SYBEX Sample Chapter

Mastering™ Windows® 2000 Server

by Mark Minasi, Christa Anderson, Brian M. Smith and Doug Toombs

Chapter 1: Windows 2000 Server Overview

After years of talk about "Cairo" (the Microsoft code name for their "ultimate" server software) and even more years of work, Microsoft has finally shipped Windows 2000. After training us to expect roughly annual releases of new versions of NT—NT 3.1 shipped in 1993, 3.5 in 1994, 3.51 in 1995, and 4 in 1996—NT 5 finally arrived, but it was considerably later than a year after the release of NT 4. Furthermore, NT 5 arrived with a new name: Windows 2000. But the name's not all that's new.

So what took so long? Was it worth the wait? For many, the answer will be "yes." Much of NT's foundation—the internal kernel structure, how drivers are designed, how Windows 2000 multitasks—hasn't changed all that terribly much from NT 4, but network professionals really don't see that part of NT. Instead, we network types will notice that the *above-ground* structures, the tools built atop the foundation, are so different as to render Windows 2000 Server almost unrecognizable as a descendant of NT 3.*x* and 4.*x*. For comparison's sake, and to extend the structural metaphor, think of using Windows NT 3.1 Advanced Server as renting a room in someone's basement, using NT 4 as renting a 2-bedroom apartment, and using Windows 2000 Server as living in Bill Gates's new mansion on Lake Washington: more rooms than anyone can count all filled with new and wonderful electronic gadgets.

In the mansion, many of the things that you know from the basement room are unchanged—the electricity comes out of sockets in the wall, the pipes are copper or PVC, bathrooms have sinks and commodes in them—but there's so much more of it all, as well as so many new things, both useful ("Hey, cool, a garden, and automatic sprinklers for it!") and of debatable value ("What does this bidet thing do, anyway?"). That's not to say that NT's underpinnings will never change, not at all—the next (and still-unnamed) version of NT will go a step further, digging up NT's 32-bit foundation and replacing it with a 64-bit one.

The main point, however, is this: If you're an NT network administrator, be prepared for culture shock. The difference between NT 4 and Windows 2000 is at least 10 times as great as the difference between NT 3.1 and NT 4. And if you've never worked with NT in any flavor, be prepared to find Windows 2000 both delightful and frustrating—as is the case with most Microsoft software.

It would be somewhat shortsighted of me to simply say, "Here are the new features you'll find in Windows 2000," and then to just dump the features—it sort of misses the forest for the trees. So let me start off by briefly discussing the big picture and what Microsoft's trying to accomplish; then I'll move along to those new features and, finally, take a look at a few of Windows 2000's shortcomings.

Microsoft's Overall Goals for Windows 2000

The changes in Windows 2000 from NT 4 are quite significant, but they were long in coming. What was the wait all about?

Make NT an Enterprise OS

Microsoft wants your company to shut off its mainframes and do your firm's work on big servers running NT. That's why there is a version of Windows 2000 Server called Datacenter Server. Microsoft is also hoping that "enterprise" customers will exploit new Windows 2000 Server facilities such as Active Directory and Microsoft Application Server (nee MTS) and COM+ to write gobs of new and hardware-hungry distributed applications. Before they can accomplish that, however, they need to clear three hurdles: reliability, availability, and scalability.

NT Must Be More Reliable

Since their appearance in the late '70s, microcomputer-based network operating systems have been seen as fundamentally different from "big-system" OSes like IBM's MVS and OS/400, Compaq's Open VMS, and the myriad flavors of Unix. PC-based network operating systems weren't exactly seen as toys, but neither were they seen as something that one would base one's business on, if one's business was truly critical. For example, it's hard to imagine the New York Stock Exchange announcing that they'd decided to get rid of their current trading system and to replace it with a NetWare 4.1 or NT 4-based client-server system. PC-based stuff just wasn't (and largely still isn't) seen as sufficiently reliable yet to take on the big guys.

Nor is that an unfair assessment. Most of us would be a bit uncomfortable about discovering in midflight that the state-of-the-art airliner taking us across the Pacific was run by NT, or that the Social Security Administration had decided to dump their old mainframe-based software in favor of a Lotus Notes–based system running atop NT. Years ago, many firms discovered that NT servers crashed far less often if rebooted weekly; it's hard to imagine running a heart-and-lung machine on something like that.

But Microsoft wants to shed that image. They want very much to build an OS that is sufficiently industrial-strength in reliability so that one day it wouldn't be silly to suggest that AT&T's long distance network could run atop some future version of NT, Windows 2000-something. With Windows 2000, Microsoft believes that they've taken some steps in that direction.

NT Must Be More Available

A server being rebooted to change some parameters is just as down as one that is being rebooted after a Blue Screen Of Death, the symptom of a system crash that is all too familiar to NT 4 veterans. Many Windows 2000 parameters can be changed without a reboot where a change to the corresponding parameter in Windows NT 4 would require one. Unfortunately, as we will see, some of the most common parameter changes still require a reboot.

NT Must Be Able to "Scale" to Use Big Computers

Reliability's not the only big-network issue that Microsoft faces. The other one is the limit on the raw power that NT can use—to use a word that the PC industry created a few years ago, NT must be more *scalable*.

Being an "enterprise" operating system requires two different kinds of scalability which are somewhat at odds with each other: performance scalability and administrative scalability. The first asks, "If I need to do more work with NT, can I just run it on a bigger computer?" The second asks, "If I need to support more users/computers/gigabytes of hard disk/etc., can I do it without hiring more administrators?"

Performance Scalability CPUs are simply not getting all that much faster in terms of the things they can do. To create faster or higher-capacity computers, then, computer manufacturers have been putting more and more CPUs into a box. And while NT has in theory been designed to use up to 32 processors since its first incarnation, in reality, very few people have been able to get any use out of more than 4 processors. With Windows 2000, Microsoft claims to have improved the scalability of NT—although I've not yet heard anyone say with a straight face that Windows 2000 will "run like a top" on a 32-processor system.

Besides the ability to use a larger number of CPUs, there were internal restrictions within Windows NT, such as the number of users that a SAM database would allow, that simply had to go. With Active Directory, many restrictions, including this one, have been removed.

The three versions of Server support different numbers of CPUs. Windows 2000 Server supports four processors. Windows 2000 Advanced Server supports 8 processors, and Windows 2000 Datacenter Server supports 32 processors.

NOTE Oh, and if you're looking in your Webster's for a definition of *scalability*, don't bother; it's not a real word. Microsoft made it up a few years ago. Basically, *scalable* roughly means, "As the job's demands grow, you can meet them by throwing in more hardware—processors and memory—and the system will meet the needs." It's become an issue because, while NT has theoretically supported 32 processors since its inception, much of the basic NT operating system itself can't use many processors—for example, adding a ninth processor to an eight-processor domain controller won't produce any faster logins. That's also true of NT programs; depending on whom you ask, SQL Server maxes out at four or eight processors. Beyond that, adding more processors does nothing more than run up the electric bill.

