



Chapter 1

Windows Server 2003 Overview

IF YOU LIVED THROUGH the change from NT 4 Server to Windows 2000 Server, then you might be a bit gun-shy about Windows Server 2003; how much more will you have to learn, and how hard will it be? If so, then I have good news: while Server 2003 offers a lot of new stuff, there's not nearly as *much* new stuff—if 2000 was a tsunami, 2003 is just a heavy storm. (If, however, you're an NT 4 guy getting ready to move to 2003, then yes, there's a whole *lot* of new stuff to learn. But don't worry, this is the right book, and I'll make it as easy as is possible!)

Clearly explaining what Server 2003 does is the job of the entire book, but in this chapter I'll give you a quick overview of what's new. I'm mainly writing this chapter for those who already know Windows 2000 Server and are looking for a quick overview of what's new in 2003, so if you're just joining the Microsoft networking family then don't worry if some of this doesn't make sense. I promise, in the rest of the book I'll make it all clear.

Four Types of Server

Once, there was just one kind of NT Server. Under 3.1 it was called NT Advanced Server 3.1, which confused people—was there a cheaper “basic” server available?—and so Microsoft just renamed it NT Server 3.5 for its second outing, and it stayed that way through NT Server 3.51. But with NT 4 came a slightly more powerful (and expensive) version called Enterprise Edition, which offered a different memory model and clustering but not much else, so not many chose it.

Pre-Server 2003 Varieties

Under Windows 2000, the basic server was just called Windows 2000 Server, and Enterprise became Windows 2000 Advanced Server. It offered a bit more incentive to buy it than Enterprise had, but not much; its most enticing feature was a new tool called Network Load Balancing Module, something that Microsoft had purchased and decided to deny to the buyers of basic Server. (But it's now shipped in the basic Server, thankfully.)

Microsoft also started releasing a third version of Server called Datacenter Server, but you couldn't just go to the store and buy it—they only “OEMed” it, which means that they allowed vendors to buy Datacenter and tune it very specifically for their particular hardware. The only way that you're going to get a copy of Datacenter is if you spend a whole lot of money on a high-end server computer, and then you get Datacenter with it.

Should you feel left out because you can't buy a copy of Datacenter 2000 and slap it on your TurboClone3000 no-name Web server? Probably not. Yes, there are a few things that Datacenter 2000 can do that the others can't: eight-computer clusters is the main one, but for most of us the loss isn't great. Unfortunately, that changes with Windows Server 2003.

Windows Server 2003 Flavors: Web Edition Makes Four

As you'd expect, Microsoft introduced a number of new features with Windows Server 2003 but didn't make them available in all of the versions. It also added a new low-cost version, Web Edition, and reshuffled the features among the four versions. There are actually a whole pile of different versions of Server 2003 if you include the 64-bit versions, the embedded versions, and so on, but the main product grouping is the four “product editions”:

- ◆ Windows Server 2003, Standard Edition
- ◆ Windows Server 2003, Enterprise Edition
- ◆ Windows Server 2003, Datacenter Edition
- ◆ Windows Server 2003, Web Edition

I'm going to focus on Standard Edition in this book, but let's take a very quick look at each edition.

“REGULAR OLD SERVER” GETS A NAME

For the first time since 1983, the basic variety of server has a name; it is now Windows Server 2003, Standard Edition. (I suspect I may have to sue Microsoft for the extra carpal tunnel damage that I'm getting writing this book—where I could once just say “NT 4,” now I'm typing half a sentence just to identify the product.) In general, it has just about all of the features that it did back when it didn't have a name.

Standard Edition comes with a bunch of new features that are new to all of 2003's editions, as you'd expect, but it also comes with a bit of quite welcome news: Standard Edition includes Network Load Balancing (NLB). NLB's not new, as it was included in Windows 2000 Advanced Server, the more expensive version of Windows 2000 Server. But where Microsoft once required you to buy the pricier version of 2000 Server to get this very useful feature, it's now included in all four editions of Windows Server 2003. (You'll learn how to set it up in Chapter 6.) But that's not all that's new in Standard Edition—for instance, how does, “You finally get a complete e-mail server free in the box” sound? But I'm getting ahead of myself.

