

Introducing Windows Server 2003

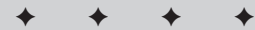
Windows Server 2003 is a complex operating system that is very different from Windows 2000 and earlier. In this book you will learn the fundamental differences between Windows Server 2003 and its predecessors. You will also gain valuable insight in the enhancements Microsoft has provided us which help with system stability, lower time needed for administration and help you understand why Windows Server 2003 has a massive performance gain over previous versions of Windows. This chapter introduces the product's architecture and provides guidelines to begin creating your strategy to adopt and support it.

Welcome to Windows Server 2003

After Windows NT 4.0 emerged in 1996, many organizations adopted the operating system, although Microsoft shipped service packs before the actual launch date for the software. This obviously begged the question — How stable can it be when service packs were created based on bugs detected by only Beta testers; surely the quantity and severity of bugs would increase now that it is gold? Many years later, people reluctantly made the leap to Windows 2000 Server so that they could reap the benefits of all the features that it contained, such as Active Directory. Unlike its predecessors, Windows Server 2003 enjoyed an extreme amount of production installations while still in Beta. This obvious support for a Beta operating system may seem rather odd considering the amount of service packs and security bulletins many have seen concerning the sibling operating systems such as Windows NT 4.0 and Windows 2000. That aside, the beta to this much-anticipated operating system has proven that it is very stable, provides a wealth of new features to aid the system administrator, and achieves an incredible performance gain over that of Windows 2000.

Windows Server 2003 doesn't give you one degree of separation from your customers right out of the box, but it does provide all the tools that you need to place yourself there such as the .NET Framework and IIS 6.0. Among these many tools are integration with the .NET Framework, improved Active Directory, improved DNS Server, and more configuration wizards than you can shake a stick at.

CHAPTER



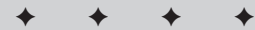
In This Chapter

Understanding the Windows Server 2003 architecture

Integrating Windows Server 2003

Understanding Windows Zero Administration (ZAW) and total cost of ownership

Windows Server 2003 collateral services



4 Part I ♦ Windows Server 2003 Architecture

In the past few decades, only the big companies could afford big iron mainframes from the likes of IBM and Digital Equipment Corp. Now, however, that firepower is in the hands of everyone with enough money to register a dot-com. We, the System Administrators, are fighting a network war in which the competition can obtain weaponry and firepower never before thought possible in computer science.

Viral warfare is surging beyond belief, with thousands of computer viruses released every month. Hackers are penetrating corporate networks all around the world. Business people are hiring geeks to bombard their competition with datagram attacks and denial-of-service bombs. And fraud is just around the next router. You need an operating system that can protect you at home and away from home, at every portal, and at every location. Today, no operating system competes with the vastness of Windows Server 2003.

Before you look into the architecture that supports Windows Server 2003, you need to understand that the Operating System is not all peaches and cream. Windows Server 2003 still has a few shortcomings and we discuss these where appropriate. We should mention here, however that a huge hurdle to overcome — besides the long-winded name, of course — is the learning curve that you face with this product. No version of Windows NT (in fact, no other server operating system) is as extensive, as deep, and as complex in many places as Windows Server 2003.

Although Windows Server 2003 was created to cater to the demand for operating systems that cost less to manage and own, yet also provide performance that can compete with operating systems that cost several times more, realizing the benefit is a long and costly journey for many. Windows Server 2003 is not the only culprit; Unix, NetWare, and the midrange systems also have a long way to go before they can truly claim to reduce the total cost of ownership — not only in terms of operating systems and software, but also in terms of all technology ownership and management.

You can decide what you want to do about Windows Server 2003 in two ways. (For a start, know that all your competitors are in the same boat. Whoever takes the plunge and adopts first is sure to be better off down the road.) You can ignore Windows Server 2003 for the next six to 12 months on the premise or misguided advice that you should wait for the OS to ship at least two service packs, or you can take the plunge now and deploy it in labs and development environments and be ready whenever the inevitable “we need it now” memo arrives.

Throughout this book, we suggest the latter approach. Put the OS into controlled development and pilot projects and deploy selective components that provide better services than what is available under NT/2000. You cannot learn the OS overnight, so learning as much as you can makes sense.

With ongoing systems to support, Windows Server 2003 typically requires a skilled network engineer or systems analyst to invest about six to eight months into the OS. And even after eight months of intense study, you still can't consider yourself an expert. Perhaps the best way to tackle the learning curve — besides spending a lot of money on courses, where end-to-end training runs into five figures per administrator, and without the cost of absence from work during the training — is to divide up the key service areas of the OS.

To a large extent, we divide this book along the following key service lines:

- ♦ Windows 2003 Architecture
- ♦ Active Directory Services
- ♦ Security Services

- ♦ Network Services
- ♦ Availability Services
- ♦ File and Print Services
- ♦ Application Services

This chapter deals with Windows 2003 architecture and introduces you to key services that fall under the *Zero Administration Windows (ZAW)* initiative.

Understanding the Windows Server 2003 Architecture

Making the effort to understand the architecture of an operating system is a lot like making the effort to understand how your car runs. Without knowing the details, you can still drive, and the vehicle gets you from A to B. But if something goes wrong, you take your car to the shop and the mechanic deals with it. The mechanic can tell you that you should have changed your oil earlier, that your tires needed balancing, or that your spark plugs are fouled. Had you known how the car operates, you would have taken more care of it and prevented excessive wear and tear. You could possibly have serviced it yourself.