Administrative Scalability/Manageability Large enterprises do not like to add headcount in their core business areas, much less just to administer Windows NT. Windows 2000 Server contains a number of facilities such as Intellimirror, designed to allow customers to support more users running with more complex desktop environments with fewer support personnel. Microsoft typically refers to this area as "Manageability," though I think "Administrative Scalability" better captures the flavor of the topic.

In this area, one of the most important additions to Windows 2000 is its support for both issuing and honoring digital certificates in place of userids and passwords for identification and authentication. The overall system needed to manage the life cycles of digital certificates and verify their authenticity and current validity is called Public Key Infrastructure (PKI). PKI-based security is both more secure and vastly more administratively scalable than userid+password-based security, but it is also much, much more technically complex.

Modernize NT

Three years can be an awfully long time in the computer business. The years since 1996 have seen the emergence of Universal Serial Bus, IEEE 1394, Fiber Channel, and 3-D video cards, just to name a few areas of technological growth, as well as the introduction of hundreds of new network cards, video boards, sound cards, SCSI host adapters, and so on. A new crop of network-aware PCs has appeared, PCs that understand networking right in their BIOSes and that are designed to be taken straight out of the box without anything on their hard drives, plugged into the network, and started up from the network rather than from any on-disk software. And on a more mundane note, nearly every PC sold in the past five years supports a hardware system called Plug and Play (PnP).

NT supports none of these things right out of the box. Some of these devices can be made to work, but some can't. Hardware support has always been something of an

afterthought in NT, and it's amazing that Microsoft shipped NT 4 without any Plug-and-Play support, save an undocumented driver that could *sometimes* make a PnP ISA board work but that more commonly simply rendered a system unusable. NT 4's off-hand support, of PC Card laptops and its near-complete lack of support for Cardbus slots forced many an NT-centric shop to put NT Server on their servers, NT Workstation on their corporate desktops…and Windows 95 on their laptops.

One of Windows 2000's goals, then—and an essential one—is to support the new types of hardware and greatly improve the way that it works on laptops.

Make NT Easier to Support

The past 10 years have seen the rise of the graphical user interface (GUI), which brought a basically uniform "look and feel" to PC applications and made learning a PC application and PCs in general so much easier for users. We've seen programming tools go from some very simple development environments that crashed more often than they worked to today's very stable 32-bit suite of programming tools, making it possible for developers to create large and powerful 32-bit applications. Users and developers are better off—sounds good, doesn't it?

Well, it is, for them. But many of us fall into a third category: support staff. And while some things have gotten better—the graphical nature of many of NT's administrative tools helped get many new admins started on a networking career—the actual job of support hasn't gotten any easier. Consider this: Would you rather rebuild a `CONFIG.SYS` file to stitch back together a damaged DOS machine from memory, or would you prefer to pick through a broken Registry trying to figure out what's ailing it?

Microsoft's competition knew that support was the Achilles' heel of both Windows and NT, and so in the mid-'90s, Sun and others began extolling the importance of considering the Total Cost of Ownership (TCO) of any desktop system. It wasn't hard to make the argument that the biggest cost of putting Windows on a desktop isn't the hardware or the software—it's the staff hours required to get it up and keep it running.

With Windows 2000, Microsoft starts to reduce desktop TCO. A group of Windows 2000 improvements called Change and Configuration Management tools makes life easier for support folks and network administrators in general.

Specific New Capabilities and Features

So much for the good intentions. What about the new goodies?

Microsoft lists pages and pages of enhancements to Windows 2000—the PR people have, after all, had over three years to cook up those lists. I'm sure they're all of value to someone, but here are the things that I find most valuable in Windows 2000,

arranged according to my three earlier categories—making NT more enterprise ready, modernizing NT, and improving its administrative tools/lowering TCO.

Making Windows 2000/NT More "Enterprising"

Several functions help push NT's latest incarnation to a place in the big leagues. In particular, the most significant "big network" changes to NT include:

- Active Directory
- Improved TCP/IP-based networking infrastructure
- More scalable security infrastructure options
- More powerful file sharing with the Distributed File System and the File Replication Service
- Freedom from drive letters with junction points and mountable drives
- More flexible online storage via the Removable Storage Manager

Active Directory

The crown jewel of Windows 2000, Active Directory is also the single most pervasive piece of the OS. Many of the things you'll read about in this book, many of the compelling features of Windows 2000, simply cannot function without Active Directory. Group policies, domain trees and forests, centralized deployment of applications, and the best features of the Distributed File System (to name a few) will not operate until you've got a system acting as an Active Directory server.

NOTE The whys and wherefores of Active Directory are complex enough that they'll get a chapter all their own. In Chapter 2, you'll read about what Active Directory is trying to accomplish, how it does so, and how you can best design the Active Directory for your enterprise.

Network Infrastructure Improvements

Anyone building an NT-based network around the TCP/IP protocol needed three important infrastructure tools:

- The Windows Internet Name Service (WINS), which helped Windows 2000–and NT-based servers and workstations locate domain controllers (which handled logins and authentication in general) as well as file and print servers.

- The Dynamic Host Configuration Protocol (DHCP), which simplified and centralized the once-onerous task of configuring TCP/IP on workstations.
- The Domain Name System (DNS), which did the same kind of job as WINS—it keeps track of names and addresses—but instead of helping workstations locate domain controllers and file/print servers, DNS helps programs like Web browsers and e-mail clients to find Web and mail servers. Some firms have avoided moving their networks to TCP/IP, staying instead with IPX (a protocol that owes its popularity to Novell's networking products) or NetBEUI (the main protocol for Microsoft networking prior to 1995). But with Windows 2000, pretty much everyone should be using TCP/IP, making DHCP, WINS, and DNS essential parts of any Windows 2000–based network.

WINS

Why did NT have two services—WINS and DNS—that kept track of names? This was the case because of a questionable choice that Microsoft made back in 1994. Of the two, WINS was the most troublesome and, for some networks, unfortunately the most vital. Thus, it was to many people quite excellent news when Microsoft announced that Windows 2000 would be the end of WINS.

Reports of its death, however, turned out to be greatly exaggerated. The actual story is that, if you have a network that is 100-percent Windows 2000, both on the workstation and server, then yes, you can stop using WINS. But most of us won't have that for years, so Windows 2000 still has a WINS service. Thankfully, it's greatly improved; one expert commented to me that it's ironic that Microsoft finally "fixed" WINS, just as they were about to kill it. Chapter 18 shows you how to set it up and make it work.

DNS

DNS was something of a sidelight under NT 4 as NT didn't really need DNS—DNS's main value was to assist Internet-oriented programs like Web, FTP, and POP3/SMTP mail clients in finding their corresponding servers. Under Windows 2000, however, DNS takes center stage. Without it, Active Directory won't work.

