select the method you want to use for transferring files from your old computer from the following options:
   - A Windows Easy Transfer cable
   - A network
   - An external hard disk or USB flash drive
5. Click This Is My New Computer.
6. Click I Need To Install It Now.
7. Choose the destination for the Windows Easy Transfer files. If you want to use a USB Flash Drive (UFD) or portable USB hard drive, make sure it's plugged into the computer. If you want to put the files on a network share, make sure the computer is on the network. In any case, any computer on which you want to run Windows Easy Transfer using these files must also have access to the same hard disk or network share.
8. Choose the path of the folder in which to put the Windows Easy Transfer files and then click OK.

After preparing the Windows Easy Transfer files, you will have a removable drive or network share that contains program files that you run on the source computer—the computer from which you are moving the user's documents and settings. Use the instructions in the following sections, depending on the scenario: Refresh Computer or Replace Computer.

Refresh Computer

This section describes how to use Windows Easy Transfer in the Refresh Computer scenario. Recall that in this scenario, you are not replacing the computer. Instead, you are formatting the hard drive and then installing Windows 7 on it. As a result, you must save user documents and settings in a temporary location. To do this, follow these steps:

1. On the user's computer, run Windows Easy Transfer. To start Windows Easy Transfer, open the path that contains the Windows Easy Transfer files (on a UFD, removable USB drive, or network share) and then double-click the Windows Easy Transfer shortcut.
2. On the Welcome To Windows Easy Transfer screen, click Next.
3. Connect the portable USB drive or network share to the computer and then click An External Hard Disk Or USB Flash Drive.
4. Click This Is My Old Computer.

5. On the Choose What To Transfer From This Computer screen, choose each user account that you want to transfer to Windows 7. Selecting Shared Items, shown in Figure 7-2, migrates public documents, music, pictures, and settings. You can customize what to transfer from each folder by clicking Customize under each one.

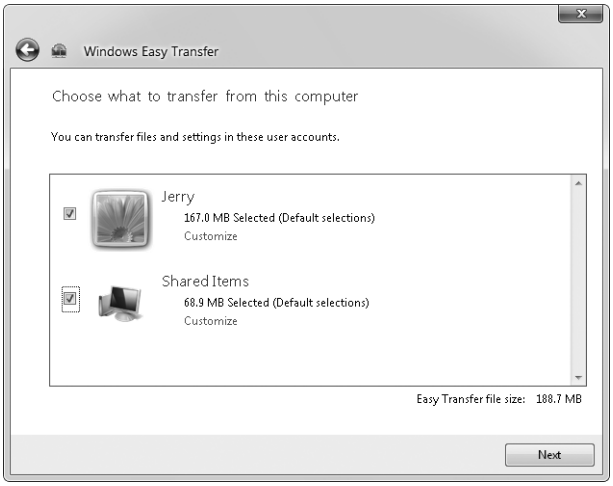


FIGURE 7-2 The Choose What To Transfer From This Computer screen

6. On the Save Your Files And Settings For Transfer screen, type and confirm a password with which you want to protect the migration data and then click Save. Microsoft recommends that you create a password to protect the information, but if you want, you also can click Save without typing and confirming a password.
7. In the Save Your Easy Transfer File dialog box, type the path of the file in which you want to store the migration data. This is not a folder path; it's the path and name of the file you want Windows Easy Transfer to create. The location can be on a network share or it can be on a portable drive. Click Save.
8. After Windows Easy Transfer finishes saving your migration data, click Next.
9. Confirm the location where Windows Easy Transfer saved your migration data and then click Next.
10. Click Close.
11. With the computer's user documents and settings safely in temporary storage, you can now install Windows 7 on the computer. Continue with the remainder of these steps after successfully installing Windows 7.
12. After installing Windows 7, connect the computer to the portable drive or network share on which you stored the migration data. In Windows Explorer, open the folder containing the migration data and then double-click the migration (.mig) file that you created before installing Windows 7. This starts Windows Easy Transfer.

13. On the Choose What To Transfer To This Computer screen, choose the accounts that you want to transfer to Windows 7 and then click Next. You can select which types of files and settings to transfer by clicking Customize. Additionally, you can map account names from the old computer to account names in Windows 7 by clicking Advanced Options.
14. Click Transfer to begin transferring the documents and settings from the migration file to the computer.
15. On the Your Transfer Is Complete screen, do either of the following and then click Close:
    - Click See What Was Transferred to see a detailed list of the accounts, documents, and settings transferred from the previous operating system to Windows 7.
    - Click See A List Of Programs You Might Want To Install On Your New Computer to see a list of applications installed on the previous operating system that you might want to reinstall in Windows 7. Install these applications after you finish migrating your documents and settings to Windows 7.
16. Click Restart Now to restart the computer. You must restart the computer for the changes to take effect.

Replace Computer

This section describes how to use Windows Easy Transfer in the Replace Computer scenario. In this scenario, you are replacing a computer running an earlier version of Windows with a new computer running Windows 7. In this case, you can certainly use the steps described in the previous section to transfer documents and settings from the old computer to temporary storage, replace the computer, and then restore documents and settings to the new computer. However, transferring documents and settings from the old computer to the new computer through the network is a simpler solution, which you can implement by following these steps:

1. Make sure both the old computer and the new computer are on the network.
2. On the new computer, complete the following steps:
    a. Close all running programs.
    b. Start Windows Easy Transfer by clicking Start, pointing to All Programs, selecting Accessories, selecting System Tools, and then selecting Windows Easy Transfer.
    c. On the Welcome To Windows Easy Transfer screen, click Next to continue.
    d. Run Windows Easy Transfer. To start Windows Easy Transfer, open the path that contains the Windows Easy Transfer files (on a UFD, removable USB drive, or network share) and then double-click Migwiz.exe. Do not open the file from the Start menu.
    e. Click A Network.
    f. Click This Is My New Computer.
    g. Click I Already Installed It On My Old Computer.

3. On the old computer, complete the following steps:
   a. Run Windows Easy Transfer. To start Windows Easy Transfer, open the path that contains the Windows Easy Transfer files (on a UFD, removable USB drive, or network share) that you created earlier and then double-click the Windows Easy Transfer shortcut.
   b. On the Welcome To Windows Easy Transfer screen, click Next.
   c. Click A Network.
   d. Click This Is My Old Computer.
   e. Record the Windows Easy Transfer key.
4. On the new computer, complete the following steps:
   a. In Windows Easy Transfer, click Next.
   b. On the Enter Your Windows Easy Transfer Key screen, type the Windows Easy Transfer key that you noted previously for the old computer and then click Next.
   c. On the Choose What To Transfer To This Computer screen, choose the accounts that you want to transfer to Windows 7 and then click Next. You can select which types of files and settings to transfer by clicking Customize. Additionally, you can map account names from the old computer to account names in Windows 7 by clicking Advanced Options.
   d. Click Transfer to begin transferring the documents and settings from the migration file to the computer.
   e. On the Your Transfer Is Complete screen, click See What Was Transferred to see a detailed list of the accounts, documents, and settings transferred from the previous operating system to Windows 7, or click See A List Of Programs You Might Want To Install On Your New Computer to see a list of applications installed on the previous operating system that you might want to reinstall in Windows 7. After reviewing the migration reports, click Close.
   f. Click Restart Now to restart the computer. You must restart the computer for the changes to take effect.

Planning User State Migration Using USMT

Thoughtful planning is a critical factor in the success of any user state migration project. By identifying the scope of the migration, you can plan storage space requirements, labor, and development time required to successfully implement the migration solution. This section describes user state migration planning topics such as using subject matter experts (SMEs), identifying and prioritizing user state data, storing user state data, and testing the effort.

NOTE **The team responsible for planning and developing user state migration must work hand-in-hand with the team responsible for application deployment. Both teams will share a lab environment, application portfolio, SMEs, and so on. For more information, see Chapter 8, "Deploying Applications." In some cases, the same IT professionals responsible for application deployment are also responsible for user state migration.**

DIRECT FROM THE SOURCE

Planning

Doug Davis, Lead Architect
Management Operations & Deployment, Microsoft Consulting Services

The main thing I have found about user state migration is that very few companies actually know which files they need to migrate. Even fewer have an idea about settings. The largest concern is, of course, lost data—the settings matter less.

Customers who use IntelliMirror features such as folder redirection and offline folders are the easiest to deal with; however, these customers are the minority. There are really only two ways to get user data and files. Asking the client which files they use never works and just drags out the process. You're left with another way that drives user feedback: to do full backups on your proof-of-concept and pilot groups and run standard USMT without any custom settings. When users ask for files to be recovered from the backup, you add them to the custom settings to be retained.

The second way takes a little bit longer and is what I call intern-ware: If you have an intern, you can give him or her this busy work. Figure out which applications are critical to you, search the registry for "open with," and cross-reference the file extensions to the program.

Choosing Subject Matter Experts

Although IT professionals in small organizations probably know each application and the settings used in the computing environment, this is highly unlikely to be the case in large organizations that potentially have thousands of applications. In large organizations, you should use SMEs to help in the planning, development, and stabilizing processes. SMEs, though not necessarily experts, are the users who are most familiar with the applications and data to migrate, and they're usually stakeholders in seeing that the process is properly performed.

Use SMEs to assist with several key tasks:

- Locating application source media, such as CDs or DVDs

- Identifying document storage locations for each application
- Identifying application configuration data and settings to migrate
- Selecting the operating system preferences to migrate
- Consulting on file relocations that will be performed as part of the migration

Identifying User State Data

User state data can consist of many elements: settings that customize the user experience, application data created by the user, e-mail and messaging settings and data, and even personal data. The following sections describe examples of user state data.

Operating System Settings

The following list describes many of the operating system files and settings that you will want to migrate. (USMT migrates most of these by default.)

- **Appearance settings** Examples include desktop background, colors, sounds, and screensaver settings.
- **User interface settings** Examples include mouse pointers, whether double-clicking a folder opens it in a new window or in the same window, and whether users must click or double-click an item to open it.
- **Windows Internet Explorer settings** Examples include home pages, favorites, cookies, security settings, and proxy settings.
- **Mail settings** Examples include mail server settings, signature files, and contact lists.

Application Data and Settings

You will find application data in a number of locations. As you inventory the applications in your environment, consider the following potential locations for application settings and data storage:

- **The Program Files folder** Many applications still store settings and data directly in the Applications folder within Program Files. As you plan the migration, consider whether or not you can safely redirect the application data to a different location. This will assist with future attempts to allow use of the application by standard (non-administrator) users.
- **A specific folder on the local disk** Many applications define a data storage location on the local disk for storage of application settings and data. This location is often the root of the system drive.
- **The user's profile folder** Many applications store data in user profile folders. Search the Documents And Settings folder (in Windows XP) or the Users folder (in Windows Vista and Windows 7) for application settings and data files.

Users' Documents

Users will store data in a variety of locations. The following strategies will help you locate users' documents:

- **Search user profile folders** The Desktop and My Documents folders are only two of many locations where you will find user data in the user profile folders. Ideally, however, these two folders are the primary location of users' documents.
- **Interview users and SMEs** Survey users and interview SMEs to determine common storage locations for documents. An intranet Web site, possibly based on Windows SharePoint Services, is an ideal data-collection tool.
- **Scan a sample of disks** Search the local disks for common document file extensions such as .doc and .xls. Although you can't scan every disk in the organization, you can scan a representative sample to give you an idea of where you'll find documents.
- **Search Recent Documents** Scan the Recent folder in users' profiles to determine the locations most frequently used to store data. This can expose some of the less intuitive storage locations. Search a representative sample of users in the organization.

DIRECT FROM THE SOURCE

USMT and ACT

Doug Davis, Lead Architect
Management Operations & Deployment, Microsoft Consulting Services

Application Compatibility Toolkit 4.0 (ACT 4.0) always grabbed the registered file extensions in the log files but never posted them to the database. In my review of the functional specification of ACT 5.0, I asked to have that log data posted to the database even if it wasn't exposed in the graphical user interface (GUI).

With a little work using SQL Server, you should be able to use ACT to find out which applications you are migrating and then sort the file extensions you need for USMT in a more logical fashion. For example, you can extract the file extensions that are a high priority from the ACT database for applications in the portfolio and then focus on migrating those applications first.

Prioritizing Migration Tasks

As you compile your user state migration requirements, prioritize them according to their impact on the organization. It's important to the success of this project to concentrate first on mission-critical data and later on preferences such as desktop wallpaper or screensaver settings. Prioritizing requirements helps the development personnel to prioritize their work. SMEs are a valuable source of input when prioritizing the migration requirements.

Choosing a Data Store Location

USMT stores user state in a data store. USMT can create the data store in a variety of locations and media during migration. By default, USMT creates a store file that contains compressed user data. It can also encrypt the data store to protect the data during the transition to the new operating system. As you prepare for user state migration, you must determine the best location for the USMT data store.

Consider the following when locating the USMT data store:

- **Hard-link migration reduces storage requirements** During a hard-link migration, USMT maps how a collection of bits on the hard disk is wired into the file system. It allows you to remove the old operating system and install Windows 7 without requiring you to create copies of the original files. After installing Windows 7, USMT restores the original file links. Hard-link migration uses significantly less disk space and takes considerably less time, but you can perform a hard-link migration only in the Refresh Computer scenario.
- **USMT cannot store multiple operations in the same file** USMT operations can collect data from more than one user but cannot store more than one operation in a single store file. In a high-volume migration, you can either locate the data store locally (which is only possible because the Windows 7 imaging process is nondestructive) or locate each data store on the network (possibly organized by computer name). Note that MDT 2010 handles the data store location automatically and provides choices for customizing it.
- **User state data can use significant space** When creating your migration plan, you can run the ScanState component of USMT with the */p:<path to a file>* command-line option to create a size estimate. If you're locating the data store on a server, rather than locally, run this command on a representative sample of computers in the environment to calculate the storage required.
- **The USMT data store must be accessible to both the source and target systems** When writing to or reading from the data store, USMT must have access to that data store. Locate the file somewhere that will be available to both computers. MDT 2010 handles this issue somewhat transparently. If you're locating the data store locally, access is not an issue.

Local Data Stores

You can locate the USMT data store on the local disk in the Refresh Computer scenario. (See Chapter 4, "Planning Deployment," for a description of deployment scenarios.) ImageX and Windows Setup are *nondestructive*, which means that they can install the operating system without destroying the data on the disk. This optimizes the speed of the migration process because network speeds and removable media speeds are factored out of the process. MDT 2010 provides the option to use local data stores.

A better option is using a hard-link migration store, which enables you to perform an in-place migration in which USMT maintains all user state on the local computer while you remove the old operating system and install the new operating system. Therefore, hard-link migration is suitable only for the Refresh Computer scenario. Using a hard-link migration store drastically improves migration performance and significantly reduces hard-disk utilization, reduces deployment costs, and enables entirely new migration scenarios.

Remote Data Stores

In the Replace Computer (side-by-side) and New Computer scenarios, you can put the USMT data store on a network server. In these scenarios, putting the data store on the network is necessary because the local data store will not be available in the postinstallation phase.

Removable Storage

You can also put USMT store files on removable media during the migration process. You can use flash disks and portable hard disks to simplify this process. Because this step adds interaction to the process, it is recommended only in bench deployments or in scenarios in which you've already factored interaction into the deployment process.

Automating USMT

The full power of the USMT is realized when you automate migration. Through the use of scripting techniques—or tools such as MDT 2010 and Microsoft System Center Configuration Manager 2007—you can automate the migration of large numbers of systems. The following list describes each option:

- **Scripting** You can execute USMT with a variety of scripting tools, including Windows PowerShell, VBScript, and batch script files. By including the appropriate command-line options, you can automate the migration process to collect and restore user state data. End users can then execute these scripts to migrate their own data.
- **Microsoft Deployment Toolkit** MDT 2010 fully enables user state migration as part of the LTI and ZTI deployment processes. You can customize the location of the data stores and customize the migration .xml files to include or exclude user state as defined by your requirements (although this is often not necessary with USMT 4.0). Using the default migration .xml files with MDT 2010 is an extremely simple, straightforward process. Thus, the only real effort is in creating custom migration .xml files for USMT, if necessary.
- **Configuration Manager** Configuration Manager can be used as is or with MDT 2010 to automate user state migration as part of operating system deployment. For more information about using USMT with Configuration Manager alone, see the System Center Configuration Manager 2007 documentation.

Testing User State Migration

After you set up the USMT migration .xml files and infrastructure, you should conduct a test pass to ensure that the solution works as planned. Test modularly: start with migration files first, followed by command-line options and automation. As you create the project plan, define testable criteria. With a test plan already established, you will be prepared to begin testing as soon as the development is complete.

Creating a Lab Environment

Establish a test lab with equipment and settings similar to systems in production in your organization. The goal is to test each scenario you will see in the production environment. Duplicate your production environment in as much detail as possible. Set up migration servers and data stores, configure source and target client systems, and prepare and place control files in the appropriate locations. Finally, execute the test and measure the results. The team responsible for user state migration should share a lab environment with the team responsible for application deployment. (For more information, see Chapter 8.)

Choosing Sample Data

If possible, conduct the migration tests on actual production data. You can copy this data from production systems as you prepare for the testing. For example, you can create images of SME computers for testing purposes. Testing on production data helps ensure that you expose the migration infrastructure to all scenarios that will be seen in the production environment.

Be sure to choose the following types of production data:

- **Operating system settings** You should test desktop and appearance settings. Users can nominate elements for testing, or you can select a sample based on the results of user surveys. Test user profile settings and user preference settings to ensure that they are properly migrated. Identify a set of user settings that you can test.
- **Application data and settings** Include application data such as configuration files and data files in your test plan. Test application registry settings and initialization files to ensure that they are properly migrated. Work with developers and SMEs to identify a representative sample of these configuration settings to test.
- **Users' documents** Choose a representative sample of user data. Include sources such as My Documents and any custom data folders found in your environment.

Running the Test

Run the USMT test using the migration .xml files and procedures that you have developed for the production environment. The goal is to simulate the production migration with as much detail as possible. Be sure to use any scripts and processes designed to automate the USMT process.

Validating the Test Results

Following the migration test, verify all user state elements that have been defined for testing. View each setting and test each migrated application. Open the e-mail client and operate applications to ensure that user customizations still exist in the new system. Identify any errors and list any items that failed to migrate. Investigate these elements to ensure that you properly configured the control files for the migration. If necessary, retest to verify that problems have been resolved.

Installing USMT

USMT 4.0 is included in the Windows Automated Installation Kit 2.0 (Windows AIK 2.0). You can download the Windows AIK from the Microsoft Download Center at *http://www.microsoft.com/downloads*. After downloading and installing the Windows AIK, the USMT source files are in C:\Program Files\Windows AIK\Tools\USMT*Platform*, where *Platform* is either amd64 or x86.

You can stage USMT directly on each client computer or on a network share. If you're using MDT 2010, it can install USMT in deployment shares automatically. MDT 2010 already contains logic for using USMT to save and restore user state data on each computer.

You use USMT in a number of ways: on a network share, on Windows PE media, on an MDT 2010 deployment share, or with Configuration Manager. The last two options enable migration during LTI and ZTI deployment projects. The following sections describe each option.

Network Share

After installing the Windows AIK on a local computer, you can copy the contents of the C:\Program Files\Windows AIK\Tools\USMT\ to a network share. Then you can run ScanState and LoadState remotely on each computer.

Windows PE Media

USMT 4.0 supports offline migration. That is, you can run USMT from a Windows PE session to save user state data without actually starting the old operating system. To support this scenario, you must copy the USMT binary files to your Windows PE media. For more information about creating Windows PE media, see Chapter 9, "Preparing Windows PE."

Microsoft Deployment Toolkit

Unlike earlier versions of MDT, MDT 2010 automatically adds USMT to deployment shares when you update them. It copies the files from the Windows AIK to the USMT folder in the deployment share. You do not have to do anything additional to install USMT in a deployment share.

Configuration Manager

You can use USMT with Configuration Manager to manage user state migrations during operating system deployment. For more information, see the System Center Configuration Manager 2007 documentation.

Understanding USMT Components

After downloading and installing the Windows AIK, the USMT source files are in C:\Program Files\Windows AIK\Tools\USMT*Platform*, where *Platform* is either amd64 or x86. The installer copies many files into this folder, including .dll files, feature manifests, and other application initialization files. (See Figure 7-3.) Most of the files support the two main executables: Scanstate.exe and Loadstate.exe.

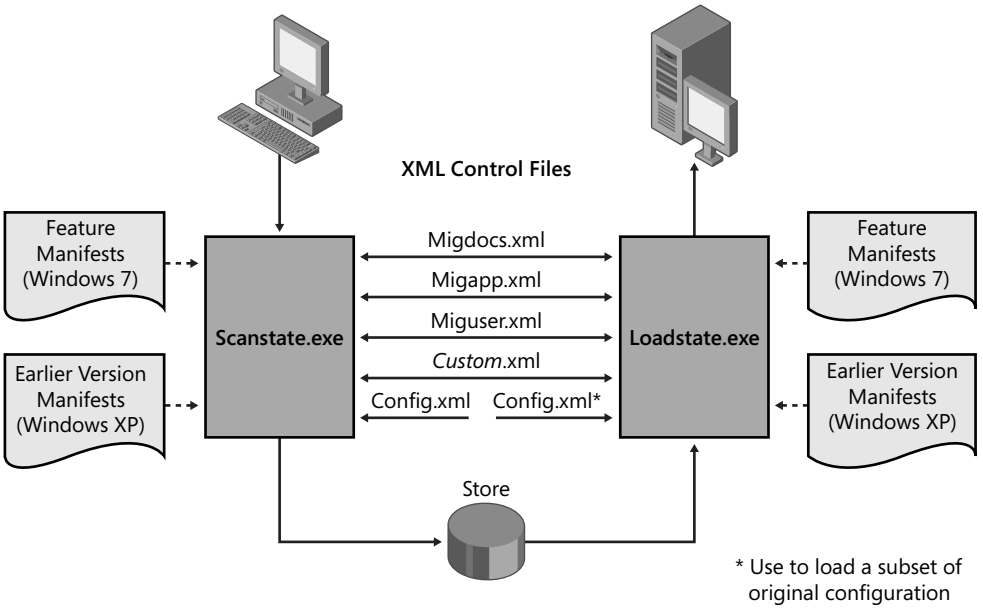


FIGURE 7-3 USMT components

In addition to ScanState and LoadState, USMT uses three XML migration files—MigApp.xml, MigDocs.xml, and MigUser.xml—to perform basic file and settings migrations based on default criteria. You can customize these files, along with custom .xml files, to migrate additional files and settings or to exclude some of the default files and settings. For more information about migration .xml files, see the section titled "Developing Migration Files" later in this chapter.

ScanState and LoadState save and restore user state data, respectively. You can run them directly from a command prompt. They provide several command-line options that control their behavior. USMT 4.0 includes an additional utility called UsmtUtils.exe. This utility helps you to determine cryptographic options for your migration. It also helps remove hard-link stores that you cannot delete otherwise due to a sharing lock.

Scanstate.exe

You use ScanState to save user state data. By default, this program places user state data into the data store location as defined by the three migration .xml files. The following describes an abbreviated syntax of ScanState, and Table 7-1 describes each command-line option.

```
Scanstate.exe [Store][/i:[path\]filename] [/config:[path\]file] [/hardlink /nocompress]
[/o] [/p[:file]] [/vsc]
```

TABLE 7-1 Scanstate.exe Command-Line Options

OPTION	DESCRIPTION
Store	Specifies a path to the data store.
/config:[path\]file	Specifies a Config.xml file. (See the section titled "Developing Migration Files" later in this chapter for more information.)
/hardlink	Enables the creation of a hard-link migration store at the location specified by *Store*. You must specify the */nocompress* option when using this option.
/i:[path\]filename	Identifies a migration .xml file to use when saving state data. You can use this option multiple times.
/nocompress	Disables data compression. Use this option only with the */hardlink* option or when testing in a lab environment.
/o	Overwrites existing data in the data store.
/p[:file]	Creates a size estimate in the path specified. When used without a path, it creates a size estimate file called USMTsize.txt in the location specified by *Store*.
/vsc	Enables use of the Volume Shadow Copy service to migrate files that are locked or in use during migration.

NOTE **ScanState supports many other command-line options. For a complete list of these options, see the USMT.chm help file in the Windows AIK.**

Loadstate.exe

You use LoadState to restore user state from the data store. By default, this program restores user state to the location from which ScanState originally saved it—unless one of the migration .xml files redirects it. You must specify the same migration .xml files to LoadState that you did to ScanState. The following describes an abbreviated syntax of LoadState, and Table 7-2 describes each command-line option.

```
Loadstate.exe [Store][/i:[path\]filename] [/hardlink /nocompress]
```

TABLE 7-2 Loadstate.exe Command-Line Options

OPTION	DESCRIPTION
Store	Specifies a path to the data store.
/i:[path\]filename	Identifies a migration .xml file to use when restoring user state data. You can use this option multiple times.
/config:[path\]file	Specifies a Config.xml file. (See the section titled "Developing Migration Files" later in this chapter for more information.)
/hardlink	Enables the creation of a hard-link migration store at the location specified by *Store*. You must specify the */nocompress* option when using this option.
/nocompress	Disables data compression. Use this option only with the */hardlink* option or when testing in a lab environment.

NOTE **LoadState supports many other command-line options. For a complete list of these options, see the USMT.chm help file in the Windows AIK.**

Migration Files

Both ScanState and LoadState use three migration .xml files to control migrations. In addition to these three files, you can specify one or more custom .xml files to migrate custom applications or customize the standard migrations. The following section, "Developing Migration Files," describes the .xml files that come with USMT and how to build custom migration files.

Developing Migration Files

USMT ships with three standard migration .xml files. You can customize these files to control the behavior of USMT during migration. In addition to the three standard files, you can develop custom .xml files to migrate special application settings and files. The three migration .xml files included with USMT are:

- **MigApp.xml** Contains rules to migrate application settings.
- **MigDocs.xml** Contains rules that can find user documents on a computer automatically without creating extensive custom migration .xml files. Use this migration file if the data set is unknown. Don't use this migration file and MigUser.xml together.
- **MigUser.xml** Contains results to migrate user profiles and user data. Don't use this migration file and MigDocs.xml together.

Customizing USMT

You manage USMT through command-line options and the migration .xml files. You could modify the default files to control some aspects of the migration, but this is not recommended. The better option is to create custom .xml files to migrate specific application settings and data. The following list describes customization points for USMT:

- **Command-Line Control** You can use command-line options, such as */ui* and */ue*, to include and exclude specific users during the migration process. You can also specify custom .xml files and manage encryption and compression options.
- **Customizing the Migration XML Files** You can modify the migration .xml files to exclude portions of a standard migration or to redirect data and settings during the migration process. This capability is helpful for scenarios in which you want to consolidate migrated data, but a better alternative to customizing the existing migration files is creating custom migration files.
- **Generating Config.xml** You can generate a Config.xml file to exclude an entire feature from the migration. For example, you can exclude the entire Documents folder or exclude all of the settings for a specific application. Using this file to exclude features is easier than modifying the migration .xml files because you don't have to understand the migration rules or syntax. Using this file is also the only way to exclude operating system settings when migrating to Windows 7. For more information about Config.xml, see the USMT.chm help file in the Windows AIK.

NOTE **You can use migration .xml files that you created for USMT 3.0 with USMT 4.0. To use new USMT features, you must refresh your migration files to use new XML elements and command-line options.**

Control File Syntax

The default migration .xml files use XML elements to control migration behavior. These files cover the most common applications, documents, and settings. If you want to migrate settings and application data that the default migration .xml files don't cover, you should create a custom .xml file. The full XML reference for USMT is in the USMT.chm help file in the Windows AIK. Additionally, the XML reference in USMT.chm contains good examples that you can use as your starting point for creating custom migration .xml files.

NOTE **The best practice is to create custom migration .xml files instead of adding application data and settings to the default migration .xml files. Doing so makes maintaining those settings easier over time and prevents confusion.**

Deploying Migration Files

The following list describes how to deploy custom migration .xml files for stand-alone use, with MDT 2010, and with Configuration Manager:

- **Stand-alone use** You can store the migration .xml files in the USMT program folder or place them in a central location. You must specify the full path to each migration .xml file (Scanstate *server**share**computer* /I:*server**share**migration*.xml).
- **Microsoft Deployment Toolkit** MDT 2010 has a specific organization for deployment shares. You must store custom migration .xml files in the USMT*platform* folder of the deployment share, where *platform* is either x86 or x64.
- **Configuration Manager** Configuration Manager uses USMT to migrate user state data during operating system deployments. You can specify the location of migration .xml files and data stores during the configuration of Configuration Manager. See the System Center Configuration Manager 2007 documentation for more information.

Using USMT in Microsoft Deployment Toolkit

User state migrations can be started and controlled in a number of ways. Among these are direct command-line execution, scripting, MDT 2010, and Configuration Manager. The section titled "Understanding USMT Components" earlier in this chapter describes the command-line options for running USMT directly or driving it by using scripts. This section describes how to enable USMT in MDT 2010, as well as how to add custom migration .xml files to MDT 2010.

HOW IT WORKS

State Migration in MDT 2010

Chapter 6, "Developing Disk Images," describes the task sequence and Task Sequencer that MDT 2010 uses for deploying Windows 7. The default task sequence separates the process into two phases. One of the preinstallation phases is State Capture; one of the postinstallation phases is State Restore. The entire state migration work is tucked into these two phases.

In the State Capture phase, the Capture User State step runs ZTIUserState.wsf /Capture to capture user state. It uses settings from the deployment share's CustomSettings.ini file or the MDT 2010 database. In the State Restore phase, the Restore User State step runs ZTIUserState.wsf /Restore to restore the state data captured in the Capture User Step.

For the */capture* command-line option, ZTIUserState.wsf reads its settings (UDDShare, UDDir, and so on) from the environment and then chooses the best place to create the data store based upon UserDataLocation. In the final step, the script executes ScanState with the command-line arguments that it assembled from

the data in the environment, adding the command-line options for hard-link migration. For the *restore* command-line option, ZTIUserState.wsf retrieves information about the data store it created from the environment and then runs LoadState using the command line that it assembled from that information, also adding command-line options for hard-link migration.

Specifying the Data Store Location

Performing hard-link migrations is the recommended action in Refresh Computer scenarios, and this is the default behavior of MDT 2010. For other scenarios, you can create the data stores within the MDT 2010 deployment share. However, creating a share for the data stores on a separate server is better than putting the data stores in the deployment share because it spreads the load and allows you to dedicate resources to user state migration more easily.

After creating the share for the data stores, you configure the data store location by customizing properties in each deployment share's CustomSettings.ini file, as shown in Figure 7-4. To configure CustomSettings.ini, right-click a deployment share in Deployment Workbench and click Properties; then configure CustomSettings.ini on the Rules tab. You can also customize these properties in the MDT 2010 database. Table 7-3 describes these properties. For more information about CustomSettings.ini and the MDT 2010 database, see Chapter 12, "Deploying with Microsoft Deployment Toolkit."

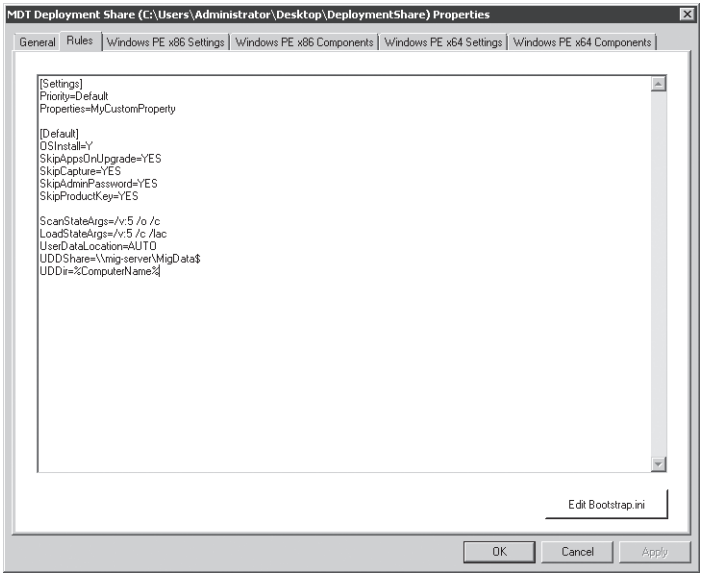


FIGURE 7-4 Configuring USMT settings in CustomSettings.ini

TABLE 7-3 USMT Properties in MDT 2010

PROPERTY	CONTROLS
LoadStateArgs=*arguments*	The arguments passed to LoadState. MDT 2010 inserts the appropriate logging, progress, and data store parameters. If this value is not included in the settings file, ZTIUserState.wsf uses LoadStateArgs=*/v:5 /c /lac*.
ScanStateArgs=*arguments*	The arguments passed to ScanState. MDT 2010 inserts the appropriate logging, progress, and data store parameters. If this value is not included in the settings file, ZTIUserState.wsf uses ScanStateArgs=*/v:5 /o /c*. Use the property *USMTMigFiles* to specify the .xml files to be used by Scanstate.exe instead of using the */i* parameter in the *ScanStateArgs* property. This prevents the ZTIUserState script from potentially duplicating the same list of .xml files.
UDDShare=*Path*	The network share in which to create data stores, such as *UDDShare*=*server**MigData$*. This value is ignored when performing hard-link migrations.
UDDir=*Folder*	The folder where the user state migration data is stored. This folder exists beneath the network shared folder specified in UDDShare. For example, UDDir=%ComputerName%. This value is ignored when performing hard-link migrations.
UserDataLocation=[*blank* | AUTO | NETWORK | NONE]	The location in which user state migration data is stored: ■ *BLANK* For LTI, the Windows Deployment Wizard prompts for the storage location. For ZTI, this is the same as setting the property to NONE. ■ *AUTO* MDT 2010 performs a hard-link migration. ■ *NETWORK* MDT 2010 creates the data store in the location designated by the UDDShare and UDDir properties.
UDProfiles=*Profile1*, *Profile2*, *ProfileN*	A list of user profiles to save during the MDT 2010 State Capture phase by Scanstate.exe, such as UDProfiles=Administrator, Patrice, Dave.

NOTE **You can also use removable media and local data stores during a user state migration by not setting the UserDataLocation value. The Windows Deployment Wizard will prompt you for the user data location. See the Toolkit Reference in MDT 2010 for more details about these properties.**

Adding Custom Migration Files

MDT 2010 will use only the MigApp.xml and MigDocs.xml files unless you indicate the path to your custom .xml files. As with other properties in MDT 2010, you can configure them in each deployment point's CustomSettings.ini file or add them to the MDT 2010 database.

Set the property USMTMigFiles to the name of each custom migration .xml file. If you don't configure this property, MDT 2010 uses the default migration files: MigApp.xml and MigDocs.xml. If you do configure this option, MDT 2010 uses only the files specified in the property. Therefore, if you configure this property, it must also include the default migration .xml files. For example, the following line in CustomSettings.ini adds Custom.xml to the default .xml files.

```
USMTMigFiles1=MigApp.xml
USMTMigFiles2=MigDocs.xml
USMTMigFiles4=Custom.xml
```

NOTE **Do not try to customize the script that drives the USMT process (ZTIUserState.wsf) to add migration .xml files by adding the */i* command-line option. This can potentially cause the script to work improperly and may make upgrading to future versions of MDT problematic. Add custom migration .xml files only by customizing the *USMTMigFiles* property.**

Summary

Migrating user state is an important aspect of desktop deployment because it minimizes lost productivity and increases user satisfaction. User state migration requires thorough planning and a good understanding of user, application, and system settings, as well as knowledge of the location of data files in your environment. SMEs can assist with the identification of files and settings for migration, and you should test all migration projects extensively to ensure that they will function properly in your production environment.

USMT offers the most powerful migration options for high-volume deployment projects. As a user state migration engine, USMT has support built in to MDT 2010. In fact, you can provide support for migrating most common data and settings with not much more effort than customizing each deployment share's CustomSettings.ini. By creating custom migration .xml files, you can add support for corner cases and custom applications that your organization uses.

Additional Resources

These resources contain additional information and tools related to this chapter.

- Chapter 12, "Deploying with Microsoft Deployment Toolkit," includes more information on using MDT 2010 to migrate users.
- Chapter 15, "Managing Users and User Data," includes details on roaming user profiles and Folder Redirection.
- USMT.chm in the Windows AIK includes detailed information about ScanState and LoadState command-line options, creating XML migration files, and other more advanced scenarios like offline migration.
- User State Migration Tool 4.0 documentation on Microsoft TechNet at *http://technet.microsoft.com/en-us/library/dd560801.aspx*.

CHAPTER 8

Deploying Applications

Deploying applications is an important aspect of desktop deployment. Choosing the applications you want to deploy, and deciding how to deploy them, affects the choices that you make when deploying the operating system. For example, will you include applications in the operating system image or deploy them later? Including applications in the operating system image provides good performance but low flexibility; deploying them later provides poorer performance but higher flexibility and lower maintenance costs.

During the planning stages of your deployment project, you must identify each application used in the environment. Then you must prioritize the application inventory so that you can focus on the most important applications first and possibly eliminate applications that are duplicates, no longer in use, or unsupported by the Windows 7 operating system.

After creating a prioritized application inventory, you must find and mitigate compatibility issues (a process described in Chapter 5, "Testing Application Compatibility"). Then you determine how to deploy the applications. This chapter helps you make these decisions and use the tools that Microsoft provides for deploying applications, including Microsoft Deployment Toolkit 2010 (MDT 2010). See Chapter 5 to learn how to create an application inventory, test applications for compatibility with Windows 7, and resolve the compatibility issues that you find.

Preparing the Lab

Planning application deployment requires a lab environment for application repackaging. Within an organization, different teams that work on deployment (image engineering, application packaging, and so on) can and often should share a single lab environment. Sharing a lab enables teams to share deliverables and integration-test their work with other components more easily. In a shared lab environment, however, each team must have its own workspace on the file server and dedicated computers on which to work.

Although the lab must have access to the Internet, it should be insulated from the production network. However, if you don't install any server features like Dynamic Host Configuration Protocol (DHCP), separating the lab from the production network is not a rigid requirement. Application repackaging does not require that the lab mirror the production network. The lab must provide storage space for application source files and repackaged applications.

The following list describes the recommended requirements for a lab used to repackage applications:

- A lab server configured as follows:
  - Windows Server 2008 or Windows Server 2008 R2
  - An Active Directory Domain Services domain
  - DHCP services
  - Domain Name System (DNS) services
  - Windows Internet Naming Service (WINS) services (optional)
  - Microsoft SQL Server 2005 or SQL Server 2008
  - Microsoft Virtual Server 2005, Microsoft Virtual PC 2007, Microsoft Windows Virtual PC, or Microsoft Hyper-V
- Lab test accounts (for standard users and an administrator)
- Network hardware to provide connectivity (consider the routing and bandwidth so that moving large files doesn't impact users on the production network)
- Internet access (for downloading updates, files, and so on)
- Test computers that accurately reflect production computers
- Source files for all applications to be tested and repackaged
- Software repackaging tools

NOTE **MDT 2010 provides prescriptive guidance for building and using a deployment lab. For more information, see the "Getting Started Guide" in MDT 2010.**

Planning Deployment

Creating an application inventory is the main task you must complete when planning application deployment. You use the inventory to prioritize applications—determining which are not compatible with Windows 7, which you must repackage for automatic installation, and so on. The Application Compatibility Toolkit (ACT) provides tools for collecting an application inventory based on the production network. For more information about using ACT to inventory applications, see Chapter 5.

After creating an application inventory, you must take the following planning steps for each application in the list:

- **Priorities** Prioritize the application inventory so that you can focus on the most important applications first. Focus on the applications that help your organization provide products and services to customers. While you are prioritizing the inventory, you might discover duplicate applications (different versions of the same application or different applications fulfilling the same purpose) that you can eliminate. You may also discover many applications that were used for a short-term project and are no longer required.
- **Categories** Categorize each application in the inventory as a core application or a supplemental application. A core application is common to most computers (virus scanners, management agents, and so on), whereas a supplemental application is not. Chapter 5 recommends additional ways in which you can categorize applications, such as by department, geography, cost center, worker type, and so on.
- **Installation method** Determine how to install the application automatically. Whether the application is a core or supplemental application, you achieve the best results by completely automating the installation. You cannot automate the installation of some legacy applications; you must repackage them. If so, the best time to choose a repackaging technology is while planning deployment. For more information about repackaging technologies, see the section titled "Repackaging Legacy Applications" later in this chapter.
- **Determine responsibility** Determine who owns and is responsible for the installation and support of each application. Does IT own the application or does the user's organization own it?
- **Subject matter experts** You will not have the in-depth understanding of all applications in the organization that you will need to repackage them all. Therefore, for each application, identify a subject matter expert (SME) who can help you make important decisions. A good SME is not necessarily a highly technical person. A good SME is the person most familiar with an application, its history in the organization, how the organization uses it, where to find the media, and so on.