The same can be said about an operating system, although it is a lot more complex than a car's engine. If you understand the various components of the kernel (the OS), the file system, and how the OS uses processors, memory, hardware, and so on, you are better at administering the machine.

Operating system modes

Windows 2003, built on Windows 2000 Server, is a modular, component-based operating system. All objects in the operating system expose interfaces that other objects and processes interact with to obtain functionality and services. These components work together to perform specific operating-system tasks.

The Windows 2003 architecture contains two major layers: *user mode* and *kernel mode*. The modes and the various subsystems are as shown in Figure 1-1.

Note

The system architecture is essentially the same across the Standard, Enterprise, Datacenter, and Web server editions.

User mode

The Windows 2003 user mode layer is typically an application-support layer for both Microsoft and third-party software, consisting of both environment and integral subsystems. It is the part of the operating system on which independent software vendors can make operating-system calls against published APIs and object-oriented components. All applications and services are installed into the user mode layer.

6 Part I ♦ Windows Server 2003 Architecture

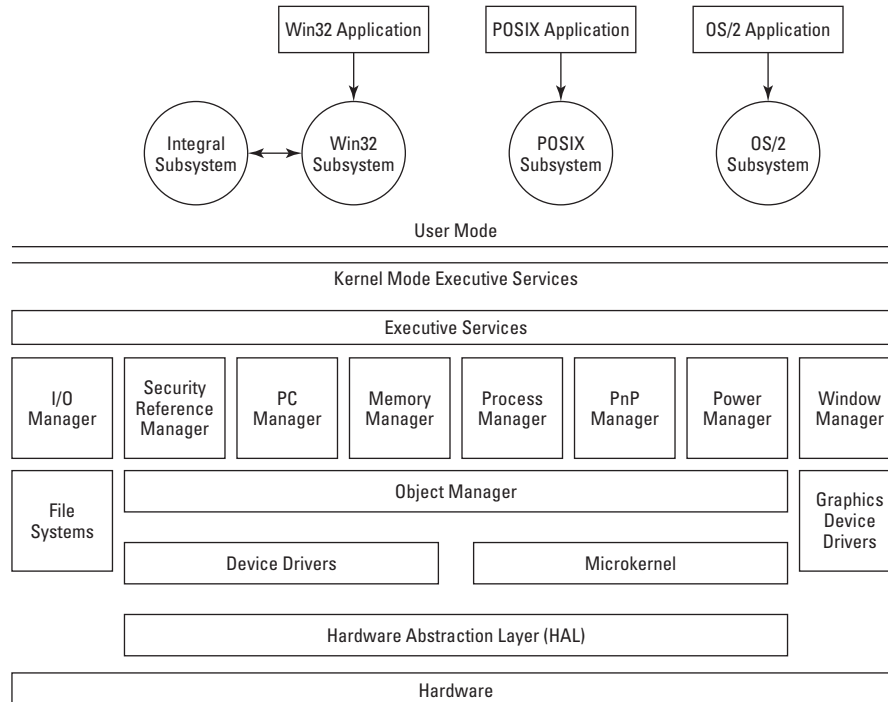


Figure 1-1: The Windows Server 2003 system architecture (simple).

Environment subsystems

The *environment subsystems* provide the capability to run applications that are written for various operating systems. The environment subsystems are designed to intercept the calls that applications make to a particular OS API and then to convert these calls into a format understood by Windows 2003. The converted API calls are then passed on to the operating-system components that need to deal with requests. The return codes or returned information that these applications depend on are then converted back to a format understood by the application.

These subsystems are not new in Windows 2003, and they have been greatly improved over the years compared to those of Windows NT. Reports in some cases indicate that the applications run better on Windows 2003 than they do on the operating systems they were intended for. Many applications are also more secure in Windows 2003. Windows 2003, for example, without affecting server stability, terminates DOS applications that would typically crash a machine just running DOS. Table 1-1 lists the Windows 2003 environment or application subsystems.

Table 1-1: Environment Subsystems

<i>Environment Subsystem</i>	<i>Purpose</i>
Windows 2003 Win32 (32-bit)	Supports Win32-based applications. This subsystem is also responsible for 16-bit Windows and DOS applications. All application I/O and GUI functionality is handled here. This subsystem has been greatly enhanced to support Terminal Services.
OS/2	Supports 16-bit OS/2 applications (mainly Microsoft OS/2).
POSIX	Supports POSIX-compliant applications (usually Unix).

The non-Win32 subsystems provide a basic support for non-Win32 legacy applications and no more. No real demand exists for either subsystem, and they have been maintained only to run the simplest of utilities that make very direct and POSIX- or OS/2-compliant function calls, usually in C. The POSIX subsystem, for example, caters to the likes of Unix utilities `vi` and `grep`.

The POSIX subsystem is not retained as a means, for example, of advanced integration of Unix and Windows 2003, such as for running a Unix shell on Windows 2003. For that level, you need to install Unix Services.

