Realtime
publishers
"Leading the Conversation"

The Definitive Guide™ To

Building a Windows Server 2008 Infrastructure

triCerat

Greg Shields

Introduction to Realtimepublishers

by Don Jones, Series Editor

For several years, now, Realtime has produced dozens and dozens of high-quality books that just happen to be delivered in electronic format—at no cost to you, the reader. We've made this unique publishing model work through the generous support and cooperation of our sponsors, who agree to bear each book's production expenses for the benefit of our readers.

Although we've always offered our publications to you for free, don't think for a moment that quality is anything less than our top priority. My job is to make sure that our books are as good as—and in most cases better than—any printed book that would cost you $40 or more. Our electronic publishing model offers several advantages over printed books: You receive chapters literally as fast as our authors produce them (hence the "realtime" aspect of our model), and we can update chapters to reflect the latest changes in technology.

I want to point out that our books are by no means paid advertisements or white papers. We're an independent publishing company, and an important aspect of my job is to make sure that our authors are free to voice their expertise and opinions without reservation or restriction. We maintain complete editorial control of our publications, and I'm proud that we've produced so many quality books over the past years.

I want to extend an invitation to visit us at http://nexus.realtimepublishers.com, especially if you've received this publication from a friend or colleague. We have a wide variety of additional books on a range of topics, and you're sure to find something that's of interest to you—and it won't cost you a thing. We hope you'll continue to come to Realtime for your educational needs far into the future.

Until then, enjoy.

Don Jones

Copyright Statement

Chapter 1: Introduction & Installation of Windows Server 2008

When you think about Windows Server 2008, what comes to mind?

A new look and feel, more like Vista and less like Windows XP. An improved Server Manager administrative console that aggregates most administrative tasks into a single interface. A focus on componentization, breaking apart what we used to think of as Windows into multiple, installable components.

An improved installation routine, moving all the questions ahead of the actual installation process. A new Initial Configuration Tasks wizard that consolidates many of the initial server personalization activities under a single console. An improved event viewer with better viewing and management of system events and the ability to send event information from machine to machine. A new task scheduler that takes away much of the guesswork in creating scheduled tasks.

An improved Performance Monitor that takes total server reliability into account. A host of improvements to Windows Backup, including capabilities for bare-metal restoration. An enhanced Disk Management subsystem, finally granting administrators the ability to shrink volumes in addition to expanding them. An integrated memory diagnostics tool that checks hardware RAM for physical errors.

A segregation of typical server elements into Roles, Role Services, and Features for better separation of duties. An upgrade to Internet Information Services, adding centralized manageability and a reduced attack surface area. A new File Services Role, formalizing the traditional service of serving files. A set of long-desired improvements to Terminal Services, enhancing virtually every component of remote application support. A set of new and improved network security capabilities, like Network Access Protection and Server and Domain Isolation, that protect a network's vulnerable interior.

A set of enhancements to Active Directory (AD), improving domain controllers, adding the new Read-Only Domain Controller role, and adding greater overall directory services resiliency An added capability for native hardware virtualization right out of the box, as well as the ability to eliminate the graphical user interface (GUI) altogether.

Anything else? All this and more are some of the new and improved features of Microsoft's latest Windows Server operating system (OS). Get ready because Windows Server 2008 is here, and within this guide, we'll talk about how best to use this new server OS to build a Windows infrastructure.

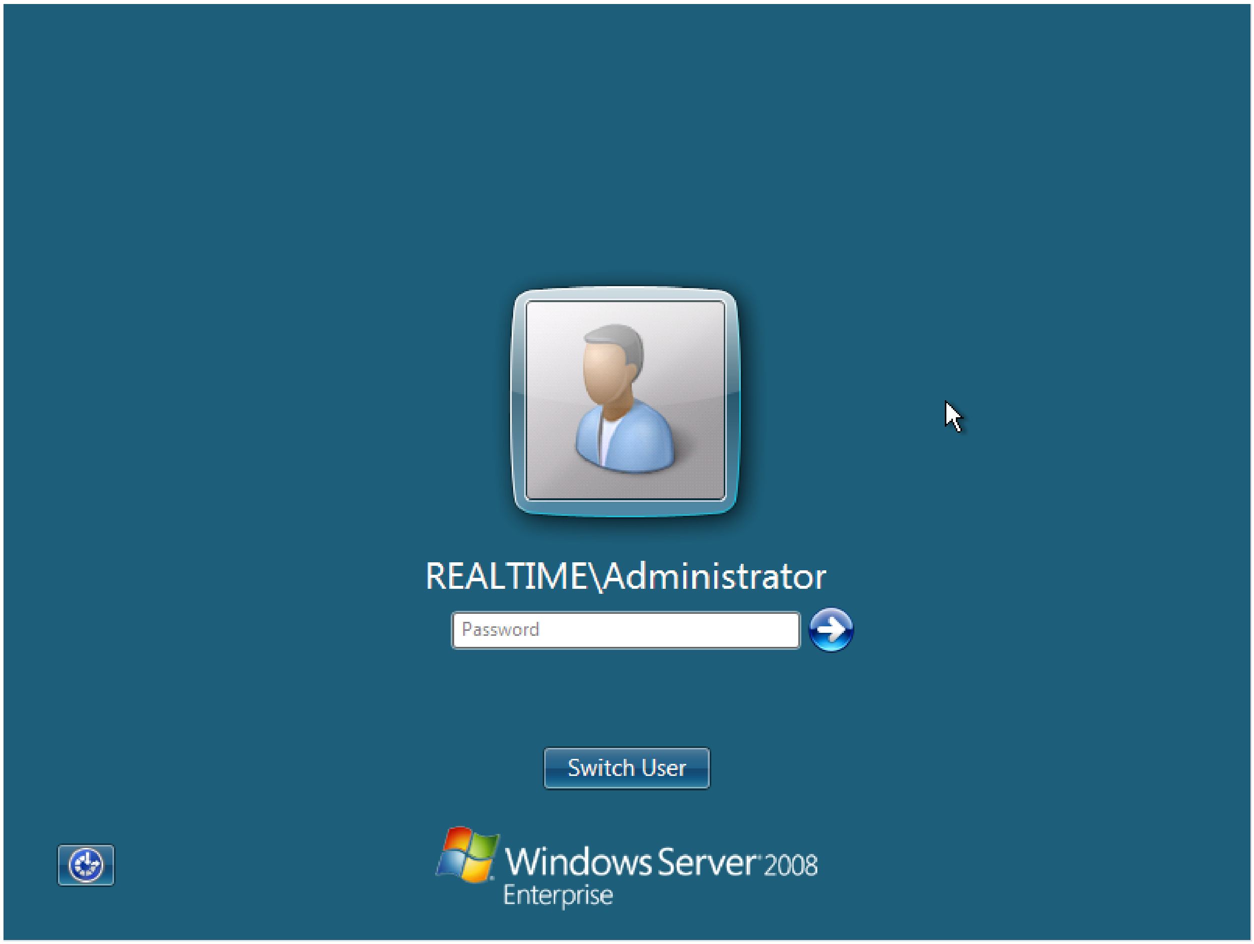


Figure 1.1: Server 2008's logon screen, like much of its GUI, more closely resembles Vista than Windows XP.

Five Years Later, a New Windows OS

With all of the new and improved capabilities discussed in the intro, we're still just scratching the surface of Windows Server 2008, the first new Microsoft server OS to be released in 5 years. With all these new capabilities, Windows Server 2008 stands to be Microsoft's best OS yet. That statement might sound like hyperbole, but considering the previous feature list combined with what you'll find throughout this guide, there are a number of compelling features that are likely to drive you to quickly implement Server 2008.

If you haven't taken a look at Windows Server 2008, take the time to download and install the software into a test environment. Server 2008 shares much of its codebase with Windows Vista, so you'll notice immediately upon installation the similarities between the Server 2008 and Vista GUIs. Though Server 2008 doesn't natively include the high-gloss and resource intensive "Aero" interface, its components haven't moved around much between these two iterations. Because of this, you'll find that making the jump from Vista to Server 2008 will be much easier than making the jump from Windows XP to Vista.

Where the real compelling features arrive with Server 2008, at least according to Microsoft's marketing, are in its improved security and manageability as well as "providing a solid foundation for business workloads." Server 2008's entire code development process was the first server OS to be completed fully under the banner of Microsoft's *Trustworthy Computing Initiative*. As part of this initiative, Server 2008's code was created with established secure coding frameworks and security best practices throughout its architecture and development. The result is a more secure OS, a reduction in its attack surface, and a reduced exposure to the outside world.

Also greatly enhanced is this OS's capacity for management. You'll find new and better mechanisms for managing your servers such as Server Manager and the Remote Server Administration Tools. You'll experience improved and better centralized control over server and workstation configuration through an enhanced Group Policy engine. You'll even find yourself learning more about command-line management through scripting languages like PowerShell and new OS roles like Server Core. A major focus of this OS upgrade is around making Windows easier to manage, which reduces its total cost of ownership (TCO) and speeds your ability to enact change and fix problems when necessary.

The Intent of this Guide

Within the next 10 chapters, we'll be talking at length about all these new capabilities. We'll link each of them to an understanding of how they fit best within your infrastructure. Some of these new features will change Windows administration as we know it. Others may see a lot of hype in the market but really aren't all that compelling. Some have yet to shine but have some potential for the future.

As the title of this guide suggests, the intent is to provide you with the knowledge you need to build a Windows Server 2008 infrastructure. That means my target as author is to assist you in learning the parts of Server 2008 that are important to accomplishing that goal. At the same time, I hope to illustrate the features from which you might want to steer clear or merely put on the back-burner for a little while. As you can see in the introduction to this chapter, there's a lot in Server 2008 to digest. Figuring out which pieces will integrate best is critical to successfully deploying it within your computing environment.