- **Configuration** Based on feedback from each application's SME, document the desired configuration of each application. You can capture the desired configuration in transforms that you create for Windows Installer–based applications or within packages that you create when repackaging older applications. Configuring older applications is usually as easy as importing Registration Entries (.reg) files on the destination computer after deployment.

ACT 5.5 provides data organization features that supersede the application inventory templates in earlier versions of MDT. With ACT 5.5, you can categorize applications a number of ways: by priority, risk, department, type, vendor, complexity, and so on. You can also create your own categories for organizing the application inventory. For more information, see Chapter 5.

Priorities

After creating an application inventory, the next step is to prioritize the list. Prioritizing the application inventory is not a task that you perform unilaterally. Instead, you will want to involve other team members, management, and user representatives in the review of priorities.

The priority levels you choose to use might include the following:

- **High** High-priority applications are most likely mission-critical or core applications. These are applications that are pervasive in the organization or are complex and must be addressed first. Examples of high-priority applications include virus scanners, management agents, Microsoft Office, and so on.
- **Medium** Medium-priority applications are nice to have but not essential. These are applications that are not as pervasive or complex as high-priority applications. For example, a custom mailing-list program might be a medium-priority application, because you can replicate the functionality in another application. To test whether an application is indeed a medium priority, answer this question: What's the worst that would happen if all the high-priority applications are deployed, but not this application? If you foresee no major consequences, the application is a medium priority.
- **Low** Low-priority applications are applications that deserve no attention in the process. Examples of low-priority applications are duplicate applications, applications that users have brought from home and installed themselves, and applications that are no longer in use. When prioritizing an application as low, record the reason for that status in case you must defend the decision later.

Prioritizing the application list helps you focus on the applications in an orderly fashion. Within each priority, you can also rank applications by order of importance. Ranking applications in an organization using thousands of applications is a foreboding task, however. Instead, you might want to rank only the high-priority applications or repeat the prioritization process with only the high-priority applications.

Categories

After prioritizing the application list, you must categorize each high- and medium-priority application. You can drop the low-priority applications from the list, as you have no intention of addressing them. The following categories help you determine the best way to deploy an application:

- **Core applications** Core applications are applications common to most of the computers in the organization (typically 80 percent or more) or applications that must be available the first time you start a computer after installing the operating system. For example, virus scanners and security software are usually core applications because they must run the first time you start the computer. Mail clients are core applications because they are common to all users and computers. The following list contains specific examples of what most organizations might consider core applications:
  - Adobe Acrobat Reader
  - Corporate screen savers
  - Database drivers and connectivity software
  - Macromedia Flash Player
  - Macromedia Shockwave
  - Microsoft Office
  - Network and client management software, such as OpenManage clients
  - Terminal emulation applications, such as TN3270
  - Various antivirus packages
  - Various Windows Internet Explorer plug-ins
  - Various Microsoft Office Outlook plug-ins
- **Supplemental applications** Supplemental applications are applications that aren't core applications. These are applications that are not common to most computers in the organization (department-specific applications) and aren't required when you first start the computer after installing a new operating system image. Examples of supplemental applications include applications that are department specific, such as accounting software, or role specific, such as dictation software. The following list contains examples of what most organizations consider supplemental applications:
  - Microsoft Data Analyzer 3.5
  - SQL Server 2005 Client Tools
  - Microsoft Visual Studio 2005 and Visual Studio 2008
  - Various Computer-Aided Design (CAD) applications
  - Various Enterprise Resource Planning (ERP) systems

Installation Methods

For each high- and medium-priority application, you must determine the best way to install it. For each, consider the following:

- **Automatic installation** Most applications provide a way to install automatically. For example, if the application is a Windows Installer package file (with the .msi file extension), you can install the application automatically. The section titled "Automating Installation" later in this chapter describes how to install applications packaged with common technologies automatically. In this case, you don't need to repackage the application unless you want to deploy a configuration that isn't possible otherwise.
- **Repackaged application** If an application does not provide a way to install automatically, you can repackage it to automate and customize installation by using one of the packaging technologies described in the section titled "Repackaging Legacy Applications" later in this chapter. Repackaging applications is a complex process and is quite often the most costly and tedious part of any deployment project. Make the decision to repackage applications only after exhausting other possibilities. Doing so requires technical experience with repackaging applications or using third-party companies to repackage the application for you.
- **Screen scraping** You can automate most applications with interactive installers by using a tool that simulates keystrokes, such as Windows Script Host. (See the section titled "Windows Script Host" later in this chapter for more information.) Understand that this method is more of a hack than a polished solution, but sometimes you're left with no other choice. Occasionally, the installation procedure may require the user to use the mouse or otherwise perform some complex task that cannot be automated easily. In these circumstances, automating the installation process may not be feasible.

For each application, record the installation method. Does the application already support automated installation? If so, record the command required to install the application. Are you required to repackage the application? If so, record the packaging technology you'll use and the command required to install the application. If you will use screen scraping to install the application, indicate that decision in the application inventory.

Subject Matter Experts

In a small organization with a few applications, you might know them all very well. In a large organization with thousands of applications, you will know very few of them well enough to make good decisions about repackaging applications. Therefore, for each application you must identify a SME. This SME should be an expert with the application, having the most experience with it. In other words, each application's SME will have insight into how the organization installs, configures, and uses that application. The SME will know the application's history and where to find the application's source media. Record the name and e-mail alias of each application's SME in the application inventory.

Configurations

During planning, with the SME's help, you should review each application and record the following:

- The location of the installation media. Often, the SME is the best source of information about the location of the source media, such as CDs, disks, and so on.
- Settings that differ from the application's default settings that are required to deploy the application in a desired configuration.
- External connections. For example, does the application require a connection to a database, mainframe, Web site, or other application server?
- Constraints associated with the application.
- Deployment compatibility. Is the application compatible with disk imaging and Sysprep? Is the application compatible with 32-bit systems? 64-bit systems?
- Application dependencies. Does the application depend on any patches or other applications?

Choosing a Deployment Strategy

Most companies share a common goal: create a corporate-standard desktop configuration based on a common image for each operating system version. They want to apply a common image to any desktop in any region at any time and then customize that image quickly to provide services to users.

In reality, most organizations build and maintain many images—sometimes even hundreds of images. By making technical and support compromises and disciplined hardware purchases, and by using advanced scripting techniques, some organizations have reduced the number of images they maintain to between one and three. These organizations tend to have the sophisticated software distribution infrastructures necessary to deploy applications—often before first use—and keep them updated.

Business requirements usually drive the need to reduce the number of images that an organization maintains. Of course, the primary business requirement is to reduce ownership costs. The following list describes costs associated with building, maintaining, and deploying disk images:

- **Development costs** Development costs include creating a well-engineered image to lower future support costs and improve security and reliability. They also include creating a predictable work environment for maximum productivity balanced with flexibility. Higher levels of automation lower development costs.
- **Test costs** Test costs include testing time and labor costs for the standard image, the applications that might reside inside it, and those applications applied after deployment. Test costs also include the development time required to stabilize disk images.

- **Storage costs** Storage costs include storage of the deployment shares, disk images, migration data, and backup images. Storage costs can be significant, depending on the number of disk images, number of computers in each deployment run, and so on.
- **Network costs** Network costs include moving disk images to deployment shares and to desktops.

As the size of image files increases, costs increase. Large images have more updating, testing, distribution, network, and storage costs associated with them. Even though you update only a small portion of the image, you must distribute the entire file.

Thick Images

Thick images are monolithic images that contain core applications and other files. Part of the image-development process is installing core applications prior to capturing the disk image, as shown in Figure 8-1. To date, most organizations that use disk imaging to deploy operating systems are building thick images.

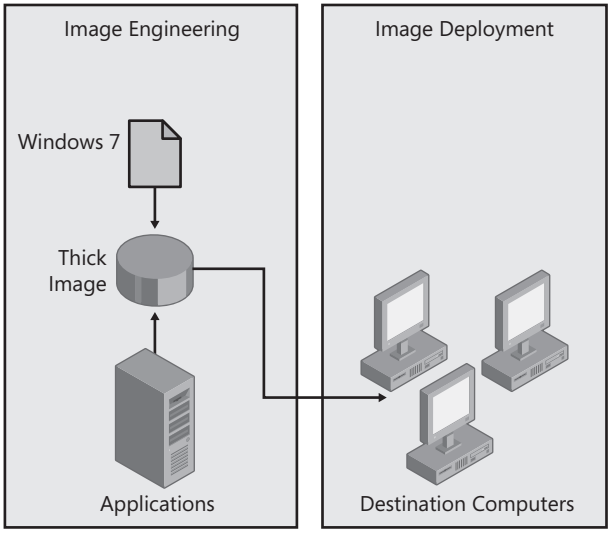


FIGURE 8-1 The thick image process

The advantage of thick images is deployment speed and simplicity. You create a disk image that contains core applications and thus have only a single step to deploy the disk image and core applications to the destination computer. Thick images also can be less costly to develop, as advanced scripting techniques are not often required to build them. In fact, you can build thick images by using MDT 2010 with little or no scripting work. Finally, in thick images, core applications are available on first start.

The disadvantages of thick images are maintenance, storage, and network costs, which rise with thick images. For example, updating a thick image with a new version of an application

requires you to rebuild, retest, and redistribute the image. Thick images require more storage and use more network resources in a short span of time to transfer.

If you choose to build thick images that include applications, you will want to install the applications during the disk-imaging process. In this case, see the following sections later in this chapter:

- See "Automating Installation" to learn how to install applications silently.
- See "Injecting in a Disk Image" to learn how to add applications to the deployment shares you create by using MDT 2010 and capturing them in a disk image.

Thin Images

The key to reducing image count, size, and cost is compromise. The more you put in an image, the less common and bigger it becomes. Big images are less attractive to deploy over a network, more difficult to update regularly, more difficult to test, and more expensive to store. By compromising on what you include in images, you reduce the number you maintain and you reduce their size. Ideally, you build and maintain a single, worldwide image that you customize post-deployment. A key compromise is when you choose to build *thin images*.

Thin images contain few if any core applications. You install applications separately from the disk image, as shown in Figure 8-2. Installing the applications separately from the image usually takes more time at the desktop and possibly more total bytes transferred over the network, but spread out over a longer period of time than a single large image transfer. You can mitigate the network transfer by using trickle-down technology that many software distribution infrastructures provide, such as Background Intelligent Transfer Service (BITS).

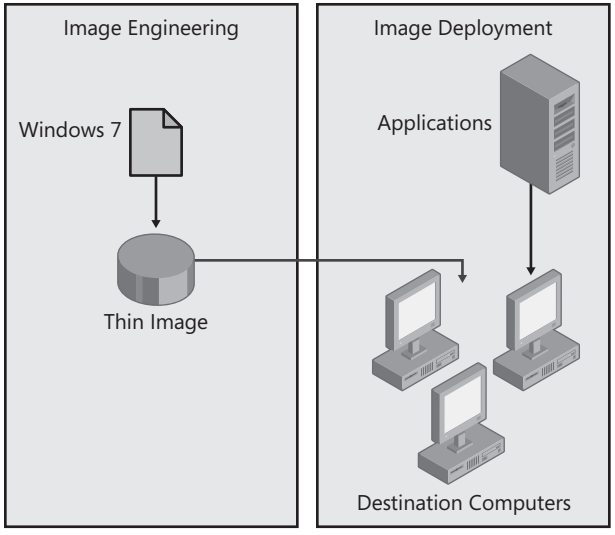


FIGURE 8-2 The thin image process

Thin images have many advantages. First, they cost less to build, maintain, and test. Second, network and storage costs associated with the disk image are lower because the image file is physically smaller. The primary disadvantage of thin images is that postinstallation configuration can be more complex to develop initially, but this is offset by the reduction in costs to build successive images. Deploying applications outside the disk image often requires scripting and usually requires a software distribution infrastructure. Another disadvantage of thin images is that core applications aren't available on first start, which might be necessary in high-security scenarios.

If you choose to build thin images that do not include applications, you should have a systems-management infrastructure, such as Microsoft System Center Configuration Manager 2007, in place to deploy applications. To use a thin image strategy, you will use this infrastructure to deploy applications after installing the thin image. You can also use this infrastructure for other postinstallation configuration tasks, such as customizing operating system settings.

Hybrid Images

Hybrid images mix thin- and thick-image strategies. In a hybrid image, you configure the disk image to install applications on first run, giving the illusion of a thick image but installing the applications from a network source. Hybrid images have most of the advantages of thin images. However, they aren't as complex to develop and do not require a software distribution infrastructure. They do require longer installation times, however, which can raise initial deployment costs.

An alternative is to build one-off thick images from a thin image. In this case, you build a reference thin image. After the thin image is complete, you add core applications and then capture, test, and distribute a thick image. Testing is minimized because creating the thick images from the thin image is essentially the same as a regular deployment. Be wary of applications that are not compatible with the disk-imaging process, however.

If you choose to build hybrid images, you will store applications on the network but include the commands to install them when you deploy the disk image. This is different than installing the applications in the disk image. You are deferring application installs that would normally occur during the disk-imaging process to the image-deployment process. They become a postinstallation task. Also, if you have a systems-management infrastructure in place, you will likely use it to install supplemental applications post-deployment. In this scenario, see the following sections of this chapter:

- See "Automating Installation" to learn how to install applications silently.
- See "Injecting in a Disk Image" to learn how to add applications to deployment shares you create by using MDT 2010 and install them during deployment.

Automating Installation

To achieve a fully automated deployment process, the packages you install must support unattended installation. Many setup programs support */s* or */q* command-line options for silent or quiet installations; others don't.

Often you can find out if the package supports unattended installation by typing ***setup /?*** at the command prompt, where *setup* is the file name of the setup program. If the setup program doesn't provide clues, you need to know which vendor's product was used to create the package. You can usually tell by running the setup program and looking for logos, for example, or checking the file properties. Armed with that information, read the following sections to learn how to install packages created by different packaging software automatically. Table 8-1 summarizes the necessary commands.

TABLE 8-1 Unattended Package Installation

PACKAGE TYPE	COMMAND FOR UNATTENDED INSTALLATION
Windows Installer	**msiexec.exe /i package.msi /qn ALLUSERS=2**
InstallShield Windows Installer	**setup.exe /s /v"/qn"** Optionally, you can extract the Windows Installer database from the compressed file and use the command **msiexec.exe /i setup.msi ISSETUPDRIVEN=1 /qn** to install it.
Legacy InstallShield	**setup.exe /s /sms** To create the Setup.iss file necessary to run setup silently, type **setup.exe /r** to create a Setup.iss from your responses to the setup program's dialog boxes and then copy Setup.iss from %SystemRoot% to the folder containing the package.
Legacy InstallShield PackageForTheWeb	**setup.exe /a /s /sms** To create the Setup.iss file necessary to run setup silently, type **setup.exe /a /r** to create the Setup.iss based on your responses and then copy Setup.iss from %SystemRoot% to the folder containing the package.
Legacy Wise Installation System	**setup.exe /s**

Useful Deployment Web Sites

The following Web sites are outstanding resources for automating the installation of applications, as well as other deployment topics:

- AppDeploy.com at *http://www.appdeploy.com*

  This Web site provides comprehensive information about deploying applications that are packaged using a variety of technologies.

- SourceForge at *http://unattended.sourceforge.net*

  This visually nondescript Web site contains a wealth of information, including information about automating the installation of many legacy installers.

- Real Men Don't Click at *http://isg.ee.ethz.ch/tools/realmen*

  Don't let the name or odd URL detract from this Web site's usefulness. It describes how to automate a variety of processes, including software installation.

- Acresso Software at *http://www.acresso.com/services/education/publications_3812.htm*

  This Web page contains the e-book *The Administrator Shortcut Guide to Software Packaging for Desktop Migrations*. This guide is an excellent resource for learning about packaging applications for deployment.

Windows Installer

Windows Installer is an installation and configuration service that helps reduce ownership costs by providing a component-based application installation architecture. Installation is consistent across all applications packaged for Windows Installer. Packages are easily customizable, installations are protected from errors, and a rollback mechanism provides for recovery in case of failure. Windows Installer supports application and feature advertising. Windows Installer provides many other benefits, and most Independent Software Vendors (ISVs) are now using it to package their applications. Windows 7 includes Windows Installer 5.0. For more information about its new features, see *http://msdn.microsoft.com/en-us/library/aa372796.aspx*.

Windows Installer 5.0 is compatible with User Account Control (UAC) in Windows 7. By using elevated installation, an administrator can authorize Windows Installer to install applications or security updates on behalf of users who aren't members of the Administrators group. For more information about UAC, see Chapter 24, "Managing Client Protection."

Windows Installer packages provide the following to enable flexible application deployment:

- **Command-line options** You use command-line options to specify options, file names, and path names, as well as control the action of the installation at run time.

- **Properties (variables) on the command line** Properties are variables that Windows Installer uses during an installation. You can set a subset of these, called public properties, on the command line.
- **Transforms** A transform is a collection of changes you can apply to a base Windows Installer package (.msi) file. You can customize applications by using Windows Installer transform (.mst) files. You configure transforms to modify a Windows Installer package to dynamically affect installation behavior according to your requirements. You associate transforms with a Windows Installer package at deployment time. Transforms for Windows Installer package files are similar to answer files that you might have used to automate the installation of an operating system such as Windows Vista.

The number of applications packaged as Windows Installer databases is multiplying rapidly. Nearly all software vendors are packaging their applications using this technology. And what often looks like a self-contained, self-extracting setup program with a file name such as Setup.exe is often a file that decompresses to a Windows Installer database. You can usually extract the database by using a tool such as WinZip (from WinZip Computing at *http://www.winzip.com*) or by running the setup program and looking in %UserProfile%\Local Settings\Temp for the package file. Windows Installer databases have the .msi file extension.

To install Windows Installer databases unattended using Msiexec.exe, use the */qb* command-line option for a basic user interface or the */qn* command-line option for no user interface. Also, to ensure that the package installs for all users, add the *ALLUSERS=2* property. For example, the command

```
msiexec.exe /i program.msi /qn ALLUSERS=2
```

installs the package file *Program*.msi with no user interaction and for use by all users who share the computer.

NOTE **You can learn more about Windows Installer at *http://msdn2.microsoft.com/en-us/library/aa372866.aspx*. For a list of command-line options, see *http://technet2.microsoft.com/WindowsServer/en/library/9361d377-9011-4e21-8011-db371fa220ba1033.mspx?mfr=true*.**

InstallShield

Some Windows Installer databases that Macrovision InstallShield (*http://www.acresso.com/products/is/installshield-overview.htm*) creates require that you install them by running Setup.exe. Trying to install the .msi file using Msiexec.exe results in a message that you must run Setup.exe to start the installation. When the developer uses InstallShield Script, this requirement is enforced to ensure that the needed version of the InstallShield Script Engine (ISScript.msi) is installed on the computer before proceeding. If it is not detected, the required version of InstallShield

Script Engine is installed automatically before starting Windows Installer. You can automate this installation a couple of ways:

- Use InstallShield's command-line support that Setup.exe offers. Not only does Setup.exe provide command-line option support, but you may also pass options to the Windows Installer setup database by using the */v* command-line option. Following */v,* you may specify any options you want to pass to the Windows Installer setup database within double quotation marks. For example, the following command installs the application silently and passes the */qn* option.

  ```
  setup.exe /s /v"/qn"
  ```

- Deploy the InstallShield Script Engine separately as part of your core applications before any setup files that require it. You may then safely bypass running Setup.exe by installing the Windows Installer setup database with Msiexec.exe and including the *ISSETUPDRIVEN* public property. You can extract the embedded Windows Installer setup database by looking in the %Temp% folder after the welcome message for the installation wizard is displayed. Then, use the following command to install it.

  ```
  msiexec.exe /i setup.msi ISSETUPDRIVEN=1 /qn
  ```

Legacy InstallShield

Packages created using legacy InstallShield technologies usually have the file name Setup.exe. To create an unattended installation for a legacy InstallShield package, you need to create an InstallShield script, which has the .iss file extension. Many applications come with such a file, but they are also easy to create.

To create an InstallShield response file, perform the following steps:

1. Run the setup program using the */r* command-line option. This creates a Setup.iss file based on how you configure the installation as you step through the setup program. The result is the file Setup.iss in %SystemRoot%.
2. Copy Setup.iss from %SystemRoot% to the folder containing the package.
3. Run the setup program using the */s* command-line option. The setup program runs silently using the responses provided by the Setup.iss file.

IMPORTANT **Packages created by InstallShield will spawn a separate process and then return immediately to the calling program. This means that the setup program runs asynchronously, even if you start the setup program using *start /wait*. You can add the */sms* command-line option to force the setup program to pause until installation is finished, however, making the process synchronous.**

Legacy InstallShield PackageForTheWeb

PackageForTheWeb is an InstallShield-packaged application contained in a self-contained, self-extracting file. You create a Setup.iss file and use it in almost the same way as described in the previous section. The difference is that you must use the */a* command-line option to pass the command-line options to the setup program after the file extracts its contents. For example, a file that you downloaded called Prog.exe will expand its contents into the temporary folder and then run Setup.exe when finished. To pass command-line options to Setup.exe, you must use the */a* command-line option. The following procedure demonstrates how this extra option changes the steps.

To create an InstallShield PackageForTheWeb response file, perform the following steps:

1. Run the setup program using the */a /r* command-line options: Type **setup.exe /a /r**. This creates a Setup.iss file based on the way you configure the installation as you step through the setup program. The Setup.iss file is in %SystemRoot%.
2. Copy Setup.iss from %SystemRoot% to the folder containing the package.
3. Run the setup program using the */a /s* command-line options: Type **setup.exe /a /s**. The setup program runs silently using the responses in the Setup.iss file.

Legacy Wise Installation System

Packages created using the legacy Wise Installation System recognize the */s* command-line option for unattended installation. No tool is available to script the installation, however.

Windows Script Host

Some applications cannot be automated with command-line options. These applications might provide a wizard-based setup routine but require the user to click buttons or press keys on the keyboard to install the application. If a user can complete the installation by using only the keyboard, you can automate the installation by creating a script (a series of text commands) that simulates keystrokes. This technique is called *screen scraping*.

You can screen scrape by using Windows Script Host. Specifically, you use the *SendKeys()* method to send keystrokes to an application. For more information about the *SendKeys()* method and an example that you can use to quickly create your own screen-scraping scripts, see *http://windowssdk.msdn.microsoft.com/en-us/library/8c6yea83.aspx*.

ON THE COMPANION MEDIA **The companion media contains the sample script Sendkeys.vbs, which provides a shell for using the *SendKeys()* method without having to write your own script. It accepts two command-line options: sendkeys.vbs *program textfile*, where *program* is the path and file name of the program you want to drive, and *textfile* is the path and file name of the text file containing the keystrokes, one keystroke per line, to send to the program. See *http://windowssdk.msdn.microsoft.com/en-us/library/8c6yea83.aspx* for a list of key codes. If you need to pause before sending more keystrokes, add a line to the file that contains *sleep*. Each line that contains *sleep* will pause for 1 second. The file Sendkeys.txt is a sample *textfile* you can use with Sendkeys.vbs; for example, type sendkeys.vbs notepad.exe sendkeys.txt and watch what happens.**

Repackaging Legacy Applications

Some legacy installers don't support silent installations, and some that do support silent installations don't provide a way to script settings. No legacy installers provide the management capabilities that Windows Installer provides.

If you have an application that is not designed for Windows Installer and does not support another automated installation technique, you can repackage it into the Windows Installer setup database so that you can use the features of Windows Installer to distribute and manage the application. A repackaged application combines the entire feature set of the application into a single feature. After repackaging an application, you use Windows Installer to install it. However, repackaged applications lack the flexibility to customize the application installation efficiently.

WARNING **Do not repackage Microsoft Office. The Office package files include logic that customizes the installation for the destination computer and user. Repackaging the package file loses this logic, potentially preventing the package from installing correctly in some configurations.**

The Repackaging Process

Windows Installer provides no functionality for repackaging applications. However, numerous vendors sell repackaging products for Windows Installer. See the next section, "Repackaging Tools," for a list of vendors.

Repackaging is not new. Organizations have historically repackaged applications to customize their installation and configuration. However, Windows Installer transforms eliminate the need to repackage Windows Installer–based applications just to customize them. In fact, repackaging applications that already install from a Windows Installer setup database is bad practice and is not supported.

Repackaging an application is a process that compares snapshots to determine the contents of the new package. The following steps provide an overview of the repackaging process:

1. Take a snapshot of the computer's current configuration.
2. Install the application.
3. Take a second snapshot of the computer's new configuration.
4. Create a package that contains the differences between the two snapshots. The repackaging tool detects all of the differences between the two snapshots, including all changes to the registry and file system. Because numerous processes are running in Windows 7 at any time, the package file will likely contain settings and files related to processes outside of the application.
5. Clean the package to remove noise (unnecessary files and settings).

WARNING **Don't let the simplicity of these five steps trick you into believing that repackaging is easy. Application repackaging is very often the most expensive part of any deployment project. When you undertake the repackaging of an organization's applications, you can count on a labor- and resource-intensive effort, particularly in organizations with thousands of applications, many of which the organization must repackage. Budget, plan, and schedule accordingly.**

Repackaging Tools

You must use tools that are not included with Windows Installer to create Windows Installer packages. The following list includes some of the variety of tools available:

- **AdminStudio** Available in multiple versions, including a free download, AdminStudio is a powerful and flexible repackaging tool. The following versions are available:
  - **AdminStudio Configuration Manager Edition** This free download from Microsoft integrates with System Center Configuration Manager 2007 to simplify repackaging. AdminStudio Configuration Manager Edition prepares legacy Setup.exe packages for deployment by converting them to Windows Installer .msi packages. To download AdminStudio Configuration Manager Edition, see *http://technet.microsoft.com/en-us/configmgr/bb932316.aspx*.
  - **AdminStudio Professional Edition** This full version of AdminStudio is a complete solution for packaging, customizing, testing, and distributing applications. The full version includes all the features included with AdminStudio Configuration Manager Edition, plus additional features. To download a trial version of AdminStudio, see the AdminStudio software overview page at *http://www.acresso.com/products/as/adminstudio-overview.htm*.

- **Wise Package Studio** Wise offers products for repackaging, testing, and configuring the deployment of applications. See *http://www.symantec.com/business/package-studio* for more information.

Injecting in a Disk Image

This section describes how to add applications to deployment shares you build with MDT 2010, and then inject those applications into disk images or install them when deploying the disk image. If you're not using MDT 2010 to build and deploy Windows 7, see Chapter 4, "Planning Deployment," to learn why using MDT 2010 is a better way to deploy Windows 7 than using the Windows Automated Installation Kit (Windows AIK) alone.

When planning application deployment, you choose between three deployment strategies: thick image, thin image, and hybrid image, as we described earlier in this chapter. If you're using a thin-image strategy, you won't be injecting applications into disk images. Instead, you'll use a systems-management infrastructure such as System Center Configuration Manager 2007 to deploy applications after installing the thin disk image. If you're using a thick-image strategy, you will install applications when you create the disk image. In other words, you will add the application installations to the MDT 2010 task sequence that you use to create the disk image. This method should be a last resort, as it's more difficult to maintain and slower to deploy. If you're using a hybrid image strategy, you will install applications during deployment. In this case, you will add the application installations to the MDT 2010 task sequence that you're deploying to destination computers, or you will add application installations to the MDT 2010 database.

NOTE **This chapter does not describe how to start or use Deployment Workbench. For more information about using Deployment Workbench, see Chapter 6, "Developing Disk Images."**

DIRECT FROM THE SOURCE

Infrastructure

Doug Davis, Lead Architect
Management Operations & Deployment, Microsoft Consulting Services

One question I hear repeatedly regarding deployment space concerns the amount of infrastructure required. Even with a moderately large (thick) image, customers still need to deploy additional applications. I typically suggest dynamic application distribution—applications that the user had before are dynamically reinstalled on the new configuration before the user logs on to the computer.

However, this requires a stable infrastructure. On average, three applications will need to be added for each computer—three applications not already included in the thick image. On average, 4,805 files per computer will be migrated by using the User State Migration Tool (USMT), and 900 megabytes (MB) will be transferred per computer. Therefore, a 1,000-computer deployment would require the following infrastructure:

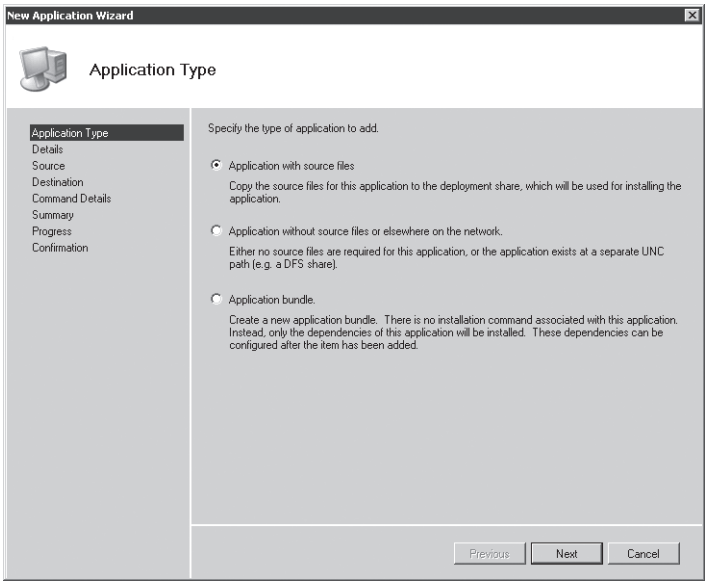
- **Computers: 1,000**
- **Applications: 2,952**
- **Files: 4,805,594**
- **Gigabytes: 977.60**

Adding Applications

When you add an application to a deployment share, you're simply describing for MDT 2010 how to install the application by using the command line and optionally copying the application source files to the deployment share. If you don't copy the application source files to the deployment share, MDT 2010 installs the application from the source location you specify, such as a network share.

To add an application to a deployment share, perform the following steps:

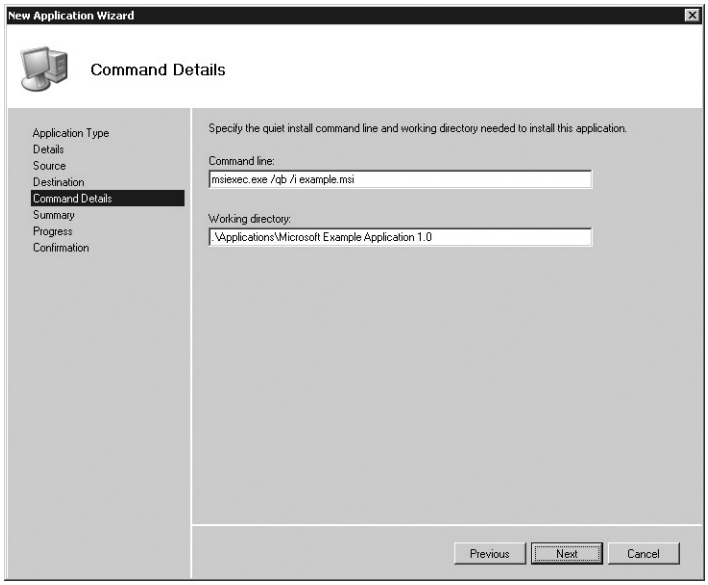
1. In the Deployment Workbench console tree, right-click Applications and then select New Application to begin the New Application Wizard. The Applications option is under Deployment Share. In MDT 2010, you must create a deployment share before adding applications to it. For more information about creating deployment shares, see Chapter 6.
2. On the Application Type page, do one of the following and then click Next:
   - Click the Application With Source Files option. Choosing this option copies the application source files to the deployment share. During deployment, MDT 2010 installs the application from source files it copied to the deployment share.
   - Click the Application Without Source Files Or Elsewhere On The Network option. Choosing this option does not copy the application source files to the deployment share. During deployment, MDT 2010 installs the application from another location on the network. You also choose this option to run a command that requires no application source files.
   - Click the Application Bundle option. This option creates essentially a dummy application with which you can associate other applications (dependencies). If you select the Application Bundle option during deployment, MDT 2010 will install all of its dependencies. For more information about dependencies, see the section titled "Creating Dependencies" later in this chapter.



3. On the Details page, provide the following information about the application and then click Next:
   a. In the Publisher box, type the name of the application's publisher (optional).
   b. In the Application Name box, type the name of the application.
   c. In the Version box, type a version label for the application (optional).
   d. In the Languages box, type the languages supported by the application (optional).
4. On the Source page, type the path of the folder containing the application you want to add and then click Next. If you've chosen to copy the application source files to the deployment share, Deployment Workbench copies everything in this folder to the deployment share; otherwise, it adds this path to the application's metadata as the application's installation path.

> ***NOTE*** **If you select the Move The Files To The Deployment Share Instead Of Copying Them check box, the New Application Wizard will move the source files instead of copying them. Use this option if you want to stage applications on the local hard disk before moving them into the deployment share.**

5. On the Destination page, type the name of the folder to create for the application within the deployment share and then click Next. The default value is the publisher, application name, and version label concatenated.
6. On the Command Details page, type the command to use to install the application silently, and then click Next. For example, type **msiexec /qb /i *program*.msi**. The command is relative to the working directory specified in the Working Directory box.



7. On the Summary page, review the application details and then click Next.
8. On the Confirmation page, click Finish.

After adding an application to the deployment share, you see it in the Applications details pane. You also see it in the deployment share in Applications*subfolder*, where *subfolder* is the destination you specified when adding the application.

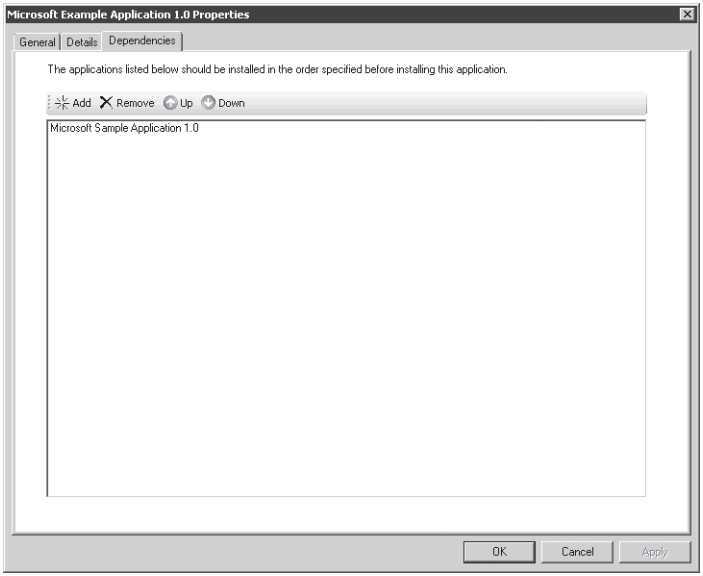
Creating Dependencies

Often, an application has dependencies. For example, application A is dependent on application B if you must install application B before installing application A. MDT 2010 allows you to specify application dependencies for each application you add to the deployment share. You can make an application dependent only on other applications that you've added to the deployment share.

To add dependencies to an application, perform the following steps:

1. In the Deployment Workbench console tree, click Applications.
2. In the details pane, right-click the application that has a dependency on another application and then click Properties.
3. On the Dependencies tab, shown on the following page, do the following:
   - To add an application to the dependencies list, click Add, select an application, and then click OK. Deployment Workbench only displays applications in this list that you've already added to the deployment share.
   - To remove an application from the dependencies list, select an application in the dependencies list and then click Remove.

- To reorder the applications in the dependencies list, select an application in the dependencies list and then click Up or click Down. MDT 2010 installs the dependent applications in the order specified by the dependencies list.



Installing Applications

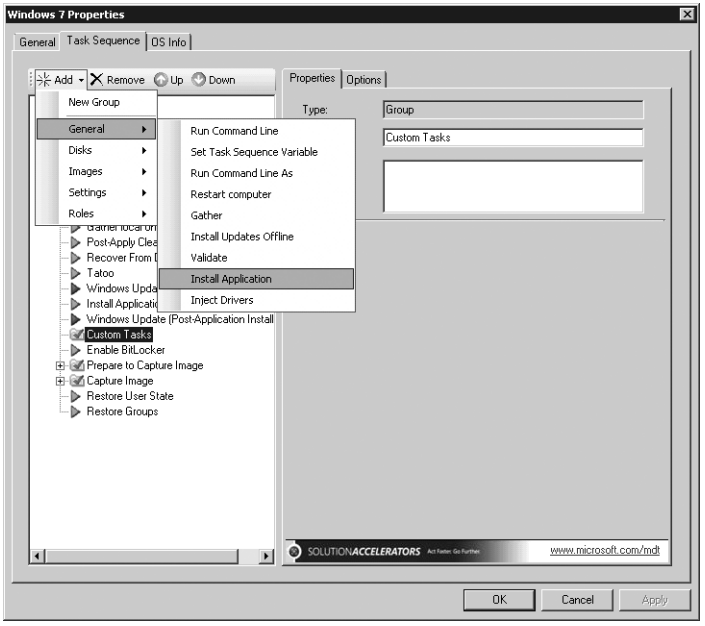
In MDT 2010, the task sequence specifies the tasks that run during deployment and their order. You can install applications during the imaging process by adding a step to the task sequence that installs the application at the appropriate time. For more information about customizing the task sequence, see Chapter 6. Although this approach is useful for injecting applications into a disk image, using the MDT 2010 database or CustomSettings.ini is more appropriate during deployment in production. For more information, see Chapter 12, "Deploying with Microsoft Deployment Toolkit."

Without creating additional groups in the task sequence, the best place to add application installs is to the Custom Tasks group, which MDT 2010 creates in each task sequence's default task sequence. The instructions in this section show you how to install an application as a step under this group.

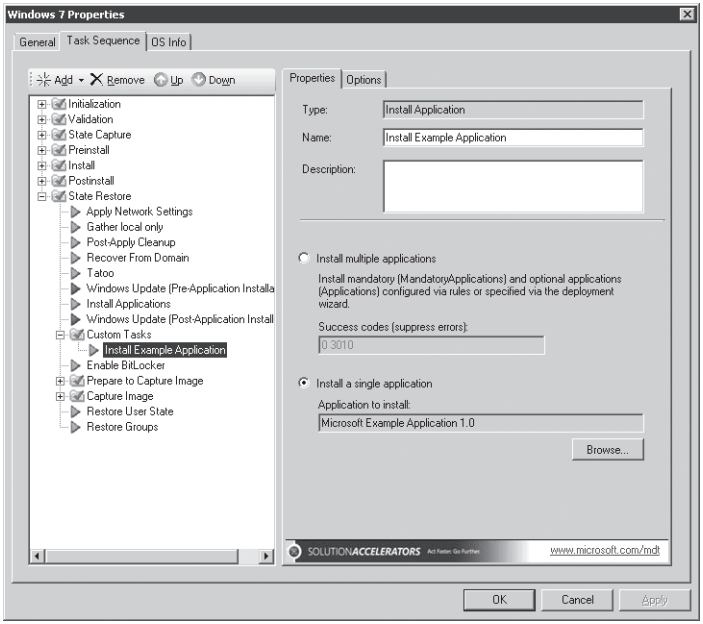
> ***NOTE*** **If you add an application to the deployment share without installing it via the task sequence, the Windows Deployment Wizard will allow the user to install the application optionally during deployment. Also, you can choose applications to install automatically during a Zero Touch Installation by configuring the deployment share to install the application automatically.**

To add an application installation to a task sequence, perform the following steps:

1. In the Deployment Workbench console tree, click Task Sequences, which is located under Deployment Share. In MDT 2010, you must create a deployment share before adding applications to it. For more information about creating deployment shares, see Chapter 6.
2. In the details pane, right-click the task sequence in which you want to install an application and then click Properties.
3. On the Task Sequence tab, shown here, click Custom Tasks in the task sequence and then click Add, click General, and then click Install Application.



4. Click the Install Application task that you just added to the task sequence, select the Install A Single Application option, click Browse, choose an application, and then click OK, as shown here.



NOTE **In MDT 2010, the task sequence is very flexible. For example, you can install applications at almost any point during the State Restore phase. You can filter application installation tasks on a variety of variables. For more information about editing task sequences in MDT 2010, see Chapter 6.**

Summary

Careful planning is the most important task you must undertake when deploying applications with Windows 7. The first step is building an application inventory. Then you must prioritize, categorize, and document the installation of each application. MDT 2010 and ACT provide tools that help with this step.