Several limitations and restrictions are imposed on non-Windows applications running on Windows 2003, by the underlying Operating System. This is demonstrated in the following list, which for the most part also includes user mode, Win32-based applications:

- ♦ **Software has no direct access to hardware.** In other words, if an application requests hard-disk space, it is barred from accessing hardware for such information. Instead, it accesses user mode objects that talk to kernel mode objects, which talk down the operating system stack to the Hardware Abstraction Layer. The information is then passed all the way up the stack into the interface. This processing is often known as *handoff processing*. The function in the Win32 code essentially gets a return value, and developers have no need to talk to the hardware. This is good for developers and the operating system. APIs that check the validity of the call protect the OS, and developers get exposed to a simple call-level interface, which typically requires only a single line of code and not 10,000 lines.
- ♦ **Software has no direct access to device drivers.** The philosophy outlined previously applies to device drivers as well. Hardware manufacturers build the drivers for Windows 2003 that access the hardware. The drivers, too, are prevented from going directly to the hardware, interfacing instead with abstraction objects provided by the device-driver APIs.
- ♦ **Software is restricted to an assigned address space in memory.** This constraint protects the operating system from rogue applications that would attempt to access whatever memory they can. This is impossible in Windows 2003, so an application can mess up only in the address space that it is assigned.
- ♦ **Windows 2003, as does Windows 2000, uses hard disk space as quasi-RAM.** Applications are oblivious to the source or type of memory; it is transparent to them. *Virtual memory* is a combination of all memory in the system and combines both physical memory in the machine and a swap file that is used to store information that cannot reside in hardware RAM.

8 Part I ♦ Windows Server 2003 Architecture

- ♦ **Applications in the user mode subsystems run as a lower-priority process than any services or routines running in the kernel mode.** This also means that they do not get preference for access to the CPU over kernel mode processes.

Integral subsystems

The *integral subsystems* are used to perform certain critical operating system functions. Table 1-2 lists these services.

Table 1-2: Integral Subsystems

<i>Integral Subsystem</i>	<i>Purpose</i>
Security subsystem	Performs the services related to user rights and access control to all network and OS objects defined or abstracted in some way in the OS. It also handles the logon requests and begins the logon authentication process.
Server service	This service is what makes Windows 2003 a network operating system. All network services are rooted in this service.
Workstation service	The service is similar in purpose to the server service. It is oriented more to user access of the network. (You can operate and even work at a machine that has this service disabled.)

You need to manage very little with respect to these systems. These services are accessible in the Service Control Manager and can be started and stopped manually.

Kernel mode

The Windows 2003 *kernel mode* is the layer that has access to system data and hardware. It comprises several components (refer to Figure 1-1).

The Windows 2003 Executive

The *Executive* is the collective noun for all executive services. It houses much of the I/O routines in the OS and performs the key object-management functions, especially security. The Executive also contains the systems services components (which are accessible to both OS modes) and the internal kernel mode routines (which are not accessible to any code running in user mode). The kernel mode components are as follows:

- ♦ **I/O Manager:** This component manages the input to and from the devices on the machine. In particular, it includes the following services:
 - **File System:** Translates file-system requests into device-specific calls.
 - **Device Drivers:** Manages the device drivers that directly access hardware.
 - **Cache Manager:** Buried in the I/O manager code, it manages I/O performance by caching disk reads. It also caches write and read requests and handles offline or background writes to the hardware.

- ♦ **Security Reference Monitor:** This component enforces security policies on the computer.
- ♦ **Interprocess Communication Manager (IPC):** This component makes its presence felt in many places in the OS. It is essentially responsible for communications between client and server processes. It comprises the Local Procedure Call (LPC) facility, which manages communications between clients and server processes that exist on the same computer, and the Remote Procedure Call (RPC) facility, which manages communications between clients and servers on separate machines.
- ♦ **Memory Manager or Virtual Memory Manager (VMM):** This component manages virtual memory. It provides a virtual address space for each process that manifests and protects that space to maintain system integrity. It also controls the demand for access to the hard disk for virtual RAM, which is known as paging. (See the section “Windows 2003 memory management,” later in this chapter, for details.)
- ♦ **Process Manager:** This component creates and terminates processes and threads that are spawned by both systems services and applications.
- ♦ **Plug and Play Manager:** This component is new to Windows 2003. It provides the plug and play services and communicates with the various device drivers for configuration and services related to the hardware.
- ♦ **Power Manager:** This component controls the management of power in the system. It works with the various power-management APIs and manages events related to power-management requests.
- ♦ **Window Manager and Graphical Device Interface (GDI):** The driver, `Win32K.sys`, combines the services of both components and manages the display system, as follows:
 - **Window Manager:** This component manages screen output and window displays. It also handles I/O data from the mouse and keyboard.
 - **GDI:** This component, the hardest interface to code against and keep supplied with memory in the days of Win16, handles the drawing and manipulation of graphics on-screen and interfaces with components that hand off these objects to printer objects and other graphics rendering devices.
- ♦ **Object Manager:** This engine manages the system objects. It creates them, manages them, and deletes them after they are no longer needed, and it manages the resources, such as memory, that need to be allocated to them.

In addition to these services (and as indicated in Figure 1-1), three other central core components complete the makeup of the kernel mode: the *Device Drivers* component, the *Microkernel*, and the *Hardware Abstraction Layer (HAL)*.