Another of my intents with this guide is to revisit topics that have become established dogma within Windows administrator circles. Due to the feature sets and capabilities we've been using over the past 5 years, some topics—such as domain design, application delivery, and vulnerability management—have become established standard operating procedure for IT everywhere. With the introduction of Server 2008, some of those "known facts" change a little. Some get completely thrown out the window. Some of them remain mostly the same, but with a few changes to your operating procedure. My goal with this guide is to help you understand where you might want to update your thinking when building and administering your Windows infrastructure.

Throughout all of this, we'll be taking a look at some of the biggest new and improved elements that make up Server 2008. We'll peel back the layers of this new OS to expose some of the most exciting new capabilities you'll want to implement immediately within your infrastructure. Most importantly, we'll learn the steps necessary to get it up and running as well as manage it over the long term.

Ten Elements of Server 2008

This guide is broken into ten chapters, which are ordered in such a way that we begin with the information most critical to getting you started. That first set of information will help you get Server 2008 into your environment while at the same time understanding its core management and capabilities. Once we understand its basic structure, we'll move on to the specific feature sets in an order you're likely to want to implement them.

After chapters on installation and core management, we'll continue with a chapter on AD and domain controllers. Then we move to the no-longer-unexciting File Services role with some much-needed conversation on the topic of storage management. We'll then duck into Server Core, Microsoft's svelte new GUI-less OS role. You'll learn why Server Core is compelling and where you might consider implementing it in your network.

Our next stop is to central manageability with Group Policy. The Group Policy engine as well as individual policies and troubleshooting becomes a lot easier with Server 2008. Following this are two chapters on Terminal Services, quite possibly Server 2008's greatest addition. There, we'll talk about installing and managing the new Terminal Services as well as advanced topics associated with application delivery and transport security.

Following is a chapter on system security that will include a discussion on the Windows Firewall with Advanced Security, an oft-ignored functionality in Server 2003 and Windows XP. With Server 2008, the Windows Firewall gains much improved management capabilities that make it easier to use and maintain. We conclude the guide with a discussion on clustering with Windows Server Failover Clustering. Clusters are an excellent way to improve your overall service uptime, and the new clustering capabilities in Server 2008 make them easier than ever before to configure and administer. Let's take a focused look now at these upcoming chapters and what you can expect from each.

Chapter 1: Introduction & Installation of Windows Server 2008

In this, our first chapter, we'll introduce the guide as well as set up the structure of the upcoming chapters. Focused towards the new installation features, we'll discuss each of the three potential mechanisms for getting Server 2008 installed onto physical hardware. These range from the traditional manual installation through scripted installations and concludes with rapid deployment options such as Windows Deployment Services and the Windows Automated Installation Kit. We'll also discuss the new Initial Configuration Tasks wizard that speeds the deployment of new servers.

Chapter 2: Server Manager

Once the OS is installed onto hardware and the initial configuration is complete, we've got to somehow manage that installation. In Server 2008, much of that management is consolidated into a new tool called Server Manager. This tool centralizes many of the traditionally segregated management consoles into a unified location. We'll discuss the separation of a server's responsibilities into Roles, Role Services, and Features, and learn about the uses of each. We'll then dive deep into some of the components you'll see in Server Manager, like Event Viewer, the Reliability & Performance Monitor, Task Scheduler, and Disk Management among others.

Chapter 3: Active Directory Design & Domain Controller Management

I've argued with many that the low-hanging fruit for early Server 2008 implementations will be an environment's domain controllers. As domain controllers are typically few in number and don't often include many third-party tools, they make an excellent starting point for building your Server 2008 infrastructure.

In this chapter, we'll discuss some of the new thinking when it comes to AD domain design. We'll then dive into the domain controller creation process itself, talking about the enhancements to DCPROMO as well as the new Read-Only Domain Controller option intended for branch offices. We'll conclude with a discussion on AD backups and restores as well as the new fine-grained password policies that allow you to create multiple password policies within a single domain.

Chapter 4: File Servers & Storage Management

File Servers are in some ways the least respected servers in the environment. Quietly humming away in the corners of your data center, these servers require little administration other than the occasional permission change or group update. But these servers are at the core of your Windows infrastructure. Storing and protecting users' files and folders is one of the most critical tasks of the IT department.

In Server 2008, this hasn't gone unnoticed. What has been a traditionally organic process of serving up files and folders is now formalized into the File Services Role. This role comes equipped natively with advanced capabilities like Network File Services, Distributed File System, the File Server Resource Manager, and iSCSI support. This chapter is dedicated to educating you on all these new capabilities as well as how best to administer them.

Chapter 5: Server Core

If you listen to the press, Server Core is a hot topic that'll change completely the way we administer certain classes of Windows servers. More like a Linux box than a Windows box, Server Core's user interface is minimal to the point of non-existence. If you've thus far been a fan of command-line management, you'll immediately understand the benefits of this new server role. Conversely, if you've relegated yourself to administration through the graphical interface only, using Server Core will incur a steep learning curve. Server Core supports only a limited set of roles, with further limitations within those roles. This chapter will show you how to install and manage it as well as detail where it has the potential for best use as part of your Windows infrastructure.

Chapter 6: Managing & Customizing Group Policy

Group Policy is one of Microsoft's greatest centralized management toolsets. Using Group Policy alone, you can configure and otherwise lock down the configuration of the servers and clients on your network.

But Group Policy has always included complexities as well. Documenting your configured Group Policy settings has always been challenging. Too many policies have traditionally led to SYSVOL bloat. Figuring out just the correct policy to apply and later troubleshooting that policy when it doesn't apply has historically been challenging. All of these and more are topics that get a lot easier in Server 2008. In this chapter, we'll discuss the changes to Group Policy that fix many of these long-held issues as well as a few more—like multiple local Group Policy Objects (GPOs) and Group Policy Preferences—that extend policy application even further.

Chapter 7: Introduction to Terminal Services

Terminal Services is arguably Server 2008's most compelling and exciting feature set. At the very least, it's my personal favorite. In this, the first of our two chapters on Terminal Services, we'll talk about why that's the case. With a host of new capabilities, some that administrators have been waiting for since Server 2003 R2, this chapter will show you those that you'll want to implement immediately. We'll discuss the use of features previously only available in Citrix Presentation Server like TS RemoteApps, TS Web Access, and TS Gateway. We'll also discuss the best ways to manage your Terminal Servers including profiles, licensing, and resource management.

Chapter 8: Advanced Topics in Terminal Services

Continuing our two-chapter series on Terminal Services, Chapter 8 will dig deep into some of the advanced topics in Terminal Services. Specifically dealing with the complexities of application delivery, we'll talk about the best ways to get server-based applications deployed to your users. Whether that is through TS Web Access or via directly installed application links, you'll appreciate the new ways to make apps available to your users. We'll also discuss at length the process of securing the Remote Desktop Protocol (RDP) through TS Gateway as well as implementing a high-end but low-cost load-balancing capability for your Terminal Servers using TS Session Broker.

Chapter 9: Securing Servers & the Domain

Security is one of the major focuses for this OS release. Thus, no guide on Server 2008 is complete without a comprehensive chapter on securing servers and your Windows domain. In Chapter 9, we'll revisit the concept of componentization and how it is realized in Server 2008. We'll discuss some of the new and improved security features enjoyed by this new OS, like the Security Configuration Wizard, User Account Control, Windows Service Hardening, and BitLocker Drive Encryption.

We'll also talk at length about the Windows Firewall with Advanced Security, a new feature shared with Windows Vista. Adding Server 2008 to your infrastructure gives you critical management capabilities that make easy the administration of this no-added-cost security product.

Chapter 10: Windows Failover Clustering

Lastly, we'll conclude our guide with Chapter 10 on Windows failover clustering. Server 2008's release includes a number of big improvements to clustering with the Windows Server Failover Clustering Feature. As many organizations have not yet used Windows-based clustering due to previous problems or the high cost of implementing previous versions, this chapter will include comprehensive instruction on building, administering, and fitting clusters best into your Server 2008 infrastructure.

Why Should You Upgrade?

I'll admit that I'm personally excited about the Server 2008 release. Though initial sales of Windows Vista have been slower than expected, there is a high likelihood that Server 2008's adoption will be a different event entirely. A big statement, but the reasons and decisions to implement a server OS are different than those when considering a desktop OS upgrade. Let's take a look at some of the pros and cons associated with upgrading both types of OSs and gain an understanding of why I believe you'll move quickly to adopting Server 2008 in your infrastructure.

	Reasons to Upgrade	Reasons Not to Upgrade
Windows Server 2008	Smaller number of instances Instance purposes are well-known Server OS considered more reliable Fewer applications = fewer conflicts Enhanced security Vendor-tested driver sets Known/fewer hardware configurations	Time and cost Concern for application conflicts Potential for outage Interconnectedness between services
Windows Vista	Better security, especially for laptops Improved search capability Improved (though different) GUI	Time and cost More instances across the network More applications = more conflicts Hardware age and upgrade requirements Driver incompatibility Fear (from administrators) Fear (from users)

Table 1.1: A list of reasons to consider before upgrading servers and workstations.

Of those listed, there are a few considerations that should be particularly highlighted as key reasons Server 2008 adoption is likely to be fast. First, in general, organizations tend to deploy a much smaller number of servers than workstations. Thus, there are fewer instances overall that require upgrading. Also notable is that server OSs can be upgraded in a stepwise fashion with much less impact to the environment than a wholesale upgrade to desktops. When desktops are upgraded, there are certain benefits—such as potential application of Group Policies—that are not realized until the entire set is upgraded. Servers don't suffer from this same limitation.