Another key planning step is determining the right type of deployment strategy for your organization. Thick images are monolithic images that contain core applications and other files. They are large and costly to maintain and deploy. Thin images are bare images. You install applications post-deployment using a systems-management infrastructure, such as System Center Configuration Manager 2007. Hybrid images use a combination of both strategies. The deployment strategy you choose determines how you build images.

After careful planning, you repackage the applications that don't provide an automated installation and document the installation commands for those that do. Then add applications to your MDT 2010 deployment share and add steps to the task sequence that installs the

application when you build the disk image (thick image) or when you deploy the disk image (hybrid image).

> ***NOTE*** **If you're not using MDT 2010 to deploy Windows 7, see Chapter 4 to learn why using MDT 2010 is a better way to deploy Windows 7 than using the Windows AIK alone. If you're not using MDT 2010, see *Windows Automated Installation Kit User's Guide* to learn how to install applications by using an answer file.**

Additional Resources

These resources contain additional information and tools related to this chapter.

Related Information

- Chapter 2, "Security in Windows 7," includes more information about how Windows 7 security features affect applications.
- Chapter 5, "Testing Application Compatibility," describes how to use ACT 5.5 to create an application inventory, analyze it, and then mitigate compatibility issues.
- Chapter 6, "Developing Disk Images," includes more information about building custom Windows 7 disk images that include applications.
- Chapter 7, "Migrating User State Data," includes more information about migrating application settings from earlier versions of Windows to Windows 7.
- The "2007 Office Resource Kit," found at *http://technet.microsoft.com/en-us/library/cc303401.aspx*, includes more information about customizing and deploying the 2007 Microsoft Office system.
- "SendKeys Method," found at *http://windowssdk.msdn.microsoft.com/en-us/library/8c6yea83.aspx*, includes more information about using Windows Script Host as a screen-scraping tool to automate application installations.
- "Application Compatibility," found at *http://technet.microsoft.com/en-us/windows/aa905066.aspx*, includes more information about downloading and using ACT to resolve compatibility issues.

On the Companion Media

- Sendkeys.vbs
- Sendkeys.txt

CHAPTER 9

Preparing Windows PE

Half the job of installing the Windows 7 operating system or building disk images is starting the computer and preparing for installation. You use Windows Preinstallation Environment (Windows PE) 3.0 to start computers, which is similar to using MS-DOS in the old days. Windows PE allows you to fully automate the preparation and installation process. This chapter describes how to use, customize, and automate Windows PE for the purpose of installing Windows 7 in business environments.

Earlier versions of Windows PE, including Windows PE 2004 and Windows PE 2005, were available only to Software Assurance (SA) customers. Windows 7 installation is entirely based on Windows PE and imaging by using ImageX; therefore, Windows PE 3.0 is freely available as part of the Windows Automated Installation Kit (Windows AIK) 2.0. Windows PE is highly customizable. You can use the *Windows PE User's Guide*, included in the Windows AIK, to accomplish most tasks. This chapter describes the most common ways to customize Windows PE, as well as how to start it in various scenarios.

In most circumstances, you should use Microsoft Deployment Toolkit 2010 (MDT 2010) to deploy Windows 7. In this case, you can use Deployment Workbench to customize Windows 7 and automatically generate images that you can use to start Windows PE with a variety of media. Although the information in this chapter does describe how to customize Windows PE manually, Microsoft recommends that you use MDT 2010 to generate Windows PE images in most cases.

Exploring Windows PE

Windows PE, which is supplied with Windows 7 and in the Windows AIK, is the installation engine for Windows 7. It is directly bootable from CD, DVD, and universal serial bus (USB) flash drives (UFDs). You can also start Windows PE by using Windows Deployment Services and the Pre-Boot Execution Environment (PXE) extensions to Dynamic Host Configuration Protocol (DHCP) (if supported by the network adapters of your computers).

Windows PE is a minimal Windows operating system that provides limited services based on the Windows 7 kernel. It also provides the minimal set of features required to run Windows 7 Setup, install Windows 7 from networks, script basic repetitive tasks, and validate hardware. For example, with Windows PE, you can use powerful batch scripts, Windows Script Host (WSH) scripts, and HTML Applications (HTAs) to fully automate computer preparation and Windows 7 installation, rather than the limited batch commands in MS-DOS. Examples of what you can do with Windows PE include:

- Create and format disk partitions, including NTFS file system (NTFS) partitions, without rebooting the computer before installing Windows 7 on them. Formatting disks with NTFS by using an MS-DOS–bootable disk required third-party utilities. Windows PE replaces the MS-DOS–bootable disk in this scenario, allowing you to format disks with NTFS without using third-party utilities. Also, the file system utilities that Windows PE provides are scriptable, so you can completely automate the setup preparation process.
- Access network shares to run preparation tools or install Windows 7. Windows PE provides network access comparable to Windows 7. In fact, Windows PE provides the same network drivers that come with Windows 7, allowing you to access the network quickly and easily. Customizing MS-DOS–bootable disks to access network shares was time consuming and tedious.
- Use all the mass-storage devices that rely on Windows 7 device drivers. Windows PE includes the same mass-storage device drivers that Windows 7 provides, so you no longer have to customize MS-DOS–bootable disks for use with specialized mass-storage devices. Once again, Windows PE allows you to focus on important jobs rather than on maintaining MS-DOS–bootable disks.
- Customize Windows PE by using techniques and technologies that are already familiar to you. Windows PE is based on Windows 7, so you are already familiar with the techniques and tools used to customize Windows PE. You can customize it in a variety of scenarios:
  - Addition of hardware-specific device drivers
  - Automation through use of Unattend.xml answer files
  - Execution of scripts (batch, WSH, and HTA) to perform specific actions

The following sections provide more detail about the features and limitations of Windows PE. They focus specifically on using Windows PE in high-volume deployment scenarios, rather than in manufacturing environments.

DIRECT FROM THE SOURCE

Windows PE 3.0

Michael Niehaus, Lead Developer for Microsoft Deployment Toolkit
Management and Infrastructure Solutions

Windows PE 3.0, the new version that will be released with Windows 7, is an important part of the deployment process. Even the standard DVD-based installation of Windows 7 uses Windows PE 3.0, and most organizations will be using it (often customized for the organization's specific needs) as part of their deployment processes.

Compared to MS-DOS–based deployment, Windows PE 3.0 brings numerous benefits, including less time spent trying to find 16-bit real-mode drivers. (It's not even possible to find these any more for some newer network cards and mass storage adapters.) Better performance from 32-bit and 64-bit networking stacks and tools, as well as large memory support, are also advantages. And don't forget support for tools such as WSH, VBScript, and hypertext applications.

Windows PE has been available for a few years (the latest version, Windows PE 2.1, was released at the same time as Windows Vista and Windows Server 2008). Previous versions required you to have SA on your Windows desktop operating system licenses. With Windows PE 3.0, that's not the case. All organizations will be able to download Windows PE 3.0 from *http://www.microsoft.com* and use it freely for the purposes of deploying licensed copies of Windows 7.

Like Windows 7 itself, Windows PE 3.0 is provided as an image that is modular and can be serviced both online and offline. As with Windows PE 2.1, several optional features can be added. New tools like Deployment Image Servicing and Management (DISM) are provided for servicing Windows PE 3.0. You can use DISM to add packages and drivers, including mass storage devices, which no longer require any special handling.

Capabilities

Windows PE is a bootable image that you can start by using removable media (CD, DVD, or UFD). You can also use Windows Deployment Services to start Windows PE. Because the Windows 7 deployment tools do not work in 16-bit environments, Windows PE replaces the MS-DOS–bootable disk in *all* deployment scenarios. It's a lightweight 32-bit or 64-bit environment that supports the same set of networking and mass-storage device drivers that Windows 7 supports, and it provides access to similar features, including NTFS and standalone Distributed File System (DFS). Windows PE includes the following features:

- **Hardware independence** Windows PE is a hardware-independent Windows environment for both x86 and x64 architectures. You can use the same preinstallation environment on all desktop computers and servers without creating and maintaining different bootable disks for different hardware configurations.
- **APIs and scripting capabilities** Windows PE contains a subset of the Win32 application programming interfaces (APIs); a command interpreter capable of running batch scripts; and support for adding WSH, HTA, and Microsoft ActiveX Data Objects to create custom tools or scripts. The scripting capabilities in Windows PE far exceed the capabilities of MS-DOS–bootable disks. For example, the command interpreter in Windows PE supports a more robust batch-scripting language than does MS-DOS, allowing you to use more advanced scripts.
- **Network access** Windows PE uses Transmission Control Protocol/Internet Protocol (TCP/IP) to provide network access and supports standard network drivers for running Windows 7 Setup and installing images from the network to the computer. You can easily add or remove network drivers from a customized version of Windows PE. In contrast, customizing MS-DOS–bootable disks to access network shares is frustrating, mostly because you need to build and maintain numerous disks. Windows PE alleviates this frustration by supporting the network drivers that Windows 7 supports, and Windows PE is easier to customize with additional network drivers.
- **Mass-storage devices** Windows PE includes support for all mass-storage devices that Windows 7 supports. As new devices become available, you can easily add or remove drivers into a customized version of Windows PE. Customizing an MS-DOS–bootable disk to access atypical mass-storage devices requires tracking down and installing the 16-bit device drivers. However, Windows PE supports many of these mass-storage devices out of the box. And customizing Windows PE to support additional mass-storage devices is easier because it uses standard, readily available Windows device drivers.
- **Disk management** Windows PE includes native support for creating, deleting, formatting, and managing NTFS partitions. Also, Windows PE provides full, unrestricted access to NTFS file systems. With Windows PE, you don't have to restart the computer after formatting a disk.
- **Support for the PXE protocol** If the computer supports PXE, you can start it automatically from a Windows PE image located on a Windows Deployment Services server—and Windows Deployment Services doesn't install the Windows PE image on the computer's hard disk. Starting Windows PE from the network makes it a convenient tool to use in all deployment scenarios. Also, you can customize a Windows PE image for recovery and troubleshooting purposes, and adding it to Windows Deployment Services makes it a convenient tool to use in production.

NOTE **You must build a custom Windows PE image from the Windows PE source files, as described in the section titled "Customizing Windows PE" later in this chapter.**

You manage and deploy Windows PE by using the tools included in Windows 7 and the Windows AIK. This toolkit includes the *Windows PE User's Guide* and tools such as:

- **BCDboot.exe** Provides initialization of the boot configuration data (BCD) store, and it enables you to copy boot environment files to the system partition during image deployment.
- **Bootsect.exe** Updates the master boot code for hard disk partitions to alternate between BOOTMGR and NTLDR. This enables you to preinstall Windows 7 from Windows XP.
- **DiskPart.exe** A text-mode command interpreter in Windows 7 that enables you to manage disks, partitions, or volumes by using scripts or direct input at a command prompt.
- **Drvload.exe** A command-line tool for adding out-of-the-box drivers to a booted Windows PE image. It takes one or more driver .inf files as inputs.
- **Oscdimg.exe** A command-line tool for creating an image (.iso) file of a customized 32-bit or 64-bit version of Windows PE. You can then burn the .iso file to a CD-ROM.
- **Dism.exe** A command-line tool that can create and modify a Windows PE 3.0 or Windows 7 image.
- **ImageX.exe** A command-line tool that enables you to capture, modify, and apply file-based disk images for rapid deployment. It can also work with other technologies that use .wim files, such as Setup for Windows 7 and Windows Deployment Services.
- **Winpeshl.ini** The default interface for Windows PE is a command prompt. You can customize Winpeshl.ini to run your own shell application.
- **Wpeinit.exe** A command-line tool that initializes Windows PE every time it boots. Wpeinit replaced the initialization function previously supported by the *Factory.exe -winpe* command in earlier versions of Windows PE.
- **Wpeutil.exe** A command-line tool that enables you to run various commands in a Windows PE session.

NOTE **The *Windows PE User's Guide* (Winpe.chm) provides complete, portable documentation of the command-line options for all of the tools discussed in this chapter. This Help file is located in the Windows AIK 2.0, which you can download from the Microsoft Download Center at *http://www.microsoft.com/downloads*.**

Limitations

Windows PE has the following limitations:

- To reduce its size, Windows PE includes only a subset of the available Win32 APIs: I/O (disk and network) and core Win32 APIs.

- Windows PE doesn't fit on floppy disks, but you can write a custom Windows PE image to a bootable CD or DVD.
- Windows PE supports TCP/IP and NetBIOS over TCP/IP for network connectivity, but it doesn't support other protocols, such as Internetwork Packet Exchange/Sequenced Packet Exchange (IPX/SPX).
- The Windows on Windows 32 (WOW32) subsystem allows 16-bit applications to run on the 32-bit Windows platform. The WOW32 subsystem isn't available in Windows PE, so 16-bit applications won't run in 32-bit versions of Windows PE. Similarly, in the x64 version of Windows PE, the Windows on Windows 64 (WOW64) subsystem is not available, so applications must be fully 64-bit compliant.
- To install 64-bit Windows 7, you must use 64-bit Windows PE. Likewise, installing 32-bit Windows 7 requires 32-bit Windows PE.
- Drive letter assignments aren't persistent between sessions. After you restart Windows PE, the drive letter assignments will be in the default order.
- Changes to the registry aren't persistent between sessions. To make permanent changes to the registry, you must edit the registry offline by mounting the image with ImageX and then loading hive files into Registry Editor.
- Windows PE supports DFS name resolution to stand-alone DFS roots only.
- You can't access files or folders on a computer running Windows PE from another computer. Likewise, Windows PE can't act as a terminal server, so you can't connect to it by using Remote Desktop.
- Windows PE requires a VESA (Video Electronics Standards Association)-compatible display device and will use the highest screen resolution that it can determine is supported. If the operating system can't detect video settings, it uses a resolution of 640 by 480 pixels.
- Windows PE doesn't support the Microsoft .NET Framework or the Common Language Runtime (CLR).
- Windows PE does not support the installation of Windows Installer package (.msi) files.
- Windows PE does not support 802.1x.
- To prevent its use as a pirated operating system, Windows PE automatically reboots after 72 hours.

New Features of Windows PE 3.0

The following features are changes to Windows PE 3.0 since Windows PE 2.1, the version that shipped with Windows Vista:

- **Deployment Image Servicing and Management (DISM)** DISM is a new command-line tool that you can use to customize a Windows PE 3.0 image offline. DISM replaces Pkgmgr.exe, Intlcfg.exe, and PEImg.exe.

- **Smaller default size** Windows PE version 2.1 contains staged optional features that add additional size and required additional effort to remove. The Windows PE 3.0 default image contains only the minimum resources to support most deployment scenarios. You can add optional features by using DISM.
- **Serviceable** In Windows Vista, you cannot modify a Windows PE image after running *peimg /prep* against it. Windows PE 3.0 images can be serviced at any time by using DISM. The PEImg.exe tool is not supported in Windows PE 3.0.
- **Image optimization** Previous versions of Windows PE provide limited support for optimizing (reducing) the size of an image. Using the new DISM */apply-profiles* command, you can reduce the contents of a Windows PE 3.0 image to only those files necessary to support a given set of applications.
- **System drive letter** You can assign any letter to the system drive using the new DISM */Set-TargetPath* command.
- **Mounted images** Windows PE 3.0 supports mounting a Windows Imaging (WIM) file.
- **Hyper-V support** Windows PE 3.0 includes all Hyper-V drivers except display drivers. This enables Windows PE to run in Hyper-V. Supported features include mass storage, mouse integration, and network adapters.
- **Customizable scratch space** You can now customize the RAM scratch space as 32, 64, 128, 256, or 512 megabytes (MB).

Setting Up the Environment

You will need to build an environment for customizing Windows PE images before deployment. Having everything in the appropriate location will simplify the task of creating builds and will help you establish repeatable methods for creating and updating builds.

Create this environment on a technician or lab computer. If you're using MDT 2010 to deploy Windows 7, configure the Windows PE customization environment on the build server. In fact, installing and configuring MDT 2010 installs all of the requirements for building custom Windows PE images.

Installing the Windows AIK 2.0

Windows PE 3.0 ships with the Windows AIK 2.0, which is available from the Microsoft Download Center at *http://www.microsoft.com/downloads*. Install the Windows AIK on your Windows PE build system from the installation DVD. (Microsoft provides the Windows AIK as a downloadable .iso image.) Installing the Windows AIK 2.0 is a requirement for installing and using MDT 2010. Therefore, a build server containing MDT 2010 already has the files necessary to build and customize Windows PE images. For more information about installing MDT 2010 and the Windows AIK, see Chapter 4, "Planning Deployment." Windows 7 and Windows Server 2008 R2 already contain all of the software prerequisites for the Windows AIK 2.0.

To install the Windows AIK, perform the following steps:

1. From the Windows AIK media or a folder containing the Windows AIK, run **waik*platform*.msi,** where *platform* is either x86 or amd64.
2. Accept the end-user license agreement (EULA) and choose the default location for the installation files. You must use the default installation location if you're using MDT 2010. The examples in this chapter are based on a default Windows AIK installation.
3. Complete the installation wizard to install Windows AIK.

NOTE **The Windows Installer file Waik*platform*.msi includes the Windows AIK tools. The file Winpe.cab actually includes the Windows PE source files. To install Windows PE, Winpe.cab must be in the same folder as the .msi file.**

Configuring the Build Environment

The Windows AIK will install the Windows PE build and imaging tools to the following folders:

- **C:\Program Files\Windows AIK\Tools** Contains Windows AIK program files
- **C:\Program Files\Windows AIK\Tools*platform*** Contains ImageX program files for different processor architectures
- **C:\Program Files\Windows AIK\Tools\PETools** Contains Window PE source files
- **C:\Program Files\Windows AIK\Tools\Servicing** Contains servicing files

The Windows AIK also provides a Deployment Tools command prompt that opens on the Windows AIK tools folders (shown in Figure 9-1). You can use commands within this command prompt interface to create your Windows PE build environment. The build environment is a copy of the build scripts and Windows PE source files that you customize and then use to create a new Windows PE image file. The Copype.cmd script is designed to create the build environment. Use the following syntax to create the Windows PE environment, where *platform* is either x86 or amd64 and *destination* is the folder to which you want to copy the files.

```
copype.cmd platform destination
```

NOTE **To follow the examples in this chapter, run the command *copype x86 c:\winpe_x86*. You can use an alternative location for your build environment, but you will need to make the appropriate modifications to the examples provided with this chapter. Additionally, you can replace x86 with x64 if you want to build a 64-bit version of Windows PE, which is necessary when you are installing the 64-bit version of Windows 7.**

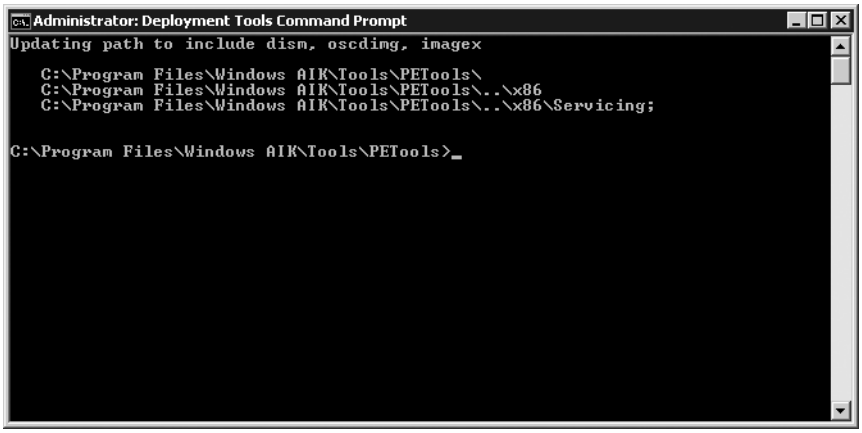


FIGURE 9-1 Use the Windows PE Tools command prompt to work with Windows PE.

Removing the Build Environment

When DISM installs features, it can modify the access control lists (ACLs) of the Windows PE build files and folders, making it difficult to remove them in the future. You can work around this by using the Windows Server 2008 tool Takeown.exe to take ownership of the affected resources.

To remove the Windows PE build environment, perform the following steps:

1. Take ownership of the folder structure using the *Takeown* command.

   ```
   takeown /F c:\winpe_x86\* /R
   ```

2. Use the *Change ACLs* (cacls) command to give yourself permission to remove the folders (*user* is your user account).

   ```
   cacls c:\winpe_x86\* /T /G user:F
   ```

3. Remove the folder.

   ```
   rd /s /q c:\winpe_x86\
   ```

Working with Windows PE

Most Windows PE tasks have just a few basic steps. Applications and customizations might vary the process somewhat, but the basic process is the same. This section gives you an overview of the Windows PE build process. In later sections, you learn how to customize Windows PE in greater depth.

Mounting Windows PE

After you create the Windows PE build environment, the first step in customizing the Windows PE–based image is to mount it so that you can service it by using DISM. An example of the command to mount the base image is shown here (where 1 is the image number within Winpe.wim to be mounted and C:*Winpe_x86**Mount* is the path on which to mount it):

```
Dism /Mount-Wim /WimFile:C:\Winpe_x86\Winpe.wim /Index:1 /MountDir:C:\Winpe_x86\Mount
```

> ***NOTE*** **With previous versions of Windows PE, you couldn't service the image after you ran *peimg /prep*. This is no longer true with Windows PE 3.0. You can now mount and service Windows PE images as required.**

Adding Packages

The next step is to add the packages that you require. You add packages by using the DISM */Add-Package* option. Additionally, for every feature you want to add to Windows PE, you must add a language-neutral package and a language-specific package. In a default installation of the Windows AIK, you find the language-neutral packages in the folder C:\Program Files\Windows AIK\Tools\PETools\x86\WinPE_FPs and the language-specific packages in C:\Program Files\Windows AIK\Tools\PETools\x86\WinPE_FPs*language*, where *language* is the language identifier (such as *en-US* for U.S. English).

To add a package to the Windows PE image, perform the following steps:

1. Look up in Table 9-1 the names of the packages that you want to install.
2. Add the language-neutral package to the Windows PE image by running the following command at the Deployment Tools command prompt.

   ```
   dism /image:C:\winpe_x86\mount /Add-Package /PackagePath:"C:\Program Files\Windows
   AIK\Tools\PETools\x86\WinPE_FPs\package.cab"
   ```

3. Add the language-specific package to the Windows PE image (look up the actual file name in C:\Program Files\Windows AIK\Tools\PETools\x86\WinPE_FPs\en-us).

   ```
   dism /image:C:\winpe_x86\mount /Add-Package /PackagePath:"C:\Program Files\Windows
   AIK\Tools\PETools\x86\WinPE_FPs\language\package_language.cab"
   ```

For example, the following two commands install the WinPE-Scripting language-neutral and language-specific packages into a Windows PE image mounted at C:\Winpe_x86\Mount.

```
dism /image:C:\winpe_x86\mount /Add-Package /PackagePath:"C:\Program Files\Windows AIK
\Tools\PETools\x86\WinPE_FPs\winpe-scripting.cab"

dism /image:C:\winpe_x86\mount /Add-Package /PackagePath:"C:\Program Files\Windows AIK
\Tools\PETools\x86\WinPE_FPs\en-us\winpe-scripting_en-us.cab"
```

After adding packages to the Windows PE image, you can verify them by listing the image's packages. To list the packages in a Windows PE image mounted to C:\Winpe_x86\Mount, run the following command at the Deployment Tools command prompt.

```
dism /image:c:\winpe_x86\mount /Get-Packages
```

TABLE 9-1 Windows PE Packages

PACKAGE	DESCRIPTION
WinPE-FONTSupport-<language>	Provides additional font support for the following languages: ja-JP, ko-KR, zh-CN, zh-HK, and zh-TW.
WinPE-HTA	Provides HTA support. Enables the creation of graphical user interface (GUI) applications using the Windows Internet Explorer script engine and HTML services.
Winpe-LegacySetup	Contains the Media Setup package. All Setup files from the \Sources folder on the Windows media. Add this package when servicing Setup or the \Sources folder on the Windows media. Must be added with the Setup package. To add a new Boot.wim to the media, add either child package in addition to the Setup and Media packages.
WinPE-MDAC	Provides Microsoft Data Access Component support. Enables queries to SQL servers with Active Directory Objects. For example, you can build a dynamic Unattend.xml file from unique system information.
WinPE-PPPoE	Enables Point-to-Point Protocol over Ethernet (PPPoE) support. Create, connect, disconnect, and delete PPPoE connections from Windows PE.
WinPE-Scripting	Provides WSH support. Enables batch file processing using WSH script objects.
WinPE-Setup	Contains the Setup package. All Setup files from the \Sources folder common to the client and server. This package is the parent package of *winpe-setup-client* and *winpe-setup-server*. You must install *winpe-setup* before you install the child packages.
WinPE-Setup-Client	Contains the Client Setup package. The client branding files for Setup. Must be added after the WinPE-Setup package.
WinPE-Setup-Server	Contains the Server Setup package. The server branding files for Setup. Must be added after the WinPE-Setup package.
WinPE-SRT	Contains the Windows Recovery Environment (Windows RE) package. Provides a recovery platform for automatic system diagnosis and repair and the creation of custom recovery solutions.

PACKAGE	DESCRIPTION
WinPE-WMI	Provides Windows Management Instrumentation (WMI) support. A subset of the WMI providers that enables minimal system diagnostics.
WinPE-WDS-Tools	Contains the Windows Deployment Services tools package. Includes APIs to enable a multicast scenario with a custom Windows Deployment Services client and Image Capture utility.

NOTE **Previous versions of Windows PE included pre-staged optional packages that you had to remove if you didn't want to include them in the image. These included the WinPE-HTA-Package, WinPE-Scripting-Package, and WinPE-MDAC-Package packages. Windows PE 3.0 does not include pre-staged optional features, helping reduce the footprint of Windows PE. You must add all of the packages that you require.**

Copying Applications

You can also copy applications into the Windows PE image so that you can use them during the Windows 7 implementation process. To copy an application into a Windows PE image, use operating system copy commands to copy the application to the appropriate location.

```
xcopy /chery myapp.exe "c:\winpe_x86\mount\program files\myapp\myapp.exe"
```

Adding Device Drivers

Windows PE can use Windows 7 device drivers to provide hardware support for Windows 7 installation processes. Use the DISM */Add-Drive* option to add device drivers to the Windows PE image (*inf_file* is the path and file name of the device driver's .inf file).

```
Dism /image:c:\winpe_x86\mount /Add-Driver /Driver:inf_file
```

Windows PE can also add device drivers dynamically when running. Use the Drvload.exe command to load device drivers while operating.

```
drvload.exe path[,path]
```

Installing Updates

You install updates to Windows PE using the same process by which you add features: You use the DISM */Add-Package* option. Run the following command at the Deployment Tools command prompt, where *update_file* is the path and file name of the update.

```
dism /image:C:\winpe_x86\mount /Add-Package /PackagePath:update_file
```

Committing the Changes

After customizing the mounted Windows PE image, you must dismount the image and commit your changes. This saves your changes into the Winpe.wim file that you mounted. Before dismounting the image, make sure you close any open files and Windows Explorer windows that might prevent you from successfully committing your changes.

To capture the Windows PE image, use the following command.

```
dism /unmount-Wim /MountDir:C:\winpe_x86\mount /Commit
```

Creating Bootable Media

Many Windows maintenance and troubleshooting utilities can make use of Windows PE, including utilities created for managing disks and recovering systems. Windows RE is one example of a recovery tool that uses Windows PE. Many other utilities created by third-party manufacturers also use Windows PE.

This section covers the creation of bootable Windows PE media based on CDs, DVDs, UFDs, and hard disks. You can use all of these technologies for Windows 7 deployment, creating an array of possible solutions for corporate deployments.

Staging a Boot Image

The Windows PE boot image needs supporting files to be made bootable. If you copy your Winpe.wim file to the ISO\Sources folder of the build directory and rename it to Boot.wim, you can create your bootable Windows PE by using the entire ISO folder hierarchy. A completed ISO folder hierarchy looks similar to Figure 9-2.

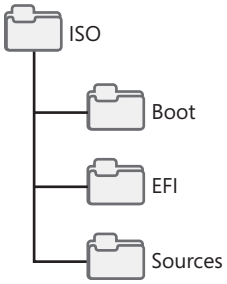


FIGURE 9-2 Windows PE ISO folder hierarchy

To stage a captured Windows PE boot image, copy Winpe.wim from c:\winpe_x86 to the c:\winpe_x86\ISO\Sources folder of the Windows PE build directory.

```
xcopy /chery c:\winpe_x86\Winpe.wim c:\winpe_x86\ISO\Sources\boot.wim
```

Creating Bootable CD/DVD Media

After the boot image is properly staged, you can create a bootable CD or DVD that uses your Windows PE image.

To create a bootable Windows PE CD or DVD, perform the following steps:

1. Use the Oscdimg.exe command to create an .iso image that can be burned onto a CD or DVD.

```
oscdimg -n -bc:\winpe_x86\etfsboot.com c:\winpe_x86\ISO c:\winpe_x86\winpe_x86.iso
```

2. Using a CD/DVD burning application, burn the .iso image to a CD or DVD.

Creating Bootable UFD Media

UFDs are available that have the capacity to hold an entire custom Windows 7 deployment. The first step, however, is to make your bootable Windows PE media. After you've accomplished this, you can copy any custom images and Unattend.xml files you have made to the UFD for deployment.

To create bootable Windows PE UFD media, perform the following steps:

1. Insert your bootable UFD device into an available USB port on your system.
2. Use the DiskPart utility to prepare the device for loading Windows PE. To run DiskPart, type **diskpart** at the command prompt and then press Enter.
3. Run the commands shown in Table 9-2 to prepare the UFD.

TABLE 9-2 Preparing a UFD for Windows PE

COMMAND	DESCRIPTION
list disk	Lists available disks.
select disk n	*n* is the UFD you are preparing. Be sure to select the correct disk when using DiskPart. DiskPart will clean your primary hard disk as easily as it will clean your UFD device.
clean	Removes the current partition structures.
create partition primary size=size	*size* is the size of the disk as shown in the list. If you omit *size*, DiskPart will use all of the available space for the partition.
select partition 1	Selects the partition you created in the previous command.
active	Marks the new partition as active.
format fs=FAT32	Formats the UFD partition with the FAT32 file system.
assign	Assigns the next available drive letter to your UFD.
exit	Quits DiskPart.

4. Copy the contents of the ISO folder to your UFD, where e:\ is the drive letter assigned to the UFD device.

```
xcopy /chery c:\winpe_x86\ISO\*.* e:\
```

5. Safely remove your UFD.

NOTE **Some UFD devices do not support this preparation process. If necessary, use the UFD device manufacturer's processes and utilities to make the disk bootable.**

Making Your UFD Bootable

Creating a bootable UFD requires careful work. First, the computer's BIOS must support booting from a UFD. Second, many UFDs are not bootable and need to be converted before use. They are shipped with a flag value set to cause Windows to detect them as removable media devices rather than USB disk devices.

To make your UFD bootable, consult with the device manufacturer to obtain directions or utilities that will convert the device. Many manufacturers make these instructions available through their product support systems. Ask specifically how to switch the removable media flag. This action will cause Windows to detect the device as a USB hard disk drive and will allow you to proceed with the preparations for creating a bootable UFD.

Booting from a Hard Disk Drive

Although it might seem strange to be booting Windows PE from a hard disk drive, you can do this to perform refresh installations of Windows 7. By loading Windows PE onto the hard disk and booting it to RAM, you can repartition your systems disks and install the new Windows 7 image.

To boot Windows PE from a hard disk drive, perform the following steps:

1. Boot your computer from prepared Windows PE media.
2. Using DiskPart, prepare the computer's hard disk for installation of Windows PE. Use the DiskPart commands shown in Table 9-3.

TABLE 9-3 Preparing a Hard Drive for Windows PE

COMMAND	DESCRIPTION
select disk 0	0 is the primary hard disk drive.
clean	Removes the current partition structures.

COMMAND	DESCRIPTION
create partition primary size=size	*size* is a partition size large enough to hold the Windows PE source files.
select partition 1	Selects the partition created by the previous command.
active	Marks the new partition as active.
format	Formats the new partition.
exit	Quits DiskPart.

3. Copy the Windows PE files from your Windows PE media to your hard disk.

```
xcopy /chery x:\*.* c:\
```

Customizing Windows PE

Most Windows PE customization tasks will involve the processes described in the previous section. First, you will mount the image by using DISM. Then, you will add packages, applications, and updates. Last, you will dismount the image and commit your changes.

Other tasks you might see when customizing your Windows PE implementation include adding hardware-specific device drivers and customizing the actual settings used by Windows PE when it runs. This section covers the installation of device drivers and details changes that you can make to base Windows PE configuration settings. Additional information on automating Windows PE is covered in the section titled "Automating Windows PE" later in this chapter.

Windows PE supports four configuration files to control startup and operation. These files can be configured to launch custom shell environments or execute specified actions:

- **BCD** The BCD file stores the boot settings for Windows PE. This file is edited with the Windows 7 command-line tool, BCDEdit.
- **Winpeshl.ini** During startup, you can start custom shell environments using the Winpeshl.ini file. This file is located in the %SystemRoot%\System32 folder of the Windows PE image. You can configure this file with the path and the executable name of the custom shell application.
- **Startnet.cmd** Windows PE uses the Startnet.cmd file to configure network startup activities. By default, the Wpeinit command is called to initialize Plug and Play devices and start the network connection. You can also add other commands to this script to customize activities during startup.
- **Unattend.xml** Windows PE operates in the windowsPE setup configuration pass of a Windows 7 installation. In this pass, Windows PE uses the appropriate sections of the Unattend.xml file to control its actions. Windows PE looks in the root of the boot device for this file. You can also specify its location by using the Startnet.cmd script or by using Wpeutil.exe with the appropriate command-line options.

Your final environment can run custom application shells (see Figure 9-3).

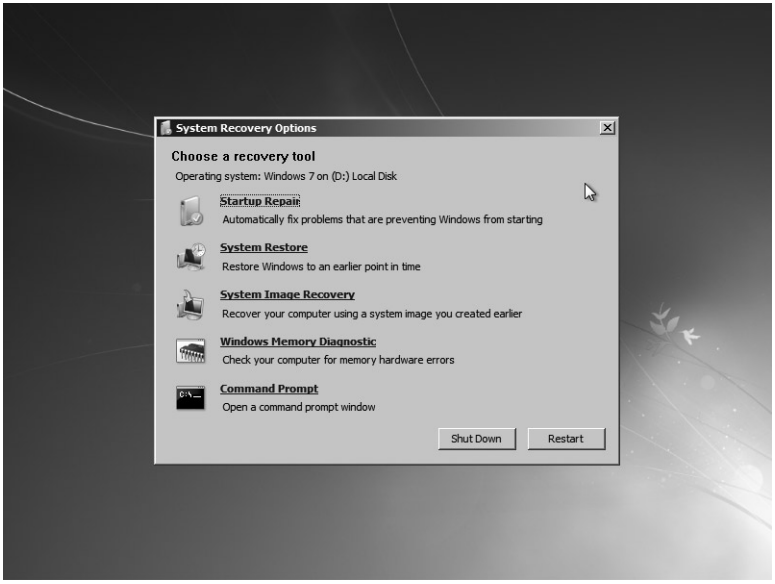


FIGURE 9-3 Windows RE running on Windows PE

Automating Windows PE

Most Windows PE automation is done by customizing Unattend.xml, the Windows 7 unattended answer file. Use the Windows System Image Manager (Windows SIM) to create and edit this file. Unattend.xml allows you to control automation tasks in all the major installation passes. You can put it in the root of your Windows PE media to automate the installation process.

Automating with Unattend.xml

Windows SIM is the primary tool for creating and modifying Unattend.xml. It is designed to validate each automation step against the actual files in an image to ensure that you use proper syntax to build the automation. This provides an extra measure of assurance that unattended installations will work as planned.

When beginning the process of creating an answer file, be sure to create a catalog file if necessary to allow Windows SIM to validate your choices against the image file. Add answer file options to the answer file by right-clicking a feature and choosing Add Setting To Pass 1 WindowsPE. The setting will then appear in the answer file pane, where you can configure it. When you complete answer file customization, you can validate the answer file by clicking Tools and then choosing Validate Answer File. Any settings that are not configured or that use invalid configuration settings will be listed in the Messages pane. When you are satisfied with

the answer file, save it to the root folder of your Windows PE media. When you boot a system using this media, the answer file is automatically detected and will control the operation of Windows PE.

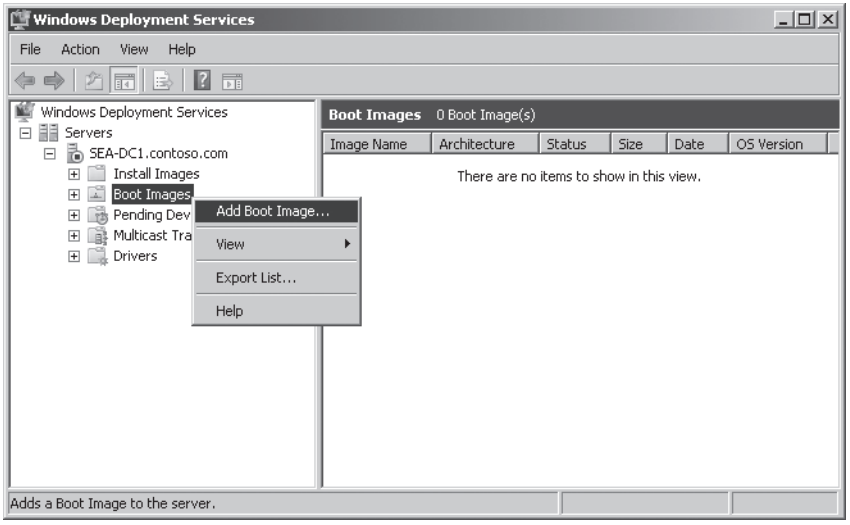
> ***NOTE*** **For more detailed information on using Windows SIM to create automation files, see the *Windows Automated Installation Kit User's Guide*.**

Adding Images to Windows Deployment Services

When you have completed building a Windows PE image, you can use Windows Deployment Services to deploy it to clients. This allows you to use the PXE boot automation services of Windows Deployment Services to replace portable media as the primary method of initiating Windows 7 installations.

To add a Windows PE boot image to Windows Deployment Services, perform the following steps:

1. In the Windows Deployment Services administration console, shown here, expand your Windows Deployment Services server and right-click Boot Images.



2. Click Add Boot Image to start the Add Image Wizard.
3. Follow the instructions in the wizard to select and import your custom Windows PE image.

> ***NOTE*** **For more information on the configuration and operation of Windows Deployment Services, see Chapter 10, "Configuring Windows Deployment Services."**

Using Windows PE with Microsoft Deployment Toolkit

MDT 2010 provides an infrastructure solution for automating the deployment of Windows 7. Part of the infrastructure is the support for automatically customizing and building Windows PE images. You manage the actual process of building the Windows PE image files by using wizards and scripting, greatly simplifying the process of adding device drivers and packages, automating settings, and prepping and capturing the deployment image.

You use Deployment Workbench to manage most operations regarding the creation and deployment of Windows 7 images and applications. This scripted environment is able to dynamically update Windows PE as updates are made to the Windows 7 distribution.

Chapter 6, "Developing Disk Images," describes how to use Deployment Workbench to create deployment shares. Deployment shares automatically generate Windows PE images when you update them. You can customize a deployment share's Windows PE image and choose which types of Windows PE images the deployment share generates when you update it. For more information on the Windows PE customization options available in MDT 2010, see the section titled "Updating the Deployment Share" in Chapter 6.

Summary

Windows PE 3.0 is the only preinstallation platform for installing Windows 7. Windows PE is publicly available in the Windows AIK.

You can approach using Windows PE in two ways. You can customize it through MDT 2010, which is the most appropriate approach if you're using MDT 2010 to deploy Windows 7. Alternatively, you can customize Windows PE manually by using the tools available in the Windows AIK. You can customize Windows PE to fit almost any deployment scenario by adding device drivers and packages, scripts and HTAs, and so on.

You can also start Windows PE in multiple ways. First, you can burn your custom Windows PE image to a CD or DVD and then start the computer using the disk. Second, you can put the Windows PE image on a bootable UFD and then use the UFD to start the computer. Last (and the most convenient option), you can add the custom Windows PE boot image to a Windows Deployment Services server and then start computers remotely.

Additional Resources

These resources contain additional information and tools related to this chapter.

- *Windows Automated Installation Kit User's Guide* (WAIK.chm)
- *Windows PE User's Guide* (WinPE.chm)
- Chapter 6, "Developing Disk Images"
- Chapter 10, "Configuring Windows Deployment Services"

CHAPTER 10

Configuring Windows Deployment Services

Windows Deployment Services in Windows Server 2008 and Windows Server 2008 R2 is the updated and redesigned version of Remote Installation Services (RIS), which was first introduced in Microsoft Windows 2000 Server. You can use Windows Deployment Services to rapidly deploy the Windows 7 operating system by using Pre-Boot Execution Environment (PXE). Using Windows Deployment Services, you can deploy Windows 7 over a network. You can also use Windows Deployment Services to start remote computers using Windows Preinstallation Environment (Windows PE) boot images and then install Windows 7 using customized, scripted deployment solutions, such as Microsoft Deployment Toolkit 2010 (MDT 2010).