Device Drivers

This component simply translates driver calls into the actual routines that manipulate the hardware.

Microkernel

This component is the core of the operating system. (Some regard it as the operating system itself, with everything else just services.) It manages process threads that are spawned to the microprocessor, thread scheduling, multitasking, and so on. The Windows 2003 Microkernel is preemptive, which means, essentially, that threads can be interrupted or rescheduled.

Hardware Abstraction Layer

The *Hardware Abstraction Layer*, or *HAL*, essentially hides the hardware interface details for the other services and components. In other words, it is an abstraction layer above the actual hardware, and all calls to the hardware are made through the HAL. The HAL contains the necessary hardware code that handles hardware-specific I/O interfaces, hardware interrupts, and so on. This layer is also responsible for both the Intel-specific and Alpha-specific support that enables a single executive to run on either processor.

Windows 2003 processing architecture

Windows Server 2003 is built around a *symmetric multiprocessing (SMP)* architecture. This means that, first, the operating system can operate on multiple CPUs, and, second, it can make the CPUs available to all processes as needed. In other words, if one CPU is completely occupied, additional threads spawned by the applications or services can be processed on other available CPUs.

Windows 2003 combines its multitasking and multithreading capabilities with its SMP capabilities. And if the threads waiting for execution are backed up, the OS schedules the processors to pick up the waiting threads. The thread execution load is evenly allocated to the available CPUs. Symmetric multiprocessing thus ensures that the operating system uses all available processor resources, which naturally speeds up processing time.

Windows Server 2003 Standard edition supports four-way (four CPUs) symmetric multiprocessing. The Enterprise server edition supports eight-way SMP, and Datacenter server supports up to 32-way SMP, while the Web edition limits you to a maximum of 2 CPUs. If you have the muscle, you can get the code from Microsoft, under hefty contract, to compile the OS to your SMP specifications.

Windows 2003 memory management

Windows 2003's handling of memory is almost identical to that of Windows 2000 Server in that it has been vastly improved over Windows NT 4.0. It consists of a memory model based on a flat, linear, albeit still 32-bit, address space. Two types of memory are used in the Windows 2003 operating system. First is *physical* memory, which includes the memory in the RAM chips installed on the system motherboards, typically in the form of SDRam, DDRam or RAMBus RAM. Second is *virtual* memory, which is a combination of all memory in the system and how it is made available to the OS.

The *Virtual Memory Manager (VMM)* is used to manage system memory. It manages and combines all physical memory in a system in such a way that applications and the operating system have more memory available to them than is provided in the actual RAM chips installed in the system.

The VMM also protects the memory resources by providing a barrier that prevents one process from violating the memory address space of another process, a key problem of the older operating systems such as DOS and earlier versions of Windows.

Every memory byte, whether physical or virtual, is represented by a unique address. Physical RAM has limitations because Windows 2003 can address the memory only according to the amount of physical RAM in the system. But virtual addressing is another story. Windows 2003 can support up to 2GB of RAM in the Web edition, 4GB with Windows Server 2003 standard edition, 64GB in the Enterprise edition and 512GB with Datacenter edition running with 64-bit processors.

The VMM manages the memory and has the following two major functions:

- ♦ The VMM maintains a memory-mapped table that can keep track of the list of virtual addresses assigned to each process. And it coordinates where the actual data mapped to the addresses resides. In other words, it acts as a translator service, mapping virtual memory to physical memory. This function is transparent to the applications, which continue to behave as if they have access to physical memory.
- ♦ If RAM is maxed out, the VMM moves the memory contents to the hard disk, as and when required. This process is known as *paging*.

Thus Windows 2003 basically has access to a 4GB address space, although the space is virtual and can be made up of both RAM and hard-disk space. Although you talk about a 4GB address space, this space is actually relative to how the system uses memory. In actual fact, the address space available to applications is only 2GB and is even less than that, because the 2GB assignment is shared by all processes running in user mode, and the other 2GB assignment is reserved for kernel mode threads.

You talk about an *upper* and a *lower portion* of the 4GB space, both containing 2GB addressing. The upper portion is reserved for kernel mode processes only, and the lower space is reserved for both user mode and kernel mode processes. The upper portion also reserves certain lower regions of its address space that are directly mapped to hardware.

The lower portion is also maintained in paging pools. It has a nonpaged pool and a paged pool. The paged pool can be swapped out to disk and is usually assigned to applications. The nonpaged pool must remain in physical RAM. The size of each page is 4K.

Paging in-depth

Paging is the process of moving data in and out of physical memory. If the physical memory pool becomes full and Windows needs more, the VMM reallocates data that is not needed in the physical memory out to the disk in a repository known as the *page file*.

Each process is assigned address space in pages that are either identified as *valid* or *invalid pages*. The valid pages are located in the physical memory and are available to the application. The invalid pages are not available to any application. The invalid pages are stored on disk.

Whenever applications need access to data that has since been moved to offline memory in an invalid page, the system acknowledges this need in what is known as a *page fault*. A page fault process is similar to a thread of execution that takes a different route in terms of routines after it encounters an error or exception. In this case, the fault is handled intentionally, and the VMM “traps” the fault, accesses the data in the page file that relates to it, and restores it to RAM. Other data that is now no longer needed is bumped out and sent offline to disk. This is one of the reasons why fast and reliable hard disks are recommended in data- and memory-intensive applications.