Second, the hardware in servers is typically more well-known than with desktops. Their resource use is also more well-known and often undersubscribed. Servers typically operate in normal conditions with known workloads. Industry-wide, these workloads average about 7% utilization, which means that there are plenty of available resources to support the enhanced needs of an OS upgrade. Specific to the hardware itself, server-class hardware is usually more rigorously tested with driver sets than is desktop and laptop hardware, so driver sets tend to be more mature.

Specific to the features that will drive a quick adoption, we can group them into three rough categories:

- Componentization means a more stable OS instance.
- Enhanced security means a better posture for preventing failures in your environment.
- Enhanced manageability means that your job gets easier.

Let's look at each of these in turn.

Componentization

We'll talk at length about Server 2008's componentization in Chapter 2 on Server Manager and again in Chapter 9 on security. But, for now, know that Server 2008's codebase has been reconfigured to segregate each of the traditional activities supported by a server. Each activity has been isolated into a Role, a Role Service that supports a Role, or a separate Feature.

Further separation has been done by restricting each component's executables and supporting files. When a component is not installed onto the server, that component's files and registry keys are not even present on the server. Fewer files and registry keys mean less of a chance for conflict on that server. It also means there is no chance that a security exploit could make use of this code. Reducing the total file count and registry size means less responsibility for patching that server. If your server doesn't include a component that has a known vulnerability, you don't need to patch that server.

Security

The tenet of componentization dovetails perfectly into the improved security posture you'll see with Server 2008. Server 2008's code has been reworked to reduce the instances of user-level code operating within the kernel level. This further isolates kernel code from the potential for external attack.

Back in Server 2003, Microsoft released a tool called the Security Configuration Wizard (SCW). This tool could lock down a Windows server to just the components necessary to perform its daily activities. So, for example, if the server is a DNS server, the SCW would lock down all the unnecessary ports while keeping open those necessary for DNS functionality. The SCW got little attention by most administrators, in my opinion due to its complexity.

In Server 2008, much of the early work done with the SCW has been rolled up into the individual Roles, Role Services, and Features themselves. So when you install a Role Service, the necessary elements are opened to enable that Role Service to function. This goes further down the "default off" path started with Windows XP SP2 and Server 2003.

In addition to this, the behavior of services is updated to restrict their access. Services in Server 2008 can be assigned a distinct SID. That SID can be assigned individual access permissions across the machine—and indeed across multiple machines. Service privileges have been limited to only the necessary access. Fewer services run with full SYSTEM privileges, reducing the ability for external attacks to use running services as a launch pad for doing damage.

Other security features baked into the system include Address Space Layout Randomization, which prevents exploits from attacking system code based on a known location in RAM and improved Kernel Patch Protection, which helps eliminate ghostware such as rootkits from updating kernel code to their own nefarious ends. All these combine to give Server 2008 the smallest attack surface right out of the box of any Microsoft server OS to date.

Manageability

Many of the security features of Server 2008 occur under the covers. So it is within the manageability features where you the administrator will visibly see the most changes. Server Manager is one example of this, where dozens of previously segregated management consoles have been aggregated into a single interface. Other examples include the Remote Server Administration Toolkit, which brings to your management workstation much of the functionality of Server Manager. Group Policy's move to the new ADMX file format for administrative templates along with its new Group Policy processing engine mean a greater likelihood of successful policy application.

Continuing along this theme, a vastly improved Event Viewer means that problems are more easily recognized. Using it, more data across multiple systems can be analyzed all on the same screen. AD itself is reconfigured as a restartable service, which improves your ability to fix it when it has problems. Even IIS gets a new management console and rights schema that means your Web site administrators no longer require Administrator rights to the entire system to do their job.

Why should you upgrade? Simply because the experience of managing your Windows infrastructure becomes more stable, more secure, and ultimately more manageable once you do.

Introducing Windows Server 2008

Let's move away a bit from my grandiose predictions of your adoption of Server 2008 to a more down-to-earth discussion about its specific capabilities. Server 2008 comes in eight flavors, though you're likely only to use two or three of these editions in most places within your infrastructure. In this section, we'll talk about the various editions you'll need to know about as well as their hardware and licensing requirements.

Windows Server Editions

As stated earlier, Windows Server 2008 comes in eight editions. The differences among editions can be relatively slight, though the cost differences between some editions can be large. So be aware of the kind of server you need before you make a purchase. If you don't require the additional functionality that Enterprise Edition offers, you might not want to pay the multiple thousand dollar difference between it and Standard Edition. If you don't plan to use Hyper-V, the new Microsoft virtualization engine, you might not want to purchase it.

> ✎ That being said, the cost difference between Server 2008 with and without Hyper-V at the time of this writing is only $28. So for an extra thirty-odd bucks, it might be worth the cost.

In Table 1.2, I've included the list of editions currently announced as of this writing. Also there is a short description of the edition and how it compares with its neighbors.

Windows Server Edition	Description
Windows Server 2008 Standard	Standard Edition is the core of each of the other editions. Standard Edition supports nearly all functionality of Server 2008 with the exception of those listed in Enterprise Edition. Be aware that Server Core is not considered an "edition" but instead an "installation option." Thus, purchasing any version of Server 2008 nets you the ability to install it as a Server Core instance.
Windows Server 2008 Enterprise	Enterprise Edition adds to Standard Edition the ability to address more processors and RAM, a benefit to systems like database servers that require excessive quantities of memory or additional processors beyond those supported in Standard Edition. Enterprise Edition also adds the Windows Server Failover Clustering Role, the ability to hot-add memory, and additional AD features such as Active Directory Federation Services and the advanced pieces of Active Directory Certificate Services. Enterprise Edition lifts restrictions on use for standalone DFS roots as well as RRAS and Terminal Services Gateway connections. In a very compelling move, Enterprise Edition bestows special virtual server license benefits. For every physical instance of Enterprise Edition licensed in an infrastructure, four virtual instances can be hosted on the same machine.
Windows Server 2008 Datacenter	Datacenter Edition is intended to be the workhorse edition for workloads with the greatest needs in the environment. Datacenter Edition provides no extra Roles over and above Enterprise Edition. However, it does allow for a much greater amount of resources to be addressed by the system, scaling to 32 processors for x86 systems and 64 processors for x64 systems. Datacenter Edition also adds the ability to hot-replace memory as well as hot-add and replace processors. One of Datacenter's most compelling benefits also has to do with virtual server licenses. For every physical instance of Datacenter Edition licensed in an infrastructure, an unlimited number of virtual licenses can be hosted on the same machine.
Windows Server 2008 for Itanium-Based Systems	Windows Server for Itanium-based systems is a special edition of Server 2008 designed for installation to systems with Itanium processor architectures only. This edition is similar in resource capabilities to Datacenter Edition but is limited to very specific Roles and Features, supporting only the Web Services (IIS) and Application Server Roles.
Windows Web Server 2008	Also limited in functionality is Web Server Edition. This edition is limited to installing only the Web Server (IIS) Role and is intended for use exclusively as a Web server. Consequently, it has a much lower price point than other editions. Web Server Edition resource limits are the same as with Standard Edition.

Windows Server Edition	Description
Windows Server [illegible] Windows Server [illegible] Windows Server [illegible]	All three of the most common editions – Standard, Enterprise, and Datacenter – can all be purchased without Microsoft's native hardware virtualization support, called Hyper-V. The regular editions of these three support the use of Microsoft's hardware virtualization. These slightly less expensive editions do not include that support.

Table 1.2: Windows Server editions and explanations of the benefits of each.

Hardware Requirements and Limitations

As times change, so do hardware requirements. With every new edition of an OS, it seems like more resources are necessary to power that OS. Server 2008 is no exception. However, it can be argued that the hardware requirements for Server 2008 have less to do with a greater need for them by the OS and more to do with a right-sizing of what constitutes the true "minimum requirement." You be the judge.

For all editions, the following minimum and recommended hardware resources are suggested:

- *Processor*—1GHz minimum for x86 processors. 1.4GHz minimum for x64 processors. 2GHz or faster recommended for all architectures.
- *Memory*—512MB RAM minimum. 2.0GB RAM or greater recommended.
- *Available Disk Space*—10GB space minimum. 40GB or greater recommended. Less space is required for Server Core installations.

Table 1.3 shows the upper limits on those same resources for each edition.

	Maximum x86 Processors	Maximum x64 / IA64 Processors	Maximum x86 RAM	Maximum x64 / IA64 RAM
Standard Edition	4	4	4GB	32GB
Enterprise Edition	8	8	64GB	2TB
Datacenter Edition	32	64	64GB	2TB
Itanium Edition	N/A	64	N/A	2TB
Web Edition	4	4	4GB	32GB

Table 1.3: Windows Server Editions and physical resource maximums.

Supported Upgrade Paths

Though the safest way to upgrade any network service is usually through a complete reinstallation of the OS, it is often beneficial to undergo an upgrade for some services. One example of a service that is typically upgraded at least once involves your domain controllers. This is done so that AD itself can be upgraded as part of the process. There are some limitations to which OSs and editions can be upgraded. Those limitations are listed in Table 1.4.

If you have this OS and Edition	You can upgrade to this OS and Edition
Windows Server 2003 R2 Standard Edition (Gold, SP1, or SP2) Windows Server 2008 Standard Beta 3	Windows Server 2008 Standard Edition Windows Server 2008 Enterprise Edition
Windows Server 2003 R2 Enterprise Edition (Gold, SP1, or SP2) Windows Server 2008 Enterprise Beta 3	Windows Server 2008 Enterprise Edition
Windows Server 2003 R2 Datacenter Edition (Gold, SP1, or SP2) Windows Server 2008 Datacenter Beta 3	Windows Server 2008 Datacenter Edition

Table 1.4: Windows Server versions and editions and upgrade capabilities.