Windows Deployment Services delivers a better in-box deployment solution than RIS. It provides platform features that allow for custom solutions, including remote boot capabilities; a plug-in model for PXE server extensibility; and a client-server communication protocol for diagnostics, logging, and image enumeration. Also, Windows Deployment Services uses the Windows Imaging (.wim) file format and provides a greatly improved management experience through the Microsoft Management Console (MMC) and scriptable command-line tools. For organizations that already have a RIS implementation deployed, Windows Deployment Services maintains parity with RIS by providing both coexistence and migration paths for RIS. First, Windows Deployment Services continues to support RIS images in legacy or mixed mode. Second, Windows Deployment Services provides tools to migrate RIS images to the .wim file format.

This chapter describes the architecture of Windows Deployment Services and the requirements for using it. It also describes the key features of Windows Deployment Services and how to use them in specific scenarios, including how MDT 2010 uses Windows Deployment Services to start destination computers and install operating systems. Finally, the chapter describes the improvements to Windows Deployment Services introduced in Windows Server 2008 R2.

Introducing Windows Deployment Services

Windows Deployment Services supports remote, on-demand deployment of Windows 7 and Windows PE images located in a central image store. It is available as an add-on to Windows Server 2003 systems running RIS and is the native remote installation technology provided with Windows Server 2008 and Windows Server 2008 R2.

Windows Deployment Services images are collected from client master systems and stored using the single instancing provided by the .wim imaging format. Clients can be booted from PXE-compliant network adapters or by using remote client boot disks. The Windows Deployment Services client boots into a customized Windows PE image, and the user can select the installation image from a list of images stored on the server. Windows Deployment Services installations can also be scripted for unattended installation support and to support Lite Touch Installation (LTI) and Zero Touch Installation (ZTI) scenarios.

Service Architecture

The Windows Deployment Services architecture has three major categories of features:

- **Management features** Management features are a set of tools that you use to manage the server, operating system images, and client computer accounts. The Windows Deployment Services MMC snap-in is a management feature, and the command-line interface is another.

- **Server features** Server features include a PXE server for network booting a client to load and install an operating system. Server features also include a shared folder and image repository that contains boot images, installation images, a Trivial File Transfer Protocol (TFTP) server, a multicast server, a driver provisioning server, and files that you need specifically for network boot.
- **Client features** Client features include a graphical user interface (GUI) that runs within Windows PE and communicates with the server features to select and install an operating system image.

Figure 10-1 illustrates the various features of Windows Deployment Services. The following sections describe the image store, PXE server, management, and client features in more detail.

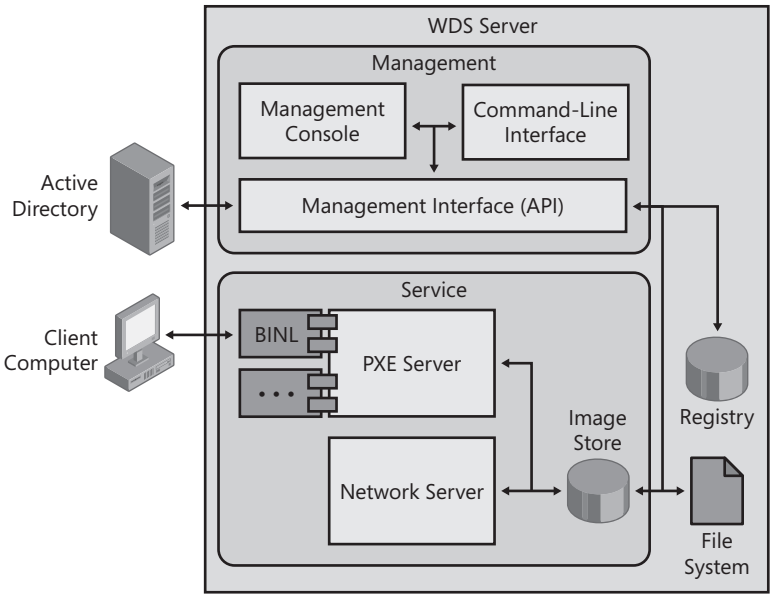


FIGURE 10-1 Windows Deployment Services architecture

Image Store

Figure 10-2 describes how Windows Deployment Services organizes the image store.

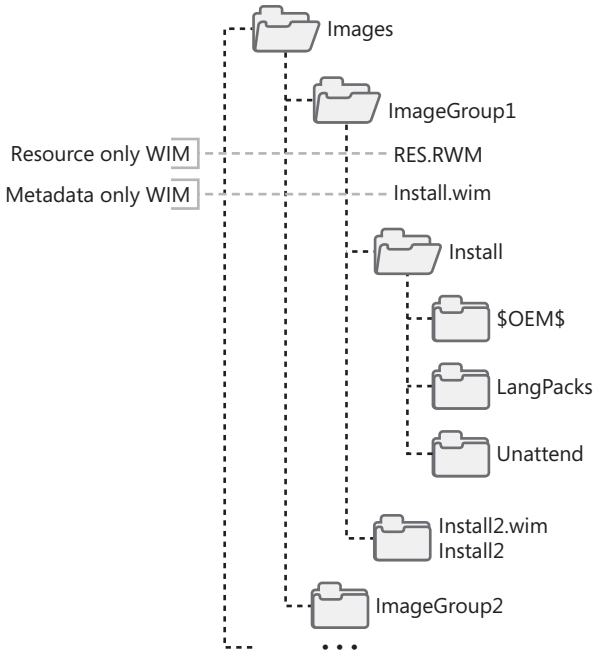


FIGURE 10-2 Windows Deployment Services image store organization

Organizing images into groups, as shown in Figure 10-2, provides two benefits. First, image groups allow you to better manage and organize images. For example, you can manage the security of an entire image group rather than managing the security of individual images. Second, image groups provide units of single instancing. This means that all the images within an image group use Single Instance Storage (SIS) to significantly compress their contents. The file Res.rwm contains all of the file resources for the image group, and this file uses SIS. Each image file (Install.wim and Install2.wim in Figure 10-2) contains only metadata that describes the image file contents based on the contents of Res.rwm.

Windows Deployment Services references images by their group name and image file name. For example, the image ImageGroup1\Install2.wim refers to the image file Install2.wim in the group ImageGroup1.

PXE Services

The Windows Deployment Services PXE server is built on a unified and scalable architecture. As shown in Figure 10-3, it uses plug-ins to provide access to the data store. The PXE server supports one or more plug-ins, and each plug-in can use any data store. Windows Deployment Services provides a default BINL plug-in, as shown earlier in Figure 10-1.

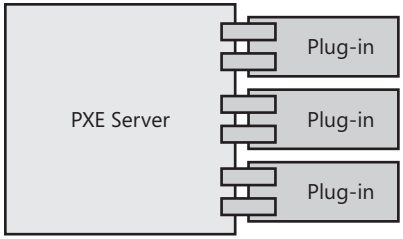


FIGURE 10-3 Windows Deployment Services PXE server

NOTE **Windows Deployment Services in Windows Server 2008 R2 also adds a provider called WDSSIPR (SImple PRovider), which is installed with the Transport Server role service and uses an .ini file as a data store.**

Developers can use published application programming interfaces (APIs) to create PXE server plug-ins. You can find these APIs in the Windows Vista Software Development Kit (SDK). The SDK also includes samples that developers can use to create their own plug-ins. For example, a developer can create a PXE server plug-in that works without requiring Active Directory Domain Services (AD DS) and reads settings from a Microsoft SQL Server database.

Management

Windows Deployment Services provides two management tools that significantly simplify management tasks. The first tool is an MMC console that provides a GUI for common management tasks. After installing Windows Deployment Services, you start this console by clicking Windows Deployment Services in the Administrative Tools folder of the Start menu. Examples of common tasks that you can perform using this console include adding images and configuring server settings. The second management tool provided by Windows Deployment Services is the Wdsutil command-line tool. Wdsutil provides all the management functionality that the console provides and more. You can use Wdsutil to perform individual management tasks; you can also use it to automate management tasks by scripting Wdsutil commands. Both tools use the management API that Windows Deployment Services provides, and both tools enable remote administration of Windows Deployment Services servers.

Other management utilities for Windows Deployment Services include:

- **Capture utility** The Windows Deployment Services capture utility captures images to the .wim file format. It includes a light version of the ImageX */capture* functionality and provides a GUI for it. You can use this to add the resulting .wim file to the image store.
- **Active Directory Users And Computers MMC snap-in** You can use this snap-in to administer legacy RIS functionality and configure settings on the Remote Install tab of computer accounts.

- **Risetup and Riprep** Windows Deployment Services provides updated versions of Risetup and Riprep for upgrade scenarios (available in Windows Server 2003 only).

The Windows Deployment Services management console (Figure 10-4) provides significant administrative control. You can add and remove servers. You can configure a variety of options, including computer-naming rules, Dynamic Host Configuration Protocol (DHCP) settings, PXE response settings, and so on. You can add and remove installation and boot images. You can also organize images into image groups. The Windows Deployment Services management console gives you full control over your image groups and the images you add to them. You can configure permissions for each image group and for individual images, too. You can also associate an answer file with each individual image. The Windows Deployment Services management console helps you better manage images for different platforms. For example, you can associate different boot programs and boot images with the x86, x64, and ia64 platforms. You can also associate a global answer file with each platform.

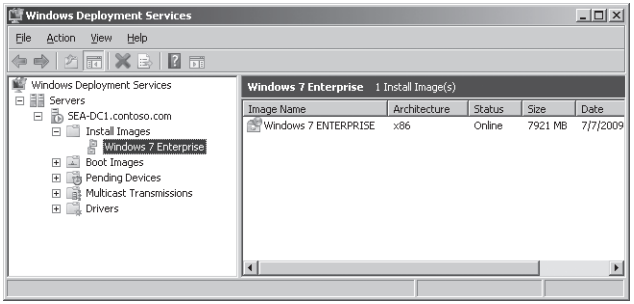


FIGURE 10-4 Windows Deployment Services management console

NOTE **Windows Deployment Services in Windows Server 2008 R2 also allows the administrator to choose the preferred boot behavior, such as whether to require an F12 keypress to boot or not.**

Client

The Windows Deployment Services client is a special version of Windows Setup that runs only within Windows PE. In other words, when you deploy Windows 7 to the destination computer using Windows Deployment Services, the Windows Deployment Services client runs within Windows PE on the client computer. This approach allows deployment of Windows 7 as well as images of previous versions of Windows. Note, however, that the version of Windows PE you use has to be at least as recent as the operating system you are deploying. For example, Windows PE 3.0 will deploy Windows 7, Windows Vista, and Windows XP, but Windows PE 2.1 will deploy only Windows Vista and Windows XP.

The Windows Deployment Services client drives the setup experience as follows:

- **Language selection** For Windows 7, the client prompts the user to choose a language. This choice applies to the setup user interface and the operating system installation. The user can also install additional language packs (Windows 7 Enterprise and Windows 7 Ultimate Edition operating systems only).
- **Credentials gathering** The client prompts the user for required credentials to connect to the image store on the Windows Deployment Services server.
- **Image selection** The client displays a list of images available to the user and allows the user to choose an image to install on the destination computer.
- **Disk configuration** The client allows the user to partition and format the destination computer's hard disks. The client provides the same options as Windows Setup.

However, you can automate all of the settings that the client prompts for. To automate these settings, you use Windows System Image Manager (Windows SIM) to create an Unattend.xml file. For more information about creating answer files, see the *Windows Automated Installation Kit User's Guide*, which is installed as part of the Windows Automated Installation Kit (Windows AIK) 2.0.

Operating Modes

To provide a clear path between legacy RIS functionality and Windows Deployment Services functionality, Windows Deployment Services supports three modes (legacy and mixed modes are available only in Windows Server 2003):

- **Legacy mode** This mode uses the Client Installation Wizard (OSChooser) and Riprep (sector-based) images. This mode is compatible with RIS. Moving from RIS-only functionality to legacy mode happens when you install the Windows Deployment Services update on a server running RIS.
- **Mixed mode** This mode supports both OSChooser and Windows PE for boot environments and Riprep and ImageX imaging. Moving from legacy mode to mixed mode happens when you configure Windows Deployment Services and add .wim image files to it.
- **Native mode** This mode supports only the Windows PE boot environment and .wim image files. The final move to native mode occurs after you have converted all legacy images to the .wim image file format and have disabled the OSChooser functionality.

Your choice of operating mode will depend on which client operating systems you are deploying and your investment into legacy Riprep images. You don't need to abandon your current deployment images; operating in mixed mode allows you to continue to deploy legacy RIS images from OSChooser. It also allows you to deploy new .wim images of Windows 7 using Windows PE.

The mode used by Windows Deployment Services is not a simple selection in a dialog box. Each mode is activated in a specific way. The following sections describe each mode in more detail and how to configure each mode.

Legacy Mode

In Windows Deployment Services, legacy mode is functionally equivalent to that of RIS (Windows Deployment Services binaries with RIS functionality). In legacy mode, only OSChooser will be present as the boot operating system. Therefore, only Risetup and Riprep images are supported. You will not be using the Windows Deployment Services management tools; rather, legacy RIS utilities will be the only way to manage the server. Legacy mode is available only on Windows Server 2003.

You configure legacy mode by first installing RIS on Windows Server 2003 and optionally adding legacy images to it. Then, you install the Windows Deployment Services update, as described in the section titled "Installing Windows Deployment Services" later in this chapter. You do not configure Windows Deployment Services by using Wdsutil or the Windows Deployment Services management console.

To configure Windows Deployment Services in legacy mode in Windows Server 2003, perform the following steps:

1. Install the RIS optional feature on Windows Server 2003 Service Pack 1 (SP1) or later and then configure it by running Risetup. Optionally, you can add images to it.
2. If needed, install the Windows Deployment Services update. (Windows Server 2003 SP2 and later installs this update by default.) The Windows AIK 1.1 includes the Windows Deployment Services update for Windows Server 2003 SP1.

Mixed Mode

Mixed mode describes a server state in which both OSChooser and Windows PE boot images are available. In mixed mode, access to the old Risetup and Riprep images is possible through OSChooser. Additionally, you can access the .wim image files via a Windows PE boot image. A boot menu allows users to choose RIS or Windows PE. You will use legacy management tools to manage Risetup and Riprep images and the Windows Deployment Services management tools to manage all facets of the server, including the .wim image files. Windows Deployment Services mixed mode is available only on Windows Server 2003.

You configure mixed mode by first installing RIS on Windows Server 2003 and adding legacy images to it. Then, you install the Windows Deployment Services update, as described in the section titled "Installing Windows Deployment Services" later in this chapter. Last, you run Wdsutil or use the Windows Deployment Services management console to configure Windows Deployment Services and then optionally add .wim images to the image store.

To configure Windows Deployment Services in mixed mode in Windows Server 2003, perform the following steps:

1. Install the RIS optional feature on Windows Server 2003 SP1 or later and then configure it by running Risetup. Optionally, you can add images to it.
2. If needed, install the Windows Deployment Services update. (Windows Server 2003 SP2 and later installs this update by default.)

3. Run **wdsutil /initialize-server** or configure the server in the Windows Deployment Services management console.

Native Mode

Native mode describes a Windows Deployment Services server with only Windows PE boot images. In this mode, OSChooser is not available, and Windows Deployment Services deploys only .wim image files to client computers. You use the Windows Deployment Services management console or Wdsutil to manage Windows Deployment Services in native mode. Native mode is available on Windows Server 2003, Windows Server 2008, and Windows Server 2008 R2. Native mode is the only mode supported on Windows Server 2008 and Windows Server 2008 R2.

To configure Windows Deployment Services in native mode in Windows Server 2003, perform the following steps:

1. Install the RIS optional feature on Windows Server 2003 SP1. Do not configure the RIS service or add images to it.
2. If needed, install the Windows Deployment Services update. (Windows Server 2003 SP2 and later installs this update by default.)
3. Run **wdsutil /initialize-server** or configure the server in the Windows Deployment Services management console.

> ***NOTE*** **The Windows Deployment Services server may be forced to enter native mode from any other mode. This is a one-way operation and is accomplished by using the Wdsutil management utility. The command** `wdsutil /Set-Server /ForceNative` **changes the Windows Deployment Services server to native mode.**

Planning for Windows Deployment Services

Windows Deployment Services doesn't have significant requirements for the system on which you install it, but you need to put some thought into which services and applications must exist in your environment to support Windows Deployment Services, including the actual server requirements, client computer requirements, and network requirements.

Windows Deployment Services supports booting computers directly from a boot image over the network. This image boots using the PXE boot specification and needs to be able to receive broadcast messages from PXE clients. This will require some planning to make sure clients will be able to find and communicate with the Windows Deployment Services server. As a result, you must consider the Windows Deployment Services requirements for DHCP and routing. This section discusses requirements you need to consider for Windows Deployment Services.

Choosing a Version of Windows Deployment Services

Windows Deployment Services is included as an installable server role in Windows Server 2008 and Windows Server 2008 R2. Windows Deployment Services is also available as a separate update for Windows Server 2003 SP1. (This update is included in Windows Server 2003 SP2.) The version of Windows Deployment Services that you use in your environment will depend upon your business needs, budget, and existing network infrastructure.

Supported Operating Systems

The Windows operating systems that can be deployed vary with the version of Windows Deployment Services used. The Windows Deployment Services role in Windows Server 2008 R2 can be used to deploy the following operating systems:

- Windows XP
- Windows Server 2003
- Windows Vista SP1
- Windows Server 2008
- Windows 7
- Windows Server 2008 R2

The Windows Deployment Services role in Windows Server 2008 can deploy all the operating systems listed previously, as well as Windows 2000.

The operating systems that you can deploy using the Windows Deployment Services update for Windows Server 2003 SP1 and later depend upon whether Windows Deployment Services is running in legacy, mixed, or native mode. Specifically:

- **Legacy mode** Supports installing Windows 2000, Windows XP, and Windows Server 2003
- **Mixed mode** Supports installing Windows 2000, Windows XP, Windows Server 2003, Windows Vista, and Windows Server 2008 (You can also deploy Windows 7 and Windows Server 2008 R2 in this mode, as long as you have a Windows PE 3.0 boot image.)
- **Native mode** Supports installing Windows 2000 Professional Edition, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, and Windows Server 2008 R2

Supported Image Types

The types of Windows images that can be deployed also vary with the version of Windows Deployment Services used. The Windows Deployment Services role in Windows Server 2008 R2 can deploy the following types of images:

- Windows Imaging (WIM) file format images
- Virtual hard disk (VHD) images

The Windows Deployment Services role in Windows Server 2008 can deploy only WIM images for a new installation of Windows Server 2008. If you upgrade to Windows Server 2008 from Windows Server 2003, you can also convert RIPREP images to WIM images. RISETUP images are not supported, however.

The types of Windows images you can deploy using the Windows Deployment Services update for Windows Server 2003 SP1 and later depend on the mode in which Windows Deployment Services is running, specifically:

- **Legacy mode** RISETUP and RIPREP images
- **Mixed mode** RISETUP, RIPREP, and WIM images
- **Native mode** WIM images only

Boot Environment

The boot environment used for deployment varies with the version of Windows Deployment Services used. The Windows Deployment Services role in Windows Server 2008 R2 uses Windows PE 3.0 as its boot environment. The Windows Deployment Services role in Windows Server 2008 uses Windows PE 2.1. The boot environment used by the Windows Deployment Services update for Windows Server 2003 SP1 and later depends on the mode in which Windows Deployment Services is running, specifically:

- **Legacy mode** OSChooser
- **Mixed mode** OSChooser and Windows PE 2.0, 2.1, or 3.0
- **Native mode** Windows PE 2.0, 2.1, or 3.0

New Features of Windows Deployment Services in Windows Server 2008 R2

The Windows Deployment Services role in Windows Server 2008 R2 has been improved with the following new features:

- **Dynamic driver provisioning** You can now use Windows Deployment Services to add driver packages to boot images so you can deploy these packages to client computers during deployment. For more information on dynamic driver provisioning, see the section titled "Managing and Deploying Driver Packages" later in this chapter.
- **Improved multicasting** Windows Deployment Services can now automatically disconnect slow clients and divide transmissions into multiple streams based on client speeds. Windows Deployment Services also now includes support for IPv6 multicasting. For more information, see the section titled "Creating Multicast Transmissions" later in this chapter.
- **Native booting to VHD images** In Windows 7 and Windows Server 2008 R2, you can now use a VHD as a running operating system without any other parent operating system, virtual machine, or hypervisor. For example, you can deploy a Windows 7 .wim file to a VHD and then copy the .vhd file to client computers. After you do this,

the Windows 7 boot manager must be configured to boot directly into the VHD. Note, however, that if you simply deploy Windows 7 into a VHD, you'll go through the Sysprep specialize pass, which prevents you from using the VHD on physical machines. The workaround for this is to first use the Wim2vhd tool available from *http://code.msdn.microsoft.com/wim2vhd*, create a VHD, and then use ImageX to apply the contents of the WIM into the VHD.

VHD images are not intended to replace WIM images for general deployment purposes. Furthermore, beginning with Windows Server 2008 R2, Windows Deployment Services now supports deploying VHD images in addition to deploying WIM images. Specifically, when you deploy a VHD through Windows Deployment Services, the Bootmgr entries are automatically fixed, so there is no extra step. For example, you can use Windows Deployment Services to deploy VHD images during an unattended installation. For more information on native booting to VHD images, see "Understanding Virtual Hard Disks with Native Boot" in the Windows Client TechCenter on Microsoft TechNet at *http://technet.microsoft.com/en-us/library/dd799282.aspx*. For more information on deploying VHD images using Windows Deployment Services, see "Deploying Virtual Hard Disk Images" in the Windows Server TechCenter on Microsoft TechNet at *http://technet.microsoft.com/en-us/library/dd363560.aspx*.

- **PXE provider for Transport Server** Windows Deployment Services now includes a PXE provider for the Transport Server role service. This lets you use a stand-alone Transport Server to boot from the network or to multicast data without the need of AD DS or Domain Name System (DNS).
- **Additional EFI support** Windows Deployment Services now supports network booting of x64-based computers that use EFI.

MORE INFO **For additional information concerning these new features, see** ***http://technet.microsoft.com/en-us/library/dd735188.aspx.***

Server Requirements

The hardware requirements for running Windows Server 2003 or Windows Server 2008 are sufficient to support most Windows Deployment Services installations. If you are supporting a large number of images or if you are expecting greater-than-normal client load, investigate adding additional memory for performance and additional hard drive space for image storage. Adding more network adapters can also help with the TFTP download phase if you have a large client load.

The following list describes the software and service requirements for installing and using Windows Deployment Services:

- **AD DS** A Windows Deployment Services server must be either a member of an AD DS domain or a domain controller for a domain. AD DS is used by Windows Deployment

Services to track Windows Deployment Services clients and Windows Deployment Services servers. In addition, systems can be preconfigured in AD DS, instructing Windows Deployment Services on how to image them. Note that AD DS is required only for Deployment Server, not Transport Server.

- **DHCP** You must have a working DHCP server with an active scope on the network because Windows Deployment Services uses PXE, which in turn uses DHCP. The DHCP server does not have to be on the Windows Deployment Services server. The type of DHCP server is not critical for Windows Deployment Services to function properly. To operate Windows Deployment Services and DCHP on the same server, see the section titled "DHCP Requirements" later in this chapter. Note that if you are using Transport Server for multicast only (no PXE), then you don't need DHCP.
- **DNS** A working DNS server on the network is required to run Windows Deployment Services. The DNS server does not have to be running on the Windows Deployment Services server. DNS is used to locate AD DS domain controllers and Windows Deployment Services servers.
- **Installation media** Windows 7 media or a network location that contains the contents of the media are required to install Windows 7 using Windows Deployment Services.
- **An NTFS partition on the Windows Deployment Services server** The server running Windows Deployment Services requires an NTFS File System (NTFS) partition for the image store. You should not create the image store on the partition containing the operating system files, so an additional partition is necessary.
- **SP1 or later version and RIS installed (Windows Server 2003 only)** If you're installing Windows Deployment Services on a server running Windows Server 2003, you must install RIS for the Windows Deployment Services update package to be run. Windows Deployment Services also requires at least SP1.

NOTE **Installing and administering Windows Deployment Services requires the administrator to be a member of the local Administrators group on the Windows Deployment Services server. In addition, most administrative tasks for Windows Deployment Services require Domain Admins credentials.**

Client Computer Requirements

The client computer requirements to support installation using Windows Deployment Services will vary based on how you intend to use Windows Deployment Services. The following list outlines the requirements for PXE booting to Windows Deployment Services and installing images:

- **Hardware requirements** The client must meet the minimum hardware requirements of the operating system you're installing. The client must also have enough memory

to run Windows PE (384 megabytes [MB] required, 512 MB recommended), because Windows Deployment Services uses Windows PE to start the client computer.

- **PXE DHCP-based boot ROM version .99 or later network adapter** To boot directly from the Windows Deployment Services server, the client's network adapter must contain a PXE boot ROM. If this is not the case, the client can be booted using a DVD boot disk, a Windows PE boot image copied to the computer's hard disk, or a USB flash drive (UFD). See the section titled "Preparing Discover Images" later in this chapter.

  All computers meeting the NetPC or PC98 specifications should have the ability to boot from the network adapter. Investigate the basic input/output system (BIOS) settings of the client to determine whether you can enable a Boot From Network option. When the option is enabled, the client should briefly display an option to press F12 to boot from the network during each startup.

- **Network access to the Windows Deployment Services server** The client must have broadcast access to the Windows Deployment Services server to enable PXE booting. Windows PE boot disks can allow you to boot to Windows PE using Windows Deployment Services as an image store without broadcast access.

NOTE **The account performing the installation must be a member of the Domain Users AD DS security group. Domain Users have permission to join computers to the domain.**

DHCP Requirements

Windows Deployment Services will configure accessible DHCP servers during installation, adding required scope options to the DHCP scopes. It may be necessary under some circumstances to modify DHCP servers manually to support advanced Windows Deployment Services scenarios. The following list describes how to manage DHCP scope modifications:

- **Microsoft DHCP and Windows Deployment Services on the same server** When Windows Deployment Services is installed on the same physical server as the DHCP service, the Windows Deployment Services PXE server and the DHCP server will both attempt to listen on port 67 for DHCP requests. To prevent this, the Windows Deployment Services PXE server must be configured not to listen on this port. (See Figure 10-5.) This allows booting PXE clients to learn about the presence of the Windows Deployment Services PXE server from the DHCP response generated by the DHCP server.

- **Microsoft DHCP and Windows Deployment Services on separate servers with the clients on the same subnet as the Windows Deployment Services server** When Windows Deployment Services and Microsoft DHCP exist on different servers, no additional settings are required. Both servers respond to DHCP requests. The DHCP server responds with an IP address offer; the Windows Deployment Services PXE server responds with the PXE boot information.

- **Microsoft DHCP and Windows Deployment Services on separate servers with the clients on a different subnet from the Windows Deployment Services server** The recommended approach in this scenario is to use IP Helper tables on the router or switch to forward PXE requests to the Windows Deployment Services server (as well as the DHCP server). An alternative approach is to configure DHCP options 66 and 67 on all scopes to specify the Windows Deployment Services server and the path to the boot program.
- **Third-party DHCP and Windows Deployment Services on separate servers** No additional action should be required for Windows Deployment Services to coexist with third-party DHCP servers. The Windows Deployment Services PXE server will respond with boot file location information only, allowing DHCP to service the IP address request.

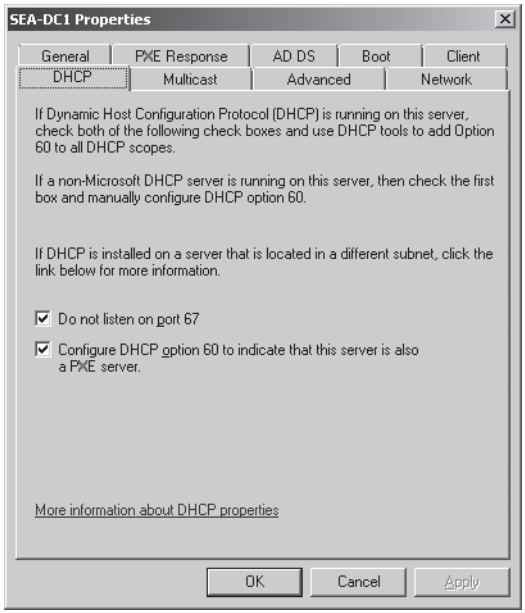


FIGURE 10-5 Configuring DHCP options in Windows Deployment Services

NOTE **RIS requires the RIS server to be authorized as a DHCP server in AD DS. This is not required to operate Windows Deployment Services.**

Routing Requirements

When DHCP and Windows Deployment Services are located on different subnets or if clients are located on a different subnet than the Windows Deployment Services server, IP Helpers must be configured on network routers to enable forwarding of DHCP and PXE boot requests to the appropriate servers. (See Figure 10-6.)

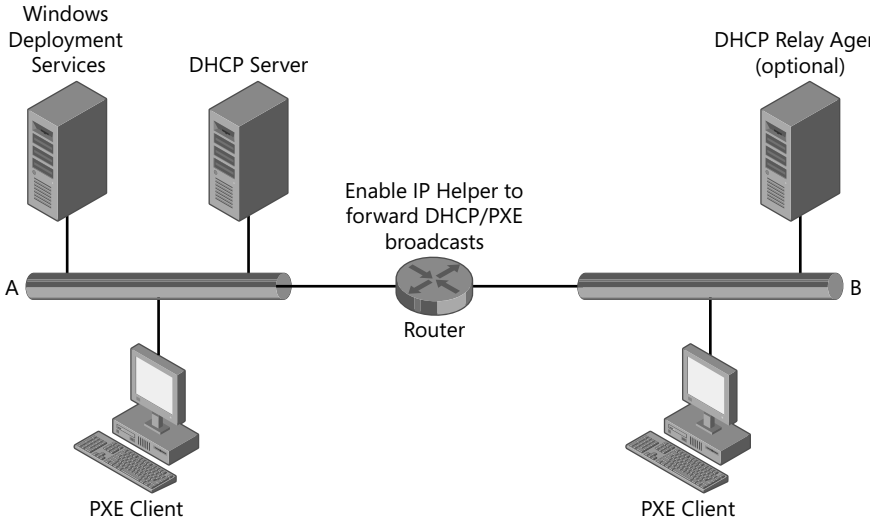


FIGURE 10-6 Windows Deployment Services on multiple subnets

NOTE **An alternative to enabling IP Helpers on your routers is to install a DHCP relay agent on the remote network, configuring appropriate scope options to allow the remote clients to locate the Windows Deployment Services server.**

Capacity Requirements

Windows Deployment Services servers can generate a lot of network traffic when servicing multiple, simultaneous client requests. Plan for this network load by designing your deployment network for sufficient capacity. You can deploy multiple Windows Deployment Services servers or use multicasting (requires Windows Server 2008 or later versions) in environments that experience significant installation activity. Note that beyond about 25 to 50 simultaneous clients, the bottleneck becomes TFTP, which is unicast and is required to download Windows PE. (Windows Deployment Services supports multicast download of Windows PE only for x64 Unified Extensible Firmware Interface [UEFI] machines). You can allocate access to Windows Deployment Services by using DHCP scopes and IP subnetting. You can also configure IP Helper tables to direct clients to one or another Windows Deployment Services server based on client network ID.

Installing Windows Deployment Services

Windows Deployment Services is installed as an update to Windows Server 2003 or added as a server role in Windows Server 2008 R2. The following procedures outline the basic installation steps for Windows Deployment Services. Refer to the appropriate guidance (listed in the

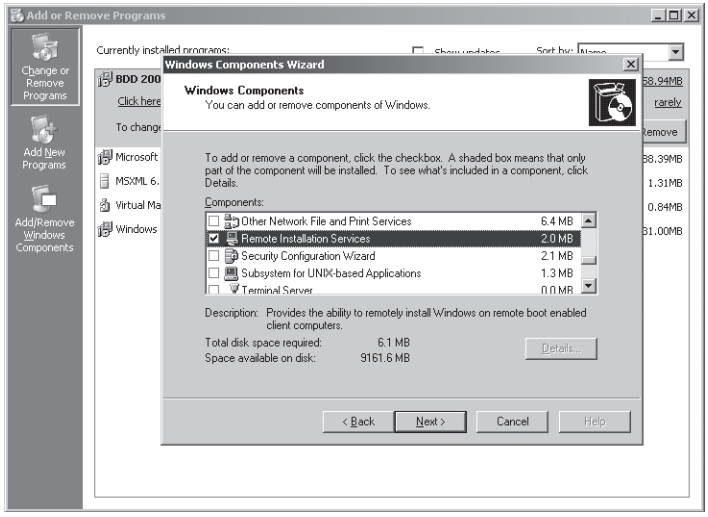
section titled "Additional Resources" at the end of this chapter) for complete instructions and planning advice.

Windows Server 2003

To completely install Windows Deployment Services on a computer running Windows Server 2003, you must first install RIS. After RIS is installed, you install the Windows Deployment Services update or Windows Server 2003 SP2 (which contains the update). The Windows AIK also includes the Windows Deployment Services update, which you can install on any server after extracting the file from the Windows AIK media.

To install RIS on Windows Server 2003, perform the following steps:

1. In the Add Or Remove Programs utility in Control Panel, click Add/Remove Windows Components.
2. Select the check box next to Remote Installation Services, as shown here, and then click Next.

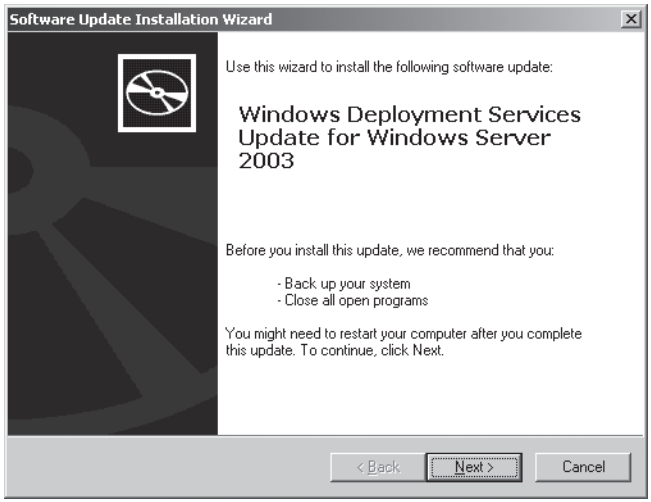


NOTE **In Windows Server 2003 SP2, the Remote Installation Services feature is named Windows Deployment Services.**

To install the Windows Deployment Services update, perform the following steps:

1. Run the Windows Deployment Services update from the Windows AIK. The file is windows-deployment-service-update-*platform*.exe, where *platform* is either x86 or x64, and is found in the WDS folder on the Windows AIK DVD. (If you have already installed SP2 for Windows Server 2003, you do not need to perform this task.)

2. On the Windows Deployment Services Setup Wizard Welcome page, shown here, click Next.



3. On the Microsoft Software License Terms page, click I Accept The Terms In The License Agreement. Click Next.
4. The Updating Your System page displays installation progress.
5. On the Completion page, click Finish to restart the computer.

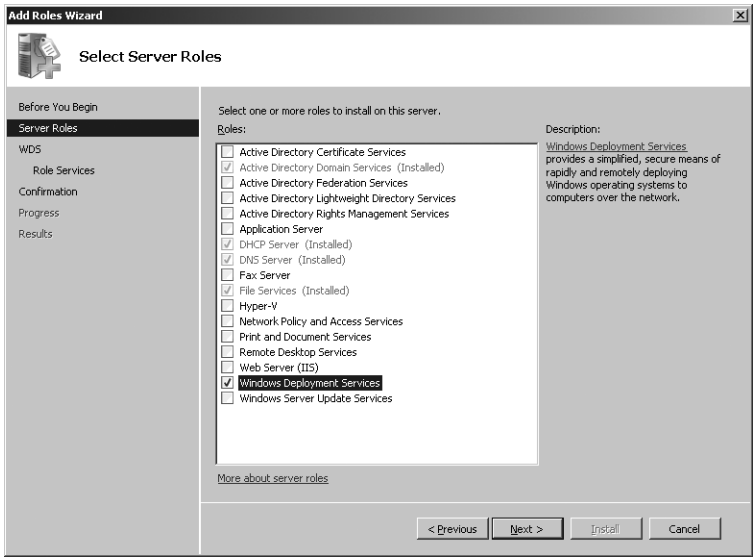
NOTE **Unless you plan to use Riprep legacy images, you can proceed with the configuration of Windows Deployment Services at this point. To enable Windows Deployment Services mixed mode, ensure that you do not install this update until at least one Riprep image is installed on the RIS server. For more information on the installation and configuration of RIS, see "Designing RIS Installations" in the *Windows Server 2003 Resource Kit*.**

Windows Server 2008 R2

You can install Windows Deployment Services by using the Add Roles Wizard, located in Server Manager.

To add the Windows Deployment Services server role, perform the following steps:

1. Start the Add Roles Wizard from Server Manager.
2. Click Next to skip the Before You Begin screen.
3. Select the Windows Deployment Services role, as shown here, and click Next.



4. Additional information on installing and using Windows Deployment Services is displayed.
5. Click Next when you are ready to proceed.
6. On the Select Role Services page, click Next to install both the Deployment Server and the Transport Server role services. The Deployment Server role service contains all of the core Windows Deployment Services functionality. The Transport Server role service contains the core networking features.
7. On the Confirm Installation Selections page, click Install.
8. Windows Deployment Services is installed.
9. Click Close to complete the Add Roles Wizard.

Configuring Windows Deployment Services

After Windows Deployment Services is installed, you will need to add the server to the management console and then configure it. Windows Deployment Services automatically adds the local computer to the console. If you want to add a remote server, you must add it.

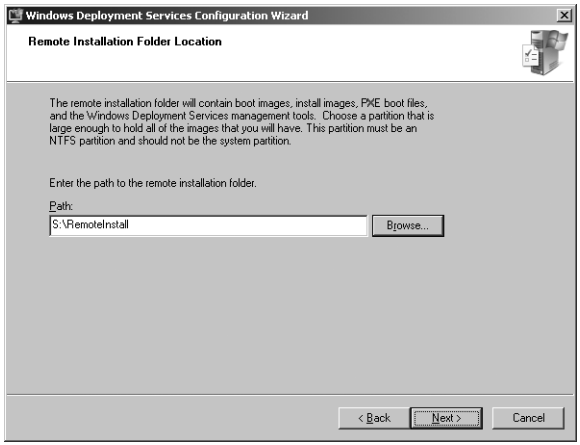
To add a server to Windows Deployment Services, perform the following steps:

1. Open the Windows Deployment Services management console by selecting Windows Deployment Services from Administrative Tools. You can also use the Windows Deployment Services node under Roles in Server Manager.
2. Right-click Servers in the Windows Deployment Services console tree and then click Add Server.

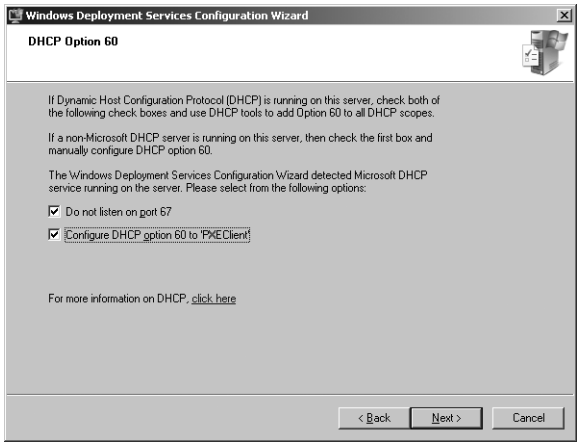
3. In the Add Server dialog box, choose a computer to add to the console. The server will be added and will now need to be configured.