NT 4's DNS server was a pleasure to work with, although that's just my opinion: I've spoken with people who tell me that it couldn't handle high volume loads. *I* didn't have any bad experiences with it, so I can't comment. NT 4's DNS wrapped a well-designed GUI around a standard DNS implementation, making basic DNS tasks simpler than they would be for a Unix DNS implementation at the time. Windows 2000 takes that a step further with improved wizards. First-time DNS administrators will find that Windows 2000's DNS server almost does all the hand-holding you could need.

Additionally, Windows 2000's DNS supports dynamic updates, a process wherein adding information about new machines to a DNS database can be automated. Based on the Internet standard document RFC 2136 (the Internet's standards are described

in documents called Request for Comments, or RFCs), it combines the best of NT 4's WINS and DNS servers. The DNS server also supports another Internet standard, RFC 2052, which greatly expands the kind of information that DNS servers can hold onto. For example, a pre-2052 DNS server could tell you what machines acted as mail servers for a given Internet domain, but not which machines were Web or FTP servers. 2052-compliant DNS servers can do that, and more: Active Directory now uses RFC 2052 to allow DNS to help workstations find domain controllers and other Active Directory–specific server types.

NOTE Chapter 18 covers how Active Directory uses RFC 2052 in more detail.

DHCP

DHCP frees network administrators from having to walk around and visit every single desktop in order to configure the TCP/IP protocol. The basic idea is that a workstation broadcasts over the network, seeking an IP address (every computer on an intranet must have a unique IP address); a DHCP server hears the plea and assigns that computer its own unique IP address.

The End of Rogue DHCP Servers This is in general great, but now and then some dodo would decide to "practice" with DHCP by setting up a DHCP server on some PC. The budding new administrator's new DHCP server would then start handing out completely bogus addresses to unsuspecting workstations. Those workstations would then have IP addresses, but they'd be worthless ones, and as a result those workstations would be unable to function on the company's network.

With Windows 2000, however, not just anyone can create a DHCP server. Now, DHCP servers must be authorized in the Active Directory before they're allowed to start handing out addresses. This is a great advance, the end of what we used to call "rogue" DHCP servers.

DHCP Works with DNS to Register Clients You read before that the new DNS supports dynamic updates, a process standardized in RFC 2136 whereby the DNS server will automatically collect address information about machines on the network. This is an improvement over NT 4's DNS server because that DNS server couldn't automatically collect DNS information about machines—you, the administrator, had to type the names and IP addresses of new machines into the DNS Manager administration tool.

Windows 2000's DNS server collects its information about machines on the network with the help of those machines. When a machine starts up, one of the things it's doing while booting up—one of the reasons that booting modern PCs takes so

long—is contacting the DNS server to tell the DNS server that the machine exists. In effect, each workstation and server on the network must know to *register* itself with the DNS server.

Unfortunately, as RFC 2136 is a fairly recent development in the DNS world, most existing operating systems—DOS, Windows for Workgroups, Windows 9*x*, NT 3.*x*, and 4.*x*—do not know to register themselves with a DNS server. That's where Windows 2000's DHCP server helps out. You can optionally tell the DHCP server to handle the DNS registrations for non-2136–aware workstations. This is a very useful new feature because, without it, dynamic updates wouldn't be worth much except for the rare firm that runs solely Windows 2000 on its desktops, laptops, and servers.

NOTE You can read more about DHCP in Chapter 18.

Quality of Service

The Internet's underlying protocols, TCP/IP, have something of an egalitarian nature; when the Net's busy, it's first come, first served. But the protocols have always had a built-in capability that would theoretically allow an Internet operator to give greater priority to one user over another, to dial in a better response time for some than for others. That's called Quality of Service, or QoS. It was always there but not really implemented as it sort of ran against the way the Net was run.

The growth of corporate intranets, however, changes that story. Network operators in corporate networks aren't serving a mass public; rather, they're serving a diverse and hierarchical organization whose leaders may well want to be able to say, "We direct that this individual get more bandwidth and faster access to network resources than this other individual." That's possible if you're using expensive Cisco routers—but now you can do it if you use Windows 2000 machines as your IP routers as well.

New Security Infrastructure

As one security expert once said to me, "We knew that NT had 'made it' when hackers started targeting it." Hardly a month goes by without word of a new security hole in NT 4 and the hot fixes that are intended to plug that hole. Patch a plaster wall with Spackle enough and eventually you have to wonder if you've got a plaster wall or a Spackle wall—so Microsoft must have decided early on that one of the things that Windows 2000 couldn't live without was a new security system.

So they built *two*.

Originally, Windows 2000 was supposed to replace NT 4's authentication system, known as NTLM (for NT LAN Manager), with a system popular in the Unix world

called Kerberos. Kerberos is well understood and works well in large-scale systems, assisting Microsoft in their "scalability" (there's that nonword again) goal.

Partway through the Windows 2000 development process, Microsoft decided to supplement Kerberos with a *third* security system, a public key system based on the X.509 standard. They did that mainly because a public key system is considered far more scalable than either an NTLM or Kerberos system. Several companies offer hardware readers that allow users to log in by inserting credit card–sized devices called *smart cards* into the readers.

Kerberos and public key provide as a side effect a feature that NT administrators have asked after for a long time—transitive trust relationships.

Distributed File System

NT's first and probably still most prevalent job is as a file server. And as time has gone on and versions have appeared, it's gotten better at it. Some benchmarks have rated it as fast or faster than NetWare, the guys to beat. And where NT 4's file server software was largely unable to deliver throughput faster than 90Mbps, Windows 2000 can transfer data almost 10 times faster.

Disconnecting Physical Locations from Names

But NT's file server system is hampered by the way it addresses shares on servers. A share named DATA on a server named WALLY would be accessed as `\\WALLY\DATA`. Although that makes sense, it's limiting. Suppose the WALLY server goes up in a puff of smoke? We install a new server, perhaps named SALLY rather than WALLY, restore the data from WALLY, and re-create the DATA share. But now it's `\\SALLY\DATA` rather than `\\WALLY\DATA`, and configurations that are hardwired to look for and expect `\\WALLY\DATA` will fail. In other words, if a share's physical location changes, so must its "logical" location—its name. It'd be nice to be able to give a share a name that it could keep no matter what server it happened to be on.

Windows 2000 takes NT beyond that with the Distributed File System. In combination with Active Directory, Dfs—note the lowercase in the acronym; apparently someone already owned *DFS* when Microsoft started working on the Distributed File System—allows you to give all of your shares names like `\\`*`domainname`*`\`*`sharename`* rather than `\\`*`servername`*`\`*`sharename`*. You needn't know the name of the file server that the share is on.

Fault Tolerance

You probably know that Windows 2000 offers you many ways to add reliability to your network through RAID storage and two-system computer clusters. RAID boxes aren't cheap, and clusters require a lot of hardware (two identical machines, external SCSI storage, extra network cards, and either the Advanced or Datacenter edition of

Windows 2000 Server). But there are some very inexpensive fault tolerance options for Windows 2000 networks as well; Dfs provides one.