> Be aware that upgrading to a Server Core installation is not supported for any OS or edition. Server Core installs a wholly different set of files to build its OS. Thus, even though it is considered an installation “option,” it is really much like a completely different OS entirely.

Licensing and Activation

If you’re used to using the same license keys for installing multiple copies of Windows Server, you’ll sorely miss this capability in Server 2008. The licensing mechanisms, which Microsoft titles Volume Activation 2.0, have grown less user-friendly with the upgrade to Server 2008. License keys entered for a particular server must be activated either over the Internet or through another mechanism such as calling the Microsoft Activation Center.

> Due to this change in how Windows is licensed, be cautious when you’re creating demonstration, test, and evaluation machines. If you activate the license for a short-lived evaluation machine, that activation goes against your total count of activations. In many cases, if you’re planning to use that evaluation machine for only a short time—such as when you’re testing out the things you learn in this guide—consider not activating it. You’ll need to rebuild the server every so often once the evaluation period expires. But as you’ll learn later in this chapter, the process to build a new Windows server is very quick and easy.
>
> Trust me on this. Before really realizing this new way of operating with evaluation systems, I found myself contacting the Microsoft Activation Center a few times to get an activation struck from my permanent record. Let’s just say that that process can be quite painful.

Two options are available for the infrastructure of license assignment. Depending on the needs of your organization and the type of licensing arrangement you have with Microsoft, you will be assigned one of these two possible options:

- *Multiple Activation Key (MAK)*—With MAKs, a single key can be used for multiple systems. The use of each key must be activated either through the Internet or another form of contact to the Microsoft Activation Center. A specific number of activations are enabled for any particular MAK. Activations over the limit assigned to the key will be rejected at the point of activation. With MAKs, it is also possible to activate multiple computers at the same time using the Volume Activation Management Tool. This tool allows for a group of computers to be activated at the same time through a single connection or call to the Microsoft Activation Center.
- *Key Management Service (KMS)*—A KMS key is used along with a central activation service that is hosted within your infrastructure. KMS activations are typically used in environments with large numbers of needed activations and the need to locally manage the count of activations. In this configuration, the key is used only to enable the KMS service. The KMS service then manages the activation of servers within the infrastructure. Computers that make use of KMS must be reactivated every 6 months to maintain licensing requirements. Reactivation is done automatically by the KMS server to all activated systems on the network. In order to use a KMS service, a minimum of 25 workstations or 5 servers must be on the network.

> Because of this, any server or workstation that is activated through a KMS must be in network contact with the KMS server at least once during every 6-month period. This can be a problem for servers that are activated using KMS and then relocated to another network unreachable by the KMS service.

So there you have it. You now have everything you need to start installing Server 2008 onto your available hardware. You will still need the media of course.

Installing Server 2008

Now that we've come to understand the requirements that Server 2008 is going to place onto our hardware, let's take some time to analyze the different possible ways that we can get it installed. Traditionally, the most often used mechanism for completing an installation has been the manual drop-in-the-disk-and-go method. This installation mechanism has long been the most time-consuming and painful way to install an OS, but it is usually the most successful. It's also arguably the easiest.

Here's why. Microsoft's installation routine in previous versions had a tendency to prompt for input throughout its process. In the beginning, the installation would request information about the drive to install and the type of formatting for the partition. Later on, it would prompt for questions about time zones and keyboard layouts. Even further down the line would be questions about networking configuration. The end result is that for a manual installation, you're effectively required to babysit to completion. I myself feel like I've lost a few years of my life waiting for server installation timers to count to zero. I'm sure you have as well.

Manual Installations

I tell this story because Server 2008 changes this manual installation process in a highly exciting way. Simply put, with the manual installation routine in Server 2008, all the questions required to complete the installation are asked either at the beginning of the process or once the process is fully complete. Take a look at Figure 1.2. You'll see the excessively simple *Install now* button that arrives once you boot the server from the installation media and after you answer the first three questions about language, time, and currency format, and input method.

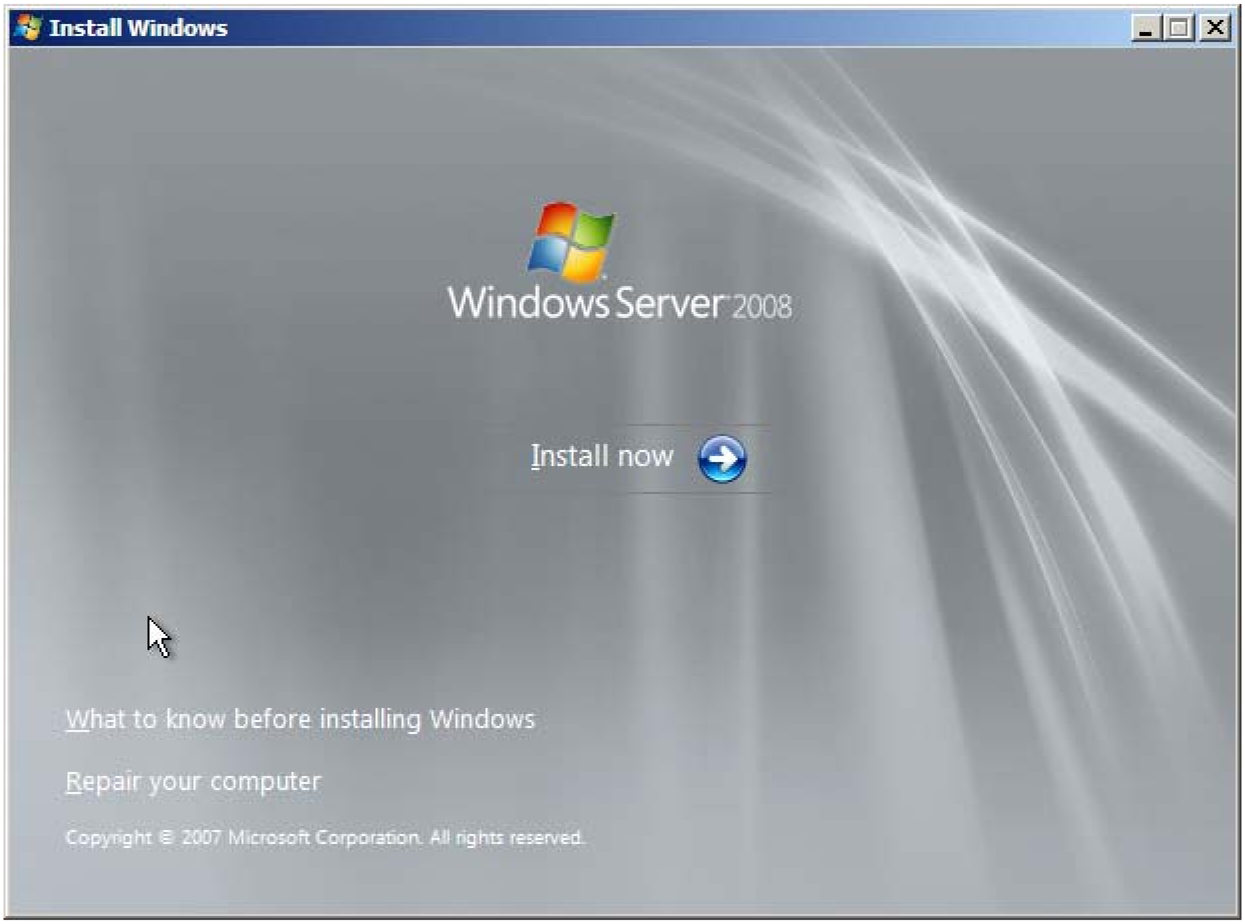

Install Windows
Windows Server 2008
Install now
What to know before installing Windows
Repair your computer
Copyright © 2007 Microsoft Corporation. All rights reserved.

Figure 1.2: The excessively simple* Install now *button for Server 2008 manual installations.

> Also note the line in the lower-left titled *Repair your computer*. We'll discuss the use of this in Chapter 2 when we talk about server backups and restore. But keep this location in your mind for later, as it is within this location where bare-metal restores and other repair operations can be launched.

Once you click to begin the installation, you'll be asked to set only a very few initial configurations:

- Your product key
- The edition of Server 2008
- The end-user licensing agreement acceptance
- Whether you want a clean installation or an upgrade installation
- The disk you want the installation to target