WEB EDITION DEBUTS

The newest and fourth option for Server is Web Edition. The idea is that Microsoft really wants their Web server, IIS, to completely crush, overtake, and overwhelm the competition: Apache and Sun Web servers. So they ripped a bunch of things out of Server and offered it to hardware vendors as an

OEM-only copy of Windows Server 2003. It can only address 2GB of RAM (NT has always been able to access 4 or more GB) and cannot

- ◆ Be a domain controller, although it can join a domain
- ◆ Support Macintosh clients, save as a Web server
- ◆ Be accessed remotely via Terminal Services, although it has Remote Desktop, like XP
- ◆ Provide Internet Connection Sharing or Net Bridging
- ◆ Be a DHCP or fax server

So it's unlikely that you'll actually see a copy of Web Edition, but if you do, then don't imagine that you'll be able to build a whole network around it. As its name suggests, it's pretty much intended as a platform for cheap Web servers.

WHAT YOU'RE MISSING: ENTERPRISE AND DATACENTER FEATURES

Back in the NT 4 days, Microsoft introduced a more expensive version of Server called NT 4 Server, Enterprise Edition. It supported clusters and a larger memory model. When Windows 2000 Server came around, Microsoft renamed it Windows 2000 Advanced Server. With Server 2003, Microsoft still offers this higher-end version of Server, but with yet another name change. Now it's called Windows Server 2003, Enterprise Edition. Yes, you read that right: once it was Enterprise Edition, then it became Advanced Server, and now it's back to Enterprise Edition. (Don't shoot me, I just report this stuff.)

Enterprise Edition still does clusters—four-PC clusters now. It also lets you boot a server from a Storage Area Network (SAN), hot-install memory like Datacenter can, and run with four processors.

With Windows Server 2003, Microsoft has finally made me covetous of Datacenter. It has this incredibly cool tool called Windows Resource Manager that basically lets you do the kind of system management that you could do on the mainframe years and years ago. How'd you like to say to your system, "Don't let SQL Server ever use more than 50 percent of the CPU power or 70 percent of the RAM?" WRM lets you do that, and it only ships with Datacenter. Datacenter also now supports eight-PC clusters as well as hot-installing RAM—yup, that's right, you just open the top of the server *while it is running* and insert a new memory module, wait a second or two and poof! the system now recognizes the new RAM, no reboot required.

XP Support Comes to Server

For the first time in a long time, Microsoft shipped NT in two parts, delivering NT Workstation version 5.1—that is, Windows XP Professional and its sadly eviscerated sibling, XP Home—over a year earlier than its NT Server counterpart, Windows Server 2003. I don't think that Microsoft originally intended for there to be a year and a half interregnum, but that unintended extra time let Microsoft make Windows Server 2003 much more than "XP Server"—it's NT Server version 5.2.

XP was a nice upgrade from 2000 Professional but not a great one, not a must-upgrade for current Windows 2000 Professional systems, but a very attractive step up for those running NT 4 or Windows 9x/Me on their desktops. Okay, I might have understated things a bit there—let's go back and italicize that "very." And for people running—auggh—Wintendo (9x and Me) put that "very"

in double-sized bold text. (This assumes, of course, that you have the minimum reasonable hardware to run XP—128MB RAM and a 600MHz processor.) But, again, if you're already running 2000 Pro and you want some you-are-a-fool-if-your-company-doesn't-upgrade-to-XP reasons, then I can't help.

But that doesn't mean that XP didn't introduce some neat features, and now with the introduction of Windows Server 2003, the server side of the NT house has them as well.