If you have a file share that you want to be available despite network misfortune and failure, then one way to accomplish that is with a *fault tolerant Dfs share*. To create one, just create two or more file shares that contain the same information, then tell Dfs to treat them like one share. So, for example, in a domain named ROCKS, you might have a share named STUFF on a server named S1 and a share named STUFF on a server named S2. To the outside world, however, only one share would be visible as `\\ROCKS\STUFF`. Then, when someone tries to access `\\ROCKS\STUFF`, Dfs will basically flip a coin and either send her to `\\S1\STUFF` or `\\S2\STUFF`. It's not full-blown fault tolerance—if S1 goes down, nothing automatically transfers people from `\\S1\STUFF` to `\\S2\STUFF`—but it's a low-cost way to increase the chance that a given share will be available, even under network "fire."

File Replication Service

Fault tolerant Dfs requires that you maintain several network shares all containing the same information. That can be a lot of work, but then fault tolerant Dfs sounds like it could be worth it.

For example, as you'll read later, Windows 2000 makes deploying applications from a central location or a few central locations possible. So instead of having to visit hundreds of desktops to install Office 2000, you can instead put Office 2000's distribution files on a server and set up everyone's system to install Office 2000 from that server. Hmmm… hundreds of people all trying to download an application package from one file share, all at the same time, won't be very satisfactory.

It'd be better to have exactly the same application package copied to perhaps 10 other shares. You *could*, of course, create the 10 shares and copy the package to each one—but you needn't. Windows 2000 includes the File Replication Service, or FRS. FRS is a vastly improved version of an old NT feature called Directory Replication. Anyone who's ever tried to use NT 4's Directory Replication knows that it needed work—FRS is the happy result.

Junction Points and Mounted Drives

All of this helpful misdirection in file shares—the ability to disconnect file share names from their physical locations—is pretty useful. In fact, it'd be nice to be able to start doing some of that physical/logical misdirection on *local* drives—and you can.

NT's always been hampered by the fact that it can only support 26 storage volumes, A: through Z:. Tying storage volumes to letters in the alphabet was a great idea when CP/M (an early pre-PC microcompute operating system) started doing it back in 1978, but nowadays it seems more a bug than a feature.

With NT 4, you created partitions on drives and then assigned drive letters to those partitions. With Windows 2000, in contrast, you can tie any number of drive partitions to a single drive letter. The trick is this: You first create a folder (a subdirectory) in any existing NTFS drive (NTFS is a file format that—not surprisingly—only NT supports, rather than the FAT file system that DOS uses or the FAT32 file system that Windows 95/98 use). You can then associate—*mount* is the Windows 2000 term—any drive partition with that folder.

Thus, for example, suppose you've got a drive D:, which is NTFS. (If you want to follow along with this example—although you needn't in order to understand it—you'll need a drive D: formatted as NTFS and an H: drive formatted in any way, it doesn't matter.) You've got a bunch of partitions on your system and you're up to drive P:. You'd like to free up drive letter H: so you can use it to map to a home directory. Well, under NT 4, you *could* just highlight the partition currently assigned to H: and change its drive letter to Q: or some other still-unused letter.

Under Windows 2000, however, you can both free up the H: drive letter *and* keep access to the partition, *without* having to use another drive letter. First, create an empty directory on D:. (It needn't be D:; any NTFS drive will do.) Just for the sake of example, call it `D:\OLDH`—again, any directory name will do. Then you'd go into the Disk Manager, Windows 2000's version of the Disk Administrator. You do that by right-clicking My Computer, choosing Manage, opening up Storage in the left pane, and then opening up Disk Management inside *that*. Find the H: partition, right-click it, and choose Change Drive Letter and Path.

Where NT 4 only allowed you to associate *one* drive letter with a partition, Windows 2000 lets you associate a partition with as many drive letters as you like. Now, that won't help us much because we're trying to get *rid* of a drive letter—but the very same dialog box that allows you to add a new drive letter also lets you associate a partition with "an empty folder that supports drive paths"—in other words, with `D:\OLDH`. Once you've added `D:\OLDH` as an acceptable "name" for the partition, you can type either **DIR H:** or **DIR D:\OLDH** and you'll see the same files, because both names refer to the same directory.

But the plan was to free up H:, and we haven't done that yet. Returning to the Disk Manager, again right-click the H: partition and choose Change Drive Letter and Path. This time, you'll see that H: has two acceptable names, `H:` and `D:\OLDH`. Highlight H: and choose delete. Once you reboot (yes, you've got to reboot for this change to take effect; some things never change), H: will be free and you'll only be able to access the partition's data through `D:\OLDH`. And if you didn't want to do all of that with a GUI tool, there's a command-line tool named MOUNTVOL that allows you to mount and unmount drives.

NOTE You can read more about mounting disks in Chapter 7, "Managing Windows 2000 Storage."

Remote Storage

As you've already read, the Distributed File System and the File Replication Service appeared in Windows 2000. As you'll read later, Windows 2000 includes disk quotas (finally) and there's a better Backup. Clearly, storage was an issue for the Windows 2000 design team. But perhaps the most unusual new storage-related capability is Remote Storage, a program whose goal is to allow you to mix tape drive space and hard disk space as if they were one thing.

The idea with Remote Storage is this. Suppose you have a 24GB hard disk on your server; perhaps it's a nice amount of storage but not quite enough for your users' needs. Suppose also that you've got a tape backup device, a carousel device that can automatically mount any one of 16 tapes into the tape drive without the need for human intervention. Perhaps it's a DLT loader and each tape can store 20 gigabytes of data; that works out to about 320GB of tape storage and, again, 24GB of hard disk storage. Here's what Remote Storage lets you do:

It lets you lie about the amount of hard disk space you have.

You essentially advertise that you've got a volume containing 320 plus 24, or 344, gigabytes of online storage space. As people save data to that volume, Remote Storage first saves the data to the hard disk. But eventually, of course, all of that user data fills up the hard disk; at that point, Remote Storage shows off its value. Remote Storage searches the hard disk and finds which files have lain untouched for the longest time. A file could have, for example, been saved eight months ago by some user but not read or modified since. Remote Storage takes these infrequently accessed files and moves them from the hard disk onto the tape drives, freeing up hard disk space.

Ah, but what happens if someone decides to go looking for that file that was untouched for eight months? Remote Storage has been claiming that the file is ready and available at any time. If some user tries to access the file, Remote Storage finds the file on tape and puts it back on the hard disk, where the user can get to it. Yes, it's slow, but the fact is that many files are created and never reexamined, which means there's a good chance that putting the file on tape and off the hard disk will never inconvenience anyone.