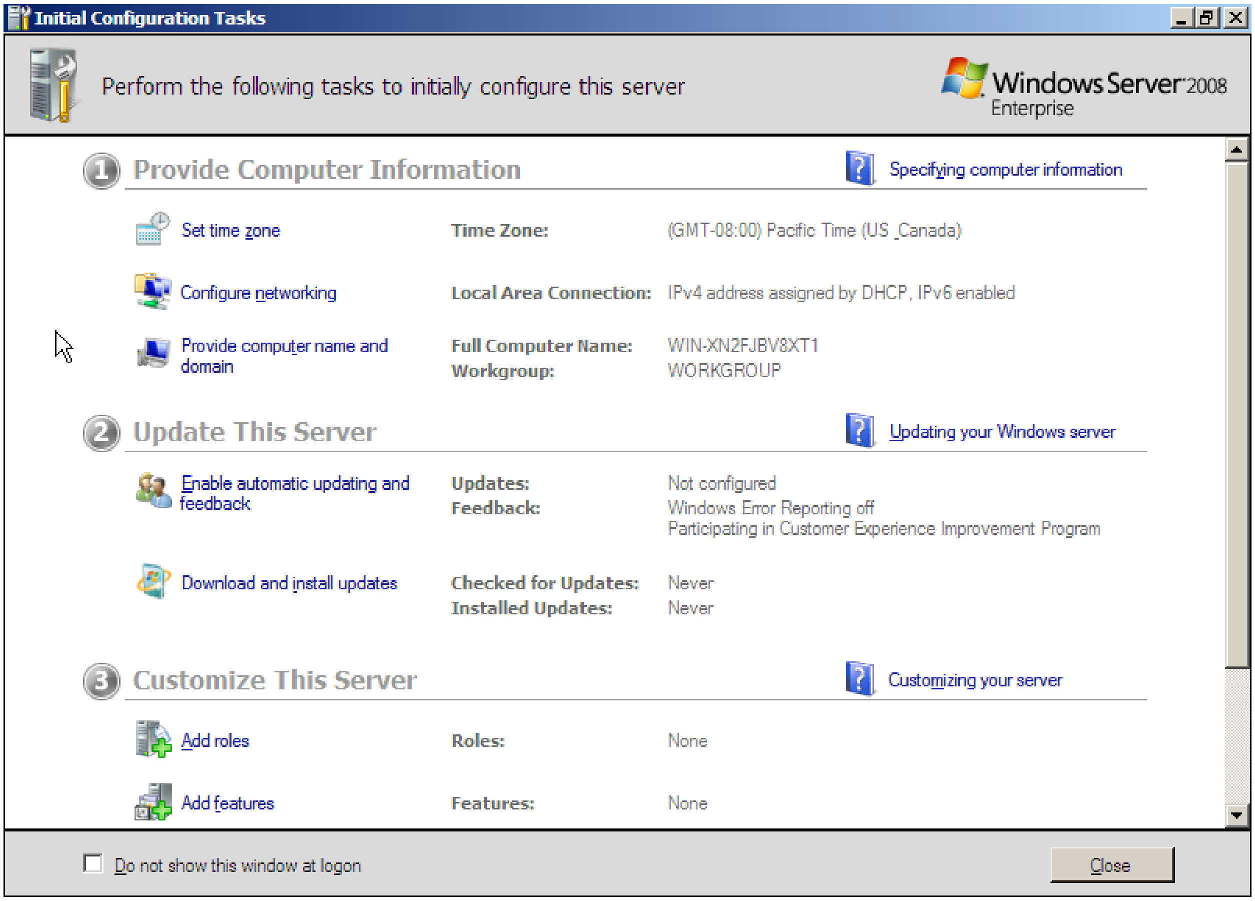
Once you've completed those few steps, the installation begins. Now you can walk away from the machine.

Initial Configuration Tasks

Depending on whether you're installing the server with or without the Server Core "option," the installation can range from a few minutes to several. Control is not returned until the installation is complete, the server has completed its post-installation reboot, and the system is ready for the initial login.

> Remember that a server can be installed either with or without the Server Core option. Once installed in one configuration, the only way to switch to the other is through a complete reinstall. We'll talk in more detail about Server Core in Chapter 5.

Once this process completes, you'll first be prompted to set the password for the Administrator account. An error message will appear on the screen that explains *The user's password must be changed before logging on the first time*. What this effectively means is that the local setting for *User must change this password at next logon* is set for the Administrator account as a component of the installation. Once this password is set, the system will automatically log you in as Administrator.



Initial Configuration Tasks

Perform the following tasks to initially configure this server

Windows Server 2008 Enterprise

1 Provide Computer Information — Specifying computer information

Set time zone — Time Zone: (GMT-08:00) Pacific Time (US Canada)

Configure networking — Local Area Connection: IPv4 address assigned by DHCP, IPv6 enabled

Provide computer name and domain — Full Computer Name: WIN-XN2FJBV8XT1
Workgroup: WORKGROUP

2 Update This Server — Updating your Windows server

Enable automatic updating and feedback — Updates: Not configured
Feedback: Windows Error Reporting off
Participating in Customer Experience Improvement Program

Download and install updates — Checked for Updates: Never
Installed Updates: Never

3 Customize This Server — Customizing your server

Add roles — Roles: None

Add features — Features: None

☐ Do not show this window at logon

Figure 1.3: The Initial Configuration Tasks wizard appears immediately after the initial post-installation logon.

Once you've logged in, you'll be greeted with the Server 2008 desktop. Shortly after logon, a new wizard will appear called *Initial Configuration Tasks* that looks similar to the image in Figure 1.3. The first thing you'll notice is that in order to eliminate all the mid-install questions, Microsoft has taken some license with the answers to these former questions. As an example, in order to complete the installation, the server requires an initial computer name. But rather than asking you for that information during the installation routine, with Server 2008, Microsoft enters placeholder information for you. See in Figure 1.3 that the initial computer name for this server is set to WIN-XN2FJBV8XT1.

The Initial Configuration Tasks wizard is broken into three sections, each of which includes necessary initial configurations that were previously done elsewhere using different management consoles. The wizard itself isn't so much a new console for making these changes. It is little more than a series of pointers to the standard wizards for each of the other configurations. Thus, for example, when you click the link to *Configure networking*, Initial Configuration Tasks brings forward the *Network Connections* control panel.

Most settings within the three sections are self-explanatory. You'll need to set the proper name for the computer and configure its networking, time, and time zone. For the second section, you'll need to enable automatic updates and download and install any available updates to patch the server. The third section discusses the installation of Roles and Features as well as enabling Remote Desktop or configuring the Windows Firewall. For the most part, leave the Roles and Features configuration alone. Installing those is better served through Server Manager, which we'll talk about in the next chapter. Once you've completed the necessary sections in Initial Configuration Tasks, click Close.

> ✎ One element available in Server 2003, though not with the same level of exposure, is the configuration of feedback to Microsoft. If you click the link for *Enable automatic updating and feedback*, and then select *Manually configure settings*, you'll be greeted with a screen that allows you to configure update settings as well as feedback settings.
>
> In addition to configuring for necessary software updates, this screen provides a single location where Windows Error Reporting and the Windows Customer Experience Improvement Program settings are configured. Be aware of how you configure these settings. Unlike much of the rest of the Server 2008 OS, these settings can enable the submission of information about system errors and your usage information back to Microsoft. If your corporate policy mandates that this information is not released, enabling participation in these programs can violate that policy. The programs are designed with the intent of collecting information about where Windows is having problems and how it is being used. According to Microsoft, personal information is not collected.

Windows Preinstallation Environment

Throughout this installation process, it should be evident that the blue installation screens of previous OSs are no longer present. This is due to the integration of the Windows Preinstallation Environment (WinPE) directly into the installation routine for Windows. WinPE is an installation tool that has been available for certain enterprise-level customers since the initial release of Windows XP. But its availability for all installations of Windows arrived only with the release of Windows Vista.

The text-only blue installation screens of previous versions have been removed for a number of reasons (and not only for their lack of aesthetics)—not the least of which are the requirements for certain drivers to include "real mode" DOS-compatible drivers to support bootstrapping the installation routine. Many manufacturers have eliminated support for these types of drivers. Because of this, an installation bootstrapper that could provide the same level of OS support as Windows itself was necessary. WinPE has its roots in the Windows OS, and thus can support rich installation options like a full TCP/IP stack, mapped drives, and internal customization. It also works well with the automated deployment mechanisms we'll discuss in the next section.

Automated Installations

With Vista and Server 2008, a number of automated installation capabilities are possible for speeding the process of getting OSs applied to hardware. Two methods for doing an automated installation are commonly used: scripted and image-based installations. Both are supported through Microsoft toolsets:

- *Scripted installations.* Scripted installations make use of the regular installation media used in manual installations. Added to that media is an unattended installation file that includes the answers to questions asked as part of the installation or through the Initial Configuration Tasks wizard. The benefit of scripted installations is that they tend to be device-independent. Because the standard media-based installation of Windows will locate and install most of the correct drivers required by the server, only the unattended installation script is necessary to answer questions about specific configuration parameters. Thus, the same unattended installation script that is used to install to an HP server can often be used to install to a Dell server or an IBM server.
- *Image-based installations.* Scripted installations are great for heterogeneous environments but they suffer from two issues: First, the scripts are challenging to develop, as we'll learn in a second. Second, the time to complete a scripted installation can be lengthy. Depending on the level of customization needed, the time to complete an installation can consume as much or more time as for a manual installation. Image-based installations solve this through the creation of a "standard image," the copying of which to the target system automatically generates an OS instance. Because this "standard image" is an already-configured server, complete with patches, applications, and customizations, the single-step process of installing the image results in a fully realized server instance. With scripted installations, applications and patches over and above the initial OS installation must be applied after the initial installation is complete. The downside of image-based installations is that driver sets between hardware types are often different. Because images include the driver set of the source machine as part of the image, it is challenging and sometimes impossible to use an image from one hardware configuration onto another.