The VMM performs the following series of housekeeping chores as part of the paging routines:

- ♦ The VMM manages the data in the page file on disk on a first-in, first-out basis. In other words, data that has been on disk the longest is the first to make it back to physical memory after RAM frees up. The VMM continues to move the data back to RAM as long as RAM keeps freeing up and until no data is left in the page file. The data that the VMM keeps tabs on in this fashion is known as the *working set*.

12 Part I ♦ Windows Server 2003 Architecture

- ♦ The VMM performs what is known as *fetching* as it brings back data from the page file. In addition, the VMM also performs what is known as *page file clustering*. Page file clustering means that, as the VMM fetches, it also brings back some of the surrounding data in the page file, on the premise that data immediately before and after the required data may be needed in the next instant as well. This process speeds up data I/O from the page file.
- ♦ The VMM is intelligent enough to work out that, if no space is present in RAM to place fetched data, it must first move out other recent data to the page file before attempting to place the fetched data back into faster RAM.

The parameters in which the VMM operates and the factors, such as the size of the page file, can be managed and controlled by you. We discuss this further in the performance management and troubleshooting techniques discussed in Chapter 24.

The Zero Administration Windows Initiative

The *Zero Administration Windows (ZAW)* initiative was a bold move to reduce the Total Cost of Ownership (TCO) and administration of Windows networks or environments. Though this was introduced in Windows 2000, take one look at the size of the Windows 2003 Resource Kit and decide whether your administrative burden has been reduced in any way. As we did, you were probably wondering just whether ZAW was a figment of Microsoft's imagination.

ZAW, however, is very much alive and apparent in Windows 2003. Consider the following to avoid having a heart attack: Understand and accept that you have a learning curve to climb in comprehending Windows 2003 in general and Windows Server 2003 in particular. The ZAW technologies, which have been added by the boatload to Windows 2003, do in fact reduce administration. We know what you are thinking: "How many nights must I stay up armed with pizzas and dozens of cans of soda before I figure out how this all works?" Here's the comforting statement from our team, which spent about 5,000 hours trying to make heads or tails of Windows Server 2003 for this book: ZAW is here and now in Windows 2003, but you must put it together now to achieve the long-term benefits.

After you put together all the pieces to your own satisfaction and understand how the new technologies come together, you begin to see ZAW emerge. Take it from us that Windows 2003 is the first operating system ever that is truly client/server. It can also be thin-client/server, fat-client/server (some call it rich-client/server), client/thin-server, and client/fat-server. Windows 2003 can also be client-client and server-server in many different variations.

When we say "truly client/server," we mean that the client operating-system processes, no matter whether they consist of a remote workstation running on Windows XP Professional or a server operating system, are very tightly integrated and meshed with the server operating processes and features. This is true regardless of the physical location of the server. This is not only apparent in the capability of a user to log on to any computer running Windows 2003 and find his desktop exactly as he left it and with access to all resources required and used previously, but also in the transparent availability of these resources. This is made possible by several key technologies that we discuss in the following sections, the first of which is Active Directory.

Active Directory

Active Directory is extensively discussed in Chapter 2 and in Part III in this book, so we do not go into too much detail other than to say that all the configuration and preferences relating to the services that we discuss in this section are stored in Active Directory. If you face a learning

curve, Active Directory is where it starts. Unfortunately for larger businesses, Active Directory involves more than what is apparent to small businesses.

Active Directory is very much the hub of the network. Without Active Directory, you really do not have a Windows 2003 network. Though you will still run across quirks here and there, the number of tools that have been added since the Windows 2000 implementation is impressive. As time progresses, I'm sure it will grow ever more powerful with a whole new set of tools and utilities to help the Server Administrator.

Microsoft Management Console

The *Microsoft Management Console (MMC)* was deployed on Windows NT to support BackOffice applications such as Exchange, IIS, and SNA Server (now known as Host Integration Server). In Windows 2003, the MMC is used system-wide for managing just about everything on Windows Server 2003. A management module, known as a *snap-in*, exists or is created for each service. Each snap-in offers peculiar features and choices, depending on the service targeted for configuration.



We cover the MMC in Chapter 7.

Server and client in unison: IntelliMirror

Several technologies work to improve the integration between client and server. The *IntelliMirror* is a group of technologies that enables a user's settings, preferences, applications, and security to follow that user to other computers on the network. IntelliMirror also extends to laptops running Windows XP Professional and enables the user to maintain a disconnected state that is automatically restored seamlessly whenever the user reconnects to the network.