I worked with mainframe systems that did things like this years ago and it was quite convenient—files untouched for six months or so would be said to be "migrated" to tape. I could "un-migrate" the tapes, and of course that would take a while, but it wasn't that much of a nuisance and it helped keep the mainframe's disks free.

NOTE You can read more about Remote Storage in Chapter 7, "Managing Windows 2000 Storage."

Modernizing NT

NT is an operating system first introduced in the '90s, so it couldn't have needed all *that* much modernizing. But it was getting awfully embarrassing not to be able to Plug and Play, so Microsoft fixed that. And while they were at it, what's a new release of Windows or NT without a bit of fiddling with the user interface?

Win2K Can Plug and Play

In what may be the feature awaited for the second-longest time (disks are no doubt the longest-awaited feature), Windows 2000 finally offers a version of NT that knows how to do Plug And Play.

That's good news, but, as when PnP first appeared in Windows 95 and ever since, sometimes the playing doesn't happen right after the plugging. Sometimes it works that way, but inserting a new board into a system often still requires a knowledge of interrupt request levels (IRQs) and other hardware characteristics, as well as a bit of CMOS spelunking. Still, it's nice to be able to finally shut up those Windows 95 guys smirking about how easy it is to add new cards to their systems.

NOTE You can read more about this in a special chapter devoted solely to adding new hardware to a Windows 2000 system, Chapter 6.

NT Gets a User Inter-Facelift

Windows 95 introduced a brand-new, more Macintosh-like user interface to the Windows world. NT 4 followed that but didn't exactly copy the Windows 95 UI, instead improving upon it. Internet Explorer 4 brought Active Desktop, which brought a more Web-like feel to the Windows/NT desktop, although at an often unacceptable cost in performance. Perhaps Active Desktop's best innovation was the Quick Launch bar, a portion of the Taskbar that can hold any number of tiny icons representing oft-used programs: One click and the program starts. Windows 98's user interface built further

upon that, and Windows 2000's desktop offers even more new features, many of which are quite useful.

For example, as time goes on, your Start-Programs menu will probably actually get *smaller*. Windows 2000 tracks how often you use programs, and if you don't use a program for a while, the program disappears off the Start-Programs menu. It doesn't disappear forever, however—instead, Windows 2000 displays a set of chevrons at the end of the menu. To see the programs (and even groups) that have disappeared because of disuse, just click the chevrons and the entire program menu returns. The Control Panel also uses this frequency-of-use information; as you no doubt know from experience with Windows 9*x* and/or NT 4, the Control Panel's Add/Remove Programs allows you to uninstall programs. That's still true with Windows 2000, but in addition to telling you what programs you can uninstall, Windows 2000 tells you how often you *use* that program. Pretty neat—if you need some more disk space and you're trying to choose which program to remove in order to *get* that space, the Control Panel even gives you useful hints about which programs you won't miss!

The "user interfacelift" isn't an unalloyed blessing, however. When I first installed beta 2 of Windows 2000, it took me about 10 minutes to find the Network Control Panel. After many years, I was used to just opening up the Control Panel, then opening the Network applet—but here, no go. Instead, I right-click on My Network Places (the name for Network Neighborhood's replacement), choose Properties, then find Local Area Connection in the resulting screen, then right-click *that*, and choose Properties again. Intuitive, no? Well, okay, intuitive NO. In any case, there's enough things that have moved around that it seemed a good idea to include a short chapter on where everything's moved to, so if you can't find the Network Control Panel, or can't figure out where to turn off a service, or are baffled about where to go to partition a hard disk, turn to Chapter 4.

Lowering TCO and Warming Administrators' Hearts

Okay, I hear you thinking, "So now Windows 2000 lets us build bigger NT networks than before—heck, maybe there's a couple of bucks in overtime to be made from larger networks—and now there's Plug and Play, great, so long as there are drivers, and by the way, many NT 4 drivers will not work under Windows 2000, so there had *better* be drivers—and now the new user interface has hidden or rearranged all of the tools that I know and lo… well, like."

So you're probably thinking, "Tell me again why I'm going to like this."

You're going to like Windows 2000 because it's got a bunch of new tools. Several tools, like the Remote Installation Services (RIS), Terminal Services, the Group Policy Editor, and the Microsoft Installer Service, will make rollouts easier; they'll simplify getting an operating system on a new computer and then simplify getting applications

onto that computer. Some tools, like (again) Terminal Services, Windows 2000's new built-in telnet server, and Windows Management Instrumentation, will make remote control easier. As you'll see, it's far easier to administer Windows 2000 servers from a distance than it ever was to administer NT 4 servers remotely. And some tools, such as disk quotas, client-side caching, RUNAS, a more powerful command line, and the Internet Connection Sharing feature, are either very effective administrative tools or just plain cool.

Remote Installation Services

Those choosing to put NT not only on their servers but on their workstations as well have never had an easy time of it. Rolling out DOS or Windows 9*x* to hundreds of similarly equipped machines is relatively simple: Set up the operating system on one "model" computer, get it configured the way your firm needs it, and then essentially "clone" that entire configuration byte-by-byte from the model computer's hard disk to the hard disks of all of the similarly equipped computers. From there, all that needs doing is usually a bit of fiddling on each of the new workstations to customize and make each machine unique in some way. Products like Ghost and Drive Image Professional are excellent tools for getting that job done, in effect "Xeroxing" a master disk image from the central model computer to other computers.

Unfortunately, NT has never lent itself to that. Its secure nature has always required that an administrator run NT's Setup program separately on every would-be NT system, making big NT Workstation rollouts a painful process. The Ghost and Drive Image Pro folks have built some tools to try to allow administrators to use those mass-copying programs to get NT onto a computer's hard disk, but those solutions have never been sanctioned by Microsoft, putting anyone who uses them in a kind of support "Twilight Zone."

Windows 2000 solves that problem by providing a new service called the Remote Installation Services. As with Ghost-like programs, RIS directs you to first create a workstation the way that you want it configured, then a wizard (RIPRep) copies that workstation's disk image to a server—it can be any Windows 2000 server. (Unfortunately, RIS won't help you install Windows 2000 Server, just Windows 2000 Professional.) Just take a new computer out of the box, then attach it to the network, and boot it with a floppy whose image ships with Windows 2000. It asks you to identify the user who will work at that computer, and from that point on, it's a hands-off installation. RIS copies the disk image down to the new computer, runs a hardware detection to ensure that the system gets the correct drivers, the new computer reboots, and Windows 2000 Professional (for some reason, Microsoft chose to name Windows 2000 Workstation "Windows 2000 Professional") is up and running on the new system.

NOTE You can read more about RIS in Chapter 3.

Windows Terminal Server Becomes Standard

Centralized systems like mainframes were great for support people because all of the user data and configuration information resided on a small number of central locations. Solving a user's problem was then easier as most support calls could be handled from one location. Centralized systems also meant easy backup.