To initially prepare the Windows Deployment Services server, perform the following steps:

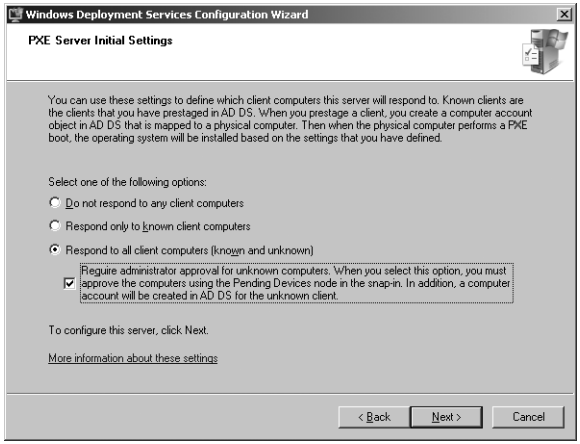
1. In the Windows Deployment Services console tree, right-click the server and click Configure Server.
2. On the Windows Deployment Services Configuration Wizard Welcome page, make sure that your environment meets the requirements and then click Next.
3. Enter a path for the image store, as shown here, and then click Next. The folder should be on a partition other than the partition containing the system files. If you choose to create the image store on the system drive, a warning message will appear. Click Yes to continue or click No to choose a new installation location (recommended).



4. Configure DHCP Option 60 settings, as shown here, and then click Next. (Depending upon your configuration, this screen may or may not be displayed.) See the section titled "DHCP Requirements" earlier in this chapter for information on how to properly configure these settings.



5. Set a PXE Server Initial Settings policy, as shown here, and then click Next.



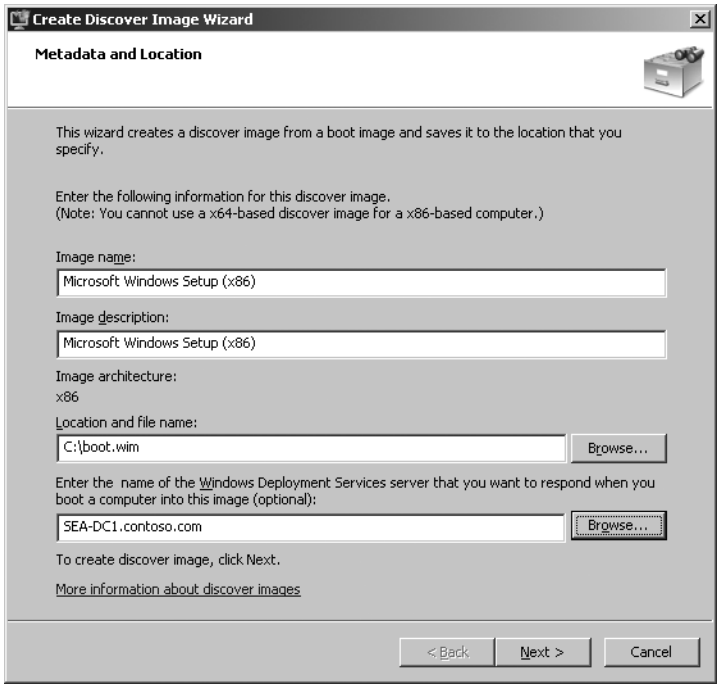
6. On the Configuration Complete page, you can add images to the server (default) or clear the Add Images To The Windows Deployment Services Server Now check box if you want to add images at another time. To add images to your server, see the section titled "Importing Images" later in this chapter.

Preparing Discover Images

For client computers that do not support PXE booting, you can create boot disks using a CD or DVD, a hard disk, or a UFD. You can create these disks by using the Windows Deployment Services administration tools or the Windows PE administration tools from the Windows AIK. The process begins by creating a Windows PE boot image using the Windows Deployment Services console or Wdsutil. After this image is created, a bootable disk is made using the *Oscdimg* command from the Windows AIK.

To create a discover image using the management console, perform the following steps:

1. In the Windows Deployment Services management console, click Boot Images. Boot Images is under Servers, *server_name*, where *server_name* is the name of the Windows Deployment Services server.
2. Right-click a boot image that you previously added to Windows Deployment Services to use as a discover image and then click Create Discover Boot Image.
3. On the Metadata And Location page, type a name and description for the discover image, as shown on the following page. Then choose the location in which to create the image and the Windows Deployment Services server to respond to it. Click Next.



4. Click Finish.

To create a discover image using Wdsutil, perform the following steps:

1. Run the following command using elevated credentials.

   ```
   Wdsutil /new-discoverimage /image:boot_image/architecture:architecture
   /destinationimage /filepath:discover_image
   ```

   Boot_image is the name of the boot image you want to use to create the discover image (not the file name), and *discover_image* is the file path and file name of the new Windows PE boot image. *Architecture* is either x86 or x64.

To create a bootable DVD using the discover image, perform the following steps:

1. To create a Windows PE build environment, open a command prompt and run the following commands.

   ```
   Md c:\Winpe\Boot
   Md c:\Winpe\Sources
   ```

2. Copy the discover image created in the previous procedures to the \Sources folder of the build environment with the following command.

   ```
   Copy d:\sources\boot.wim c:\Winpe\Sources
   ```

3. Copy boot files from the Windows AIK with the following command, where *architecture* is the processor architecture for the computer being used (either x86 or x64).

```
Xcopy c:\Program Files\Windows AIK\tools\architecture\boot c:\WinPE\boot
```

4. Run the following command in the folder C:\Program files\Windows AIK\tools *architecture*, where *architecture* is x86 or x64.

```
Oscdimg –n -bc:\winpe\boot\etfsboot.com c:\winpe c:\winpe.iso
```

5. Burn the .iso file Winpe.iso to a DVD by using a third-party DVD mastering program.

NOTE **For more information on creating bootable media, see Chapter 9, "Preparing Windows PE."**

Importing Images

After you have installed and configured the Windows Deployment Services service, you can add more Windows PE boot images (Boot.wim) and Windows 7 install images (Install.wim). This process is straightforward: The files Boot.wim and Install.wim from the \Sources folder on Windows 7 media are used for this purpose. For example, you can add the boot image that MDT 2010 creates to Windows Deployment Services, allowing you to connect to deployment points and run MDT 2010 task sequences across the network.

NOTE **For more information on creating custom boot and install images that you can use with Windows Deployment Services, see Chapter 9 and Chapter 6, "Developing Disk Images."**

Importing Boot Images

To prepare to service client computers, you must import a Windows PE boot image. Although Windows Deployment Services in Windows Server 2008 and later versions includes the boot loader code, it does not include the actual Windows PE boot image. You can import boot images directly from the Windows 7 or Windows Server 2008 R2 source files. You can also customize boot images with hooks into services, such as MDT 2010. For example, MDT 2010 builds custom Windows PE boot images that connect to MDT 2010 deployment points to install operating system builds. You can add these custom Windows PE boot images to Windows Deployment Services to streamline the LTI deployment process.

To import a Windows 7 boot image, perform the following steps:

1. Insert a Windows 7 DVD into the server's DVD-ROM drive or make an installation source available to the server over the network.

2. Right-click the Boot Images folder and then click Add Boot Image. Boot Images is located under Servers, *server_name*, where *server_name* is the name of the Windows Deployment Services server to which you're adding the boot image.
3. On the Image File page, click Browse to select the boot image and then click Open. For example, you can select the default boot image \Sources\Boot.wim on the Windows 7 media.
4. On the Image File page, click Next.
5. On the Image Metadata page, type a name and description of the image and then click Next. The default name and description is derived from the contents of the boot image file.
6. On the Summary page, click Next to add the image to Windows Deployment Services.
7. When the import task is completed, click Finish.

Importing Install Images

Windows 7 includes an installation image on the media. The installation image (Install.wim) can include multiple editions of Windows 7. You can import one or more of these editions into Windows Deployment Services for deployment over the network.

ON THE COMPANION MEDIA **This book's companion media includes a sample script, VRKAddInstallImage.vbs, that demonstrates how to script the addition of installation images to Windows Deployment Services. A similar script, VRKListImages.vbs, demonstrates how to write a script that iterates install images. These scripts are samples only and should be customized to meet the specific needs of your deployment environment.**

To import a Windows 7 install image, perform the following steps:

1. Insert a Windows 7 DVD into the server's DVD-ROM drive or make an installation source available to the server over the network.
2. Right-click the Install Images folder in the Windows Deployment Services management console and then click Add Image Group. Install Images is under Servers, *server_name*, where *server_name* is the name of the Windows Deployment Services server to which you're adding the installation image.
3. Name the Image Group and then click OK. This creates a folder for image import. It also allows you to group similar images together for optimal use of disk space and security.
4. Right-click Install Images and then click Add Install Image.
5. Choose the Image Group you created in the previous steps and then click Next.

6. In the Image File page, click Browse, choose the Install.wim file you're adding to the server, and then click Open. This file is located in the \Sources folder of the Windows 7 DVD. Click Next to continue.
7. Choose the image(s) you want to import from the selections presented on the List Of Available Images page. (Be sure to select only images for which you have licenses.) Click Next.
8. Click Next on the Summary page to begin the import process. The process can take several minutes to finish.
9. When the import task is completed, click Finish.

NOTE **Copying the source files to the local hard drive first and then importing the image into Windows from the local source files is faster than importing the image directly from the DVD.**

Managing and Deploying Driver Packages

A new feature of Windows Deployment Services in Windows Server 2008 R2 is the ability to manage and deploy driver packages when performing deployment. Specifically, you can:

- Add driver packages to a Window Deployment Services server and deploy these driver packages to different client computers based on filtering criteria.
- Add boot-critical driver packages to boot images (supported for Windows 7 and Windows Server 2008 R2 images only).

These new features make it simpler to ensure that the appropriate drivers are available during a deployment.

Deploying Driver Packages to Clients

You can use Windows Deployment Services in Windows Server 2008 R2 to deploy driver packages to client computers using the following methods:

- **Method 1** Make all driver packages available to all clients. This is the simplest approach, and each type of client will use Plug and Play to install the driver package it needs. This method assumes that the devices that need the driver packages are connected to or attached to the clients before you deploy Windows to them. However, this method can cause problems if two or more incompatible drivers are installed on the same client. If this happens, try method 2.
- **Method 2** Create a different driver group for each type of client and add different driver packages to each driver group as needed. A *driver group* is a collection of driver packages on a Windows Deployment Services server. They use filters to define which type of client has access to the driver group based on the client's hardware and the

operating system being installed. You should use this method if you need to install specific driver packages on specific computers or if your hardware environment is too complex for method 1 above to work properly.

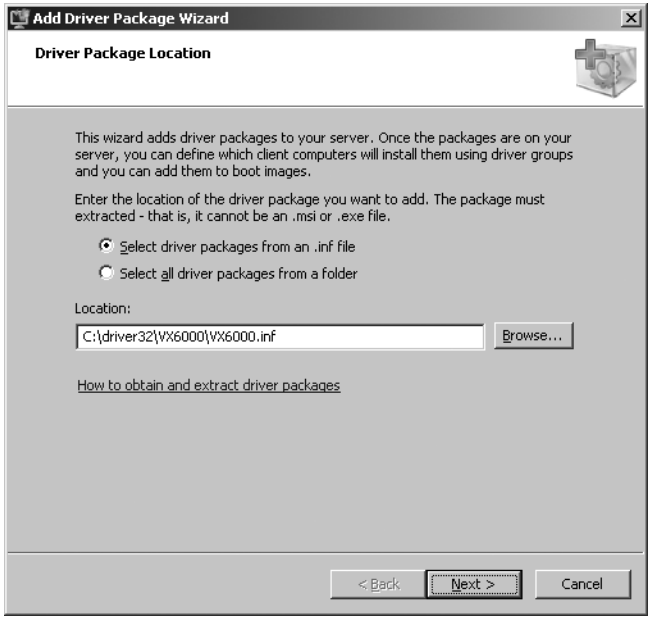
- **Method 3** Create a different driver group for each type of client and add different driver packages to each driver group as needed. Then create an additional driver group and deploy all the driver packages in it to all computers. This method is useful if you have external hardware that is not connected to clients during the installation process. Once the installation is complete, you can connect the hardware and the driver package will install.

The sections that follow describe each method in more detail.

Deploying Driver Packages to Clients Using Method 1

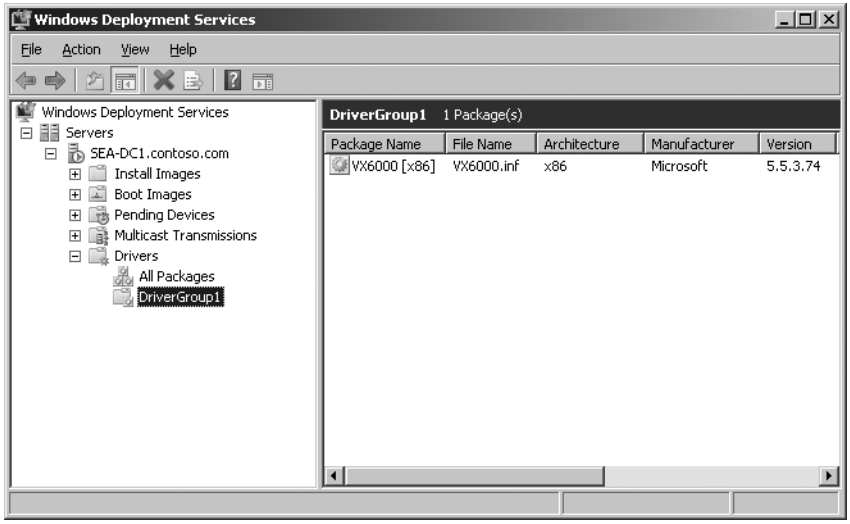
To make all driver packages available to all clients during deployment, do the following:

1. In the Windows Deployment Services console, under the *server_name* node, right-click the Drivers node and select Add Driver Package.
2. Either browse to select a folder containing the driver packages you want to deploy, or browse to select the .inf file of a single driver package you want to deploy, as shown here.



Note that you cannot deploy driver packages that are in the form of .msi or .exe files. You must extract the driver files from these packages to add them to your Windows Deployment Services server.

3. Click Next and select the driver package(s) you want to add to the Windows Deployment Services server.
4. Click Next to add the driver package to the Windows Deployment Services server.
5. Click Next and select the Select An Existing Driver Group option. Then select DriverGroup1 as the driver group to which the driver package will be added. DriverGroup1 is the default driver group and has no filters configured for it. This means that all client computers will have access to the driver packages in this driver group. Plug and Play will ensure that only those driver packages that match the client's hardware will be installed.
6. Finish the Add Driver Packages Wizard. The added driver package will be displayed in the Windows Deployment Services console under DriverGroup1, as shown here.



You can test this approach as follows:

1. Make sure that the device for which the driver package is intended is connected to or attached to a client computer.
2. Use Windows Deployment Services to deploy Windows 7 to the client computer.
3. When the install is finished, log on as an administrator and open Device Manager. Verify that the device drivers needed by the device have been installed and that the device is working properly.

Deploying Driver Packages to Clients Using Method 2

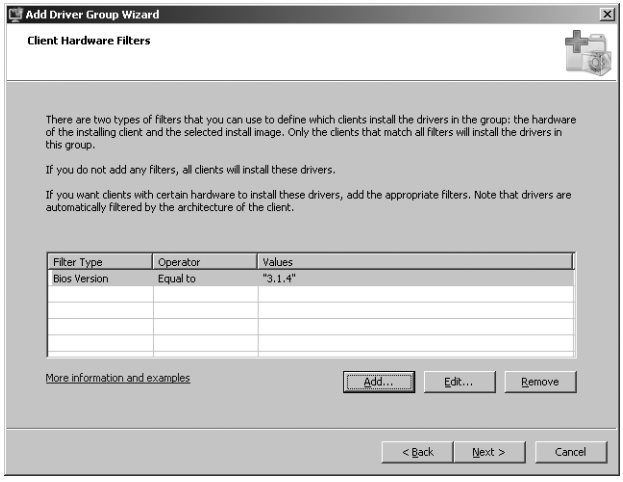
To deploy driver packages to different types of clients using driver groups that have been configured with hardware and/or install image filters, do the following:

1. In the Windows Deployment Services console, under the *server_name* node under the Drivers node, right-click on DriverGroup1 and select Disable. You must disable

DriverGroup1 when performing this method because DriverGroup1 does not have any filters configured on it, which means that all driver packages in DriverGroup1 will be deployed to all clients unless DriverGroup1 is disabled.

2. Right-click on the Drivers node and select Add Driver Group. Type a descriptive name for the driver group.
3. Click Next to display the Client Hardware Filters page of the Add Driver Group Wizard.
4. Click Add to open the Add Filter dialog box.
5. Select a filter type. The available filter types are:
   - Manufacturer
   - BIOS Vendor
   - BIOS Version
   - Chassis Type
   - UUID

   Manufacturer is the most common type of filter used, followed by Chassis Type. The others are typically used for troubleshooting.
6. Select either Equal To or Not Equal To as the operator for the filter.
7. Type a value for the filter and click Add. You can add multiple values to a filter if needed—for example, if the name of the manufacturer has multiple possible spellings.
8. Repeat steps 5 through 7 to add additional filters as needed.
9. Click OK when finished. The added filters are displayed, as shown here.



10. Click Next to display the Install Image Filters page.
11. Click Add to open the Add Filter dialog box.
12. Select a filter type. The available filter types are:
    - OS Version

- OS Edition
- OS Language

13. Select either Equal To or Not Equal To as the operator for the filter.
14. Type a value for the filter and click Add.
15. Repeat steps 12 through 14 to add additional filters as needed.
16. Click OK when finished, and then click Next to display the Packages To Install page.
17. On the Packages To Install page, leave Install Only The Driver Packages That Match A Client's Hardware selected. Click Next and then Finish to complete the Add Driver Group Wizard.
18. Now add the driver packages needed to your new driver group. You can do this in two ways:
    - For driver packages not yet added to the Windows Deployment Services server, right-click the Drivers node and select Add Driver Group. Use the Add Driver Packages Wizard to add driver packages, first to the server and then to the driver group.
    - For driver packages already added to the Windows Deployment Services server but in the wrong driver groups, right-click the driver group you just created and select Add Driver Packages To This Group. Use the Add Driver Packages To *driver_group* Wizard to add the driver packages to the driver group.

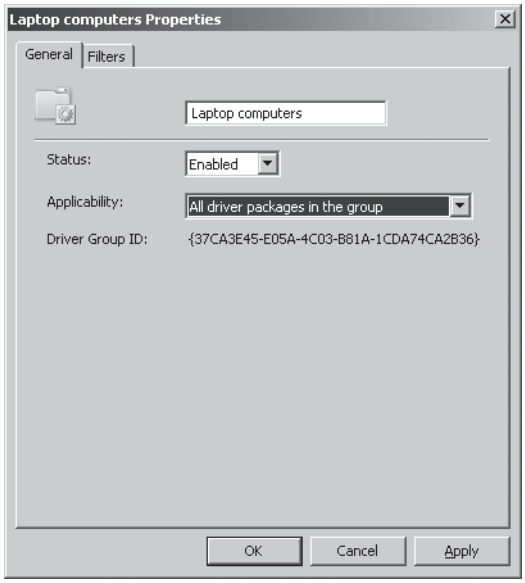
WARNING **Be sure to test this approach carefully before using it in a production environment. In particular, be careful to specify the values of filters exactly as needed—omitting a period or other character can invalidate a filter.**

Deploying Driver Packages to Clients Using Method 3

To deploy driver packages to different types of clients by skipping the running of Plug and Play enumeration, do the following:

1. Complete steps 1 through 16 of method 2, as outlined in the previous section.
2. On the Packages To Install page, select Install All Driver Packages In This Group.
3. Click Next and then Finish to complete the Add Driver Group Wizard. Then add the driver packages needed to the new driver group as described in step 18 of Method 2.

Alternatively, if you already used method 2 to create driver groups with filters and add driver packages to them, you can right-click a driver group, select Properties, and then select All Driver Packages In The Group, as shown on the following page.



WARNING **If incompatible driver packages are deployed using this method, the result can be client computers that fail to boot properly.**

Managing Driver Groups and Driver Packages

You can use Windows Deployment Services in Windows Server 2008 R2 to manage driver groups. For example, you can:

- Enable or disable a driver group.
- Duplicate a driver group. (This creates a new group with the same driver packages and filters. It doesn't make any copies of the files, but just references them again.)
- Modify the filters for a driver group.
- Configure the applicability of a driver group.

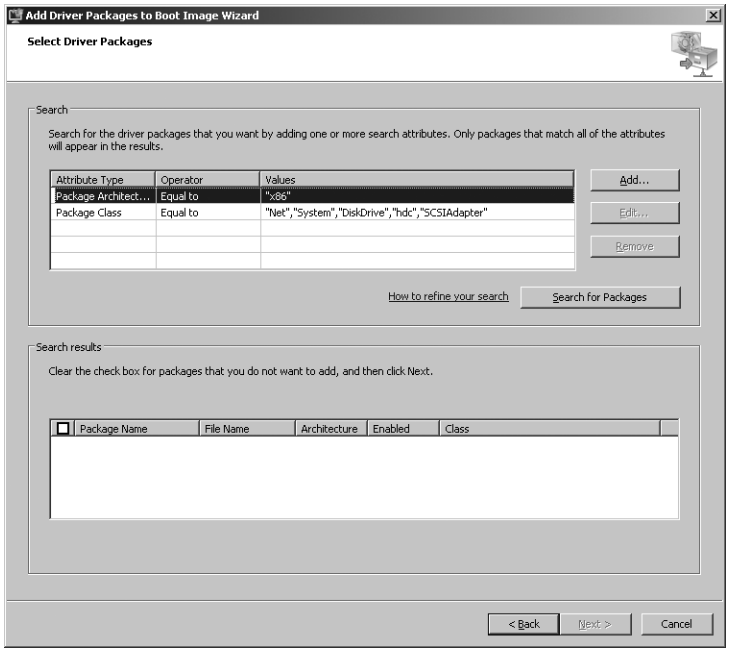
You can also use Windows Deployment Services in Windows Server 2008 R2 to manage driver packages. For example, you can:

- View the properties of a driver package, including its drivers and files.
- Configure the driver groups to which the driver package belongs.
- Enable or disable the driver package.

Adding Driver Packages to Boot Images

You can also use Windows Deployment Services in Windows Server 2008 R2 to add driver packages for boot-critical drivers to boot images. To add a driver package to a boot image, perform the following steps:

1. In the Windows Deployment Services console, under the *server_name* node under the Boot Images node, right-click a boot image and select Export Image to back up your boot image before proceeding further. This is recommended because adding an incompatible or corrupt boot-critical driver to a boot image can render the boot image unbootable and unrepairable.
2. Right-click the boot image again and select Add Driver Packages To Image to start the Add Driver Packages To *driver_group* Wizard.
3. Click Next to display the Select Driver Packages page, as shown here.



4. Click Add or Remove to add or remove filter criteria for finding driver packages that were previously added to your Windows Deployment Services server. Then click Search For Packages to display all driver packages on the server that match your filter criteria.
5. Select the driver packages you want to add to the boot image from your search results. Then, finish the wizard.

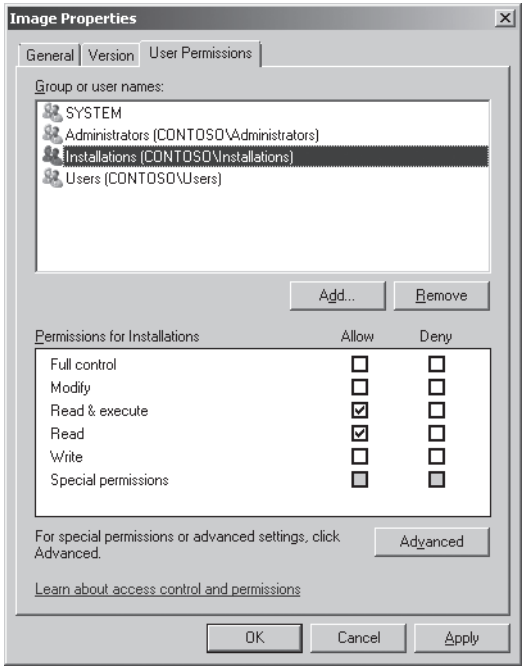
Managing Image Security

It is important to properly secure boot and installation images to prevent their unauthorized use. A fully configured image might include corporate applications and data, proprietary configurations, and even codes and keys required to activate line of business (LOB) applications.

One way to prevent unauthorized installations is by controlling the clients that are allowed to receive images. You can accomplish this through pre-staging, in which clients are registered with AD DS through the use of a globally unique identifier (GUID). Another method is to enable administrative approval for client installations. Finally, you can restrict images by user as shown in the following procedure.

To configure an image file's access control list (ACL), perform the following steps:

1. Right-click the image and then click Properties.
2. On the User Permissions tab, configure the ACL and then click OK. The image's ACL must give a user Read and Execute permissions for the user to be able to install the image. In the following screenshot, members of the Installations group can install the image secured by this ACL.



NOTE **In addition to securing individual images, you can secure image groups. Right-click an image group, click Security, and then configure the group's ACL on the Security tab. By default, images in an image group inherit the group's permissions.**

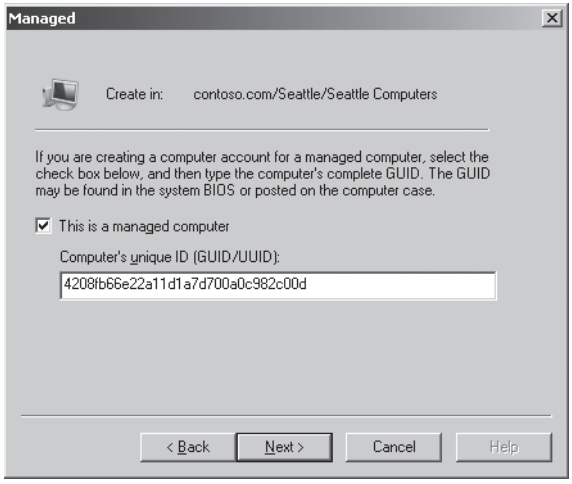
Pre-staging Client Computers

Pre-staging client computer accounts allows you to restrict Windows Deployment Services to respond only to known clients. You can also cause specific Windows Deployment Services servers to respond to the pre-staged client, assign specific install images, and control client provisioning automatically. You configured these settings earlier by setting the PXE Server Initial Settings policy when you installed Windows Deployment Services, as described in the section titled "Installing Windows Deployment Services" earlier in this chapter.

To pre-stage a client computer's account, you will need to know the computer's GUID. You can find this value in the system's BIOS, in the documentation delivered with the system, or on a tag affixed to the computer's case. This value is entered into the AD DS computer account details for the computer to pre-assign its membership in the AD DS infrastructure.

To pre-stage a client computer, perform the following steps:

1. In Active Directory Users And Computers, find the organizational unit (OU) where the computer will be staged.
2. Right-click the OU, click New, and then click Computer.
3. Type a name for the computer and then click Next. If you want, click Change to choose the user or group with permission to join this computer to the domain.
4. On the Managed page, select the check box next to This Is A Managed Computer. Type the computer's GUID, as shown here, and then click Next.



5. On the Host Server page, choose Any Available Remote Installation Server or select the Windows Deployment Services server that will serve this client. Click Next.
6. Click Finish to complete the wizard.

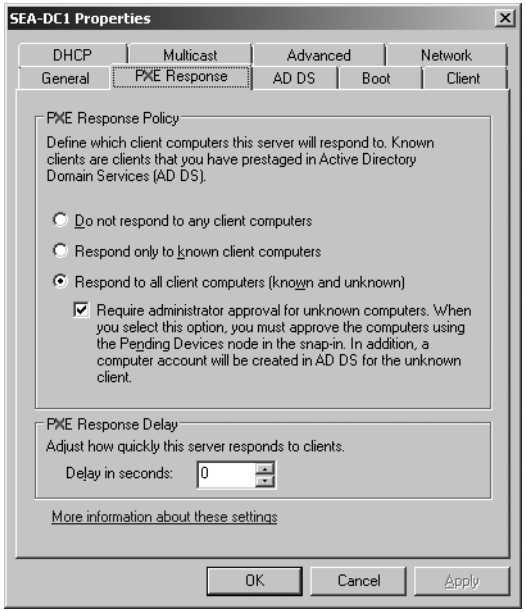
NOTE **You can also pre-stage client computer accounts using the *WDSUTIL /Add-Device* command.**

Configuring Administrator Approval

An alternative to pre-staging computers or allowing unrestricted access to Windows Deployment Services images is to require administrator approval before allowing installation. You accomplish this on the PXE Response tab of each server. You can also configure this by setting the PXE Server Initial policy when you install Windows Deployment Services, as described in the section titled "Installing Windows Deployment Services" earlier in this chapter.

To require administrative approval for unknown computers, begin by granting Domain Admin permissions to the computer account of the Windows Deployment Services server. Instructions on how to do this can be found at *http://technet.microsoft.com/en-us/library/cc754005.aspx* under the heading "Approve a Pending Computer." Then perform the following steps:

1. In the Windows Deployment Services management console, right-click the server and then click Properties.
2. On the PXE Response tab, click Respond To All Client Computers (Known And Unknown) and then select the Require Administrator Approval For Unknown Computers check box, as shown here.



Systems booted to Windows PE will enter a pending state until an administrator approves their installation. You can view systems in this state in the Pending Devices item of the Windows Deployment Services management console.

Installing Windows 7

For ease of installing Windows 7, client computers must support booting from the network. Windows Deployment Services uses PXE technology to boot from the network and start the Windows Deployment Services client. You must also ensure that the computer's BIOS is configured to boot from the network.

To install Windows 7 from Windows Deployment Services, perform the following steps:

1. Start or reboot the client computer.
2. When the client computer starts and the Windows Deployment Services boot loader prompts you to press F12, press F12 to download and start the Windows Deployment Services client. Make sure you enable network boot in the computer's BIOS.
3. On the Windows Deployment Services page, choose a locale and keyboard layout and then click Next.
4. When prompted to connect to the Windows Deployment Services server, type the user account and password to use for the connection and then click OK.
5. On the Select The Operating System You Want To Install page, choose an operating system image and then click Next.
6. On the Where Do You Want To Install Windows? page, choose a partition on which to install Windows 7 and then click Next. To repartition the disk using advanced options, click Drive Options (Advanced).
7. Windows Setup will install Windows 7, prompting for required settings that are not specified in an unattended-setup answer file.

Capturing Custom Images

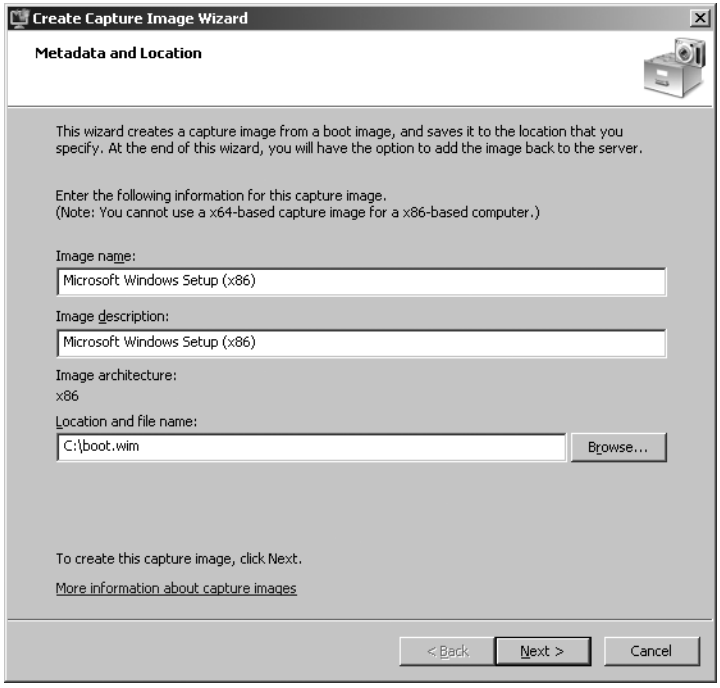
Windows Deployment Services can deploy more than just default images from the Windows 7 media. You can also create custom boot images and install images and then import them into Windows Deployment Services for automated distribution. Chapter 9 describes how to create custom Windows PE boot images. After creating a custom image, you can import it using the instructions in the section titled "Importing Images" earlier in this chapter.

To create a custom installation image for Windows Deployment Services, you must install an existing image on a reference computer, customize the reference computer as desired by adding drivers and applications, and then capture an image of the reference computer. Image capture is a two-step process. First, you must create a Windows PE capture image to support the image-capture process. Then you capture an image from a reference computer that was prepared for imaging using the Sysprep utility.

To create an image-capture image, perform the following steps:

1. Click the Boot Images item in the Windows Deployment Services console tree.

2. Right-click the image to use as a capture image and then click Create Capture Boot Image.
3. On the Metadata And Location page, type a name and description for the capture image and then specify the location and file name of the image file to create, as shown here. Click Next to create the capture image.



4. Click Finish.
5. Import the custom capture boot image by using the instructions in the section titled "Importing Images" earlier in this chapter. Note that Windows Deployment Services in Windows Server 2008 R2 includes a check box on the last page of this wizard that you can select to automatically re-import the image.

To create a custom Windows 7 install image, perform the following steps:

1. Create a master installation by installing a Windows 7 image on a reference computer and then customizing the installation to meet your requirements. You can install Windows 7 on your reference computer either from media or using Windows Deployment Services. To learn how to install Windows 7 using Windows Deployment Services, see the section titled "Installing Windows 7" earlier in this chapter.
2. From a command prompt on the master computer, change directories to \Windows \System32\Sysprep and run the following command.

```
Sysprep /oobe /generalize /reboot
```

3. When the reference computer reboots and the Windows Deployment Services boot loader prompts you to press F12, press F12 to download and start the Windows Deployment Services client. Make sure you enable network boot in the computer's BIOS.
4. In Windows Boot Manager, select the capture boot image.
5. On the Windows Deployment Wizard Image Capture Wizard, click Next.
6. On the Image Capture Source page, choose the volume to capture from the Volume To Capture list and then provide a name and description for the image. Click Next to continue. (Note that if you have omitted step 2, you won't see any volumes available at this point. This has been a major area of customer confusion.)
7. On the Image Capture Destination page, click Browse to choose the location where you want to store the captured image. In the File Name text box, type a name for the image using the .wim extension and then click Save. Click Upload Image To WDS Server, type the name of the Windows Deployment Services server, and click Connect. If prompted for credentials, provide a user name and password for an account with sufficient privileges to connect to the Windows Deployment Services server. Choose the image group in which to store the image from the Image Group list.
8. Click Finish.

NOTE **The reason for saving a local copy in step 7 rather than just uploading it to the server immediately is to minimize the chances of corruption occurring over the network.**

Creating Multicast Transmissions

Multicasting enables you to deploy an image to numerous client computers at the same time without overburdening the network. By using multicast, you transmit image data only once, drastically reducing the amount of network bandwidth that is used to deploy images from Windows Deployment Services.

Consider implementing multicasting if your organization:

- Has network routers that support multicasting.
- Is a large company that requires many concurrent client installations. If your organization deploys images to only a small number of computers at the same time, multicasting might not be the right choice.
- Wants to use network bandwidth efficiently. With this feature, images are sent over the network only once, and you can specify limitations (for example, to only use 10 percent of your bandwidth). If your organization does not have bandwidth overload issues, multicasting might not be worth the effort.

- Has enough disk space on client computers for the image to be downloaded. When multicasting, Windows Deployment Services downloads the image to the client computer instead of installing it from the server.
- Meets the requirements listed in the following section.

Multicast Prerequisites

To use multicast in your organization, it must meet all the following requirements:

- Routers that support multicasting. In particular, your network infrastructure needs to support the Internet Group Management Protocol (IGMP) to properly forward multicast traffic. Without the IGMP, multicast packets are treated as broadcast packets, which can lead to network flooding.
- At least one install image that you want to transmit on the server.
- The Boot.wim file located in the \Sources folder on Windows 7 or Windows Server 2008 R2 media.
- IGMP snooping should be enabled on all devices. This will cause your network hardware to forward multicast packets only to those devices that are requesting data. If IGMP snooping is turned off, multicast packets are treated as broadcast packets and will be sent to every device in the subnet.

Transmission Types

There are two types of multicast transmissions:

- **Auto-Cast** This option indicates that as soon as an applicable client requests an install image, a multicast transmission of the selected image begins. Then, as other clients request the same image, they are also joined to the transmission that is already started.
- **Scheduled-Cast** This option sets the start criteria for the transmission based on the number of clients that are requesting an image, a specific day and time, or both. If you do not select either of the check boxes in Scheduled-Cast, the transmission will not start until you manually start it. Note that in addition to these criteria, you can start a transmission manually at any time by right-clicking it and then clicking Start.

Performing Multicast Deployment

Multicast deployment requires using the Windows Deployment Services server role in Windows Server 2008 or Windows Server 2008 R2. The Windows Deployment Services update for Windows Server 2003 SP1 and later versions does not support multicast deployment.

Multicast deployment is supported for install images only. The Boot.wim file used for multicast deployment must be imported from Windows Server 2008, Windows Vista SP1 or later versions, Windows 7, or Windows Server 2008 R2 media.

New features of multicast deployment for the Windows Deployment Services server role in Windows Server 2008 R2 include the following:

- Enables Windows Deployment Services to automatically disconnect slow clients and to divide transmissions into multiple streams based on client speeds. Note that while multicast deployment requires Windows 3.0, auto-disconnect will work with Windows PE 2.1 or 3.0.
- Supports multicast deployment in IPv6 environments. This feature requires that the boot image comes from Windows Vista SP1 or later versions, Windows Server 2008, Windows 7, or Windows Server 2008 R2.
- Supports boot images for computers that use x64 EFI. This feature can be managed using the Wdsutil command only.

MORE INFO **For more information on performing multicast deployment using Windows Deployment Services, see "Performing Multicast Deployments" at *http://technet.microsoft.com/en-us/library/dd637994.aspx*.**

Using Windows Deployment Services with Microsoft Deployment Toolkit

For LTI, MDT 2010 generates Windows PE boot images that connect to the deployment point and starts the Windows Deployment Wizard. The Windows Deployment Wizard allows the user to select an operating build to configure, applications to install, and so on.

MDT 2010 generates boot images when you update deployment points. MDT 2010 generates .iso image files that you can burn to DVD. You find these boot images in the \DeploymentShare$\Boot folder on your MDT 2010 technician computer. The file name is LiteTouchPE_*platform*.iso, where *platform* is x86 or x64. After you burn the .iso image to DVD and then use this DVD to start destination computers.

MDT 2010 also generates Windows PE .wim boot images that you can add to Windows Deployment Services. Starting the MDT 2010 Windows PE boot images by using Windows Deployment Services is more convenient and quicker than using DVDs. You find these boot images in the \DeploymentShare$\Boot folder on your MDT 2010 technician computer. The file name is LiteTouchPE_*platform*.wim, where *platform* is x86 or x64. You can import this boot image into Windows Deployment Services using the instructions in the section titled "Importing Images" earlier in this chapter.

ON THE COMPANION MEDIA **This book's companion CD includes a sample script, VRKAddBootImage.vbs, that adds boot images to Windows Deployment Services. You can use this script to quickly add MDT 2008 boot images. These scripts are samples only and should be customized to meet the specific needs of your deployment environment.**

MDT 2010 can also use Windows 7 installation images from Windows Deployment Services. By doing so, you can use installation sources that already exist in a Windows Deployment Services server without duplicating the files in an MDT 2010 deployment share. This requires that you copy Wdsclientapi.dll, Wdscsl.dll, and Wdsimage.dll from the \Sources folder of the Windows 7 media to the C:\Program Files\Microsoft Deployment Toolkit\Bin folder. It also requires that at least one Windows 7 source must exist within the deployment share and that you must create and update a deployment share. MDT 2010 uses the setup program files from the deployment share to install the Windows 7 image from the Windows Deployment Services server.

To add images from Windows Deployment Services to an MDT 2010 deployment share, perform the following steps:

1. Add a full set of Windows 7 source files to an MDT 2010 deployment share. See Chapter 6 for more information on this topic.
2. Copy the following files from the \Sources folder of the Windows 7 media to the C:\Program Files\Microsoft Deployment Toolkit\Bin folder:
   - Wdsclientapi.dll
   - Wdscsl.dll
   - Wdsimage.dll
3. In the MDT 2010 Deployment Workbench console tree, right-click Operating Systems under your deployment share and click Import Operating System to start the New Operating System Wizard.
4. On the OS Type page, select Windows Deployment Services Images and then click Next to add an image from a Windows Deployment Services server to the distribution share.
5. On the WDS Server page, type the name of the Windows Deployment Services server from which to add the operating system images and then click Finish.
6. Deployment Workbench adds all the installation images it finds in Windows Deployment Services to the Operating Systems folder.

Summary

Windows Deployment Services provides a solution for the network-based installation of Windows 7. It's built on standard Windows 7 setup technologies, including Windows PE, .wim image files, and image-based setup. Using Windows Deployment Services can help reduce the cost and complexity of Windows 7 deployments.

In Windows Server 2008 R2, Windows Deployment Services replaces RIS. Windows Deployment Services is also available as an update for Windows Server 2003, and it provides a clear migration path from RIS for customers using legacy RIS images.