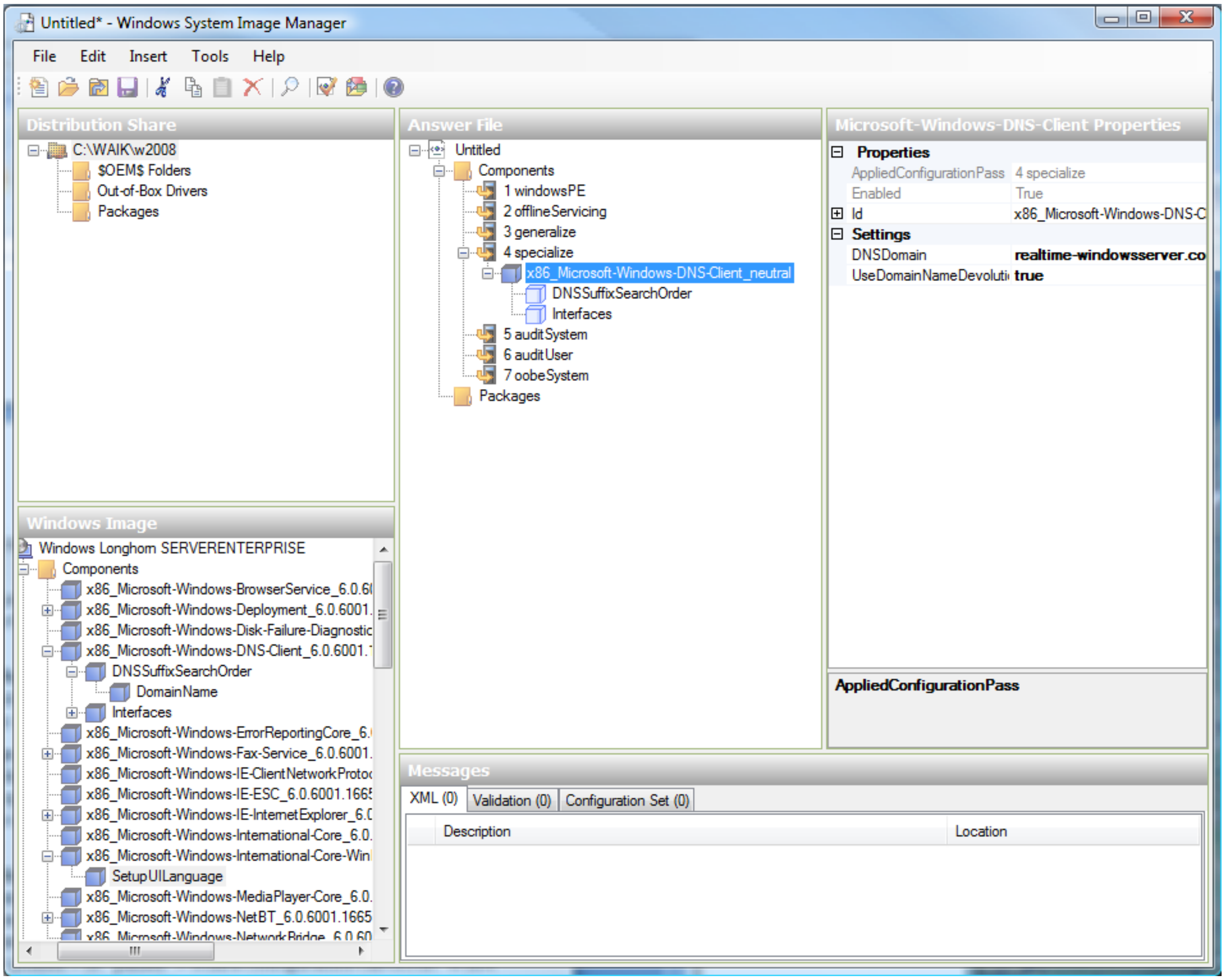
Let's start by taking a look at the process to create an unattended installation file for a scripted installation.

Scripted Installations with the Windows System Image Manager

To create an unattended installation file, you'll first need to download and install the *Windows Automated Installation Kit* (WAIK) from the Microsoft Web site. This large download of nearly a gigabyte includes the necessary components to interrogate OS media, enumerate a list of configuration options, and ultimately create an unattended installation file. Part of the WAIK is the *Windows System Image Manager* tool. WSIM replaces what used to be called *Setup Manager* in previous versions and comprises the tools used to create your unattended installation files.

> ✎ Be aware that the WAIK downloads with an .IMG extension. To install it, you will either need to burn it to a DVD or mount it using an ISO/IMG mounting tool. If your mounting tool doesn't support the .IMG extension, you can safely rename its extension to .ISO.

Once created, you will use that file along with the Server 2008 installation media to boot a candidate system. The file, which must be named *autounattend.xml*, is typically copied to FAT32-formatted removable media like a floppy disk or USB hard drive and inserted into the system along with the installation media as the system is booted. The system will read the installation media along with the instructions provided as part of the autounattend.xml file to complete the installation.


Untitled* - Windows System Image Manager
File
Edit
Insert
Tools
Help
Distribution Share
C:\WAIK\w2008
OEM Folders
Out-of-Box Drivers
Packages
Answer File
Untitled
Components
1 windowsPE
2 offlineServicing
3 generalize
4 specialize
x86_Microsoft-Windows-DNS-Client_neutral
DNSSuffixSearchOrder
Interfaces
5 auditSystem
6 auditUser
7 oobeSystem
Packages
Microsoft-Windows-DNS-Client Properties
Windows Image
Windows Longhorn SERVERENTERPRISE
Components
x86_Microsoft-Windows-BrowserService_6.0.60
x86_Microsoft-Windows-Deployment_6.0.6001.
x86_Microsoft-Windows-Disk-Failure-Diagnostic
x86_Microsoft-Windows-DNS-Client_6.0.6001.1
DNSSuffixSearchOrder
DomainName
Interfaces
x86_Microsoft-Windows-ErrorReportingCore_6.
x86_Microsoft-Windows-Fax-Service_6.0.6001.
x86_Microsoft-Windows-IE-ClientNetworkProto
x86_Microsoft-Windows-IE-ESC_6.0.6001.1665
x86_Microsoft-Windows-IE-InternetExplorer_6.0
x86_Microsoft-Windows-International-Core_6.0.
x86_Microsoft-Windows-International-Core-Win
SetupUILanguage
x86_Microsoft-Windows-MediaPlayer-Core_6.0.
x86_Microsoft-Windows-NetBT_6.0.6001.1665
x86_Microsoft-Windows-NetworkBridge_6.0.60
Messages
XML (0)
Validation (0)
Configuration Set (0)
Description
Location

⊟ **Properties**	
AppliedConfigurationPass	4 specialize
Enabled	True
⊞ Id	x86_Microsoft-Windows-DNS-C
⊟ **Settings**	
DNSDomain	**realtime-windowsserver.co**
UseDomainNameDevoluti	**true**

AppliedConfigurationPass

Figure 1.4: The Windows System Image Manager showing a configured setting for DNS suffix search order.

To create your *autounattend.xml* file, complete the following steps:

1. *Download and install the WAIK.* You'll want to install this onto a workstation and not the server you intend to build using the unattended installation file. The workstation you use can be your desktop.
2. *Locate and copy install.wim.* WIM, or Windows Imaging Format files, is a file-based disk image format. These files are used for booting to WinPE and ultimately installing OSs. The *install.wim* file needed by WSIM is found on the Server 2008 media in the *sources* folder and contains the metadata information necessary to do an installation of Server 2008. Copy this file to your desktop.
3. *Launch WSIM.* Upon launching WSIM, you'll see a relatively busy screen similar to what is shown in Figure 1.4, though initially your screen will start with much less information. Each of the windows within the WSIM link to each other. As an example, a distribution share (in the upper-left) can have multiple Windows Images (in the lower-left). You'll interrogate the possible settings of those Windows images and configure the ones of interest using the middle and right panes. The answer file in tree format will show the configured settings. The right-most pane will include the options both available and set for a desired configuration.
4. *Create a new distribution share.* The first step with WSIM is to create a share that will contain your working folders as you create your unattended installation file. Do this by right-clicking *Select a Distribution Share,* and clicking *Create Distribution Share*. Select an appropriate folder and click *Open*.
5. *Select a Windows image or catalog file*. Next, you'll need to locate and interrogate the install.wim file you copied from the Server 2008 media. Do this by right-clicking *Select a Windows image or catalog file* and choosing *Select Windows image*. Locate your install.wim file and click *Open*. WSIM will search the file for available installations and provide you with a list of those available. In our case, we'll install *Windows Longhorn SERVERENTERPRISE*. Yours may be slightly different. An error will appear noting that no catalog file exists and asking if you want to create a new one. Select *Yes* to continue. The process to create the catalog file can take an extended period of time. WSIM at this point is discovering and cataloging all of the potential configuration options available.
6. *Create a new answer file.* In the center pane, right-click *Create or open an answer file* and select *New answer file* to create a new answer file to be stored in your distribution share. The terms "answer file" and "unattended installation file" here are used interchangeably. You'll notice that the answer file has seven components. Each of these components, called a "pass" is one phase of the installation process. Depending on the configuration you intend to include, that configuration may be set in one of the seven possible passes.

7. *Configure the settings for your unattended installation file.* The lower-left pane will now include the list of options available for configuration in your answer file. Here's where the hard part begins. Discovering which of the options are necessary and interesting is a time-consuming process. You'll want to dig through the components to find configurations of interest. For example, if you want to set the DNS suffix search order, you can right-click *x86_Microsoft-Windows-DNS-Client_6.0.6001.16659_neutral* and select *Add setting to Pass 4 specialize*. Then highlight the new setting under *4 specialize* and in the upper-right screen look for *DNSDomain*. There, you'll want to enter the correct value for suffix search order. Continue with setting your desired configurations as necessary.

8. *Validate the answer file.* Due to linkages between settings, some settings require others to also be set. Once you have completed setting those of interest to you, you'll want to validate the answer file to ensure that it includes the necessary components. Do this by clicking *Tools* and then *Validate Answer File*. Saving the answer file also completes a validation prior to the save.

9. *Save the answer file.* From the *File* menu, select *Save Answer File*. Save the file as *autounattend.xml* to use it for a server installation.

10. *Copy the file to removable media and boot from the Server 2008 media.* Once you've completed your autounattend.xml file, you can copy it to the root of your removable media and use it to boot a candidate system along with the Server 2008 DVD. If you've done everything properly, the installation should complete and a fully-configured server will be the result.