On the other hand, centralized systems like mainframes weren't very good at highly interactive "personal productivity" applications such as word processors or spreadsheets or more modern applications like Web browsers. The decentralized nature of desktop PCs solved that problem. Unfortunately, having computers scattered geographically around an enterprise made for a tougher support job.

How, then, to have a system that allows users to run highly interactive PC-type applications and at the same time keep all of the computing and storage in a centrally located, cheaper-to-support place?

Windows Terminal Server, that's how. WTS turns an NT machine into a kind of a mainframe. You attach dumb terminals—or PCs running programs that make them look like dumb terminals—to the Terminal Server over a network or dial-up connection, and for all intents and purposes it looks as if the user's just running a standard Windows 2000 Professional desktop. But all the user's machine is doing is providing keystrokes and mouse-clicks and receiving graphic images of the desktop. Everything else—all the data and all of the computation—is going on in the centrally located Windows 2000 servers.

Now, Windows Terminal Server first shipped late in NT 4's life, but it was a separate product. Windows 2000 lets you convert *any* Windows 2000 server into a Terminal Server with just a few mouse clicks. Additionally, users on a Windows 2000 Terminal Server have more options than did users on an NT 4 Windows Terminal Server.

NOTE You can read more about Terminal Server in Chapter 14.

Group Policy Snap-in Replaces System Policies

One way to reduce TCO in a firm with dozens, hundreds, or thousands of Windows or NT desktops is to standardize those desktops and to control in some way what gets done on those desktops.

Windows NT 4 had a feature called *system policies* that let an administrator lock down a desktop to a certain extent. If applied in full, system policies would allow an administrator to create a user workstation that could run just a few applications—say, Word, Outlook, and Internet Explorer—and nothing else.

But system policies were difficult to work with and some of them just plain never worked. Furthermore, it was impossible to apply system policies to a group of *machines*—only groups of users. So Microsoft went back to the drawing board and redesigned the idea from the ground up. The result is the Group Policy snap-in. It creates and assigns "group" policies. The word *group* is in the name to underscore something missing from NT 4's system policies. It was simple to apply some kind of control to one user, but it was more difficult to apply policies to groups of users, which is really the only reasonable way to create and manage a control structure—it's far easier to manage a large enterprise wholesale, with groups, than to manage in a retail fashion, user by user.

NOTE The odd part about group policies is that they don't *apply* to groups. Instead, they apply to subunits of Windows 2000 domains called *organizational units*, which you'll read about in the next chapter. You can certainly control whether a policy affects a particular individual or machine based on what group or groups they belong to, but you can't apply a policy to a group—instead, you apply policies to organizational units. (That's only basically true, as you can also apply policies to particular domains or sites, but you'll read more about that in the next chapter.)

NT 4–style system policies furthermore required building some files with the desired policy information, placing those files on a domain controller, and having to ensure that those policy files replicated properly amongst the other domain controllers. Group policies live in the Active Directory, meaning that they get replicated automatically without any necessary fussing from the administrator.

But that's not all you'll like about group policies. The list of available system policies was relatively short, and the vast majority of those policies were of no value. In contrast, there's a rich variety of group policies in the Group Policy snap-in, and many of them will solve some common administrative nightmares.

NOTE You'll read more about group policies throughout the book, but much of the coverage appears in Chapters 8 (user accounts) and 10 (deploying applications).

Installer Service and Application Deployment

While I commented earlier that support people had gotten the short end of previous NT upgrades, that's not the case in Windows 2000. As part of their Zero Administration Windows initiative, Microsoft has built a tool into the Group Policy snap-in that allows an administrator to sit in a central location and place applications on a user's desktop without having to visit that desktop.

Previously, firms wanting to do this needed to buy and deploy the Microsoft's Systems Management Server (SMS) tool to accomplish deployment at a distance; with Windows 2000, it's built right in. But that's only the first part of the story.

We usually install programs by running the Setup program that they come with. But we never know beforehand just what the Setup program's going to do; what messes it may make on the computer. Windows 2000 has an answer for that, as well: the Installer service.

The idea with the Installer service is that you no longer run Setup programs to install applications. Instead, you feed to the Installer a file called a *Microsoft Installer* file; they're recognizable because they have the extension `.MSI`. But an MSI file isn't a program. Instead, it's a set of commands telling the Installer how to install an application—what Registry entries to create, where to copy files, what icons to place on the program menu, and so on. And you can examine an MSI file before installing it to find out what it's going to tell Installer to do—which means you can head off trouble at the pass.

NOTE You'll read more about Installer in Chapter 10.

Better Remote Control and Command Lines

One of my pet peeves with NT has always been that there are very few good remote administration tools. For example, if you want to create a file share on a remote machine, you can do it, but it's cumbersome and involves a different tool—you create a local file share from the Explorer, but you must use NT 4's Server Manager to create remote shares. Even then, Server Manager won't let you control share permissions on remote shares; for that, you've got to look to the Resource Kit and its `RMTSHARE.EXE` program. And that's just one example: In general, it seems as if NT 4's administrative tools are originally built to only control the local machine; any remote administration abilities either don't exist or have a distinctly "tacked on afterward" feel.

Windows Management Instrumentation

While Windows 2000 doesn't completely solve that problem, you'll find that most administrative tools work as well on remote computers as they do on the local machine. Virtually all hardware functions are now built around something called the Windows Management Instrumentation, or WMI, an eminently "remoteable" software interface. As a result, Device Manager lets you view and modify hardware settings not only for the computer you're sitting at, but any machine on the network that you can see (and on which you have administrative rights); the same is true for storage management. Where Disk Administrator let you format and partition disks, it only operated on locally attached disks—its successor, Disk Manager, lets you do any of those things locally or over the network. (Finally, we network administrators will have the respect we deserve! Just think: "Call *me* a geek, will ya? I'll just attach to your computer across the network and reformat your drive…." Just joking, just joking—we network types would *never* use our powers for Evil….)

Windows 2000 Includes a Telnet Server

Furthermore, every Windows 2000 Server ships with a telnet server. If you choose to run the telnet server on a server, you can then connect to that server with any telnet client.

Odd as it may sound, you may sometimes find yourself telnetting to your own local machine. Why? Because when you log in with a telnet session, you identify yourself with a name and password. That means that, if you're currently logged in to a machine as a user and you need to run some administrative-level command, you need to make the machine suddenly recognize you as an administrator. Telnetting is one way—but there's another as well, a new command called RUNAS that you'll meet in a bit.

Better Admin Tools: A More Powerful Command Line

That kind of leads me into my discussion of tools that aren't so much classifiable as rollout tools or as remote control tools; rather, they just fall into a category of "neat new administrative tools." Telnet offers me a segue.

Once connected, the telnet session then gives you a command-line prompt. From there, you can run any *command-line* application remotely. "But," you may be wondering, "what good is the command line? Can I create user accounts, reset passwords, and the like from the command line?" Well, according to Microsoft, one of the "must-do" items on its Windows 2000 things-to-do list was to ensure that you could do all of your administration from the command line, that in theory you would never have to use a GUI tool. I've not found that I can do *everything* from the command line, but there's a whole lot more that you can do from a command line, as you'll see throughout this book.