Although Windows Deployment Services provides the technology necessary to capture and remotely deploy custom operating system images, it does not provide end-to-end technology or guidance for high-volume deployment projects. It also does not provide tools or guidance for customizing the custom images you deploy with settings, applications, device drivers, and so on. MDT 2010 builds on Windows Deployment Services by adding both end-to-end guidance and tools for building and customizing images and then deploying them by using Windows Deployment Services.

Additional Resources

These resources contain additional information and tools related to this chapter.

Related Information

- The "Windows Deployment Services" section of the Windows Server TechCenter on Microsoft TechNet at *http://technet.microsoft.com/en-us/library/cc772106.aspx* for more information concerning Windows Deployment Services.
- *Infrastructure Planning and Design Guide for Windows Deployment Services* at *http://technet.microsoft.com/en-us/library/cc265612.aspx*.
- *Windows Deployment Services Getting Started Guide* at *http://go.microsoft.com/fwlink/?LinkId=84628* for step-by-step instructions on using Windows Deployment Services.
- The "Setup Deployment" forum on Microsoft TechNet at *http://go.microsoft.com/fwlink/?LinkId=87628*.
- Chapter 6, "Developing Disk Images," includes more information about building custom Windows Vista images that you can deploy using Windows Deployment Services.
- Chapter 9, "Preparing Windows PE," includes more information about creating custom Windows PE images that you can use with Windows Deployment Services.
- Chapter 12, "Deploying with Microsoft Deployment Toolkit," includes more information about using Windows Deployment Services to deploy Windows Vista with MDT 2008 and Windows Deployment Services.

On the Companion Media

- VRKAddBootImage.vbs
- VRKAddInstallImage.vbs
- VRKListImages.vbs

CHAPTER 11

Using Volume Activation

Volume Activation is a configurable solution that helps IT professionals automate and manage the product activation process on computers running the Windows Vista, Windows 7, Windows Server 2008, and Windows Server 2008 R2 operating systems licensed under a Microsoft Volume Licensing program and other programs that provide Volume License editions of Microsoft Windows. This chapter describes Volume Activation for Windows 7.

Introduction

Product activation is the process of validating software with the manufacturer. Activation confirms the genuine status of a product and that the product key is not compromised. It is analogous to the activation of credit cards or new mobile phones. Activation establishes a relationship between the software's product key and a particular installation of that software on a device.

All methods of activation used by Microsoft are designed to help protect user privacy. Data that is sent during activation is not traceable to the computer or user. The data that is gathered is used to confirm a legally licensed copy of the software. It is then aggregated for statistical analysis. Microsoft does not use this information to identify or contact the user or organization. For example, during online activations, information such as the

software version, language, and product key are sent, as well as the IP address and information about the hardware of the device. The IP address is used only to verify the location of the request, as some editions of Windows—such as Windows 7 Starter—can be activated only within certain target market geographies.

ON THE COMPANION MEDIA **The complete set of Volume Activation guides is included on the companion media. The *Volume Activation Planning Guide* provides guidance for deployment planning. The *Volume Activation Deployment Guide* includes detailed guidance for deploying Volume Activation in enterprise environments. Also, the *Volume Activation Operations Guide* describes how to support Volume Activation in enterprise environments. The *Volume Activation Technical Reference Guide* is a useful reference for Volume Activation.**

Activation Options

Licenses for Windows 7 can be obtained through one of three basic channels: retail, Original Equipment Manufacturer (OEM), or Volume Licensing. Each channel has its own unique methods of activation. Because organizations can obtain their operating systems through any of the three available channels, they can choose a combination of activation methods.

Retail

Windows 7 products acquired through a retail store are licensed individually and are activated in the same way as retail versions of Windows Vista. Each purchased copy comes with one unique product key, found on the product packaging, which is typed in during the installation of the product. The computer uses this product key to complete the activation after the installation of the operating system is complete. This activation can be accomplished either online or by telephone.

Original Equipment Manufacturer

Most OEMs sell systems that include a standard build of Windows 7. Hardware vendors perform OEM activation by associating Windows with the firmware (basic input/output system, or BIOS) of the physical computer. This process occurs before the computers are sent to the customer so that no additional actions are required of the user. This method of activation is known as OEM Activation.

OEM Activation is valid as long as the customer uses the OEM-provided image on a system. To create a customized image, customers can use the image provided by the OEM as the basis for creating the custom image. Otherwise, a different activation method must be used.

> ***NOTE*** **Some editions of Windows 7, such as Windows 7 Enterprise, are available only through the Volume Licensing channel. OEM Activation is applicable to computers purchased through OEM channels with Windows installed.**

Volume Licensing

Volume Licensing offers customized programs tailored to the size and purchasing preference of the organization. These programs provide simple, flexible, and affordable solutions that enable organizations to manage their licenses. To become a Volume Licensing customer, an organization needs to set up a Volume License agreement with Microsoft.

There are only two legal ways to acquire a full Windows desktop license for a new computer system. The first and most economical way is preinstalled through the computer hardware manufacturer. The other option is with a full, packaged retail product. Volume Licensing programs such as Open License, Select License, and Enterprise agreements cover Windows upgrades only and do not provide a full Windows desktop license. After the computers have a full Windows desktop license, a Windows Volume Licensing agreement can be acquired and used to provide version upgrade rights. For more information on Volume Licensing, go to *http://go.microsoft.com/fwlink/?LinkId=73076*.

Volume Activation is designed to allow Volume License customers to automate the activation process in a way that is transparent to users. Volume Activation applies to computers that are covered under a Volume Licensing program. It is used strictly as a tool for activation and is in no way tied to license invoicing or billing. Volume Activation provides two different models for completing volume activations: Key Management Service (KMS) and Multiple Activation Key (MAK). KMS allows organizations to activate systems within their own network, whereas MAK activates systems on a one-time basis using Microsoft's hosted activation services.

Customers can use either or both key types to activate systems in their environment. The model chosen depends on the size, network infrastructure, connectivity, and security requirements of the organization. IT professionals can choose to use just one or a combination of these activation models. For more information about choosing an activation model, see the section titled "Volume Activation Scenarios," later in this chapter.

DIRECT FROM THE SOURCE

Choosing the Activation Method

Kim Griffiths, Product Manager
Genuine Windows

Aaron Smith, Program Manager
Windows Genuine Platform Team

Which method to use? That is one of the most common questions that we hear from our customers about Volume Activation. It is a decision that you need to make before any systems are deployed. When we were designing Volume Activation, it was clear that there were a wide variety of customer deployment models and use cases that needed to be considered. For example, a well-connected, global corporate intranet would have very different requirements from a disconnected development and test lab. Accordingly, two methods were developed to give the level of flexibility that our customers needed: KMS and MAK. Customers can use one or both methods, depending on how they deploy and use their machines.

KMS is the recommended solution for most customer use cases, for a variety of reasons. First, it is automated and simple for the administrator to configure. The KMS clients detect and use the service for activation on their own, without any configuration changes to the image or end-user involvement. Second, activation happens within the customer environment. After the service is activated, all communication stays inside the organization. None of the KMS clients will ever connect to Microsoft to activate.

MAK is best suited to a smaller set of systems, individual stand-alone machines, or those that are disconnected from the corporate network. It is very similar to retail activation and can be configured as part of system provisioning, making it transparent to the end user as well.

Key Management Service

KMS activates computers on a local network, eliminating the need for individual computers to connect to Microsoft. To do this, KMS uses a client–server topology. KMS clients can locate KMS hosts by using Domain Name System (DNS) or a static configuration. KMS clients contact the KMS host by using Remote Procedure Call (RPC). KMS can be hosted on computers running the Windows 7, Windows Vista, Windows Server 2008 R2, Windows Server 2008, or Windows Server 2003 operating systems.

Minimum Computer Requirements

If you are planning to use KMS activation, the network must meet or exceed the activation threshold (the minimum number of qualifying computers that KMS requires). IT professionals must also understand how the KMS host tracks the number of computers on the network.

KMS Activation Thresholds

KMS can activate both physical computers and virtual machines (VMs). To qualify for KMS activation, a network must have a minimum number of qualifying computers, called the *activation threshold*. KMS hosts activate clients only after meeting this threshold. To ensure that the activation threshold is met, a KMS host counts the number of computers requesting activation on the network.

The Windows Server operating systems (starting with Windows Server 2008) and Windows client operating systems (starting with Windows Vista) are activated after meeting different thresholds. The Windows Server activation threshold is 5 computers, and the Windows client activation threshold is 25 computers. The threshold includes Windows client and server operating systems running on physical computers or VMs.

A KMS host responds to each valid activation request from a KMS client with the count of how many computers have contacted the KMS host for activation. Clients that receive a count below their activation threshold are not activated. For example, if the first two computers that contact the KMS host are running Windows 7, the first receives an activation count of 1, and the second receives an activation count of 2. If the next computer is a Windows 7 VM, it receives an activation count of 3, and so on. None of these computers is activated because computers running Windows 7 must receive an activation count greater than or equal to 25 to be activated. KMS clients in the grace state that are not activated because the activation count is too low will connect to the KMS host every two hours to get the current activation count and will be activated when the threshold is met.

If the next computer that contacts the KMS host is running Windows Server 2008 R2, it receives an activation count of 4, because activation counts are a combination of computers running Windows Server 2008 R2 and Windows 7. If a computer running Windows Server 2008 or Windows Server 2008 R2 receives an activation count that is greater than or equal to 5, it is activated. If a computer running Windows 7 receives an activation count greater than or equal to 25, it is activated.

Activation Count Cache

To track the activation threshold, the KMS host keeps a record of the KMS clients that request activation. The KMS host gives each KMS client a client machine identification (CMID) designation, and the KMS host saves each CMID in a table. Each activation request remains in the table for 30 days. When a client renews its activation, the cached CMID is removed from the table, a new record is created, and the 30-day period begins again. If a KMS client does not

renew its activation within 30 days, the KMS host removes the corresponding CMID from the table and reduces the activation count by 1.

The KMS host caches twice the number of CMIDs that KMS clients require to help ensure that the CMID count does not drop below the activation threshold. For example, on a network with clients running Windows 7, the KMS activation threshold is 25. The KMS host caches the CMIDs of the most recent 50 activations. The KMS activation threshold for Windows Server 2008 R2 is 5. A KMS host that is contacted only by clients running Windows Server 2008 R2 KMS would cache the 10 most recent CMIDs. If a client running Windows 7 later contacts that KMS host, KMS increases the cache size to 50 to accommodate the higher threshold. KMS never reduces the cache size.

How KMS Works

KMS activation requires Transmission Control Protocol/Internet Protocol (TCP/IP) connectivity. By default, KMS hosts and clients use DNS to publish and find the KMS. The default settings can be used, which require little to no administrative action, or KMS hosts and clients can be configured manually based on network configuration and security requirements.

KMS Activation Renewal

KMS activations are valid for 180 days. This is called the *activation validity interval*. To remain activated, KMS clients must renew their activation by connecting to the KMS host at least once every 180 days. By default, KMS client computers attempt to renew their activation every seven days. If KMS activation fails, the client will reattempt every two hours. After a client's activation is renewed, the activation validity interval begins again.

Publication of the KMS

The KMS uses service (SRV) resource records (RRs) in DNS to store and communicate the locations of KMS hosts. KMS hosts use Dynamic DNS (DDNS), if available, to publish the KMS SRV RRs. If DDNS is not available, or the KMS host does not have rights to publish the RRs, the DNS records must be published manually or IT professionals must configure client computers to connect to specific KMS hosts. The *Volume Activation Deployment Guide* at *http://go.microsoft.com/fwlink/?LinkId=150083* describes the steps necessary to publish the KMS in DNS.

> ***NOTE*** **DNS changes may take time to propagate to all DNS hosts, depending on the complexity and topology of the network.**

Client Discovery of the KMS

By default, KMS clients query DNS for KMS information. The first time a KMS client queries DNS for KMS information, it randomly chooses a KMS host from the list of SRV RRs that DNS returns.

The address of a DNS server containing the SRV RRs can be listed as a suffixed entry on KMS clients, which allows advertisement of SRV RRs for KMS in one DNS server and allows KMS clients with other primary DNS servers to find KMS.

Also, priority and weight parameters can be added to the DnsDomainPublishList registry value for KMS. Doing so allows IT professionals to establish KMS host priority groupings and weighting within each group, which specify the KMS host to try first, to balance traffic among multiple KMS hosts. Only Windows 7 and Windows Server 2008 R2 use the priority and weight parameters.

If the KMS host that a client selects does not respond, the KMS client removes that KMS host from its list of SRV RRs and randomly selects another KMS host from the list. After a KMS host responds, the KMS client caches the name of the KMS host and uses it for subsequent activation and renewal attempts. If the cached KMS host does not respond on a subsequent renewal, the KMS client discovers a new KMS host by querying DNS for KMS SRV RRs.

By default, client computers connect to the KMS host for activation by using anonymous RPCs through TCP port 1688. (IT professionals can change the default port.) After establishing a TCP session with the KMS host, the client sends a single request packet. The KMS host responds with the activation count. If the count meets or exceeds the activation threshold for that operating system, the client is activated and the session is closed. The KMS client uses this same process for renewal requests. The communication each way is 250 bytes.

Planning a KMS Deployment

The KMS does not require a dedicated server. The KMS can be co-hosted with other services, such as Active Directory Domain Services (AD DS) domain controllers and read-only domain controllers (RODCs). KMS hosts can also run on physical computers or VMs running any supported Windows operating system, including Windows Server 2003. Although a KMS host running on Windows Server 2008 R2 can activate any Windows operating system that supports Volume Activation, a KMS host running on Windows 7 can activate only Windows client operating systems. A single KMS host can support unlimited numbers of KMS clients; however, Microsoft recommends deploying a minimum of two KMS hosts for failover. Most organizations can use as few as two KMS hosts for their entire infrastructure.

> ***NOTE*** **KMS is not included automatically in Windows Server 2003. To host KMS on machines running Windows Server 2003, download and install KMS for Windows Server 2003 SP1 and later from *http://go.microsoft.com/fwlink/?LinkID=82964*. KMS is available in several languages. The 64-bit version is available at *http://go.microsoft.com/fwlink/?LinkId=83041*.**

Planning DNS Server Configuration

The default KMS auto-publishing feature requires SRV RR and DDNS support. Microsoft DNS or any other DNS server that supports SRV RRs (per Internet Engineering Task Force [IETF] RFC 2782) and dynamic updates (per RFC 2136) can support KMS client default behavior and KMS SRV RR publishing. Berkeley Internet Domain Name (BIND) versions 8.*x* and 9.*x* support both SRV records and DDNS, for example.

The KMS host must be configured so that it has the credentials needed to create and update SRV, A (IP version 4, or IPv4), and AAAA (IP version 6, or IPv6) RRs on the DDNS servers, or the records need to be created manually. The recommended solution for giving the KMS host the needed credentials is to create a security group in AD DS and add all KMS hosts to that group. In the Microsoft DNS server, ensure that this security group is given full control over the _VLMCS._TCP record on each DNS domain that will contain the KMS SRV RRs.

Activating the First KMS Host

KMS hosts on the network need to install a KMS key and then be activated with Microsoft. Installation of a KMS key enables the KMS on the KMS host. After installing the KMS key, complete the activation of the KMS host by telephone or online. Beyond this initial activation, a KMS host does not communicate any information to Microsoft.

KMS keys are installed only on KMS hosts, never on individual KMS clients. Windows 7 and Windows Server 2008 R2 have safeguards to help prevent inadvertently installing KMS keys on KMS client computers. Any time users try to install a KMS key, they see a warning, but they can continue to install the KMS key.

Activating Subsequent KMS Hosts

Each KMS key can be installed on up to six KMS hosts, which can be physical computers or VMs. After activating a KMS host, the same host can be reactivated up to nine more times with the same key.

If the organization needs more than six KMS hosts, IT professionals can request additional activations for the organization's KMS key. An example of this would be if 10 separate physical locations were under one Volume Licensing agreement, and IT wanted each location to have a local KMS host. To request this exception, call the Activation Call Center. For more information, see the Volume Licensing Web site at *http://go.microsoft.com/fwlink/?LinkID=73076*.

Upgrading Existing KMS Hosts

KMS hosts operating on Windows Server 2003, Windows Vista, or Windows Server 2008 can be configured to support KMS clients running Windows 7 and Windows Server 2008 R2. For Windows Vista and Windows Server 2008, it will be necessary to update the KMS host with a package containing the files supporting the expanded KMS client support. This package is available through the Microsoft Download Center at *http://www.microsoft.com/downloads* or through Windows Update and Windows Server Update Services (WSUS). Once the KMS host

is updated, a KMS key that is designed to support Windows 7 and Windows Server 2008 R2 can be applied as described earlier in this chapter. Note that a KMS key supporting these new versions of Windows provides backward support for all previous versions of Volume License editions of Windows acting as KMS clients.

In the case of updating a Windows Server 2003 KMS host, all necessary files are contained within the KMS 1.2 downloadable package, which is available through the Microsoft Download Center at *http://www.microsoft.com/downloads*. Once the KMS host is updated, a KMS key designed to support Windows 7 and Windows Server 2008 R2 can be applied as described earlier in this chapter. A KMS key supporting these new versions of Windows provides backward support for all previous versions of Volume License editions of Windows acting as KMS clients.

Planning KMS Clients

By default, computers running Volume Licensing editions of Windows Vista, Windows 7, Windows Server 2008, and Windows Server 2008 R2 are KMS clients, and no additional configuration is needed. KMS clients can locate a KMS host automatically by querying DNS for SRV RRs that publish the KMS. If the network environment does not use SRV RRs, a KMS client can be configured manually to use a specific KMS host. The steps needed to configure KMS clients manually are described in the *Volume Activation Deployment Guide* at *http://go.microsoft.com/fwlink/?LinkId=150083*.

Activating as a Standard User

Windows 7 does not require administrator privileges for activation. However, this change does not allow standard user accounts to remove Windows 7 from the activated state. An administrator account is required for other activation- or license-related tasks, such as rearming.

Multiple Activation Key

A MAK is used for one-time activation with Microsoft's hosted activation services. Each MAK has a predetermined number of allowed activations; this number is based on Volume Licensing agreements and does not match the organization's exact license count. Each activation using a MAK with Microsoft's hosted activation service counts toward the activation limit.

There are two ways to activate computers using a MAK:

- **MAK Independent activation** MAK Independent activation requires that each computer independently connect and be activated with Microsoft, either over the Internet or by telephone. MAK Independent activation is best suited for computers within an organization that do not maintain a connection to the corporate network.
- **MAK Proxy activation** MAK Proxy activation enables a centralized activation request on behalf of multiple computers with one connection to Microsoft. MAK Proxy activation is configured using the Volume Activation Management Tool (VAMT).

MAK Proxy activation is appropriate for environments in which security concerns may restrict direct access to the Internet or the corporate network. It is also suited for development and test labs that lack this connectivity.

MAK is recommended for computers that rarely or never connect to the corporate network and for environments in which the number of computers needing activation does not meet the KMS activation threshold. MAK can be used for individual computers or with an image that can be bulk-duplicated or installed using Microsoft deployment solutions. MAK can also be used on a computer that was configured originally to use KMS activation—useful for moving a computer off the core network to a disconnected environment.

Volume Activation Management Tool

Included in the Windows Automated Installation Kit (Windows AIK), VAMT is a stand-alone application that collects activation requests from several computers and then sends them to Microsoft in bulk. VAMT allows IT professionals to specify a group of computers to activate using AD DS, workgroup names, IP addresses, or computer names. After receiving the activation confirmation IDs, VAMT distributes them to the computers that requested activation. Because VAMT also stores these confirmation IDs locally, it can reactivate a previously activated computer after it is reimaged without recontacting Microsoft. The communication between VAMT and client computers is via Windows Management Instrumentation (WMI), so Windows Firewall on client computers must be configured to allow WMI traffic. Additionally, VAMT can be used to transition computers easily between MAK and KMS activation methods. Download Windows AIK, which includes VAMT, at *http://go.microsoft.com/fwlink/?LinkId=136976*.

MAK Architecture

MAK Independent activation installs a MAK product key on a client computer and instructs that computer to activate itself against Microsoft servers over the Internet. In MAK Proxy activation, VAMT installs a MAK product key on a client computer, obtains the Installation Identifier (IID) from the target computer, sends the IID to Microsoft on behalf of the client, and obtains a Confirmation Identifier (CID). The tool then activates the client by installing the CID.

Volume Activation Scenarios

Each Volume Activation method is best suited to a particular network configuration. To select the best activation method or methods for the organization, assess the network environment to identify how different groups of computers connect to the network. Connectivity to the corporate network, Internet access, and the number of computers that regularly connect to the corporate network are some of the important characteristics to identify. Most medium-sized to large organizations use a combination of activation methods because of the varied ways their client computers connect to their networks.

KMS is the recommended activation method for computers that are well connected to the organization's core network or that have periodic connectivity, such as computers that are offsite. MAK activation is the recommended activation method for computers that are offsite with limited connectivity or that cannot connect to the core network because of security restrictions. These include computers in lab and development environments that are isolated from the core network.

Table 11-1 lists common network configurations and the best practice recommendations for each type. Each solution factors in the number of computers and network connectivity of the activation clients.

TABLE 11-1 Volume Activation Recommendations by Scenario

NETWORK INFRASTRUCTURE	RECOMMENDATIONS	CONSIDERATIONS
Core network Well-connected LAN Most common scenario	If total computers > KMS activation threshold: ■ Small (< 100 machines): KMS host = 1 ■ Medium (> 100 machines): KMS host ≥ 1 ■ Enterprise: KMS host > 1 If total computers ≤ KMS activation threshold: ■ MAK (by telephone or Internet) ■ MAK Proxy	Minimize the number of KMS hosts Each KMS host must consistently maintain a count of total machines > KMS activation threshold KMS hosts are autonomous KMS host is activated by telephone or Internet
Isolated network Branch office, high-security network segments, perimeter networks Well-connected zoned LAN	If ports on firewalls can be opened between KMS clients and hosts: ■ Use KMS hosts in core network If policy prevents firewall modification: ■ Use local KMS hosts in an isolated network ■ MAK (by telephone or Internet) ■ MAK Proxy	Firewall configuration ■ RPC over TCP (TCP port 1688) ■ Initiated by the client Change management on firewall rule sets

NETWORK INFRASTRUCTURE	RECOMMENDATIONS	CONSIDERATIONS
Test or development lab Isolated network	If total computers > KMS activation threshold: ■ KMS host = 1 (per isolated network) If total computers ≤ KMS activation threshold: ■ No activation (reset grace period) ■ MAK (by telephone) ■ MAK Proxy performed manually	Variable configuration Limited number of computers KMS host and MAK activation through telephone; MAK Proxy performed manually
Individual disconnected computer No connectivity to the Internet or core network Roaming computers that periodically connect to the core network or connect through a virtual private network (VPN) Roaming computers with Internet access but no connection to the core network	For clients that connect periodically to the core network: ■ Use the KMS hosts in the core network For clients that never connect to the core network or have no Internet access: ■ MAK (by telephone) For networks that cannot connect to the core network: ■ If total computers > KMS activation threshold: • Small: KMS host = 1 • Medium: KMS host ≥ 1 • Enterprise: KMS host > 1 ■ If total computers ≤ KMS activation threshold, MAK Independent or MAK Proxy performed manually For clients that never connect to the core network but have Internet access: ■ MAK (by Internet)	Restricted environments or networks that cannot connect to other networks KMS host can be activated and then moved to disconnected network KMS host and MAK activation by telephone; MAK Proxy performed manually

The following sections describe examples of Volume Activation solutions in heterogeneous corporate environments that require more than one activation method. Each scenario has a recommended activation solution, but some environments may have infrastructure or policy requirements that are best suited to a different solution.

Core Network

A centralized KMS solution is recommended for computers on the core network. This solution is for networks that have well-connected computers on multiple network segments that also have a connection to the Internet. Figure 11-1 shows a core network with a KMS host. The KMS host publishes the KMS using DDNS. KMS clients query DNS for KMS SRV RRs and activate themselves after contacting the KMS host. The KMS host is activated directly through the Internet.

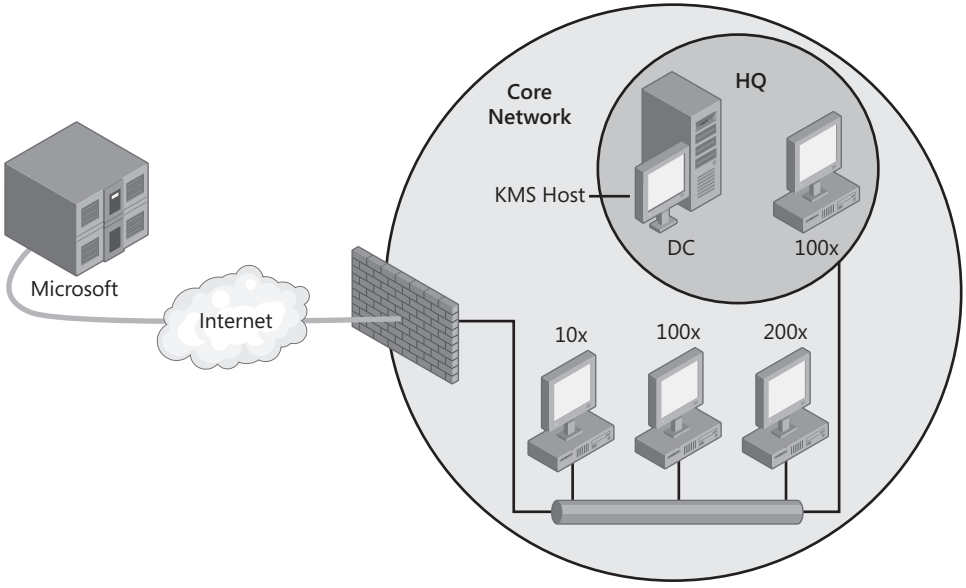


FIGURE 11-1 Core network scenario

NOTE **A KMS host can be installed on a VM, but select a VM that is unlikely to be moved to a different host computer. If the virtual KMS host is moved to a different host computer, the operating system detects the change in the underlying hardware and the KMS host must reactivate with Microsoft. KMS hosts can activate with Microsoft up to nine times.**

Isolated Networks

Many organizations have networks that are separated into multiple security zones. Some networks have a high-security zone that is isolated because it has sensitive information, whereas other networks are separated from the core network because they are in a different physical location (branch office locations).

High-Security Zone

High-security zones are network segments separated by a firewall that limits communication to and from other network segments. If the computers in a high-security zone are allowed access to the core network by allowing TCP port 1688 outbound from the high-security zone and an RPC reply inbound, activate computers in the high-security zone by using KMS hosts located in the core network. This way, the number of client computers in the high-security network does not have to meet any KMS activation threshold.

If these firewall exceptions are not authorized and the number of total computers in the high-security zone is sufficient to meet KMS activation thresholds, add a local KMS host to the high-security zone. Then, activate the KMS host in the high-security zone by telephone.

Figure 11-2 shows an environment with a corporate security policy that does not allow traffic between computers in the high-security zone and the core network. Because the high-security zone has enough computers to meet the KMS activation threshold, the high-security zone has its own local KMS host. The KMS host itself is activated by telephone.

If KMS is not appropriate because there are only a few computers in the high-security zone, MAK Independent activation is recommended. Each computer can be activated independently with Microsoft by telephone.

MAK Proxy activation using VAMT is also possible in this scenario. VAMT can discover client computers by using AD DS, computer name, IP address, or membership in a workgroup. VAMT uses WMI to install MAK product keys and CIDs and to retrieve status on MAK clients. Because this traffic is not allowed through the firewall, there must be a local VAMT host in the high-security zone and another VAMT host in another zone that has Internet access.

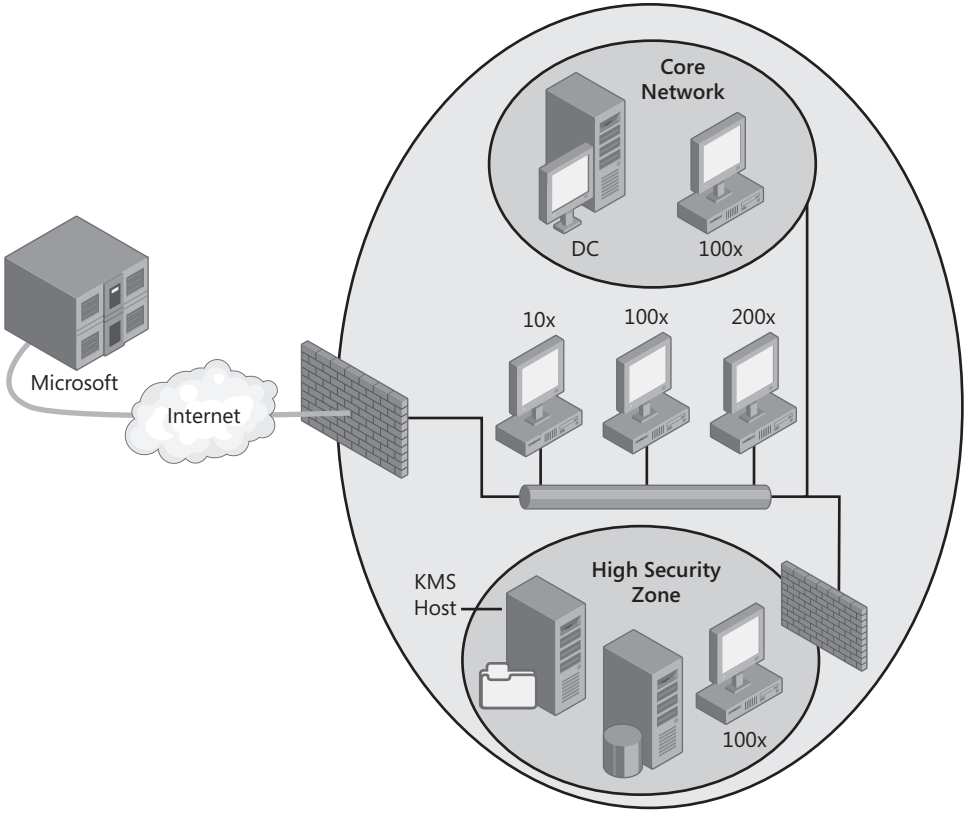


FIGURE 11-2 High-security network scenario

Branch Office Locations

Figure 11-3 shows an enterprise network that supports client computers in three branch offices. Site A uses a local KMS host because it has more than 25 client computers, and it does not have secure TCP/IP connectivity to the core network. Site B uses MAK activation because KMS does not support sites with fewer than 25 KMS client computers, and the site is not connected by a secure link to the core network. Site C uses KMS because it is connected to the core network by a secure connection over a private wide area network (WAN), and activation thresholds are met using core network KMS clients.

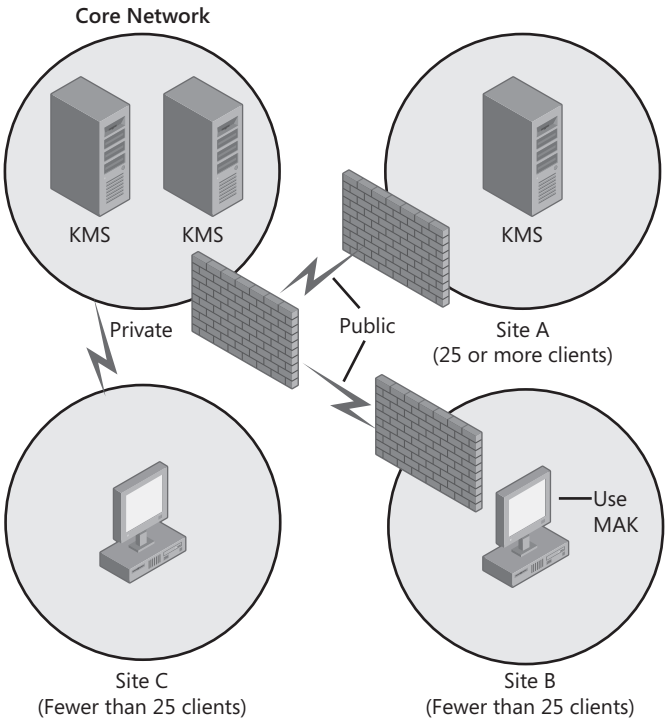


FIGURE 11-3 Branch office scenario

Individual Disconnected Computers

Some users in an organization may be in remote locations or may travel to many locations. This scenario is common for roaming clients, such as the computers of salespeople or other users who are offsite but not at branch locations. This scenario can also apply to remote branch office locations that have no connection or an intermittent connection to the core network.

Disconnected computers can use KMS or MAK, depending on how often the computers connect to the core network. Use KMS activation for computers that connect to the core network—either directly or through a VPN—at least once every 180 days and when the core network is using KMS activation. Use MAK Independent activation—by telephone or the Internet—for computers that rarely or never connect to the core network. Figure 11-4 shows disconnected clients using MAK Independent activation through the Internet and also through the telephone.

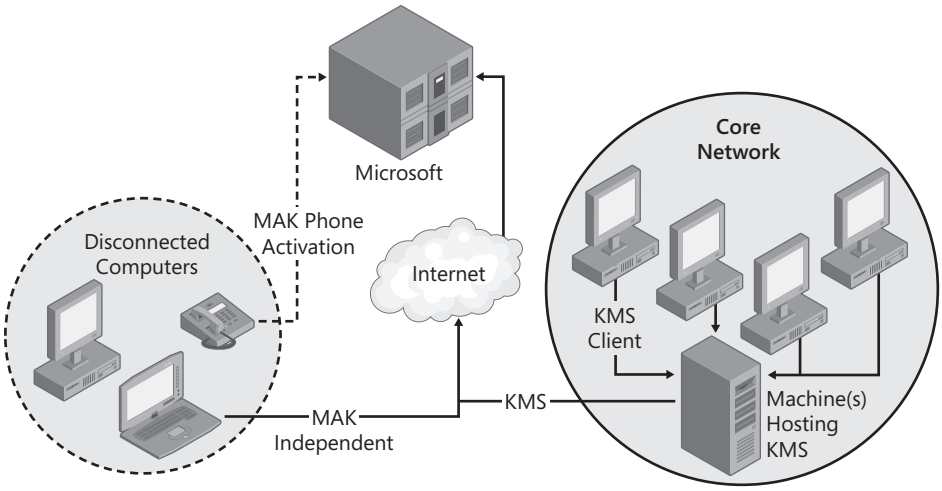


FIGURE 11-4 Disconnected computer scenario

Test/Development Labs

Lab environments usually have large numbers of VMs, and computers in labs are reconfigured frequently. First, determine whether the computers in test and development labs need activation. The initial 30-day grace period of a computer running Windows 7 or Windows Server 2008 R2 can be reset three times without activating it. Therefore, if you are rebuilding lab computers within 120 days, these computers need not be activated.

If lab computers do require activation, use KMS or MAK activation. Use KMS activation if the computers have connectivity to a core network that is using KMS. If the number of computers in the lab meets the KMS activation threshold, deploy a local KMS host.

In labs that have a high turnover of computers as well as a small number of KMS clients, it is important to monitor the KMS activation count to maintain a sufficient number of cached CMIDs on the KMS host. A KMS host caches activation requests from computers for 30 days. (See the section titled "Minimum Computer Requirements" earlier in this chapter for more information about how CMIDs affect activations.) If the lab environment needs activation but does not qualify for KMS activation, use MAK activation. MAK clients are activated by telephone or over the Internet, whichever is available to the lab.

MAK Proxy activation with VAMT can also be used in this scenario. Install VAMT in the isolated lab network and also in a network that has access to the Internet. In the isolated lab, VAMT performs discovery, obtains status, installs a MAK product key, and obtains the IID of each computer in the lab. This information can then be exported from VAMT, saved to removable media, and then the file can be imported to a computer running VAMT that has access to the Internet. VAMT sends the IIDs to Microsoft and obtains the corresponding CIDs needed to complete activation. After exporting this data to removable media, take it to the isolated lab to import the CIDs so that VAMT can complete the activations.

NOTE **In High Security mode, VAMT removes all personally identifiable information (PII) from the file that it exports. This file is a readable Extensible Markup Language (XML) file that can be reviewed in any XML or text editor.**

What If Systems Are Not Activated?

Activation is designed to provide a transparent activation experience for users. If activation does not occur immediately after the operating system is installed, Windows 7 and Windows Server 2008 R2 still provide the full functionality of the operating system for a limited amount of time (a grace period). The length of the grace period is 30 days for Windows 7 and Windows Server 2008 R2. After the grace period expires, both operating systems remind the user through notifications to activate the computer.

Grace Period

During the initial grace period, there are periodic notifications that the computer requires activation. Computers in this grace period have a set period of time to activate the operating system. Once per day, during the logon process, a notification bubble reminds the user to activate the operating system. This behavior continues until there are three days left in the grace period. For the first two of the final three days of the grace period, the notification bubble appears every four hours. During the final day of the grace period, the notification bubble appears every hour on the hour.

Grace Period Expiration

After the initial grace period expires or activation fails, Windows 7 continues to notify users that the operating system requires activation. Until the operating system is activated, reminders that the computer must be activated appear in several places throughout the product:

- Notification dialog boxes appear during logon after users enter their credentials.
- Notifications appear at the bottom of the screen above the notification area.
- A persistent desktop notification will be shown on a black desktop background.
- A reminder might appear when users open certain Windows applications.

Product Keys

Volume Activation does not change how Volume Licensing customers obtain their product keys. They can obtain MAK and KMS keys at the Volume Licensing Service Center (VLSC) Web page at *http://go.microsoft.com/fwlink/?LinkId=107544* or by calling an Activation Call Center. Service Provider License Agreement (SPLA) partners can obtain keys only by calling

an Activation Call Center. Customers in the United States can call 888-352-7140. International customers should contact their local Support Center. For the telephone numbers of Activation Call Centers worldwide, go to *http://go.microsoft.com/fwlink/?LinkId=107418*. When calling a Support Center, customers must have the Volume License agreement.

Volume Licensing customers can log on to the VLSC Web page at any time to view their KMS key information. The VLSC Web site also contains information on how to request and use MAKs. For more information about MAK and KMS keys, including information about increasing the number of allowed activations, go to the Existing Customers page at *http://go.microsoft.com/fwlink/?LinkId=74008*.

Summary

Volume Activation helps IT professionals automate and manage the product activation process on computers running Windows 7 editions that are licensed under a Volume Licensing program or other programs that provide Volume License editions of Windows. Two options are available for Volume Activation: KMS and MAK. KMS activation provides a solution that is easy to deploy and manage, requiring little interaction. Environments that don't meet the minimum requirements for KMS can use MAK activation to activate systems with the Microsoft-hosted activation service.

Additional Resources

These resources contain additional information and tools related to this chapter.

Related Information

- Genuine Microsoft Software at *http://go.microsoft.com/fwlink/?LinkId=151993*.
- Genuine Microsoft Software validation page at *http://go.microsoft.com/fwlink/?LinkId=64187*.
- "Key Management Service 1.1 (x64) for Windows Server 2003 SP1 and Later" at *http://go.microsoft.com/fwlink/?LinkId=83041*.
- "Microsoft Activation Centers Worldwide Telephone Numbers" at *http://go.microsoft.com/fwlink/?LinkId=107418*.
- Microsoft Volume Licensing at *http://go.microsoft.com/fwlink/?LinkId=73076*.
- Microsoft Volume Licensing Service Center at *http://go.microsoft.com/fwlink/?LinkId=107544*.
- "Product Activation and Key Information" at *http://go.microsoft.com/fwlink/?LinkId=74008*.
- "System Center Pack Catalog" at *http://go.microsoft.com/fwlink/?LinkID=110332*.
- "Volume Activation 2.0 Technical Guidance" at *http://go.microsoft.com/fwlink/?LinkID=75674*.
- *Volume Activation Deployment Guide* at *http://go.microsoft.com/fwlink/?LinkId=150083*.

- Volume Activation on TechNet at *http://technet.microsoft.com/en-us/windows/dd197314.aspx*.
- *Volume Activation Operations Guide* at *http://go.microsoft.com/fwlink/?LinkId=150084*.
- *Volume Activation Planning Guide* at *http://go.microsoft.com/fwlink/?LinkID=149823*.
- *Volume Activation Technical Reference Guide* at *http://go.microsoft.com/fwlink/?LinkId=152550*.
- "Windows Vista Privacy Notice Highlights" at *http://go.microsoft.com/fwlink/?LinkID=52526*.
- "Windows Automated Installation Kit (Windows AIK) for Windows 7 RC" at *http://go.microsoft.com/fwlink/?LinkId=136976*.