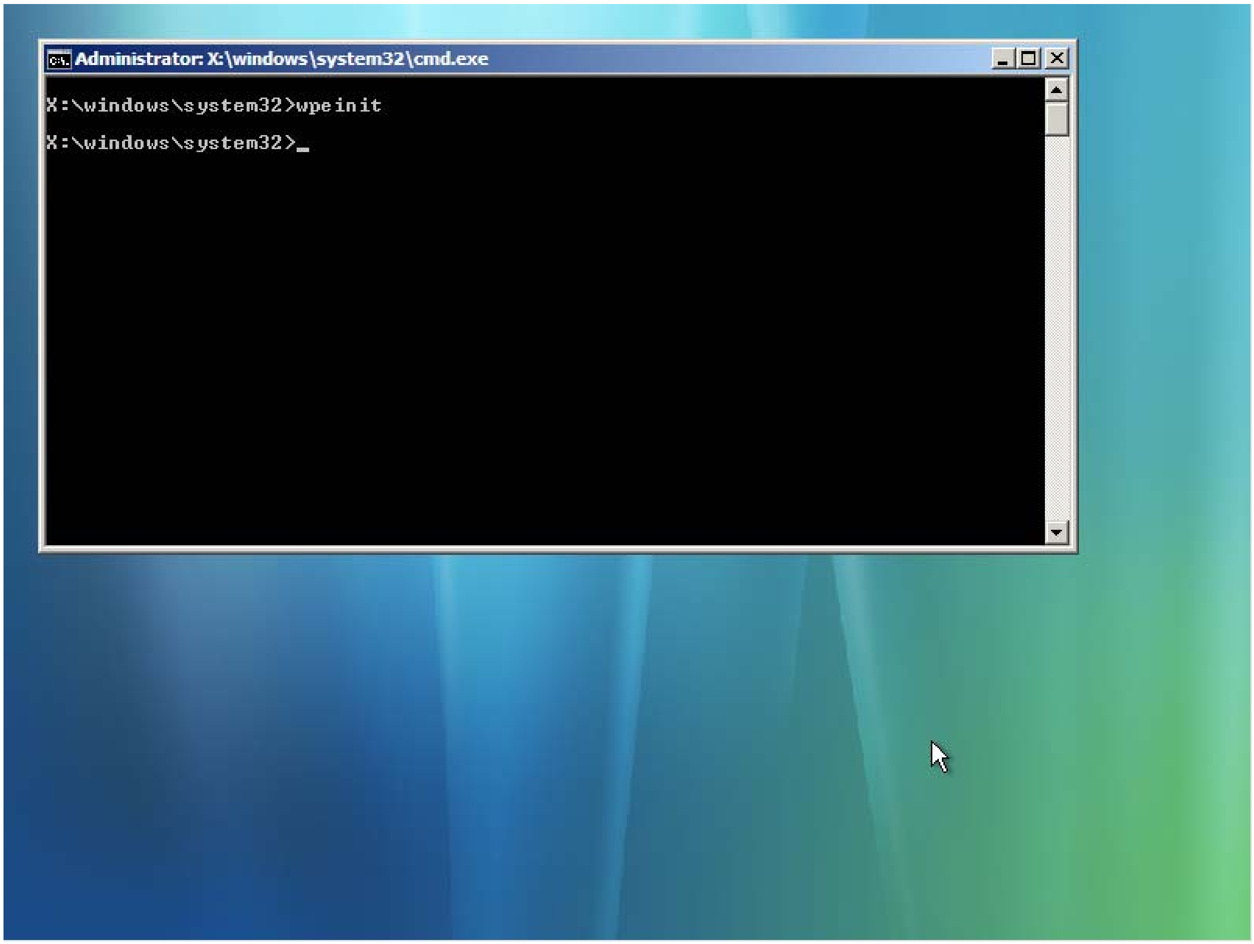
> As has been stated above, the sheer number of options available in WSIM makes complicated the process of creating an unattended installation script. Adding to this complexity are certain elements that are required for the script to function. This will be a trial-and-error process to find the correct set of configurations necessary for your environment. Specific information is available on the Internet about configurations that are required and others that are good suggestions to include.

Image-Based Installations with ImageX

The process above discusses the creation of a scripted install. But in some cases what you really want is the replication of a fully-realized server image. Using that image, you can drop copies of your reference computer (often called the "golden image" or "standard image") onto other machines in a single step. Also part of the WAIK is ImageX, which is a command-line tool used along with WinPE-bootable media to create and later deploy those images. ImageX's focus on the command line makes it a somewhat complex tool to use as well. Let's take a look at the steps you would want to use in creating and later deploying an image using ImageX and WinPE.

- *Create your reference computer*. The first step is to create a computer whose configuration will become the image you will later deploy. For your reference computer install the OS and any necessary patches. Also install and customize any applications that you want residing on your image. When you're finished you can use SysPrep to reseal the installation and eliminate any personalization information that links your image to a particular computer. Do this by running the command *C:\Windows\System32\sysprep\sysprep.exe /generalize /shutdown*.
- *Create your bootable WinPE CD*. Your next step will be to create a bootable CD with the WinPE OS. This CD will run the OS separately from the one installed to your reference computer to allow to image to be captured. The process to create this CD involves multiple steps:
    - With the WAIK installed onto your desktop, from the command prompt change to the folder *C:\Program Files\Windows AIK\Tools\PETools*. Then enter the command *copype.cmd x86 C:\{target folder}*. This will create a bootable WinPE instance for an x86 architecture computer called *winpe.wim* in the target folder.
    - This initial process does not include the executable for ImageX itself. So you'll need to mount the image you created in the previous step and add it. You can mount the boot image you just created with the command *imagex.exe /mountrw C:\{target folder}\ISO\sources\boot.wim 1 C:\{target folder}\mount*. Once you've mounted the image, navigate to *C:\{target folder}\mount*. From there, drill deeper into the *\Windows\System32* folder and copy the *imagex.exe* file from your desktop computer to this location.
    - If your source computers will require additional device drivers not on the standard Server 2008 media you'll need to pre-stage into the image's driver store. First, unpack the driver from any installation executables and find its .INF file. Then add it into the image with the command *peimg.exe /inf:{path to driver's INF file} C:\{target folder}\mount\windows*.
    - Once complete with any configurations, you'll need to unmount the image and commit your changes with the command *imagex.exe /unmount C:\{target folder}\mount /commit*.
    - Create a bootable ISO file with the folders you've created in *C:\{target folder}*. You can use any standard CD burning tool to do this. If you don't have one available, the WAIK includes a CD burning tool as well called *oscdimg.exe*. To use this tool to create your ISO, use the command *oscdimg.exe -bc:\{target folder}\etfsboot.com -n -o c:\{target folder}\ISO c:\{target folder}\winpe.iso*.
    - The result of the last command will be an ISO file that includes a bootable version of WinPE called *winpe.iso* in *C:\{target folder}*. Burn that ISO image to a CD.

- *Capture an image of your reference computer.* To do this, boot the reference computer with the CD you just created. Once the CD has booted you'll be greeted with the WinPE OS, which will look similar to Figure 1.5. You'll see here that WinPE is effectively a command-line-based OS. To capture an image of your reference computer with maximum compression, use the command *imagex.exe /capture C: C:\w2008.wim "Windows 2008 Image" /verify /compress maximum*. This will create a WIM image named *w2008.wim* in the root of your reference computer's file system. Note that since WinPE enjoys a full TCP/IP stack, it is possible to map a drive to store this image elsewhere if you wish.
- *Deploy your image.* To deploy your captured image to another computer, boot that computer with your bootable CD created earlier and map a drive to the location where your WIM image is stored. Then, to deploy the image use the command *imagex.exe /apply {path to WIM image}\w2008.wim 1 C:*.



```
X:\windows\system32>wpeinit

X:\windows\system32>_
```

Figure 1.5: WinPE's command-line-centric OS can be used for image-based installations.

Due to the CLI-centric nature of all those commands, the steps above probably look more complicated than they are in actuality. The most complex part of the process is creating the WinPE bootable CD. Once complete there, the only commands necessary are those to gather and later deploy your images.

✎ Remember too that the process used above to pre-stage drivers into the bootable CD's image can also be used for the image created by your reference computer. If you have target computers with other driver set needs, you can pre-stage them into the image you plan to deploy as well.

Windows Deployment Services

If you'd like to further automate the deployment of images once you've captured them, Microsoft provides a GUI-based tool that makes much of this relatively easy. This tool, *Windows Deployment Services*, arrives as an installable Role on your server. By installing the WDS Role to your server you can make your images available across the network for installation anywhere. WDS supports Preboot Execution Environment (PXE) instructions, which can be used by target servers that support PXE booting to eliminate the need for bootable CD's.

💣 Be aware that network equipment is often configured not to allow multicast traffic to pass through network subnet boundaries. As WDS multicast installations require this traffic to operate, you may need to ensure that your clients and server are both within the same subnet or make an exclusion within your network security policy for this kind of traffic.

After you've installed the WDS Role to your Server 2008 instance, you'll need to prepare that server for use. Do this by right-clicking the server name of interest in the WDS node of Server Manager and choosing *Configure Server*. The configuration wizard will ask for a path to store your system images as well as an initial configuration for PXE boot responses.

Now you can add your images. When prompted, either supply a path to the images you created using the steps above with ImageX. Or, if you'd rather install standard images from the Microsoft-supplied media, point the wizard to the *\sources* folder on your installation media. Once the images have been added, the screen will resemble Figure 1.6.