XP Integration

Windows 2000 Server came with a file named `adminpak.msi`, which would let you install all of the administrative tools for a 2000 network on a 2000 Pro desktop. I *loved* that, as NT Workstation never really did a great job as an administrator's desktop and I always ended up running Server as my desktop OS. But 2000 Pro was a different story; get `adminpak.msi` on the Win2K Pro box and you could do all the server administration that you wanted.

But then XP arrived.

I was perfectly happy with my Win2K desktop, but it's kind of my job to use the latest version of NT, so I upgraded to XP, only to immediately find that none of the server administration tools worked anymore—the only way to control my DNS server, AD domain controllers, DHCP server, and the like was by either keeping a Win2K machine around somewhere, walking over to the server to work on it, or just using Terminal Services to remotely control the server. It was irritating. Microsoft soon shipped a beta version of administrative tools that worked on XP, but I'm kind of leery of running my actual commercial network with beta tools, if you know what I mean.

So it's good news that Server 2003 brings a welcome addition: a new set of administrative tools that run fine on XP.

Server Understands XP Group Policies

To my mind, XP's two absolute best features from an administrator's point of view were its remote control/support and software restriction capabilities. Both of those capabilities either absolutely require or considerably benefit from group policies, but Server 2000 knew nothing about them, and so required some tweaking to support XP-specific policies on a Windows 2000–based Active Directory. That's all taken care of now.

New Free Servers: An E-Mail Server and SQL Server “Lite”

Thank you, Microsoft.

Not too many people remember this, but back when Server first came out, it wasn't all that impressive in terms of performance. But over time, it took market share away from network OSes that were, in many ways, faster, more flexible, or more reliable. How'd they do it? Many reasons, but I've always thought that there were two biggies. First, NT used the Windows interface, which meant that once you'd mastered Solitaire you were well on the way to administering an NT Server.

The second reason was that NT came with a lot of stuff free in the box. From the very beginning, NT contained software that most vendors charged for. At one time, most server OS vendors charged for the TCP/IP protocol, but NT always had it. Ditto remote access tools, or Macintosh support, or a Web server, FTP, and a dozen other things. In terms of features, Microsoft made NT an attractive proposition.

So I could never understand why they didn't include an e-mail server. Well, okay, I understood it—they wanted to sell you MS-Mail (you in the back there, stop laughing) or Exchange, and didn't want to offer a free alternative. But I've never understood that. Exchange is a mail server that, while powerful, is complex, difficult to set up, and expensive. Why not offer an e-mail server that is nothing more than an SMTP and POP3-based system? It would serve that five-person office well, and they're probably not about to buy Exchange. Nor would it keep the 100-person (or 100,000-person) enterprise from buying Exchange, as they're probably large enough that they want support of shared calendars, IMAP, mailbox forwarding, antivirus add-ons, and so on, and a super-basic POP3 service wouldn't do it.

I got my wish. Windows Server 2003 in all flavors includes a POP3 service. The other part, SMTP, has always existed, so between the two of them, you've got a complete low-end mail server. Again, there are no hooks for antivirus software, no way to set a mailbox to automatically forward somewhere else, and no way to create an autoresponder message for a mailbox ala, "Jack doesn't work here anymore, please don't send anymore mail here to his address," but it may still do the job for you.

The next goodie wasn't on my wish list, but I'll bet it was on a lot of other peoples': a free database engine. Even better, it's a free database engine that is a copy of SQL Server 2000, although with a "governor" and no administrative tools.

For years, Microsoft has offered a thing called Microsoft Database Engine or MSDE. It was never generally available to NT users, but it was available to various groups of developers. The idea with MSDE was that Microsoft took SQL Server 2000—a fairly expensive piece of software—and crippled it in three ways:

- ◆ First, they limited the database size to 2GB. That may not sound like much, but a "real" application of any size could grow beyond that in not too much time. But it's a great size for testing and developing database-driven apps, or for managing a database that will never get very big.
- ◆ Second, they put a "throttle" (Microsoft's word) on it so that if more than five people access it, it slows down. Again, it's a barrier to using this for member registration on a thousand-member Web site, but fine for testing and small networks.
- ◆ Finally, they do not ship any administrative tools for MSDE. If you want to do something as simple as changing the password on the default "sa" account, you'll have to do some scripting.