On the Companion Media

- *Volume Activation Planning Guide*
- *Volume Activation Deployment Guide*
- *Volume Activation Operations Guide*
- *Volume Activation Technical Reference Guide*

CHAPTER 12

Deploying with Microsoft Deployment Toolkit

The Windows 7 operating system and the Windows Automated Installation Kit (Windows AIK) include the low-level tools necessary to deploy the operating system. However, they don't provide a framework for managing and automating high-volume Windows 7 deployments or business logic for managing complex projects. Microsoft Deployment Toolkit 2010 (MDT 2010) provides this framework and business logic, making it Microsoft's primary tool for deploying Windows 7.

This chapter describes how to use MDT 2010 to deploy Windows 7. It assumes that you've already created a deployment share in a lab and populated it with applications, device drivers, and packages. It also assumes that you've already designed and built custom Windows 7 disk images, as described in Chapter 6, "Developing Disk Images." This chapter helps you configure and customize MDT 2010 for Lite Touch Installation (LTI). For more information about Zero Touch Installation (ZTI) by using MDT 2010 with Microsoft System Center Configuration Manager 2007, see the MDT 2010 documentation.

Introducing MDT 2010

The following sections introduce key concepts for using MDT 2010 to deploy Windows 7. Specifically, the section titled "Deployment Scenarios" describes the scenarios that MDT 2010 supports. For LTI, MDT 2010 relies entirely on MDT 2010, the Windows AIK, and potentially Windows Deployment Services.

Deployment Scenarios

The following list describes the scenarios supported by MDT 2010:

- **New Computer** A new installation of Windows is deployed to a new computer. This scenario assumes that there is no user data or profile to preserve.
- **Upgrade Computer** The current Windows operating system on the target computer is upgraded to the target operating system. The existing user state data and applications are retained (as supported by the target operating system).
- **Refresh Computer** A computer currently running a supported Windows operating system is refreshed. This scenario includes computers that must be reimaged for image standardization or to address a problem. This scenario assumes that you're preserving the existing user state data on the computer. Applications are not preserved in this scenario.
- **Replace Computer** A computer currently running a supported Windows operating system is replaced with another computer. The existing user state migration data is saved from the original computer. Then, a new installation of Windows is deployed to a new computer. Finally, the user state data is restored to the new computer.

Based on your existing environment, you can select any combination of these scenarios in the deployment. For example, if you are upgrading only existing computers, only the Refresh Computer scenario or the Upgrade Computer scenario is necessary. If you're deploying new computers for some users and upgrading the remaining computers, use the Upgrade Computer, Replace Computer, and Refresh Computer scenarios as appropriate.

Resource Access

Before starting the deployment, create additional shared folders in which to store the user state migration data and the deployment logs. You can create these shared folders on any server that is accessible to destination computers. Refer to your deployment plan to guide you on server placement. The following list describes the shared folders you should create:

- **MigData** Stores the user state migration data during the deployment process
- **Logs** Stores the deployment logs during the deployment process

> ***NOTE*** **MigData and Logs are recommended shared folder names. You can use any name for these shared folders; however, the remainder of this chapter refers to these shared folders by these names.**

During deployment to destination computers, the MDT 2010 deployment scripts connect to the deployment shares and shared folders. Create accounts for use by these scripts when accessing these resources.

After creating the additional shared folders, configure the appropriate shared folder permissions. Ensure that unauthorized users are unable to access user state migration information and the deployment logs. Only the destination computer creating the user state migration information and the deployment logs should have access to these folders.

For each shared folder, disable inheritance and remove existing permissions. Then give the domain Computers group the Create Folder/Append Data permission for each folder only, and do the same for the domain Users group. Also, add the Creator Owner group to each shared folder, giving it the Full Control permission for subfolders and files only. Also, give each group that will have administrator access to migration data and log files the same permissions.

The permissions that you set in these steps allow a target computer to connect to the appropriate share and create a new folder in which to store user state information or logs. The folder permissions prevent other users or computers from accessing the data stored in the folder.

Using LTI with MDT 2010

Prior to deploying Windows 7 by using LTI with MDT 2010, be sure to perform the following steps, as described in Chapter 6:

- Create a deployment share, possibly in a lab environment, and add the appropriate resources to it. To add applications to the deployment share, see Chapter 8, "Deploying Applications."
- In the deployment share, create and customize task sequences that install Windows 7 as required.
- Build any custom Windows 7 images required for deployment.
- Test your deployment share and custom disk images in the lab.

Chapter 6 describes how to fully stock the deployment share with applications, device drivers, packages, and operating system source files. It also describes how to create task sequences and build custom Windows 7 disk images. In this chapter, you learn how to replicate your deployment share onto the production network and how to perform an LTI deployment by using it.

Replicating a Deployment Share

For LTI, you need to replicate the deployment share in the production environment or copy it to removable media. This process enables you to develop in a controlled environment and then easily move the deployment share into the production environment when you're ready to deploy Windows 7. MDT 2010 provides both capabilities.

When you replicate a deployment share, you can replicate everything or you can choose which folders in the deployment share to replicate. You choose folders to replicate by creat-

ing selection profiles. A selection profile simply selects folders across applications, operating systems, out-of-box drivers, packages, and task sequences. You create a selection profile in advance, and then you choose that selection profile when you set up replication.

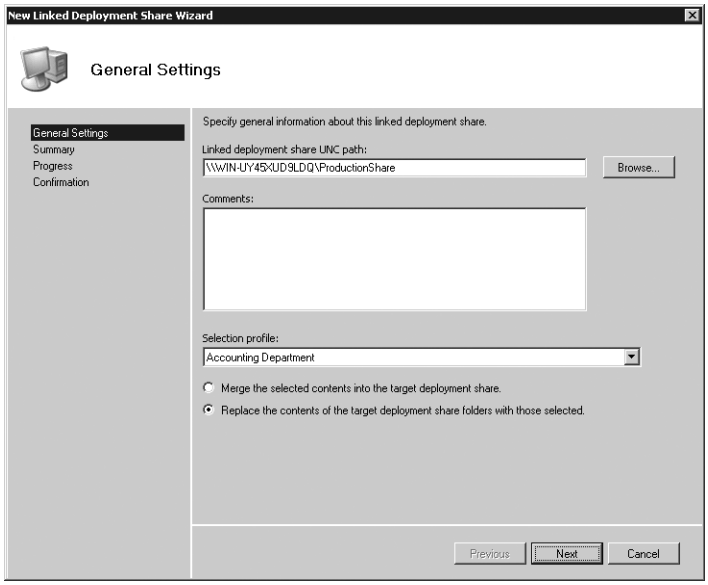
During replication, Deployment Workbench updates the boot media. The updated boot media contains an updated Bootstrap.ini file that is configured to connect to the replicated deployment share. In other words, each deployment share has its own boot media associated with it, and that boot media is configured to connect a specific share.

To create a selection profile, perform the following steps:

1. In the Deployment Workbench console tree under the Advanced Configuration folder of your deployment share, right-click Selection Profiles and click New Selection Profile.
2. In the Selection Profile Name box, type a descriptive name for the selection profile and then click Next. For example, a selection profile that selects files for deployment in a particular department might use the department name for its title.
3. On the Folder page, select the folders that you want to include in the selection profile and then click Next.
4. On the Summary page, review the details and click Next.
5. Click Finish to close the New Selection Profile Wizard.

To link deployment shares for replication, perform the following steps:

1. In the production environment, create a share in which to replicate the deployment share. Make sure that your account has full control of the product sharing.
2. In the Deployment Workbench console tree under the Advanced Configuration folder of your deployment share, right-click Linked Deployment Shares and then click New Linked Deployment Share.
3. On the General Settings page, shown on the following page, do the following and then click Next:
    a. In the Linked Deployment Share UNC Path box, type the Universal Naming Convention (UNC) path of the deployment share in the production environment.
    b. From the Selection Profile list, click the profile that contains the folders that you want to replicate to the production environment.
    c. Click the Merge The Selected Contents Into The Target Deployment Share option to merge this deployment share with the production share; alternatively, click the Replace The Contents Of The Target Deployment Share Folder With Those Selected option to replace the contents of the production share with this share.



4. On the Summary page, review the details and then click Next.
5. On the Confirmation page, click Finish to close the New Linked Deployment Share Wizard.

To replicate the lab deployment share to production, perform the following steps:

1. In the Deployment Workbench console tree under the Advanced Configuration folder of your deployment share, click Linked Deployment Shares.
2. In the Details pane, right-click the replication partnership you created previously and then click Replicate Content.
3. On the Confirmation page, click Finish to close the Replicate To Linked Deployment Share dialog box.

To link removable media to the deployment share, perform the following steps:

1. In the Deployment Workbench console tree under the Advanced Configuration folder of your deployment share, right-click Media and click New Media.
2. On the General Settings page, do the following and then click Next:
   a. In the Media Path box, type the path of the removable media to which you want to copy the deployment share.
   b. From the Selection Profile list, click the profile that contains the folders you want to replicate to the production environment.
3. On the Summary page, review the details and click Next.
4. On the Confirmation page, click Finish to close the New Media Wizard.

To replicate the lab deployment share to removable media, perform the following steps:

1. In the Deployment Workbench console tree under the Advanced Configuration folder of your deployment share, click Media.
2. In the details pane, right-click the media link you created previously and then click Update Media Content.
3. On the Confirmation page, click Finish to close the Updated Media Content dialog box.

Preparing Windows Deployment Services

In the deployment process, Windows Deployment Services servers are responsible for starting Windows Preinstallation Environment (Windows PE) on destination computers to prepare the computers for image installation. After you install and initially configure Windows Deployment Services, ensure that Windows PE images created by updating deployment shares in Deployment Workbench have the appropriate flat-file image structures and add them to the Windows Deployment Services server.

Windows Deployment Services is responsible for initiating the deployment process for Pre-Boot Execution Environment (PXE) boot-enabled destination computers. For more information about setting up and configuring the Windows Deployment Services server, see Chapter 10, "Configuring Windows Deployment Services."

Configuring Resources

In addition to the shared folders described in the section titled "Resource Access" earlier in this chapter, the MDT 2010 scripts may require access to other resources, including application or database servers, such as Microsoft SQL Server 2008. The resources that the installation requires access to depend on the applications you've added to the distribution and the customizations you've made to MDT 2010 and the task sequence.

For LTI, you need to grant access to the deployment share to the credentials specified in one the following ways:

- UserID, UserPassword, and UserDomain properties in the CustomSettings.ini file. MDT 2010 uses these credentials to connect to the deployment share and other network resources. Make sure the credentials used in these properties have Read and Execute permissions on the deployment share. By providing these credentials in CustomSettings.ini, you can fully automate the LTI installation process.
- If you don't provide the credentials in CustomSettings.ini, you provide the credentials necessary to connect to the deployment share when you start the Windows Deployment Wizard on the destination computer. Make sure that the credentials used to start the Windows Deployment Wizard have at least Read and Execute permissions on the deployment share.

Make sure that the credentials used for LTI (defined in CustomSettings.ini or used to start the Windows Deployment Wizard) have Read and Execute permissions to access the following resources:

- **Deployment share** Configure access to the deployment share created in Deployment Workbench.
- **Any resources on application or database servers** Configure access to applications or databases that are accessed through the SQLServer, SQLShare, and Database properties.

NOTE **Other connections to the same servers, such as Named Pipes and Remote Procedure Call (RPC), use the same credentials listed here. Use the ZTIConnect.wsf script to establish these connections. For more information about the ZTIConnect.wsf script, see the MDT 2010 documentation.**

Configuring CustomSettings.ini

CustomSettings.ini is the primary customization file for MDT 2010. The customizations you perform are specific to your organization. The names of the servers, default gateways for each subnet, media access control (MAC) addresses, and other details are unique to your organization, of course. The customization that you perform configures the deployment processes to run properly in your network environment. The examples in this section are provided as guides to help you in your customization. For more information on other configuration scenarios, see the MDT 2010 documentation.

The following listing shows a customized version of the CustomSettings.ini file after completing the New Deployment Share Wizard in Deployment Workbench. The initial contents of CustomSettings.ini depend on the answers given to the New Deployment Share Wizard, of course. The section titled "Customizing CustomSettings.ini" later in this chapter describes in more detail how to customize these settings for different computers.

CustomSettings.ini Modified by Deployment Workbench

```
[Settings]
Priority=Default
Properties=MyCustomProperty

[Default]
OSInstall=Y
SkipAppsOnUpgrade=YES
SkipCapture=NO
SkipAdminPassword=YES
SkipProductKey=YES
```

The CustomSettings.ini file in the listing contains the property values for all the target computers to be deployed using this version of the file. This version of the file contains no values that are unique to a specific target computer, because all of the settings are defined in the [Default] section. In this case, the target computer–specific configuration values are provided manually during the installation process by using the Windows Deployment Wizard. Table 12-1 explains the properties and corresponding values used in the listing.

NOTE **In MDT 2010, the document *Microsoft Deployment Toolkit Reference* defines these and dozens of other settings that you can define in CustomSettings.ini. Of all the guides in MDT 2010, the *Microsoft Deployment Toolkit Reference* is the most useful, particularly for IT professionals already familiar with the MDT 2010 basic concepts.**

TABLE 12-1 Explanation of CustomSettings.ini Properties for LTI

LINE IN CUSTOMSETTINGS.INI	PURPOSE
[Settings]	Indicates the start of the *[Settings]* section.
Priority=Default	Establishes the sequence in which the process parses subsections to locate values for the variables. In this example, the *[Default]* section is the only subsection that is parsed for variables.
Properties=MyCustomProperty	Indicates any additional properties to locate. The properties listed here are in addition to the properties listed in ZTIGather.xml. ZTIGather.wsf parses ZTIGather.xml to obtain a list of the properties. The property names defined here are added to them.
[Default]	Indicates the start of the *[Default]* section. The settings defined in this section apply to all computers.
OSInstall=Y	Indicates that the computer is supposed to perform an operating system deployment.
SkipAppsOnUpgrade=YES	Indicates whether the Windows Deployment Wizard prompts the user to install applications during an upgrade. If the property is set to *YES*, the wizard page is not displayed.
SkipCapture=NO	Indicates whether the Windows Deployment Wizard prompts to capture an image. If the property is set to *YES*, the wizard page is not displayed.
SkipAdminPassword=YES	Indicates whether the Windows Deployment Wizard prompts to set the local Administrator password. If the property is set to *YES*, the wizard page is skipped and not displayed.
SkipProductKey=YES	Indicates whether the Windows Deployment Wizard prompts for a product key. If the property is set to *YES*, the wizard page is skipped and not displayed.

Automating the LTI Process

You can use LTI to automate much of the deployment process. ZTI provides full deployment automation using the MDT 2010 scripts, System Center Configuration Manager 2007, and Windows Deployment Services. However, LTI is designed to work with fewer infrastructure requirements.

You can reduce (or eliminate) the wizard pages that are displayed in the Windows Deployment Wizard during the LTI deployment process. You can also skip the entire Windows Deployment Wizard by specifying the SkipWizard property in CustomSettings.ini. To skip individual wizard pages, use the following properties (see the *Microsoft Deployment Toolkit Reference* in MDT 2010 for a description of each property):

- SkipAdminPassword
- SkipApplications
- SkipAppsOnUpgrade
- SkipBDDWelcome
- SkipBitLocker
- SkipBitLockerDetails
- SkipTaskSequence
- SkipCapture
- SkipComputerBackup
- SkipComputerName
- SkipDeploymentType
- SkipDomainMembership
- SkipFinalSummary
- SkipLocaleSelection
- SkipPackageDisplay
- SkipProductKey
- SkipSummary
- SkipTimeZone
- SkipUserData

NOTE **Automating LTI by using CustomSettings.ini alone is not realistic. Defining custom settings for each computer by using CustomSettings.ini is difficult. The ideal tool to use for fully automating LTI is the MDT 2010 database, which enables you to easily associate settings with individual computers and define settings that apply to groups of computers. For more information about using the MDT 2010 database, see the section titled "Using the MDT 2010 Database" later in the chapter.**

For each wizard page that you skip, provide the values for the corresponding properties that normally are collected through the wizard page in the CustomSettings.ini and BootStrap.ini files (or by using the MDT 2010 database). For more information on the properties that you need to configure in the CustomSettings.ini and BootStrap.ini files, see the MDT 2010 documentation.

The following listing illustrates a CustomSettings.ini file used for a Refresh Computer scenario to skip all Windows Deployment Wizard pages. In this sample, the properties to provide when skipping the wizard page are immediately beneath the property that skips the wizard page.

CustomSettings.ini File for a Refresh Computer Scenario

```
[Settings]
Priority=Default
Properties=MyCustomProperty

[Default]
OSInstall=Y
ScanStateArgs=/v:5 /o /c
LoadStateArgs=/v:5 /c /lac /lae
SkipAppsOnUpgrade=Yes
SkipCapture=Yes
SkipAdminPassword=YES
SkipProductKey=YES

SkipDeploymentType=Yes
DeploymentType=REFRESH

SkipDomainMembership=Yes
JoinDomain=Americas
DomainAdmin=Administrator
DomainAdminDomain=Americas
DomainAdminPassword=

SkipUserData=yes
UserDataLocation=AUTO
UDShare=\\nyc-am-dep-01\Dellimage\OSDUsmt
UDDir=%ComputerName%

SkipComputerBackup=yes
ComputerBackuplocation=AUTO
BackupShare=\\nyc-am-dep-01\Dellimage\OSDBackup
BackupDir=%ComputerName%

SkipTaskSequence=Yes
TaskSequenceID=Enterprise

SkipComputerName=Yes
```

```
ComputerName=%ComputerName%

SkipPackageDisplay=Yes
LanguagePacks1={3af4e3ce-8122-41a2-9cf9-892145521660}
LanguagePacks2={84fc70d4-db4b-40dc-a660-d546a50bf226}

SkipLocaleSelection=Yes
UILanguage=en-US
UserLocale=en-CA
KeyboardLocale=0409:00000409

SkipTimeZone=Yes
TimeZoneName=China Standard Time

SkipApplications=Yes
Applications1={a26c6358-8db9-4615-90ff-d4511dc2feff}
Applications2={7e9d10a0-42ef-4a0a-9ee2-90eb2f4e4b98}
UserID=Administrator
UserDomain=Americas
UserPassword=P@ssw0rd

SkipBitLocker=Yes
SkipSummary=Yes
Powerusers1=Americas\JoinRis
```

Performing LTI Deployments

To deploy a computer using LTI, start the destination computer by running LiteTouch.vbs from the deployment share or by using the Windows PE boot image generated by updating the deployment share. Start the Windows PE boot image in any of three ways:

- Burn the .iso images to a DVD. This process is slow and tedious. These ISO image files reside in the \Boot folder of the deployment share.
- Copy the contents of the Windows PE boot image to a bootable USB Flash drive (UFD). This is far more convenient than DVDs, and most modern computers support booting from UFDs. For more information about creating bootable UFDs, see Chapter 9, "Preparing Windows PE."
- Add the LiteTouchPE_x86.wim or LiteTouchPE_x64.wim image files to the Boot Images item of a Windows Deployment Services server. The .wim image files are in the \Boot folder of the deployment share. For more information about installing and configuring Windows Deployment Services, see Chapter 10.

Before beginning installation, verify that the folders in the following list no longer exist on any drive on the target computer (MDT 2010 creates them on the drive with the most free space):

- **MININT** This folder is preserved through the deployment process and contains deployment state information (such as user state migration information and log files).
- **SMSTaskSequence** This folder contains state information specific to Task Sequencer.

The Windows Deployment Wizard creates and uses these folders (on the drive where the operating system is installed) during the deployment process. If a previous deployment terminates abnormally, these folders may still exist on the target computer, and if you don't remove them manually, the process will continue from the point where the process abnormally terminated instead of starting from the beginning. Be sure to remove these folders, if they exist, before initiating deployment.

To start an LTI deployment using Windows Deployment Wizard, perform the following steps:

1. Start the Windows Deployment Wizard using one of the following methods:
   - Start the wizard manually from an existing Windows installation by connecting to the appropriate deployment share (for example, *servername*\DeploymentShare$\Scripts) and typing **cscript litetouch.vbs**.
   - Start the Lite Touch Windows PE image by using a bootable DVD, bootable UFD, or Windows Deployment Services. Any images created by Deployment Workbench automatically start the Windows Deployment Wizard. See Chapter 10 to learn how to add these boot images to Windows Deployment Services.
2. If prompted by the Welcome To Windows Deployment dialog box, click Run The Deployment Wizard To Install A New Operating System and then click Next.
3. If prompted by the User Credentials dialog box, type the credentials necessary to connect to the deployment share (user name, domain, and password) and then click OK. The Windows Deployment Wizard starts automatically. You must use an account that has Read and Write access to the deployment share.
4. Follow the Windows Deployment Wizard instructions to choose a task sequence, answer prompts not skipped by CustomSettings.ini or the MDT 2010 database, and begin installation. The actual experience is based entirely on the customizations you made to CustomSettings.ini and the MDT 2010 database.

NOTE **Windows 7 can use the new Offline Domain Join feature to join a domain without a connection to it. This process requires Windows 7 and Windows Server 2008 R2. First, you provision the computer account on the domain controller, which creates a metadata file containing the information required to join the domain. Then, you transfer the metadata to the joining computer. The computer performs the domain join without having connectivity to the domain controller. For more information, type djoin.exe /? on a domain controller running Windows Server 2008 R2.**

Customizing MDT 2010

MDT 2010 customization provides the necessary configuration settings for the destination computers. The configuration settings include the values that you would normally provide if you were deploying the operating system manually. You accomplish this customization by using one or more of these options:

- Configure the CustomSettings.ini file.
- Configure the BootStrap.ini file.
- Retrieve information from the MDT 2010 database.

For LTI-based deployments, any configuration settings that you don't specify in the CustomSettings.ini file, the BootStrap.ini file, or the database must be provided when running the Windows Deployment Wizard. This gives you the flexibility to automate the LTI process fully or have the majority of configuration settings provided when running the Windows Deployment Wizard.

MORE INFO **For more information, see the following resources:**

- **For the syntax and structure of the CustomSettings.ini file, see the MDT 2010 documentation.**
- **For the syntax and structure of the BootStrap.ini file, see the MDT 2010 documentation.**

Configuring Multiple Computers

Whenever possible, apply configuration settings to multiple computers. You can define groups of computers and then apply configuration settings to the groups you define. Group-based configuration settings allow you to apply the same settings to a group of client computers. After you apply group-based settings, you can apply computer-specific configuration settings through computer-based settings.

Selecting a Grouping Method

You can use different methods to group client computers. After you determine how you want to group computers, select the appropriate properties.

Using the processing rules in MDT 2010, you can group computers based on any property that might be applied to a group of computers (such as Make, Model, DefaultGateway, and so on). Table 12-2 lists methods of grouping computers, descriptions of the methods, and the properties that you can use to group the computers.

TABLE 12-2 Grouping Methods

GROUPING METHOD	DESCRIPTION	PROPERTIES
Geographically	Group configuration settings based on resources located within a geographic region (such as a shared folder on a computer within a geographic region).	DefaultGateway
Target computer hardware attributes	Group configuration settings based on hardware attributes (such as the make of the computer or processor architecture of the target computer).	Architecture CapableArchitecture Make Model HALName
Target computer software attributes	Group configuration settings based on software attributes (such as the operating system version of the target computer).	OSVersion
Default attributes	Apply configuration settings to all target computers when the properties are not located in other sections.	Default

In most instances, you can nest computer groupings. For example, you can use the DefaultGateway property to designate the IP subnets on which a computer resides within a geographic location. You can define locations by using the user-defined properties in the *[DefaultGateway]* section, as shown in the following listing. When grouping computers by hardware configuration, you can use a variety of methods, and the script searches for the substituted value. For instance, if you specify *Priority=Make*, the script substitutes the value for *Make* that it determines through a Windows Management Instrumentation (WMI) call and looks for the corresponding section, such as *[Dell Computer Corporation]*.

Grouping with [DefaultGateway]

```
[DefaultGateway]
172.16.0.3=NYC
172.16.1.3=NYC
172.16.2.3=NYC
172.16.111.3=DALLAS
172.16.112.3=DALLAS
172.16.116.3=WASHINGTON
172.16.117.3=WASHINGTON

[NYC]
UDShare=\\NYC-AM-FIL-01\MigData
SLShare=\\NYC-AM-FIL-01\Logs
Packages1=NYC00010-Install
Packages2=NYC00011-Install
```

```
Administrator1=WOODGROVEBANK\NYC Help Desk Staff

[DALLAS]
UDShare=\\DAL-AM-FIL-01\MigData
SLShare=\\DAL-AM-FIL-01\Logs
Administrator1=WOODGROVEBANK\DAL Help Desk Staff
```

MORE INFO **You can find the complete source of the CustomSettings.ini file used in these examples in the MDT 2010 documentation.**

Applying the Properties to the Groups

After you identify the ways you want to group configuration settings, determine which properties and corresponding configuration settings you will apply to each group. Properties that you can group are properties that you can apply to multiple computers. Properties that you can apply to groups of computers include:

- BackupDir
- BackupShare
- CaptureGroups
- ComputerBackupLocation
- Packagesx
- SLShare
- UDDir
- UDShare
- UDProfiles

You should not apply properties that are specific to individual computers to groups of computers. These properties include:

- AssetTag
- HostName
- IPAddress
- OSDNewMachineName
- SerialNumber

NOTE **MDT 2010 supports dozens of properties in CustomSettings.ini. The *Microsoft Deployment Toolkit Reference* in MDT 2010 contains a complete reference of all the settings it supports.**

Configuring Individual Computers

For LTI, configuration settings that you apply to a group of computers may be sufficient. You can supply the remainder of the computer-specific settings interactively in the Windows Deployment Wizard.

If you want to automate your LTI-based deployment fully, you need to provide computer-specific configuration settings in addition to the settings that apply to groups of computers. You can use the configuration settings for individual computers to override or augment settings for groups of computers based on the priority. For more information about determining the priority of processing rules, see the MDT 2010 documentation.

Selecting an Identification Method

More than one method is available for identifying individual computers (just as when identifying groups of computers). After you select the method for identifying an individual target computer, you can select the appropriate properties.

The processing rules in MDT 2010 allow you to identify individual computers based on any property that might be applied to only one computer (such as AssetTag, MACAddress, UUID, and so on). Table 12-3 lists the methods of identifying individual computers, descriptions of the methods, and properties that you can use to identify the individual computers.

TABLE 12-3 Identifying Individual Computers

IDENTIFICATION METHOD	DESCRIPTION	PROPERTIES
Target computer hardware attributes	Identify the target computer by using the hardware configuration.	MACAddress
Target computer software attributes	Identify the target computer by using the software or firmware configuration.	Product (in conjunction with Make and Model) UUID
Target computer user-defined attributes	Identify the target computer by using attributes that are assigned to the computer but are not a part of the hardware or software configuration.	AssetTag SerialNumber

Applying the Properties to Individual Computers

After you select the methods for identifying individual computers, determine which properties and corresponding configuration settings you will apply to each destination computer. These configuration settings typically apply to only one computer because the configuration settings are unique to that computer. In instances in which a configuration setting is being applied to several computers, use group-based processing rules.

Properties that are typically applied to individual computers include:

- AssetTag
- HostName
- IPAddress
- OSDNewMachineName
- SerialNumber

If a group-based setting has a higher priority and the configuration setting is found in that group, the same configuration setting for an individual computer is ignored. For more information about deployment processing rule priority, see the MDT 2010 documentation.

Customizing CustomSettings.ini

The CustomSettings.ini file is the primary configuration file for MDT 2010. All configuration settings are specified either directly or indirectly:

- Directly, in the CustomSettings.ini file
- Indirectly, in the MDT 2010 database that is referenced in the CustomSettings.ini file

The CustomSettings.ini file syntax is very similar to many .ini files. The CustomSettings.ini file in the following listing illustrates a CustomSettings.ini file customized for a LTI-based deployment. For further explanation of the CustomSettings.ini file in the listing, see the MDT 2010 documentation.

CustomSettings.ini for LTI

```
[Settings]
Priority=Default, MACAddress
Properties=CustomProperty

[Default]
OSInstall=Y
ScanStateArgs=/v:5 /o /c
LoadStateArgs=/v:5 /c /lac
UserDataLocation=NONE
CustomProperty=TRUE

[00:0F:20:35:DE:AC]
ComputerName=HPD530-1

[00:03:FF:FE:FF:FF]
ComputerName=BVMXP
```

A CustomSettings.ini file includes:

- **Sections** Sections are identified by brackets that surround the section name (for example, *[Settings]*). In the previous listing, the sections include *[Settings]*, *[Default]*, *[00:0F:20:35:DE:AC]*, and *[00:03:FF:FE:FF:FF]*. CustomSettings.ini has the following types of sections:
  - **Required sections** Only the *[Settings]* section is required. All other sections are optional. The MDT 2010 scripts require the *[Settings]* section in CustomSettings.ini to locate the reserved properties (Priority and Properties).
  - **Optional sections** The optional sections in the CustomSettings.ini file are used to assign a group of configuration settings to groups of computers or to individual computers. In the previous listing, the configuration settings in the *[Default]* section are applied to more than one computer, and the configuration settings in the *[00:0F:20:35:DE:AC]* and *[00:03:FF:FE:FF:FF]* sections are applied to the corresponding computers.
- **Properties** Properties are variables that need to have values assigned. Properties are followed by an equals sign (=). The scripts scan the CustomSettings.ini file to locate the properties.
- **Values** Values are the configuration settings assigned to the properties. Values are preceded by an equals sign. The scripts scan the CustomSettings.ini file to locate the values. In the previous listing, the value assigned to the *LoadStateArgs* property is */v:5 /c /lac*.

MORE INFO **For more information on the syntax of the CustomSettings.ini file, see the MDT 2010 documentation**.

Customizing BootStrap.ini

Configure the BootStrap.ini file to specify property settings prior to accessing the CustomSettings.ini file. In other words, the BootStrap.ini file describes how to connect to the deployment share, which contains the CustomSettings.ini file. Configure the BootStrap.ini file to help the MDT 2010 scripts locate the appropriate MDT 2010 deployment share.

The syntax of the BootStrap.ini file is identical to the CustomSettings.ini file. The BootStrap.ini file contains a subset of the properties that are used in the CustomSettings.ini file. The following lists the common properties that are configured in BootStrap.ini:

- DeployRoot
- SkipBDDWelcome
- UserDomain
- UserID
- UserPassword
- KeyboardLocale

MORE INFO **Deployment Workbench creates the BootStrap.ini file when a deployment share is created. After the initial creation, make all further customizations manually. For more information on configuring the BootStrap.ini file syntax, see the MDT 2010 documentation.**

Using the MDT 2010 Database

You can configure the rules for LTI deployment in the MDT 2010 database by using Deployment Workbench. The benefits of using the database include:

- **A more generic version of CustomSettings.ini** Storing the configuration settings in the MDT 2010 database removes most of the detail from CustomSettings.ini. This change helps make the CustomSettings.ini file more generic so that you can use the same file in multiple deployment shares.
- **A centralized repository for all property configuration settings** Centralizing the configuration for all property settings ensures consistency across all deployment shares.

To configure the rules in the configuration database, perform the following steps:

1. Create the database by using Deployment Workbench. The following section, "Creating the MDT 2010 Database," describes this step.
2. Configure the property values in the MDT 2010 database by using the Database item in Deployment Workbench. The section titled "Configuring the MDT 2010 Database" later in this chapter describes this step in more detail.
3. Configure CustomSettings.ini to include the appropriate database queries for returning the property values stored in the MDT 2010 database. The section titled "Configuring the Database Access" later in this chapter describes this step in more detail.

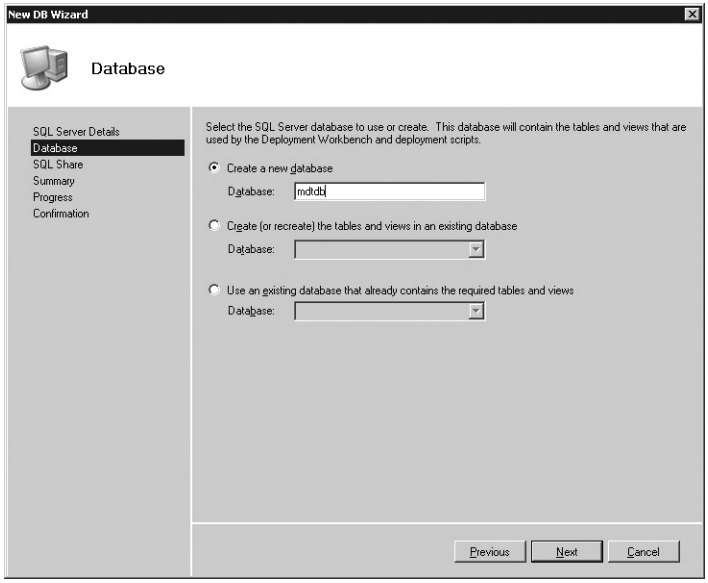
Creating the MDT 2010 Database

Before configuring the database, you must create it in SQL Server. Deployment Workbench creates this database automatically by using the New DB Wizard. Of course, this section assumes that SQL Server is already installed and configured locally or remotely in your environment and that you have permission to create databases.

To create the MDT 2010 database in SQL Server, perform the following steps:

1. In Deployment Workbench, right-click Database and then click New Database. Database is located under Advanced Configuration in the deployment share.
2. On the SQL Server Details page, in the SQL Server Name box, type the name of the server hosting SQL Server and click Next. If you want, provide an instance and port and specify the network library to use for the connection.

3. On the Database page, shown here, choose Create A New Database, type the name of the database in the Database text box, and then click Next. You can also choose to repair or connect to an existing database.



4. If you want, on the SQL Share page, type the name of any share on the server running SQL Server and then click Finish. MDT 2010 uses this share only if necessary to create a secure connection to the computer running SQL Server when using integrated security. Specify this share only if the Windows Deployment Wizard is not able to connect to SQL Server during deployment. The wizard will attempt to connect to this share using the connection credentials specified as described in the section titled "Configuring Resources" earlier in this chapter.

Configuring the MDT 2010 Database

MDT 2010 organizes the property values in the database by the method for applying them to destination computers. An item beneath the Database item in Deployment Workbench represents each method, as shown in Figure 12-1 and as listed in Table 12-4.

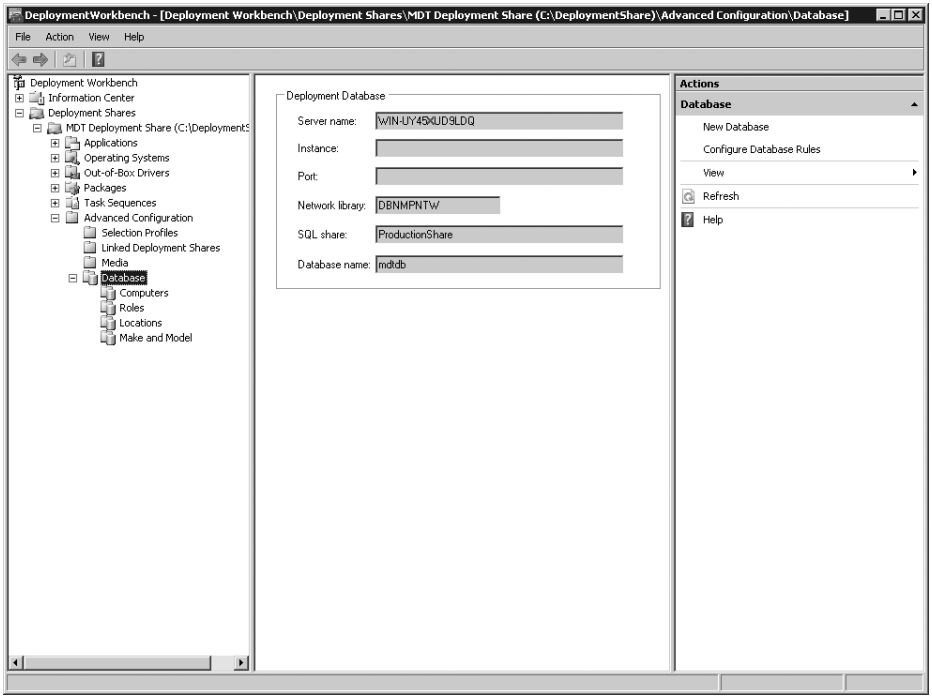


FIGURE 12-1 Database organization in Deployment Workbench

TABLE 12-4 Database Items in Deployment Workbench

NODE	ITEMS DEFINED BY THIS NODE
Computers	Specific target computers based on the AssetTag, UUID, SerialNumber, and MACAddress properties. You can associate property settings, applications, packages, roles, and administrative-level accounts with a computer. For more information on configuring this node, see the MDT 2010 documentation.
Roles	A group of computers based on the tasks performed by the users of the target computers (by using the Role property). You can associate property settings, applications, packages, and administrative-level accounts with a role. For more information on configuring this node, see the MDT 2010 documentation.
Locations	A group of computers using the DefaultGateway property of the target computers to identify a geographic location. You can associate property settings, applications, packages, roles, and administrative-level accounts with a location. For more information on configuring this node, see the MDT 2010 documentation.

NODE	ITEMS DEFINED BY THIS NODE
Make And Model	A group of computers using the Make And Model properties of the target computers. You can associate property settings, applications, packages, roles, and administrative-level accounts with target computers that are of the same make and model. For more information on configuring this node, see the MDT 2010 documentation.

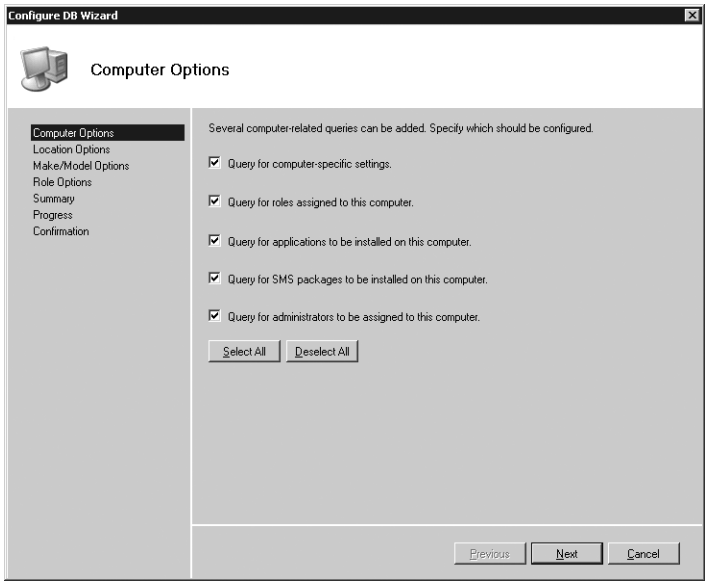
NOTE **Create the items in the Roles node before you create the other items beneath other nodes (Computers, Locations, and Make And Model), because the other nodes can be associated with roles.**

Configuring the Database Access

After you have configured the property values in the MDT 2010 database, you need to configure CustomSettings.ini to perform the appropriate database queries. You can do this easily by using the Configure DB Wizard in Deployment Workbench. Run the Configure DB Wizard for each deployment share defined in Deployment Workbench with which you want to use the database.

To configure CustomSettings.ini for database queries, perform the following steps:

1. In the Deployment Workbench console tree, right-click Database and then click Configure Database Rules. Database is under Advanced Configuration in the deployment share.
2. On the Computer Options page, shown on the following page, choose from the following options and then click Next:
   - **Query For Computer-Specific Settings** Queries the settings configured on the Details tab of the Properties dialog box of the computer item.
   - **Query For Roles Assigned To This Computer** Queries the roles associated with the computer on the Roles tab of the Properties dialog box of the computer item.
   - **Query For Applications To Be Installed On This Computer** Queries the applications to be installed on the computer, as configured on the Applications tab of the Properties dialog box of the computer item.
   - **Query For Administrators To Be Assigned To This Computer** Queries the accounts that will be made members of the local Administrators group on the target computer, as configured on the Administrators tab of the Properties dialog box of the computer item.



3. On the Location Options page, choose from the following options and then click Next:
   - **Query For Location Names Based On Default Gateways** Queries for location names based on the IP addresses of the default gateways configured on the Identity tab of the Properties dialog box of the location item.
   - **Query For Location-Specific Settings** Queries the settings configured on the Details tab of the Properties dialog box of the location item.
   - **Query For Roles Assigned For This Location** Queries the roles associated with the location on the Roles tab of the Properties dialog box of the location item.
   - **Query For Applications To Be Installed For This Location** Queries the applications to be installed on the target computers within the location configured on the Applications tab of the Properties dialog box of the location item.
   - **Query For Administrators To Be Assigned For This Location** Queries the accounts that will be made members of the local Administrators group on the target computers within the location configured on the Administrators tab of the Properties dialog box of the location item.
4. On the Select Make And Model Query Options page, choose from the following options and then click Next:
   - **Query For Model-Specific Settings** Queries the settings configured on the Details tab of the Properties dialog box of the make and model item.
   - **Query For Roles Assigned To Computers With This Make And Model** Queries the roles associated with the make and model on the Roles tab of the Properties dialog box of the make and model item.