Group Policy, discussed in Chapter 14, is mostly responsible for the mirror, and of course, all configuration is stored in the directory. Clients access the data from the directory as needed. IntelliMirror is really an umbrella term that refers to the following technologies and features:

- ♦ **Offline folders:** Offline folder technology enables you to obtain a copy of a file that is usually stored on the server and to work with that file after you're disconnected from the network. If you disconnect from the server, the file that you were working on is managed as if it is still residing on the server. For all intents and purposes, your application thinks that it *is* still connected to the server. You save normally as if saving to the network, but instead, you are actually saving to an offline resource that is a mirror of the file and the folder on the server. After you reconnect to the network, the file is synchronized again, and the latest changes to the file are saved to the server's copy.
- ♦ **Folder redirection:** This is another IntelliMirror feature that makes a folder redundant. If the server disconnects from you but you are still connected to the network, the next time that you save your file, you are redirected to another copy of the folder residing on another server.
- ♦ **Roaming profiles:** These were inherited from Windows NT profile management philosophy, but they are much more sophisticated under Windows 2003. The idea is that your user profile follows you wherever you go.
- ♦ **Remote Installation Services (RIS):** These services are provided by several components and services that makes possible remotely installing Windows XP Professional and Windows XP Home to desktops and notebook computers.

14 Part I ♦ Windows Server 2003 Architecture

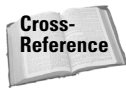
- ♦ **Application publishing and software installation and maintenance:** By using Active Directory services, you can remove and install software remotely to users' workstations.

A lot of overlap does, of course, exist between the IntelliMirror-cum-Active Directory services and *System Management Server (SMS)*. SMS manages the deployment of software over multiple sites as part of its complex change-control and change-management services. It is also an extensive scheduling and inventory management system. SMS is a BackOffice product worthy of description within its own book covers, which is why we do not cover SMS in this book.

Group Policy

Managing Windows Networks and Windows Server 2003 just got a lot easier with the new *Group Policy* technology. Group Policy is used to manage user settings, security, domain administration settings, desktop configurations, and more. In short, most of the workspace is managed through Group Policy.

Group Policy is applied at all levels of the enterprise in Active Directory, from domains down to organizational units and so on. The tool used for the job is the *Group Policy Editor (GPE)*. GPE enables you to create objects that are associated or referenced to *organizational units (OUs)* in Active Directory. *Group Policy Objects (GPOs)* can be secured with NTFS permissions in the same fashion as files and folders.



Group Policy is discussed in detail in Chapter 14.

Availability services

Availability, as it relates to information systems, is the effort to keep systems and IS services available to users and processes all the time. We usually talk in terms of 24×7 availability, which, if possible, would be 100 percent availability. But 100 percent is possible only in a perfect world, so the target that we strive for is 99.9 percent. Few systems are capable of 99.9 percent availability. Trust us; we work in an environment of mainframes, Unix servers, NT clusters, AS/400s, and many other high-end servers, and no one can claim to have achieved perfect availability.

Availability is very important for companies that have service-level agreements with their customers. *Service Level Agreement (SLA)* is an IT/IS term used to refer to the availability of host systems that customers depend on for service from their suppliers. But SL is no longer applicable only to host systems or supplier-customer relationships. It is now important to all entities that depend on systems being available "all the time." Of particular importance are the Internet services because of the increased dependence that e-commerce companies place on Windows Server 2003. If your server goes offline, that situation can result in mounting losses that can easily be calculated and related in hard-dollar amounts. Taking a server down on an Internet site is like closing the physical doors to the store, which would send your customers to the competition. Cyberstores cannot afford that.

Every server administrator thus needs to zero in on this subject with the determination to maintain high availability of servers and services at all times. In fact, all services and components in Windows 2003 should be listed on an availability chart or a risk-assessment chart. The following list enumerates several areas that have been built by Microsoft with high availability criteria on the agenda:

- ♦ Bouncing server syndrome
- ♦ Clustering and server redundancy

- ♦ Storage redundancy
- ♦ Disaster recovery
- ♦ Security

Bouncing server syndrome

We are not sure who first used the word *bounce* to refer to the act of rebooting a server. But the term has caused many good laughs. In mid-1999, we were joined by a good-natured but overly serious VMS administrator who took over the management of a nationwide network of DEC VMX machines. One day, he came over to network administration and told us that he had just received the strangest call from one of the remote centers. “They say that they need me to bounce the Coral Gables VMX, but this is the first time I have heard such a term.” And we replied: “Yes, you need to pick it up and drop it on the floor . . . do you need help?”

From that day on, our VMX administrator would often tease us about the number of times that you must “bounce” NT (a lot more than a VMX machine). Another term that is often used in IT circles is *IPL*, which stands for *Initial Program Load* (an operating-system restart) and which is rooted in legacy host systems and midrange talk. All systems require you to reboot, bounce, or IPL. Availability is rated on how often you must reboot.

Windows NT has a horrible availability rating. Just about any configuration change that you make demands a reboot. If you have administered NT for any number of years, you know that you just need to open the network configurations utilities and look at the settings and be told that you must reboot. Often, we would just ignore the warning and hit Cancel. Far too many changes, however, require you to reboot an NT server. At times, we wondered whether just breathing on the monitor would require a bounce.

Microsoft has improved the reboot record in the Windows 2003 kernel tremendously, both for new services and situations where applications and services crash. Improvement is especially noticeable in areas where you typically make a lot of changes, such as the network configuration and so on. Static IP address changes are, for example, immediate, as are reconfiguring network interface cards (NICs) and the like. A number of areas still need to be improved. Installing software (such as service packs) is a good example. Although a service pack reboot may be forgiven, reboots while installing new user applications on the server devoted to terminal users is not. A reboot after promoting a domain controller, however, is understandable. Still, we hope that later versions of Windows Server 2003 require even fewer reboots.