None of that is intended to sound negative, even though it's true the MSDE is a severely cut-down version of SQL Server 2000. The price is right and once you get past the basic lack of admin interface—the hard part—then you'll find that it's a pretty nice add-on.

General Networking Pluses

XP's new networking features made it to Windows Server 2003, with some extras as well.

NAT Traversal

First, XP introduced NAT Traversal. For those who don't know what that is, NAT Traversal tries to solve the problem of "how do I communicate from inside one NAT network to another?"

More specifically: suppose you've got a cable modem or DSL connection with a connection sharing device of some kind, like a DSL router. The DSL router has two IP addresses. First, there's the honest-to-God, fully routable IP address that it got from your Internet provider, connected to the DSL or cable modem connection. Then there's the connection to a switch that you've got all of your internal machines connected to—the old Windows 9x boxes, NT machines, 2000 systems, Macintoshes, or whatever. The DSL router's job is to share the one “legal” Internet address among several devices. But every device needs a unique IP address. Lots of devices, but just one IP address—what to do?

As you may know, DSL routers solve this problem by giving all of the internal systems—those Windows, NT, 2000, and Mac machines—IP addresses from a block of addresses set aside to be nonroutable. Anyone can use them.

NOTE *By the way, if you've never worked with IP, don't worry too much about this—read Chapter 6 on the basics of TCP/IP on Server 2003.*

There are several of these nonroutable blocks, but most DSL routers seem to use the 192.168.1.x or 192.168.0.x subnets. The DSL routers then use something called network address translation or, more correctly, port address translation (again, see Chapter 6 if this isn't familiar) to share the one routable address with all of the internal systems. How it does it is pretty simple: whenever an internal system wants to access the Internet, perhaps to browse some Web site, then that system just says to the DSL router, “Please forward this request to Internet address so-and-so,” as routers normally do. But the DSL router knows perfectly well that it *can't* do that: if it says to the Internet, “Hey, someone at 192.168.1.3 has a request,” then the first Internet router to see the message will simply refuse to route it, as the address is in a range of addresses that are, by definition, NONroutable. So the DSL router *doesn't* say “192.168.1.3 wants something”; instead, the DSL router substitutes *its* routable address. Then, when the answer to 192.168.1.3's question comes back, the DSL router remembers which machine asked the question in the first place and routes the answer to 192.168.1.3. The result is that to the general Internet, that DSL router sure seems like a demanding system, when in fact it is simply busy because it is impersonating a bunch of systems.

In any case, notice that it's possible for an internal system (one with one of those 192.168.x.x addresses) to initiate a communication with a device on the public, routable Internet, but it's NOT possible for a device on the public, routable Internet to initiate a conversation with an internal 192.168.x.x system.

Here, then, is the problem. Suppose I'm sitting at a Windows 2000 Pro box in my home that has a 192.168.x.x address, accessing the Internet via my DSL router or cable modem sharing device. You're sitting in *your* house, also using some kind of DSL router or cable modem sharing device to access the Internet. We meet on-line and decide to play some networkable game and start to set up our connection. One of us acts as the server and one as the client. The client then initiates communication with the server. That's where the problem appears. I could initiate a communication to a routable address, or YOU could initiate a communication to a routable address, but neither of us has a routable address... and so we can't communicate.

(Note that some of you might be scratching your heads saying, “Mark, I don't have that problem.” In that case, I'm guessing that you use your Windows 98 SE, Windows Me, or 2000-based system as the DSL or cable modem-sharing device. As you know if you read Chapter 6 of *Mastering Windows 2000 Server*, you can easily activate something called Internet Connection Sharing to

make your 98 SE/Me/2000 device into a simple NAT router. But if you do your gaming while sitting at that box, then NAT isn't a problem, as that particular computer has a legal IP address, recall, as *it's* the device connected to the Internet.)