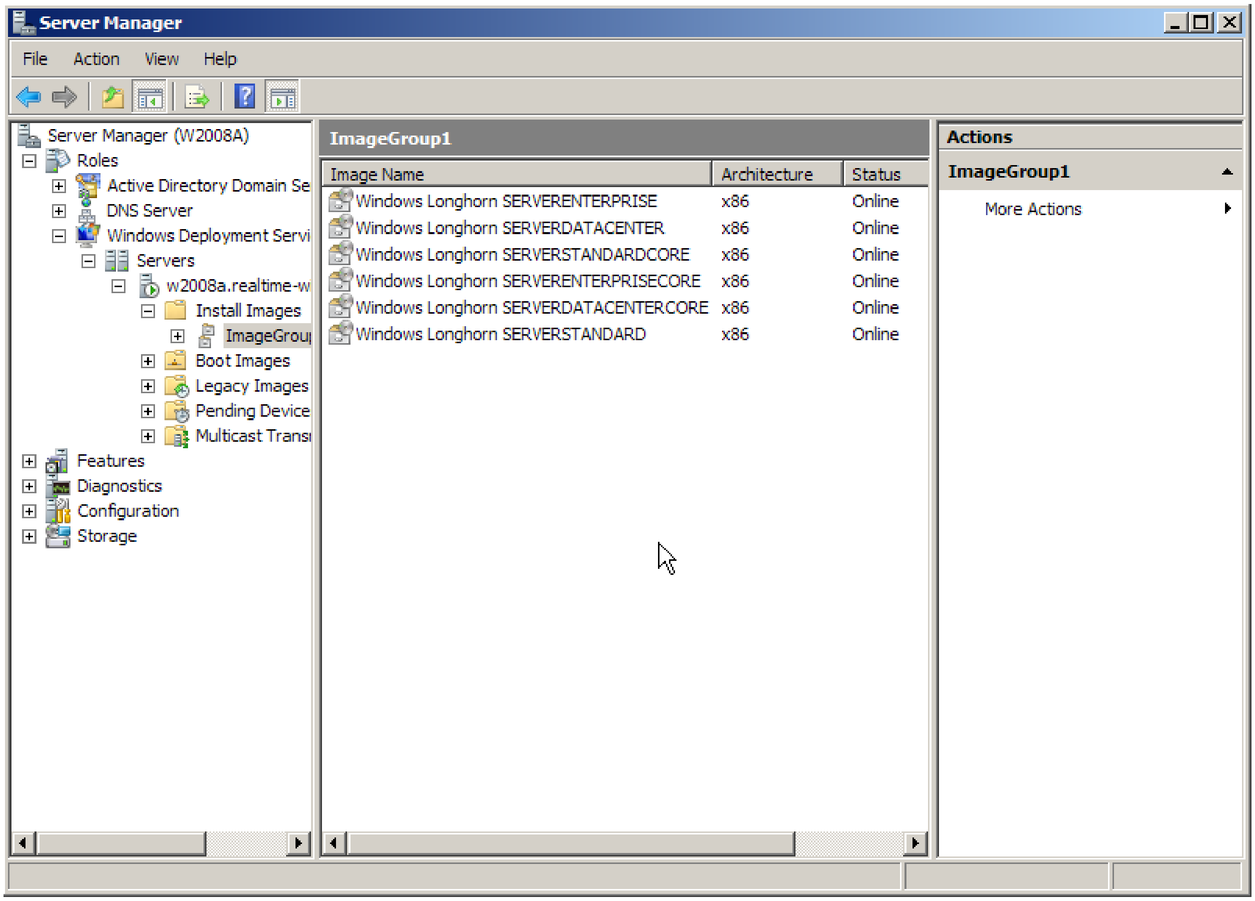


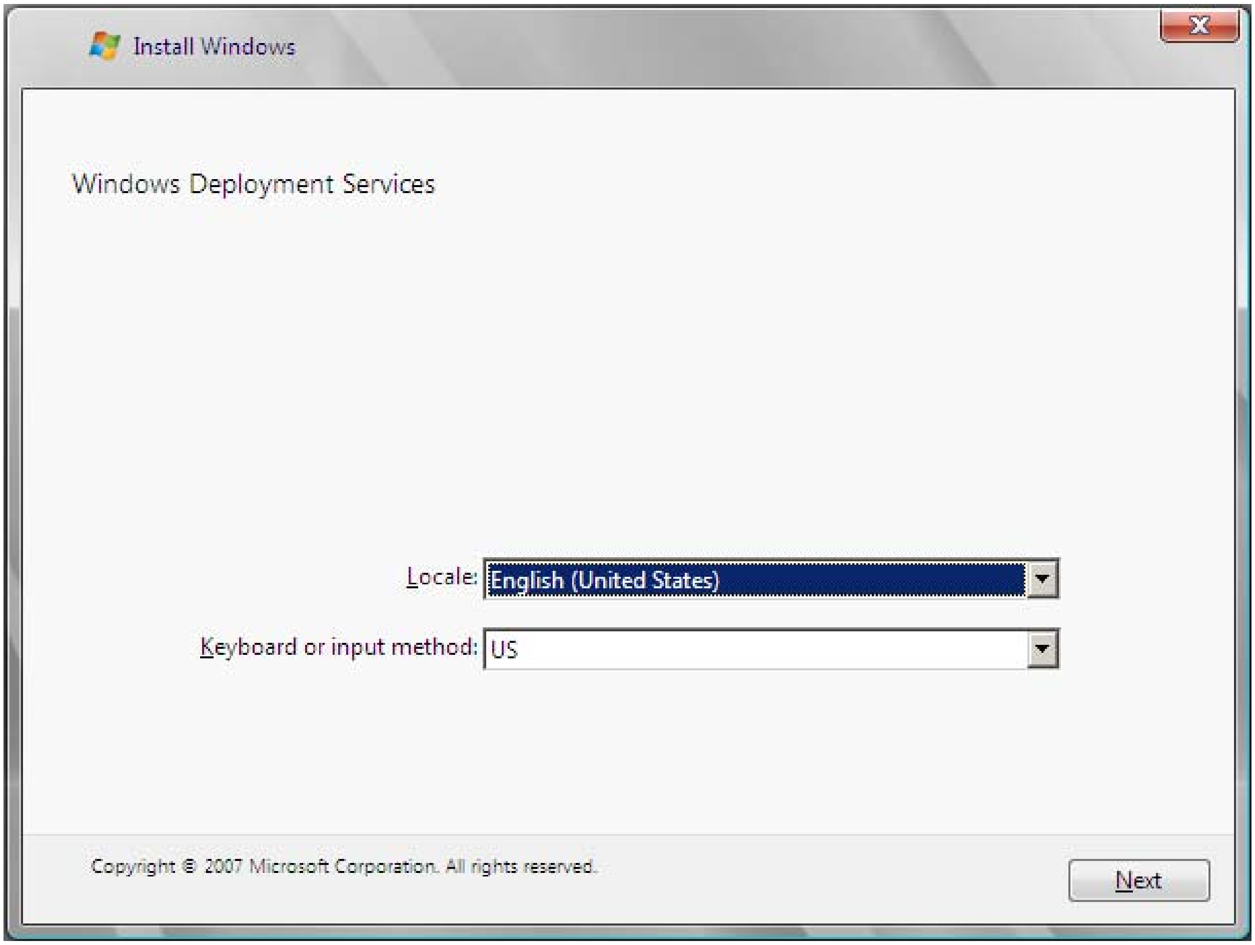
Image Name	Architecture	Status
Windows Longhorn SERVERENTERPRISE	x86	Online
Windows Longhorn SERVERDATACENTER	x86	Online
Windows Longhorn SERVERSTANDARDCORE	x86	Online
Windows Longhorn SERVERENTERPRISECORE	x86	Online
Windows Longhorn SERVERDATACENTERCORE	x86	Online
Windows Longhorn SERVERSTANDARD	x86	Online

Figure 1.6: Windows Deployment Services showing the standard installation images from the Server 2008 DVD.

The process to create a new over-the-network installation can be as easy as right-clicking *Multicast Transmissions* and selecting *Create Multicast Transmission*. You'll be prompted for a name for the transmission and to select the image you want to deploy as well as the type of multicast transmission you want to use. Two options are available for start criteria:

- *Auto-Cast.* With Auto-Cast transmissions the multicasting process starts immediately after the first client checks in. Any clients that arrive after the transmission has started will receive their missing pieces at the end of the transmission.
- *Scheduled-Cast.* With Scheduled-Cast transmissions you can configure either a number of clients or an amount of time at which you want the transmission to begin. This puts the transmission into a paused state until the correct number of clients checks in or the configured time arrives.

In order to use PXE booting to connect to the WDS server, the server's PXE response settings will need to be configured to listen for clients. This can be configured within the WDS properties pages. Clients who attempt a PXE boot should locate the WDS server and after completing the boot process you should see a screen similar to Figure 1.7. You'll see that this screen shows that the installation is being sourced from Windows Deployment Services.



Install Windows
Windows Deployment Services

Locale: English (United States)

Keyboard or input method: US

Copyright © 2007 Microsoft Corporation. All rights reserved.
Next

Figure 1.7: Once the target server has completed its PXE boot and located the WDS server, the screen above will show that the impending installation is sourced from the WDS server.

There are a number of additional options available within WDS for the configuration of PXE and the deployment of images, not the least of which is the capability to integrate a scripted installation as we discussed earlier in this chapter with an image-based installation. By right-clicking any image listed within the WDS interface and choosing *Properties*, it is possible to assign an unattended installation file with an image if desired. This allows for the combined use of both types of automated installations.

Summary

So now you know some of the best ways, both manual and automated, to get Microsoft's newest OS onto available hardware. Depending on the needs of your organization, you may choose the easy-to-start manual method or you may opt for one of the automated methods. Any of the automated methods involve a greater startup cost in terms of time to find the correct configurations you want to include as part of the automation. But once that initial work is done, it becomes easy and fast to replicate new servers across your network.

In our next chapter we'll start looking at the management of Server 2008. We'll take a hard look at Server Manager, where all the Roles, Role Services, and Features live. It is also the place where much of the rest of this guide will concern itself as we go through the new features and capabilities of Server 2008.

💣 The content of this chapter was written based on pre-release information, specifically the RC0 version of Microsoft Windows Server 2008.

Download Additional eBooks from Realtime Nexus!