Clustering and server redundancy

Windows 2003 Enterprise Server now has clustering services built in, which is a big improvement over the Cluster Server product that was shipped as an add-on to Windows NT and Windows 2000. *Clustering* is a form of fault tolerance that enables users connected to one server to automatically connect to another server if the former server fails. This situation is technically known as a *failover* in clustering terms. (We have not dealt with clustering in this book because our scope of coverage is Windows Server 2003.)

Clustering, however, is not only applicable to redundancy, but also to *load balancing*, and particularly network load balancing, which clusters network resources. With technologies such as IntelliMirror and Group Policy, your users need never know which server in a farm of 50 machines is currently servicing their needs. The distributed files system, folder redirection, offline files and folders, and more all play their part in clustering and availability.

Storage redundancy

Storage services in Windows 2003 play a critical part in availability. Windows 2003 supports all available levels of disk arrays. The distributed file systems and NTFS 5.0 have several key features that support the high availability initiative.

Disaster recovery

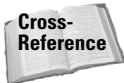
Disaster recovery is managed by using Windows 2003 remote and removable storage services to maintain reliable backup sets. The *System Recovery Console* is also a new feature that enables you to boot to an NTFS-supported command line that enables you to access NTFS volumes. In addition, Windows 2003 also boots to a menu of “safe mode” choices in the event of serious system instability and gives you the opportunity to provide a description of why the system went down for logging purposes.

Security

Security services are critical to Windows 2003. In our opinion, you cannot have enough tools to protect the network. We discuss this topic further in the following section, in depth in Chapter 3, and in places in this book where security configuration is required. You would do well to take stock of the thousands of hacker sites on the Internet and join as many security groups as possible. The age of e-terrorism is upon us, and Windows Server 2003s are among the main targets.

Distributed Security

Microsoft has loaded Windows 2003 with security services unparalleled in any other operating system. Just about every port can be encrypted and protected in some way or form . . . all the way to 128-bit keys, which is now the world-wide encryption level in the wake of the e-business and e-commerce phenomenon. Windows 2003 also supports MIT’s Kerberos version 5.0 protocol, now a de facto Internet standard that provides authentication of network logons by using symmetric key encryption and digital certificates, and the *Single Sign-On* initiative (SSO).



We devote a whole chapter to a discussion of Windows 2003 security (Chapter 3).

Interoperation and integration services

A homogenous network in a large company is a pipe dream in our opinion, and Microsoft probably concurs despite evangelizing to the contrary before the advent of the World Wide Web. All large companies deploy a hodge-podge of interconnected systems running on different platforms. The aim is to reduce the number of different systems as much as possible. At best, you can get down to supporting just Windows 2000, Windows Server 2003, Unix, AS/400, and possibly some legacy stuff. NetWare, once king of the LAN, still manages a foot in the door (especially with attorneys and high schools), and OS/2 died for most companies on December 31, 1999, a legend of the 20th century.

Microsoft has invested heavily in Unix integration, which is very strong. Microsoft has acquired new technology that, after it’s deployed into Windows 2003, all but turns Windows 2003 into a Unix box. (Okay . . . we are exaggerating just a little.) These services existed with Windows 2000 but many of you may not have heard of them because they were very prone to bugs and used very infrequently. Not only does Windows Server 2003 communicate natively with Unix by using TCP/IP, but it also runs the Unix shell, giving administrators who work in both systems the best of both Windows and Unix worlds.

Hardware support and plug and play

Do not try to count the number of drivers that have been made available to Windows 2003; you'd be at it for hours. Remember the bad old days when critics berated the first version of NT for not having support for a lot of the hardware that was being shipped on Windows 95? Now the hardware teams for Windows 2003 and Windows 98 share the device-driver testing for all operating systems because they are the same drivers. More than 2,000 printer drivers, for example, already ship with Windows 2003, as well as many other drivers that are not even available to Windows 2000.

Did we just say that Windows 2003 and Windows 98 device drivers are the same? Thanks to the long-awaited *Windows Driver Model (WDM)* initiative, all device drivers created for Windows 98 and its successor that conform to the WDM can support hardware in both operating systems because they are the same drivers.

Hardware vendors can also bring their products to market a lot faster because, if they build device drivers to the WDM, they need to add only the final code to an extensive device-driver template that has already been written by Microsoft.

The WDM also caters to media streaming and image architecture, enabling support for a wide variety of scanners, plotters, digital cameras, image-capturing equipment, and more.

Storage and File System Services

Windows 2003 Storage and File Systems Services has been vastly improved over that of 2000. Many new features were added; some of them are mind-blowing, to say the least. The following list highlights the key services that you can expect to affect your administration routines and how you view both file and storage services in the future:

- ♦ Disk Administrator
- ♦ Removable Storage
- ♦ Remote Storage
- ♦ Microsoft Dfs
- ♦ NTFS 5.0

Disk Administrator

Windows Server 2003, like Windows 2000 supports dynamic disks, which enable you to merge volumes or extend them across multiple disks. Software RAID (Redundant Array of Independent Disks) support is built in to a much-improved disk-management utility that is now an MMC snap-in. You have full control over RAID volumes and can manage the volumes for the most part without needing to reboot the server. The MMC enables you to connect to any remote server and manage its hard disk resources as if they were local. This is hard-disk nirvana all the way.