Unix's SU Comes to Windows 2000

I often find myself, as mentioned before, needing to change status in the machine's eyes. For example, suppose I'm at a user's workstation trying to figure out a computer problem that's plaguing her. I realize that something's set incorrectly on her workstation and I know how to fix it, but she's currently logged in, and she's only got user-level privilege, so I can't execute whatever administrative command I had in mind. What to do?

As mentioned earlier, I could telnet to the system as an administrator, but the telnet server only ships with Windows 2000 Server, not Professional. I *could* ask her to log off, and then I could log on with my administrative account. But I might not want to do that—sometimes I've got a roaming profile set up and I don't want to wait for the profile to download *and* I don't want to have to worry about deleting that profile off the user's machine. The answer? RUNAS.

The scenario described above, where someone's logged on to the system as a user and needs to briefly take on administrative powers, and perhaps doesn't want to have to wait for a logoff/logon sequence, is an old one in the Unix world. That's why most Unix implementations have a so-called Super User (SU) command. It lets you run *just one program* with a different set of credentials. In the Windows 2000 world, the command's name is RUNAS—in other words, "*run* this particular application *as* if someone else—presumably an administrator—were running it."

You must run RUNAS either from the command line or from Start/Run. RUNAS's syntax looks like this:

```
runas /user:username command
```

Username is the administrative username, and *command* is whatever command you want to run as an administrator. If the administrator's account is not in the same domain as the Windows 2000 Professional machine, then you may have to include the name of the administrator's domain as well. For example, if I wanted to modify a user account, I would do it with the Directory Services Administrator (which you'll meet in Chapter 8), a file named `dsa.msc`. If my administrative account were named Bigmark from a domain named LANGUYS, I could start up the DSA like so:

```
runas /user:languys\bigmark dsa.msc
```

Under Windows 2000, you have *two* ways of identifying an account—through the old NT 4–flavor "domain\username" approach, as I used earlier, or through a newer user-specific logon name called the *User Principal Name*. A UPN looks a lot like an e-mail address; for example, Bigmark's UPN might be `bigmark@languys.com`. I could use that formulation as well in my RUNAS command:

```
runas /user:bigmark@languys.com dsa.msc
```

You'll learn about UPNs in Chapters 2 and 8. Oh, and by the way, in case you were wondering, when you do a RUNAS, the system prompts you to enter a password; merely knowing an administrative account name isn't sufficient to become an administrator.

Disk Quotas

Let's get a drum roll on this one…. After years of waiting, it's now possible to control how much space a given user takes up on a given volume. You can only set quotas on NTFS volumes.

The disk quota system is fairly simple—you can only set quotas on entire volumes, not directories, so you could, for example, say that Joe couldn't use more than 400MB of space on E, but you *couldn't* say that he couldn't use more than 200MB in `E:\DATA1` and 200MB in `E:\DATA2`—you can't get directory specific.

Oddly enough, you also cannot set quotas on particular user groups. Instead, you determine a good generic quota value and set that on the volume; that's the disk space limitation for each user. So, for example, suppose you set the quota to 20MB. That means that each user's personal quota is set to 20MB. You can then override that for any particular user. Sound cumbersome? It is; if you have 1,000 users from 10 different groups that access a particular volume and you want to set each user's quota based on its group membership, there's not much to do save to hand-set each user's quota amount, one at a time. But it's free, and at least it's of more value than Proquota, the profile size quota manager available under NT 4.

Backup Continues to Improve

Few things grow as rapidly as the apparent need for storage space. In the late '70s, network file servers were often built around a single shared 10MB hard disk; nowadays, it's not unusual for a desktop *workstation* to have one thousand times that much disk space.

Hard drives have gotten larger, faster, cheaper, and more reliable. But one thing that hasn't changed is the need for backup. NT's always come with a backup program, but it's always been a bit limited. It could only back up to a tape drive, so you couldn't use the NT Backup program to back up to a Jaz drive or network drive; it didn't support robotic tape changers, carousels that could automatically change the tape in a tape drive; and it was very cumbersome to use for a full server recovery.

With Windows 2000, those three objections go away. If you want to save to tape, then of course you can do that, but now you can also save to anything with a drive letter—Jaz, Superdisk, some Web-based backup system, or the like. If you use tapes but your server's disks are larger than the capacity of a single tape, you need no longer baby-sit the server waiting for the chance to swap tapes: Windows 2000 supports many tape loaders. And if you find yourself with a dead server that you need to revive quickly, you can take the most recent backup tapes from the dead server and a new computer and quickly get the contents of those tapes onto the new computer. The new computer acts in the role of the server, making disaster recovery simpler and quicker than it was under NT 4.

Client-Side Caching/Offline Files

This next aspect of Windows 2000 is not really a server function, it's a workstation (Windows 2000 Professional) function; but it'll gladden the hearts of users and administrators alike. Called either *client-side caching* or *Offline Files* by Microsoft, this function makes the network more reliable and faster and simplifies laptop/server file synchronization for mobile users.

Offline Files acts by automatically caching often-accessed network files and storing the cached copies in a folder on a local hard drive. Your desktop computer then uses those cached copies to speed up network access (or rather, they speed *apparent* network access), as subsequent accesses of a file can be handled out of the local hard disk's cached copy rather than having to go over the network. Offline Files can also use the cached copies of the files to act as a stand-in for the network when that network has failed or isn't present—such as when you're on the road.

You'll like Offline Files for several reasons. As these oft-used cached files will reside on the local hard disk, you'll immediately see what seems to be an increase in network response speed; opening up a file that appears to be on the network but is really in a local disk directory will yield apparently stunning improvements in response time, as little or no actual network activity is actually required. It also produces the side effect of reducing network traffic, as cached files needn't be retransmitted over the LAN. Having frequently used files on a local cache directory also solves the problem of "What do I do when the network's down and I need a file from a server?" If you try to access a file on a server that's not responding (or if you're not physically connected to the network), Offline Files shifts to *offline* mode. When in offline mode, Offline Files looks on your local Offline Files network cache, and if Offline Files finds a copy of that file in the cache, it delivers the file to the user just as if the server were up, running, and attached to the user's workstation.

Anyone who's ever had to get ready for a business trip knows two of the worst things about traveling with a laptop: the agony of getting on the plane only to realize that you've forgotten one or two essential files and the irritation of having to make sure that whatever files you changed while traveling get copied back to the network servers when you return. Offline Files greatly reduces the chance of the first problem because, again, often-used files tend to automatically end up in the local network cache directory. It greatly reduces the work of the second task by automating the laptop-to-server file synchronization process.

NOTE You can read more about Offline Files in Chapter 9, "Creating and Managing Shared Folders."