- **Query For Applications To Be Installed On Computers With This Make And Model** Queries the applications to be installed on the target computers with the make and model configured on the Applications tab of the Properties dialog box of the make and model item.
- **Query For Administrators To Be Assigned To Machines With This Make And Model** Queries the accounts that will be made members of the local Administrators group on the target computer with the make and model configured on the Administrators tab of the Properties dialog box of the make and model item.

5. On the Select Role Query Options page, choose from the following options and then click Finish:
   - **Query For Role-Specific Settings** Queries the settings configured on the Details tab of the Properties dialog box of the role item.
   - **Query For Applications To Be Installed For This Role** Queries the applications to be installed on computers that perform this role, as configured on the Applications tab of the Properties dialog box of the role item.
   - **Query For Administrators To Be Assigned For This Role** Queries the accounts that will be made members of the local Administrators group on computers that perform this role, as configured on the Administrators tab of the Properties dialog box of the role item.

MORE INFO **After you complete the Configure DB Wizard, the CustomSettings.ini file is configured to perform the selected queries. For more information, see the MDT 2010 documentation. See the corresponding section for each table and view in the configuration database under "Tables and Views in the MDT DB" in *Microsoft Deployment Toolkit Reference*.**

Summary

This chapter provided step-by-step instructions for configuring MDT 2010 to deploy Windows 7 using LTI. LTI is a simple way to deploy Windows 7 in small and medium-sized businesses. It requires no infrastructure and is very easy to set up and customize.

Additional Resources

These resources contain additional information and tools related to this chapter.

- Chapter 3, "Deployment Platform," includes information about the Windows 7 installation architecture and its main components and technologies. This chapter describes how the various components interact.

- Chapter 4, "Planning Deployment," includes information about installing and preparing MDT 2010 for use. This chapter also describes how to use the MDT 2010 guidance.
- Chapter 6, "Developing Disk Images," explains how to design and develop custom Windows 7 disk images for use with MDT 2010 LTI.
- Chapter 10, "Configuring Windows Deployment Services," explains how to install and configure Windows Deployment Services and how to use it with MDT 2010.
- Chapter 11, "Using Volume Activation," includes more information about Windows 7 product keys and volume activation.
- *Microsoft Deployment Toolkit Reference* in MDT 2010 lists the properties you can configure in a deployment share.
- *Windows Automated Installation Kit User's Guide* includes detailed information about the tools and technologies included in the Windows AIK 2.0. This guide is in the file Waik.chm in the Windows AIK.

FROM

TechNet Magazine

Migrate Already! Why You Should Migrate to Windows 7 If You're Still Waffling

By Don Jones

Maybe your organization migrated to Windows Vista. Maybe you're still running Windows XP. Maybe—like many organizations—you've got a mix of both, and you're wondering if Windows 7 will be worth your time.

The answer is Yes: It is definitely worth your time. In fact, it's time to stop waffling and get back to the days when we looked forward to a new release of Windows and couldn't wait to start planning its deployment. Windows 7 is here, it's ready, and it's offering you and your users a fantastic computing experience. Here are seven reasons to start moving to Windows 7 right away.

7: Plays Better with Hardware

One huge fear in moving to Windows Vista was based on hardware compatibility—both in terms of performance and in terms of driver availability. Windows 7 does not reinvent the wheel here. Instead, it uses the same drivers that Windows Vista did—and because manufacturers have had a number of years to release drivers, almost all modern hardware is supported. Windows 7 is also less demanding on that hardware, so even some older machines that had trouble running Windows Vista may do just fine with Windows 7. In short, the hardware-related arguments that may have kept you on XP and away from Windows Vista don't apply anymore.

6: It's Supported
Don't forget that Windows XP has pretty much moved beyond its support lifecycle. Microsoft has extended long-term support a couple of times, but that operating system's time has come to an end. Microsoft can't continue to maintain Windows XP as a secure, stable, viable platform indefinitely. Windows XP debuted in 2001, and technology has continued its inexorable move forward since that time. It follows that hardware manufacturers are less likely to offer Windows XP drivers, while software developers are focused on taking advantage of new features only available in Windows Vista or Windows 7.

What if you just made the move to Windows Vista? If your purchase included Software Assurance, Windows 7 won't cost you much more, and it's an incremental visual change for users—meaning they won't feel like you've pulled the rug out from under them. Even if you don't have Software Assurance, upgrade pricing can make it worth your while for the better performance and new features that Windows 7 offers.

5: Includes Windows XP
Sure, Windows XP is gradually going away—but what if you still have applications that absolutely depend on it? In those cases, it's "Windows 7 XP Mode" to the rescue. Yes, it's really just an embedded copy of Microsoft Virtual PC with a complete copy of Windows XP included—but that's really amazing. If you need Windows XP you can have it, because Windows 7 comes with it. It's the perfect way to keep those legacy applications running, while helping users transition to Windows 7.

4: Less Annoying
Windows Vista got bad press for its sometimes-overzealous User Account Control (UAC) prompts. Windows 7 continues to use UAC to help protect your system from the abuses of unnecessary administrative privileges, but the new UAC is much less intrusive, much more intelligent, and much more likely to be left on in most organizations. UAC was another "reason" many organizations cited for avoiding Windows Vista, but that argument doesn't hold water with Windows 7. Oh, and for fans of Apple's "I'm a Mac, I'm a PC" commercials—the Macs' privilege-protection system is remarkably similar to the Windows 7 UAC.

3: BranchCache and DirectAccess

More and more companies are dealing with geographically distributed users, both in branch offices and home offices, as well as on the road. Many organizations have spent untold dollars connecting these distributed users to centralized IT resources. BranchCache and DirectAccess, two new features supported in Windows 7 and Windows Server 2008 R2, are designed to make these distributed scenarios easier.

Once set up, DirectAccess provides a fast and easy way for remote users to securely access office IT resources anywhere. It does this without the use of cumbersome Virtual Private Networks (VPNs), which typically require specialized support from Internet Service Providers and access points. Ever try to get a VPN going through the airport Wi-Fi? If you have, you'll love DirectAccess. With it, Windows 7 can establish a connection to the office before the user even logs on, providing users with the same computing experience on the road that they have in the office.

BranchCache is designed to provide faster and easier access to centralized resources from within branch offices. It's literally a cache, allowing users in branches to retrieve information once, then caching that information for future access. It's designed to lower wide area network utilization and speed up access to resources, and it works with both HTTP-based content and Server Message Block traffic (used for file and print sharing).

2: More Control

Windows 7 comes jam-packed with all the latest Group Policy improvements, many of which are accessible once you've moved your domain controllers to Windows Server 2008 R2. You'll enjoy more fine-grained and vastly expanded breadth of control, and new options for customizing Group Policy deployment and application. You can use this new control to increase the security in your environment, improve the usability of your computers and more.

Best of all, Windows 7 includes Windows PowerShell 2.0 by default. If you haven't started using Windows PowerShell, do so now. The best new feature in version 2.0 is the ability to sit at your desk and run commands on multiple remote computers in parallel—a stunningly powerful administrative model

that, frankly, is worth the price of admission to Windows 7 all by itself. Sure, Windows PowerShell 2.0 will be available for Windows Vista and even Windows XP, but it's built right into Windows 7. I like to think of Windows 7 as a giant deployment package for Windows PowerShell 2.0.

1: Earn a 'Win!'

Finally, perhaps the best reason to start moving to Windows 7 is to give your users some love. Everyone loves a shiny new OS (and Aero Glass is nothing if not shiny), and in these days of cutbacks and downturns, a new Windows on the desktop can show users that the company is still looking ahead, moving ahead and planning ahead. Users will appreciate having a new "toy," and that appreciation makes the job of every IT staffer just a bit easier.

Don Jones *is a co-founder of* ConcentratedTech.com*, where he writes regular technical content for IT professionals, and is a columnist for* TechNet Magazine.

8 Common Issues in Windows 7 Migrations

By Don Jones

So you're ready to make the jump to the all-new Windows. While most users have reported fairly fast, straightforward migrations, there are some potential "gotchas" that you should be aware of before you pop that new DVD into your computer. And of course, in a business environment there are always a few post-migration issues that you need to watch out for. Here are some of the most common potential pitfalls, and how to avoid or mitigate them.

8: You Can't Migrate from Windows XP

That's not exactly true, but it's a misconception that's getting good play on the Web. What you can't do is an *in-place upgrade* of Windows 7 on a computer running Windows XP; you can *migrate* from a Windows XP computer to a Windows 7 one, with user data and preferences intact.

Many IT professionals prefer migrations over in-place upgrades, because it results in a cleaner, easier-to-support computer in the end, and it gives you a chance to filter out any bad practices—like Elf Bowling games or an excessive number of screen savers—that may have been followed on the old computer. The Microsoft Deployment Toolkit can make an XP-to-7 migration fast and pretty painless (check out the video demonstration to see for yourself). It supports a *hard-link migration,* which allows an in-place migration from Windows XP to Windows 7. It's like an in-place upgrade, with the "fresh start" of a "from-scratch" install.

7: You Skipped Windows Vista

And now you may wish you hadn't. While Windows 7 has been praised for being a better performer with nicely evolved features, it's still based on Windows Vista. Those who have been using Windows Vista will find that most everything has stayed in the same place, with the same name and the same basic look—so they'll feel comfortable finding their Control Panel, operating the new Start menu and so forth. Windows XP users will face a bigger transition as locations, looks and settings have moved. Help prepare them by grabbing a copy of Camtasia or another screen recorder, and making short, narrated videos like, "How to change your desktop wallpaper" and "How to customize the Task bar."

6: Incompatible Applications

It seems like every environment has some old 16-bit Windows application or MS-DOS application left over from the early 1990s. While Windows XP could be tweaked to run many of these applications, Windows 7 (like Windows Vista) tries to live more in the 21st century and may have problems natively running those applications. The keyword is *natively,* because Windows 7 can run a complete copy of Windows XP running in a virtual machine. Called **Windows XP Mode**, it allows those old applications to continue running on Windows XP, while your users enjoy the broader benefits of Windows 7. Download XP Mode and watch a video of it in action here. You can also use the Application Compatibility Toolkit 3.5 to evaluate and mitigate application-compatibility issues. Anything already running on Windows Vista should have no problems.

5: Incompatible Hardware

After the somewhat shaky launch of Windows Vista, a lot of IT pros are going to be concerned about the availability of device drivers and other essentials for running their existing hardware under Windows 7. They needn't worry. Windows 7 uses the same basic drivers as Windows Vista in most cases—meaning hardware manufacturers have had years to migrate everything. True, some pre-Windows Vista hardware may not have drivers—and that's something you want to figure out *before* you start your upgrade or migration. Check hardware vendor Web sites for drivers and compatibility information, and use the Microsoft Assessment and Planning Toolkit to inventory your computers for incompatible or problematic hardware.

4: Manual Labor

Seriously, *nobody* runs around the office installing copies of Windows from DVDs anymore. That kind of manual labor is not going to make your migration project a success. Instead, read up on the latest generation of Microsoft deployment tools. True, they come with more acronyms than a can of alphabet soup, but if you wade through all that you'll find mature, straightforward tools that can make even mass migrations much easier. The Windows Automated Installation Kit (Windows AIK) comes with the Deployment Image Servicing and Management Tool to help you manage your Windows 7 deployment images over the long term, and includes the User State Migration Tool to help migrate user data and preferences. Windows Deployment Services can dynamically provision device drivers, deploy virtual hard drive images and provide better support for x64-based hardware. Finally, the Microsoft Deployment Toolkit helps you create images and automate OS and application installations as well as data migration and desktop configurations.

3: Missing Applications

Be aware that the Windows gallery applications—Windows Mail, Messenger, Address Book, Photo Gallery and Movie Maker—are not bundled in Windows 7. Note that they are available as a free download, called Windows Live Essentials. If you have users relying on those applications, now is a good time to start finding alternatives. For Personal Information Management, Microsoft Office Outlook is always an option.

2: Internet Explorer 8

Let's be clear: Internet Explorer 8 (IE8) should be hailed as an improvement, not an "issue." However, if you have poorly written intranet applications that depend on IE6-specific features and behavior, some of those may not operate 100 percent correctly with IE8. In general, IE8's compatibility mode will solve those problems, provided your users know when to turn it on (it's usually on by default for intranet addresses). However, IE8 really focuses on bringing IE into compliance with industry standards, so you may still run across an intranet app or two that doesn't behave quite right. Plan to test those intranet applications in advance so that you're not caught off-guard.

1: Volume Activation
This is the one that scares me the most. While Volume Activation is essentially unchanged from Windows Vista, a lot of folks skipped Vista, and hence never built a skill set around Volume Activation. Now's the time to do that, and you won't find it difficult. Many businesses will find themselves using Volume license keys rather than retail or multiple-activation keys—and Volume keys offer better license management, so there's a benefit in using them. But they do require you to have a Volume Activation server up and running before you start Windows 7 installations.

Ready, Set, Deploy!
All in all, Windows 7 presents a fairly unchallenging upgrade—even from Windows XP. You simply need to spend a little bit of time evaluating and planning so that you can avoid any nasty surprises. The Windows 7 Deployment FAQ and Step-by-Step Upgrade and Migration pages provide great information to help you avoid most of the pitfalls I've discussed. The product team blog provides some great insight on how and why Windows 7 came to be the way it is. And the *TechNet Magazine* article, "The 10 Things to do First for Windows 7," provides strategy and more tips designed to smooth migration to Windows7. Read up, plan and start your migration!

Don Jones *is a co-founder of* ConcentratedTech.com*, where he writes regular technical content for IT professionals, and is a columnist for* TechNet Magazine.

9 Things You Should Know About Windows 7

By William Stanek

Between Twitter, e-mail, blogs and elsewhere, I've been inundated with questions about Windows 7. Lots of people are asking me, "What do I need to know about Windows 7?" People are eager for the operating system's release and yet anxious at the same time. What they want to know is this: "So what do I really need to know as an IT professional, and what changes do I really need to look out for?" As someone who's worked in IT for more than 20 years, I can understand both the excitement and the concern.

That said, I can confidently say that Windows 7 is a step ahead, a step toward the future, and what IT professionals really need to know about Windows 7 (that you might not already know) are these nine things:

1. Using the Deployment Image Servicing and Management tool
2 Using the Problem Steps Recorder
3 How BranchCache works
4. How DirectAccess works
5. What the options are for PowerShell 2.0
6. How PowerShell scripts can be used to replace other scripts
7. How Windows Remote Management (WinRM) works with PowerShell
8. How remote administration tools are changing
9. What the options are for virtual hard disks

Everything else is icing on the cake, with two notable exceptions: application-compatibility options and virtualization options using Windows XP mode. I'd add related discussions, but these are things I cover in my article "Can't We All Just Get Along?" Let's skip the razzle dazzle and dig right in to each feature.

Deployment Image Servicing and Management Tool

Being able to work with both live operating systems and offline images makes Deployment Image Servicing and Management (DISM) one of the most valuable new administrative tools available in Windows 7. With offline images, IT professionals can use DISM to mount and unmount Windows 7 images stored in Windows Image (.WIM) files or Virtual Hard Disk (.VHD) files. Once images are mounted, IT professionals can use DISM to:

- Enumerate packages, features and drivers
- Add or remove packages, features and drivers
- Display information about installed MSI applications and installed MSP patches
- Upgrade Windows 7 to a higher edition
- Manage product keys, default languages and locale settings

Although DISM is primarily for working with offline and mounted images, IT professionals can use DISM to service online images of Windows operating systems as well. With live operating systems, use DISM to:

- Display information about installed features, drivers and packages
- Enable or disable Windows features
- Display information about the currently installed edition of Windows
- Display regional and language information and more

DISM ships with Professional, Enterprise and Ultimate editions.

Using the Problem Steps Recorder

Sure, DISM is great for IT professionals, but what about tools that can help solve problems users may be experiencing? Well, Windows 7 includes many built-in troubleshooters that can help users diagnose and resolve problems on their own. For example, users can click the Troubleshooting link in Action Center to display all available troubleshooters and then click the troubleshooter that they want to use.

When the troubleshooter starts, the user follows a series of prompts and by default, any suggested fixes can be applied automatically. If a problem cannot be resolved automatically, IT professionals may want the user to send a trouble-shooting report. To automate this process, Windows 7 includes the Problem Steps Recorder (PSR). This feature can capture step-by-step details related to the exact problem a user is having so that the support staff can more easily diagnose and resolve the issue. The basic approach works like this:

1. The user starts PSR and turns on the recording feature.
2. The user performs the action that isn't working correctly.
3. Screen captures for all the user steps are recorded automatically.
4. The user stops recording and is prompted to saved the recorded information.
5. The user sends the report containing the screen captures to a support technician.

As the report contains a fairly complete record of the problem, the support technician is more likely to be able to diagnose and resolve the problem—all without requiring the technician to access the user's computer for the initial troubleshooting.

BranchCache Power Primer

Windows BranchCache makes life easier and faster for users in branch offices who get documents and other types of files from centralized servers. With BranchCache, desktop computers in a branch office can get files from a local cache rather than having to retrieve files from remote servers. Branch caching works with files that are transferred using Server Message Block (SMB) and Hypertext Transfer Protocol (HTTP), allowing files transferred from internal file servers and intranet Web servers to be cached.

Branch caching can be configured in one of two ways:

- **Distributed cache mode** When you use distributed cache mode, desk-top computers running Windows 7 host distributed file caches for other computers at the branch office. A branch server is not needed because each local computer caches and sends out files.

- **Hosted cache mode** When you use hosted cache mode, a server running Windows Server 2008 R2 and located in the branch office hosts the local file cache. The server caches files and sends them to clients.

BranchCache can dramatically improve response times and dramatically reduce transfer times for documents, Web pages and multimedia content. For clients, you configure BranchCache through Group Policy.

DirectAccess Power Primer

Windows DirectAccess allows users to use their computer at home or away from the office exactly as they do at work. Not only are users seamlessly connected to the corporate network anytime they have Internet access, every user request for corporate resources also is securely directed to the corporate network without requiring users to connect to a Virtual Private Network (VPN).

DirectAccess takes advantage of IPv6 and IPsec to provide a secure network infrastructure. Both computers and users are authenticated before they can access the corporate network. IPsec is used to encrypt communications across the Internet. IT professionals can control which resources users can access, whether only traffic destined for the corporate network goes through the DirectAccess server or all Internet and intranet traffic goes through the DirectAccess server. IT professionals can configure network traffic routing using the Route All Traffic Through The Internal Network policy under Administrative Templates\ Network\Network Connections.

DirectAccess relies on Internet Protocol version 6 (IPv6) for end-to-end connectivity. By default, applications that only use IPv4 cannot be reached by DirectAccess clients. However, IPv6-capable applications can reach IPv4-only resources on an intranet using an IPv6/IPv4 translation device. DirectAccess implementation requires an Intra-Site Automatic Tunnel Addressing Protocol-based IPv6 infrastructure, which IT professionals can configure and set up using the DirectAccess Setup Wizard.

60-Second Tour of PowerShell 2.0, Scripts, Tools and WinRM

Windows PowerShell extends the command line by creating a command shell that has a shell/scripting language loosely based on C# and an object model based on the Microsoft .NET Framework. Both 32-bit and 64-bit environments are available, in addition to a graphical environment called the PowerShell Integrated Scripting Environment (ISE).

IT professionals can use PowerShell to more easily configure user computers. With PowerShell and the Add-Computer cmdlet, adding a computer to a domain becomes a one-step process. Ditto for creating system restore checkpoints before modifying a computer. Simply enter "checkpoint-computer" followed by a description string. Powerful, simple and definitely cool.

But one of the coolest treats is the ability to use PowerShell scripts in Group Policy. This allows you to replace other types of scripts for logon, logoff, startup and shutdown with PowerShell scripts.

If PowerShell 2.0 is a big daddy in Windows 7, Windows Remote Management (WinRM) service is the lesser-known (but equally important) stepchild. To remotely manage computers via PowerShell, you need to ensure WinRM is properly configured. Not only are PowerShell and WinRM used for remote management from a PowerShell prompt, they also are used for remote management with Microsoft Management Consoles (MMCs).

Virtual Hard Disk Power Primer

The .VHD file format allows you to create a virtual hard disk (VHD) that is encapsulated in a single file and can be used to host native file systems and support standard disk operations. In Windows 7, you can use Disk Management and DiskPart to create and manage VHDs. Once you've created a VHD, you can copy it to other computers.

Creating and attaching a VHD makes it appear to be a partition on the computer, almost as if you attached a USB external drive. IT professionals can configure computers running Windows 7 to boot from a VHD as well. To do this, you need a Windows Image (.WIM) file.

If you are using a retail image, this image needs to be system-prepared and generalized before you can boot into it. Because you cannot use BitLocker or the hibernate function on a VHD boot drive, booting from VHD is really meant for stationary computers in highly managed environments. VHD features are only in Windows 7 Enterprise and Ultimate.

There you have it: nine things you should know about Windows 7. Its new tools and features can make your job as an IT professional easier, if you take the time to learn it.

Related Links

- Managing and Servicing Your Windows Image
- Problem Steps Recorder
- Windows 7 Walkthrough: BranchCache
- BranchCache Early Adopter's Guide
- BranchCache Security Guide
- Windows 7 Walkthrough: DirectAccess
- DirectAccess Early Adopter's Guide
- DirectAccess Design Guide
- Demonstration: Windows 7 VHD Boot

William Stanek *(williamstanek.com) is a leading authority for Microsoft Windows and Windows Server technologies and the award-winning author of more than 100 books. He is an expert on, and writes about, Active Directory, Group Policy, Windows, Windows Server, Exchange Server, SQL Server, IIS, PowerShell, and Web technologies. Follow him on Twitter at* twitter.com/WilliamStanek.

Can't We All Just Get Along? Running Windows 7 in Mixed Environments

By William Stanek

As great as Windows 7 is (and I think it's pretty great), it should come as no surprise that many organizations are running or will run Windows 7 in a mixed environment. Because of this, Windows 7 is going to have to play nice with a variety of other environments, drivers, APIs and so on, from predecessors including various versions of Windows XP and Windows Vista to Linux, Unix and even the Mac OS X. With so many disparate operating systems, the issue of interoperability becomes extremely important, and you may be wondering what features Windows 7 has in order to support interoperability. So let's dig in and take a look.

Interoperability with Unix and Related Operating Systems

Like earlier releases of Windows, Client for Network File Systems (NFS) and Subsystem for Unix-based Applications (SUA) remain the primary components provided for interoperability with non-Windows operating systems. They allow both small and large enterprises to integrate their Windows systems with Unix-based systems. Client for NFS enables Windows computers to gain access to files on Unix-based computers. SUA provides a subsystem for compiling and running custom Unix-based applications and scripts on Windows computers. Also available are administration tools for managing Services for NFS on local and remote computers. As with Windows Vista, any or all of these features can be turned on or off using the Windows Features dialog box.

Once you've enabled Client for NFS and the related administrative tools, you can configure a computer to connect to Unix NFS shares that allow anonymous access. If you don't allow anonymous access, you must configure the computer to get Unix identity information from an existing User Name Mapping (UNM) server or configure one if it is not already available. At an elevated command prompt, enter **nfsadmin client** to determine what options Client for NFS is configured to use. Use the mount command, the NET USE command or the Map Network Drive feature to map a drive to a remote NFS share.

In Windows, security identifiers (SIDs) identify objects in the file system and elsewhere. In Unix, user identifiers (UIDs) and group identifiers (GIDs) identify objects in the file system and elsewhere. Whenever you work with Services for Unix, UNM is used for authentication. UNM authenticates incoming access requests and determines the effective UID and GID. To correlate Windows and Unix identities, UNM uses the Windows Security Accounts Manager (SAM) or Active Directory to identify Windows users and Unix password and group files or NIS domains to identify Unix users and groups.

There are two approaches to name mapping: simple and advanced. Simple Name Mapping automatically creates name maps for all users and groups who have the same name in your Windows and Unix environments. You can create simple name maps between Windows and Unix using the MapAdmin command with the AddDomainMap parameter. Before you use this command, copy the Unix password and group files to your computer, merge them and then filter out duplicates and any system accounts. You also can create simple name maps using Unix options in the GUI on the UNM server.

In contrast, you create advanced name maps by manually mapping Windows users and groups with their Unix counterparts. While it sounds complex, Unix options in the GUI on the UNM server make this process fairly easy and straightforward. First, you turn off Simple Name Mapping, and then you use the Advanced Maps options to manually map Windows users and groups to Unix users and groups.

For Windows 7, Microsoft made several enhancements to Client for NFS and SUA. Most of these enhancements are bug fixes that provide a better integration solution with fewer problems. Because Windows 7 is best used with Windows Server 2008 R2, it is important to know how R2 supports Services for NFS. In R2, Services for NFS supports net groups so you can create network-wide named groups of hosts and RPCSEC_GSS for enhanced security with Remote Procedure Calls. Generic Security Service Application Programming Interface (GSS-API) allows Services for NFS to use Kerberos version 5 for authentication and integrity checking.

NFS Authentication can be configured to use Kerberos v5 authentication (KRB5) or Kerberos v5 integrity checking and authentication (KRB5i). It is important to note that if you use NFS versions 2 or 3 and KRB5i, you will be unable to mount shares over the User Datagram Protocol (UDP). To use KRB5i integrity checking, you must configure the NFS client and server to use the TCP protocol. With KRB5, you can configure the client and server to use either TCP or UDP.

As far as Mac OS X goes, like Windows Server 2008, Windows Server 2008 R2 does not include Services for Macintosh. That's not necessarily a bad thing, as Mac OS X is built on Unix and includes an NFS client. You can use the NFS client to connect to NFS shares.

Interoperability with Earlier Windows Versions

Many applications will run natively in Windows 7. For applications designed for earlier releases of Windows, you have the option of configuring compatibility settings to get these older applications to run without problems. For example, 32-bit editions of Windows 7 run 16-bit and MS-DOS applications using a virtual machine that mimics the 386-enhanced mode used by Windows 3.0 and Windows 3.1. Each of these older programs runs as a thread within a single virtual machine but can be configured to run in a separate memory space. For these and other programs, you often can use the Compatibility Wizard to resolve many types of compatibility issues.

If applications don't run or you want to create a sandbox environment prior to transitioning to Windows 7, not to worry. You can use Windows XP Mode to create a Virtual PC environment that runs a full copy of Windows XP, allowing you to run applications designed for Windows XP just as if you were on a computer running Windows XP natively.

Windows XP Mode provides an additional layer of compatibility for Windows 7. This gives you more time to move to Windows 7, and it also saves any retraining that might be required to run new versions of applications in Windows 7. Windows XP Mode is designed for small and large enterprises and requires a running instance of the Professional, Enterprise or Ultimate editions of Windows 7.

Once installed, Windows XP Mode is easy to set up. During setup, a tutorial runs to help users understand how to work with Windows XP Mode. When you run Windows XP Mode the first time, a full desktop will open and you can use it to install applications. Installed applications will automatically appear on the Windows 7 Start Menu. Thereafter, users can start applications in Windows XP Mode simply by clicking the application shortcut on the Start Menu. As an example, if your Web applications have compatibility problems with Internet Explorer (IE) 7 or IE8, you can install IE6 in Windows XP Mode as a workaround. Users can then start IE6 from the Windows 7 Start Menu to seamlessly access IE6 in Windows XP Mode.

Windows XP Mode requires hardware virtualization support in the CPU, such as Intel virtualization or AMD virtualization. Virtualization support must be enabled in firmware. Although Windows XP Mode provides a fully functional Windows XP environment, it is not meant for graphics-intensive applications. As with any virtualized PC environment, you should protect the virtual OS by installing anti-virus and anti-malware programs in Windows XP mode. These anti-virus and anti-malware programs are separate from those running in the native Windows 7 environments.

For more on Windows XP Mode, check out these pages:

- Windows XP Mode FAQ
- Windows XP Mode and Windows Virtual PC Web site
- Download Windows XP Mode

That, in a nutshell, is how you can run Windows 7 in mixed environments. As you've learned about in this article, Client for NFS and SUA allow for interoperability with Unix-based computers; Windows XP Mode provides an additional layer of application compatibility that you can take advantage of whenever your computers are running Professional or higher editions of Windows 7. I hope you find this article to be useful, and you'll look for my new books from Microsoft Press, "Windows PowerShell 2.0 Administrator's Pocket Consultant," "Windows 7 Administrator's Pocket Consultant" and "Windows Server 2008 Administrator's Pocket Consultant, Second Edition" (all published in 2009).

William Stanek *(*williamstanek.com*) is a leading authority for Microsoft Windows and Windows Server technologies and the award-winning author of more than 100 books. He is an expert on and writes about Active Directory, Group Policy, Windows, Windows Server, Exchange Server, SQL Server, IIS, PowerShell and Web technologies. Follow him on Twitter at* twitter.com/WilliamStanek.

Integrating Windows 7 and Windows 2008 R2

By Alan Maddison

With the pending release of Windows 7, Microsoft has created an OS that far surpasses its earlier client operating systems, including Windows Vista and Windows XP, in both features and functionality. This fact alone is a compelling enough reason why organizations should upgrade; based on industry reports, it seems that many companies intend to do just that.

However, enterprises looking to maximize their return on investment are looking not only at the benefits of Windows 7, but also at the benefits to be gained from deploying Windows 7 together with Windows Server 2008 R2. The combination of Microsoft's latest desktop and server OSes can benefit any organization. From an administrator's perspective, this pairing raises some questions: Are there any pitfalls to be concerned about? What are some of the benefits of integrating Windows 7 and Windows Server 2008 R2?

At the simplest level, Windows 7 and Windows Server 2008 R2 integration will be seamless and require little or no effort. However, to take advantage of some key features made possible by the combination of the two, you will need to take some time to understand the requirements and the impact of implementing the technology. Here's a look at some of those key features and what this may mean to the typical systems administrator.

There are a number of key integration points to focus on in order to get the most from a rollout.

Security

Microsoft continues to make inroads in improving PC and server security, and this is particularly true with its latest OS releases. Perhaps the most significant technology from a security perspective (not to mention the management and networking benefits) is DirectAccess.

While DirectAccess will almost certainly prove to be the most complex to integrate, not least of all because it creates the possibility that your existing remote access solutions will be completely displaced, it does have some major benefits. These include the ability to integrate with Network Access Protection (NAP) to check that the remote machine complies with corporate security policies; the ability to apply application updates; and the ability to deploy software under Group Policy even when the user is not logged on.

The DirectAccess client connects automatically using an automatic protocol selection, which allows for seamless connectivity without requiring user intervention. For example, it can use IPsec tunnels if it can connect to the correct ports on the DirectAccess server. Alternatively, it supports a new protocol called IP-HTTPS, which allows tunneling of the IP packets over an SSL connection.

DirectAccess requires a Windows Server 2008 R2 DirectAccess server in your perimeter network and either Windows 7 Enterprise Edition or Ultimate Edition deployed on the client. Depending upon the number of remote users, their location and your adoption policy, you could potentially end up with a significant DirectAccess infrastructure, so be prepared to plan carefully. DirectAccess also relies on public key infrastructure (PKI) certificates for both the user and computer to ensure end-to-end encryption and authentication; make sure you incorporate these requirements. Finally, although DirectAccess is also based on IPv6, it supports translation from IPv4 networks.

BranchCache

BranchCache is another technology that can significantly impact your network. Many of you will have experienced the problems associated with branch offices that have limited or heavily utilized wide area network (WAN) connections. A typical solution has been to buy a third-party WAN accelerator, but this tends to stretch already limited budgets and can't be justified for smaller offices.

The introduction of BranchCache provides an efficient and cost-effective solution that caches data to optimize WAN traffic. BranchCache has two modes of operation: distributed and hosted cache mode. In distributed mode the cache is actually maintained across client (Windows 7) machines. In hosted cache mode an on-site Windows 2008 R2 server is responsible for hosting the cache.

Because distributed caching is limited in that it requires clients to be on the same subnet, you will usually consider hosted cache mode first. On the client side you need either Windows 7 Ultimate Edition or Enterprise Edition, with BranchCache enabled and firewall exceptions created. This can be done using Group Policy. Note that files smaller than 64KB will not be cached.

AppLocker

Also new is AppLocker, the long-awaited replacement for Software Restriction Policies (SRPs). If you have ever used SRPs you know that they were cumbersome and lacked granularity and flexibility. AppLocker constitutes a huge stride forward and provides a robust and flexible method for managing user access to applications. Because AppLocker provides the ability to control a wide variety of file types, including executables, scripts, installers and .DLLs, it's quite an effective security tool.

However, while you will almost certainly want to control AppLocker using Group Policy, if you are not familiar with or have never used SRPs, you will want to test AppLocker thoroughly on a local machine using the Local Security Policy.

AppLocker's default behavior is to block all programs and scripts not explicitly allowed that can create major problems if you make a mistake. An additional testing safeguard to consider is the AppID service, which needs to be running on the Windows 7 machine in order for AppLocker policies to apply. By keeping the service startup type of the AppID set to "manual," you can easily fix mistakes by simply restarting the test machine.

When ready to deploy, use a dedicated Group Policy Object (GPO) for AppLocker and don't combine it with SRPs—the two should never be part of the same GPO. Finally, be aware that if you implement AppLocker and are not currently using SRPs, users will likely struggle to adapt, resulting in a temporary increase in calls to the help desk. For more help with implementing AppLocker, consult the Microsoft AppLocker Step-by-Step Guide.

BitLocker

Continuing in the same vein of improvements in management and security, Microsoft has improved BitLocker with the concept of BitLocker To Go, which allows you to encrypt removable media such as USB drives and thumb drives. This is a significant enhancement that has many positive implications, particularly as it relates to protection of intellectual property and data privacy laws.

This technology is best controlled through Group Policy; you should save the encryption keys within Active Directory to ensure that you always have control of the encrypted data. More information is available at the Microsoft BitLocker Drive Encryption Step-by-Step Guide for Windows 7.

Virtualization

Virtualization is here, and is a key part of Windows 7 and Windows Server 2008 R2. While a lot of attention has been given to the Windows 7 Windows XP Mode (which is a copy of Microsoft's Virtual PC with a Windows XP virtual machine built in), that technology is targeted at small to midsize businesses, not the enterprise.

The real virtualization benefits come from different technologies. Remote Desktop Services (RDS), the next generation of the venerable Terminal Services, was introduced with Windows Server 2008 R2.

RDS provides a much more seamless remote end-user experience, which is important in the growing hosted desktop space. And Windows 7 takes this one step further with the concept of RemoteApp and Desktop Connections. These capabilities provide a dynamic feed for remote applications and can be accessed through the Start menu, just as they would a locally installed application. Windows Server 2008 R2 and Windows 7 also use RDP 7, which further enhances the capabilities of a virtualized application or desktop by supporting Aero Glass, audio input to support VoIP in remote sessions and enhanced bitmap acceleration.

Power Conservation
Finally, in support of the ever growing number of green initiatives within IT, Microsoft has introduced some power-management improvements in Windows 7. All of these enhancements can be deployed and managed centrally, via Group Policy, using Windows Server 2008 R2. While larger organizations could see some large numbers in terms of reducing costs associated with power consumption, don't forget about your users. As should be the case with all changes to a user's desktop or laptop environment, proceed with care—unless you want your help desk swamped with calls.

There's a lot to like about the Windows 7 and Windows Server 2008 R2 operating systems by themselves. Together, their new features provide an opportunity to make fundamental changes in how effectively and efficiently IT is run; and more easily, too—at least until the next upgrade.

For more information on some of the features discussed in this article, check out these links:

DirectAccess
microsoft.com/servers/directaccess.mspx

BranchCache
technet.microsoft.com/en-us/library/ee307962(WS.10).aspx

AppLocker
technet.microsoft.com/en-us/library/dd378941(WS.10).aspx

BitLocker
technet.microsoft.com/en-us/library/dd875547(WS.10).aspx

Alan Maddison *is a 15-year IT veteran who is currently a senior consultant with SBS, a division of Brocade. You can reach him at* amaddison@sbsplanet.com.

The Good Computing Seal of Approval: Windows 7 Certification

By Joshua Hoffman

One of the greatest challenges an IT professional faces is keeping up with the rapid and constant changes in the technological landscape. There's no question that the constant motion is part of what makes life in IT so interesting and exciting—especially to those of us who are gadget-inclined (or gadget-addicted). But it also makes for a difficult balance: how to continue to do all that is necessary for your job as it exists today, while also preparing for what's to come.

One of the more valuable resources available as we face that challenge is the industry-certification process. The benefits are really two-fold. First is the learning process that preparing for a certification offers. After all, the aim of any certification is to arm you with all the information necessary to tackle the subject at hand, and studying for the exams you'll need to take to earn your certification will help you do just that. Second is the professional value that a certification provides. Proof of your familiarity with specific products and technologies can often be what makes the difference between getting that interview you're after, or even getting the job.

So with many already planning deployments of Windows 7, it may be time to bring your Microsoft certifications up-to-date. For more information on getting started with certification, check out the Microsoft Learning FAQ page.

Windows 7 Certification Options

The first certification available for Windows 7 is the Microsoft Certified Technology Specialist (MCTS). Earning an MCTS certification in Windows 7 requires passing one exam: 70-680 – Windows 7, Configuration. This exam covers the following topics:

- Installing, Upgrading, and Migrating to Windows 7
- Deploying Windows 7
- Configuring Hardware and Applications
- Configuring Network Connectivity
- Configuring Access to Resources
- Configuring Mobile Computing
- Monitoring and Maintaining Systems that Run Windows 7
- Configuring Backup and Recovery Options

As you can see, getting that MCTS gains you all the fundamental information needed to begin managing Windows 7 in an enterprise environment. If you want to go further, consider pursuing the Microsoft Certified IT Professional (MCITP) certification.

The MCITP certification for Windows 7 requires both the 70-680 exam (the exam you'd take for your MCTS certification) as well as Exam 70-686: Windows 7 Desktop Administration (available by the end of 2009). Whereas the MCTS certification is intended to provide the fundamentals necessary to configure and deploy Windows 7, the MCITP is intended for advanced IT professionals who are likely to address a broader range of Windows Client issues.

Windows 7 Certification Options

Certification	Required Exam(s)	Available Courses
MCTS: Windows 7, Configuration	Exam 70-680: TS: Windows 7, Configuring	Course 50321A: Windows 7, Configuring Technology Specialist Course
MCITP: Enterprise Desktop Administrator 7	Exam 70-680: TS: Windows 7, Configuring Exam 70-686: PRO: Windows 7, Desktop Administration (available late 2009)	6292A: Installing and Configuring Windows 7 Client Course 6294A: Planning and Managing Windows 7 Desktop Deployments and Environments

How to Prepare

Everyone has his or her own style when it comes to studying for a test. Some might prefer the solo method (you know it: cram the night before the exam). I don't necessarily recommend that particular approach, but if you'd like to go it alone, perhaps the MCTS Self-Paced Training Kit (Exam 70-680): Configuring Windows 7 is best for you. It includes all the material needed to prepare for the exam, along with examples and sample questions that you can push through at your own speed.

Others might prefer in-person classroom training; in that case, the instructor-led course, Windows 7, Configuring Technology Specialist Course (Course 50321A), might be the ideal fit for you. The opportunity to move through the curriculum at a steady pace along with other colleagues allows for asking questions, discussing examples and preparing for the exam as a group, which can be helpful. For more information, options and advice on the exam process, go to the Microsoft Learning Preparing for and Taking an Exam FAQ page.

There are dozens of additional resources in the Microsoft Learning Catalog, including e-Learning courses, free articles and more. The Microsoft Learning site also has a number of special offers on training materials and resources. And of course, resources such as the TechNet forums, *TechNet Magazine* and the experience of your colleagues, can be invaluable as you prepare for the exams.

Once you're ready to go, head over to the exam registration site. Exams are administered by Prometric, an independent testing organization with more than 3,000 locations. The cost is typically $125 per exam.

Stand Out!

Good luck pursuing your Windows 7 certification. Certifications like the MCTS and the MCITP not only help you get up to speed on the latest technology, they're also valuable credentials to help you stand out in the workforce.

One final note before you register: Microsoft often provides special offers and discounts on exam fees. Be sure to check out the special offers section of its Web site. If you're going to take the exam anyway, you might as well save a few bucks!

Joshua Hoffman *is the former editor in chief of* TechNet Magazine. *He is currently a partner at HearSay Marketing, a consultancy dedicated to helping organizations understand their customers and engage their audience through a variety of traditional and social media channels. You can reach him at* joshuah@hearsaymarketing.com.