Removable Storage

Removable Storage enables you to manage removable-storage media, such as tape drives and other removable media, to such an extent that doing so is almost an art form. Tape drives are now no longer managed as part of the backup and restore services. NTBackup (`ntbackup.exe`) is still there, but in a much-improved form (and labeled as Microsoft Backup which is a scaled down version of Veritas Backup Exec) . . . although it's still not quite as professional as we

would have liked. Removable Storage enables you to create media pools that are assigned to various backup regimens and removable storage routines.

Remote Storage

Remote Storage is a service that takes files that are no longer accessed or used by the user or local process and moves them to removable storage media. An active file marker is placed in the place of the file with the data. This service thus frees up hard-disk space on demand by looking at which files can be relocated. If the user requires the files, they are returned from remote storage. The access may be slow at first, depending on the location or the technology used by the remote and removable storage service.

Microsoft Dfs

Microsoft Dfs (named *Dfs* so as not to be confused with the DFS standard) works a lot like the Unix NFS, in which folders and the directory tree are local to the network and not to any particular server. In other words, you can locate the files and folders that you need without needing to browse to any particular server or needing to map to a network drive from a workstation, as is the case with Windows NT.

NTFS 5.0

The Windows *NTFS 5.0* file system has been given extensive performance boost enhancements and supports mounted volumes, encryption, a folder hierarchy that extends multiple servers, and so on. Perhaps the most notable service that sits between the storage services and NTFS 5.0 is enforceable disk quotas which was introduced with Windows 2000. Disk quotas are set on a per-volume basis and enable you to warn and deny access to hard-disk space as or before the user reaches the quota level set.

Internet Services

Internet Information Server 6.0 is part of Windows Server 2003. You now have extended support for the SMTP and NNTP protocols besides FTP. In other words, whenever running the Internet services, the server also behaves as a mail or news host, enabling relay support and more.

The fully integrated support for IIS enables you to host multiple Web sites on one server with only one IP address. Each site can also have its own user-related databases, which thus supports multiple DNS domains.

Communication Services

What is a network without the capability to communicate? Windows 2003 is loaded with new and improved communications services. The Internet communications services have been highly optimized and advanced with e-mail, chat capability, and the wholesale support for the NNTP (Network News Transport Protocol) protocol, which is the pipeline to the newsgroups. For starters, Outlook Express, which is an Internet e-mail client with extensive support for security and attachments, is built into all versions of Windows 2003.

You also find new support for *Virtual Private Networking (VPN)* services, enabling connection to the enterprise network from remote networks such as the Internet. These services fully support both the *Point-to-Point Tunneling Protocol (PPTP)* and the *Layer Two Protocol (L2TP)*.

Terminal Services

Windows NT enabled a single interactive session from the console, usually someone sitting directly in front of the monitor attached to the server. If you needed remote access to the server, you would usually need to use pcANYWHERE or CarbonCopy. This single interactive session is now completely obsolete with Windows Server 2003. *Terminal Services*, inherited from the Windows 2000 Terminal Server, is built into all versions of Windows Server 2003. You can, without license restrictions, connect remotely to the machine using the Remote Desktop feature.

Terminal Services enables a user to establish a session on the server from a dumb terminal or with terminal-emulation software running on just about any device that can connect to the network. The model, known as thin-client/server computing, works just the same as the mainframe model of a fat server to which many terminals can attach and obtain interactive sessions. The only difference here is that the “frame” sends back the Windows 2003 desktop to the user’s terminal and not some arcane and bland collection of green characters, typical of midrange and mainframe systems.

The Windows 2003 kernel now includes a highly modified Win32 subsystem to support interactive sessions running in the server’s allocated process space. Looking at it from dizzying heights, you would essentially take the Win32 subsystem and “clone” it for each user attaching to the server and running a workstation service. Have a look back at Figure 1-1. Now just make numerous copies of the Win32 subsystem, and you have Terminal Services in action.

For the most part, Terminal Services are not managed separately from the rest of the server. You have a few user-specific settings to manage related to sessions and session activity. Terminal Services are extremely important and are expected to garner wide adoption.

Summary

This chapter serves as an introduction to Windows 2003. First, you look at the Windows 2003 System architecture. It is the same architecture as Windows 2000 and the same foundation but with some dramatic changes.

Some major paradigm shifts are demonstrated in Windows 2003. The most significant is the shift back to terminal-mainframe environments. The mainframe, however, is Windows 2003, and the terminal gives you the Windows 2003 desktop and not a screen chock full of green characters and a blinking cursor.

We also introduce you to some of the key additions to the operating system. Almost all the topics that we introduce in this chapter are extensively covered in depth in the remaining 29 chapters.

We also warn that the learning curve you are about to face is steep. Experienced Windows 2000 administrators have less of a climb than newcomers, but much is new to Windows 2000 administrators as well, such as the .NET Framework, Active Directory changes, and so on.

You have no time to waste in getting down to learning Windows Server 2003 and starting on the road to deployment and rollout.