Internet Connection Sharing

A very large percentage of us have some kind of connection to the Internet, whether it be a simple dial-up connection, cable modem, or DSL. A substantial portion of us have more than one PC in our house, which leads to one of the most common pieces of e-mail that I get: "How do I share my Internet connection with all of the computers in the house?" Once, the answer to that question was a fairly lengthy discussion of routers and proxy servers.

Now, however, the answer's easy: Just use Internet Connection Sharing (ICS). Anyone who's ever used Windows 9*x* or NT 4 to dial in to an ISP will be able to use ICS without any trouble—using it involves little more than just checking a box.

Here's how it works. You run ICS on the computer that's dialed in (or cable modemed or DSLed) to the Internet. That computer can be running either Windows 2000 Professional or Windows 2000 Server. (It can even be running the updated version of Windows 98, Win 98 Second Edition.) You check a box labeled Shared Access in your connection's properties; this activates ICS. At this point, the ICS machine acts as a DHCP server (which provides the other computers at home with their IP addresses) and as a router (which ensures that their packets get from the home LAN to the Internet and back).

NOTE ICS is a very neat feature and will no doubt be pretty popular; you can read more about it in Chapter 17.

Bad News

It's not all wine and roses with Windows 2000, however. While it's a great improvement over NT 4, it still lacks in a number of ways.

DHCP Won't Be Fault Tolerant

The Dynamic Host Configuration Protocol (DHCP) is an essential bit of network infrastructure, and when it goes down, the network is at least partially crippled. Adding some kind of fault tolerance to DHCP made good sense and Microsoft told us it would offer it. Unfortunately, however, to implement fault tolerance on DHCP you must invest tens of thousands of dollars in hardware and software for a server cluster—a great answer for a large corporation, but impractical for the rest of us.

No Fax Server Software

NT's all-in-one small business version, BackOffice Small Business Edition, shipped with a nice, basic fax server—nothing so fancy that it would put the third-party fax server folks out of business, just a nice basic system that is to fax servers what WordPad is to word processors.

For some reason, Microsoft did not ship a fax server with Windows 2000, however. There *is* fax support, but only on a workstation-by-workstation basis. Thus, you could walk over to a server equipped with a fax modem and fax something from there, but you couldn't fax from your desktop using the server. This seems odd given that Microsoft clearly has NT-ready fax code, but perhaps the fear of Justice has stayed their hand on this matter…

Requires Powerful Hardware

Every new version of NT (or Windows, for that matter) renders entire product lines of formerly useful computers useless. For example, NT 3.1, 3.5, and 3.51 ran relatively well on 486 computers, but running NT 4 on a 486 was a quixotic venture. In the same way, Windows 2000 puts the final nail in the Pentium and the MMX coffins. Yes, you *can* run Windows 2000 on a Pentium—some of this book was written on a 266MHz MMX laptop, and some of my braver (or patient) coauthors did their testing on 133MHz and 166MHz machines—but at a noticeable loss in speed. Anyone wanting to get anything done on Windows 2000 will need at least a 350MHz Pentium II and 128MB of RAM. Domain controllers will run best with two physical hard disks. Much of the same advice goes for anyone wanting to run the Workstation version of Windows 2000, Professional; at the moment all I'm doing on my Professional workstation is editing this chapter with Word 97, and I'm using 96MB of RAM—so 96MB to 128MB minimum is definitely indicated!

Hardware/DirectX Support Is Still Spotty

One of the great frustrations about NT, whether in its 3.*x* and 4.*x* versions or in its current Windows 2000 incarnation, is its relatively thin hardware support, particularly when compared to its Windows 9*x* little brother. Windows 2000 improves upon this as it supports a wider range of hardware and because Plug and Play now makes it easier to install that hardware—but there are still many boards that plain won't work.

Furthermore, Windows 2000 claims to support the DirectX interface, the interface that most modern games are written to, but in actual fact, DirectX's performance makes the few games that I've tried on Windows 2000 unplayable. "What's that?" you say, "Games are irrelevant on servers?" Well, yes, that's probably true—but if *one* much-touted but easily tested subsystem of Windows 2000 (DirectX) doesn't work,

isn't it reasonable to be concerned about the other subsystems, the ones that aren't so easy to test?

AD Is Inferior to Existing Directory Services

Directory services have been around for ages. I recall working with a competing network operating system named Banyan VINES almost 10 years ago, when Banyan introduced a directory service called StreetTalk. StreetTalk was more flexible in 1992 than Active Directory is now. For example, it's inconceivable that you cannot take two existing domains and join them into an Active Directory forest (you can read more about forests in the next chapter)—such "pruning and grafting" has been possible in Novell Directory Services (NDS) for years. Nor are Active Directory's weaknesses the fault of NT somehow; Banyan has been selling an implementation of its StreetTalk directory service for NT for at least three years, and Novell's got a version of NDS for NT as well.

Granted, Active Directory's relative weakness probably stems from the fact that it's a "version 1.0" product. But will it improve? As with all companies, Microsoft isn't primarily motivated to create good products; instead, they're motivated to sell a lot of whatever they make—and making good stuff is usually one good way to sell a lot of stuff. But that's not the only way. Sometimes the battle for market share is won by effective advertising rather than quality. And if Microsoft wins the directory services war with marketing, then there won't *be* any incentive to improve the product.

There Are Still Far Too Many Reboots

Back in 1992, I interviewed one of the higher-ups in the NT project, a fellow named Bob Muglia. Bob is a heckuva nice guy and he provided me with a lot of useful information. But I remember one comment that he made to me, a promise that we're still waiting to see fulfilled.

"NT's going to be stable," he told me. "Once you get it set up with your drivers and applications, you should never have to reboot it. If you do, then we've failed."

I've run into Bob since then on several occasions and I've never had the heart to needle him about his quote. But what he told me in 1992 made eminent sense: At minimum, an enterprise-quality operating system *doesn't need to be rebooted all the time*. And in fact, Microsoft has gone to some pains to advertise that you needn't reboot Windows 2000 as often as you did NT 4. The number of necessary reboots *has* been reduced, but it's still too much.

Having observed the positive and negative aspects, how does Windows 2000 come out in the balance? It depends on your expectations. If you wanted a vastly improved version of NT 4 with a raft of cool new doodads like Internet Connection Sharing, then you'll like Windows 2000 quite a bit. On the other hand, if you were hoping for a rock-solid, enterprise-capable network operating system that could potentially replace your existing MVS, VMS, or Unix systems, then Windows 2000 may disappoint you—

while it's good, I wouldn't feel really confident about the Dow Jones running solely on Windows 2000, nor would I be very happy about finding out that the airliner I was sitting in depended on Windows 2000. But it's definitely a positive step, a step in the direction of more power and better reliability.

The biggest part of Windows 2000, the newest and most extensive piece, is undoubtedly Active Directory. It's a whole new world, even for those already expert in NT 4—so the next chapter is an overview intended to get you started in this pivotal Windows 2000 technology.