How, then, to create a meeting of the minds in PC-land? With NAT Traversal. The idea is that if your DSL router (or other sharing device), your opponent's sharing device, and your game software understand NAT Traversal, then the two sharing devices work out the details to allow 192.168.x.x-to-192.168.x.x communications with no muss, fuss, or greasy aftertaste. And XP Pro's version of Internet Connection Sharing supports NAT Traversal, so if you replaced your DSL router with an XP Pro (or Home) box, you'd have all the more online gaming options. (And of course it's good for more than just gaming; you could use this for any peer-to-peer communications that must go through a NAT-type router, like Webcam-type videoconferencing—once there's videoconferencing software that understands NAT Traversal.)

NAT Traversal's migration to Windows Server 2003 is, then, pretty good news.

IPSec NAT Traversal

I discussed NAT Traversal as if it were mainly of interest to gamers, and I suppose that at first it was. But you could just as easily imagine 192-to-192 type network communications in business as well. Consider a business with two offices in different cities and about 50 employees in each location. They'd like to connect the offices but don't want to have to buy a dedicated leased line or frame relay between the offices, so they get DSL in each location.

In each location they end up with network addresses that look like 192.168.0. something, but they'd like to communicate from location to location. Their problem is, as you can see, exactly the same problem that the gamers in my earlier example face. So they could just put in NAT Traversal hardware and software and be done with it.

But then they'd be transmitting office-to-office data in cleartext over the Internet. An OK thing in 1993, I suppose, but a definite no-no in these modern times. Running sensitive data over the Internet is exactly what IPSec (Internet Protocol Security) was built for. IPSec (also covered in Chapter 6) converts an IP connection into an *encrypted* IP communication.

The only trouble is that IPSec and NAT don't mix. Or didn't, until Windows Server 2003.

Windows Server 2003 includes a new kind of IPSec that is NAT Traversal-aware. So you can have as many 192 networks as you like, and they can all talk to one another, and securely. Of course, this isn't free—you need firewalls and routers that are NAT Traversal-aware—which is probably one reason Microsoft has started selling network hardware, including some interesting wireless devices.

RRAS's NBT Proxy Eliminates Network Neighborhood Problems

Routing and Remote Access Service (RRAS) has always been a source of troubles, largely due to the fact that one of its main jobs is to allow networking over dial-up lines, and dial-up lines are noise-ridden, unreliable things. Another RRAS problem stems from the fact that you normally use it to connect some remote computer, like a home PC, to a distant larger network, such as your company's network, meaning that your home PC is now a network segment all by itself, and in effect the RRAS server has to act as router, authentication server, and a host of other things.

A side effect of your home system being a network segment all its own is that Network Neighborhood or My Network Places doesn't have much to show, as it normally displays the systems on the local segment. (I'm simplifying but that's basically right.) That doesn't mean that users cannot access

servers on the corporate network; unless configured otherwise, a remote user can connect to any server at the office. But people aren't comfortable using Find Computer or some other way to connect to a server, and unfortunately Network Neighborhood is the tool of choice for many when looking for a server—so an empty NetHood is disconcerting to many users.

Seeing tons of computers in NetHood while in the office and none while at home troubles some users, but Windows Server 2003 can fix that. Server 2003's RRAS server includes a feature called the NetBIOS over TCP/IP proxy or NBT proxy. This basically takes the Network Neighborhood that any system inside the office sees and ships it over to the dial-in system.

Of course, in the long run users are going to have to get used to finding servers and resources by searching the Active Directory rather than browsing NetHood, but this provides a useful interim tool.

DNS Conditional Forwarding Supports Multidomain AD-Integrated DNS

As you learned when creating your Windows 2000-based AD, or as you'll learn when you create your Windows Server 2003-based AD, AD needs a sturdy and secure DNS infrastructure. A big part of the "secure" aspect of DNS comes from a DNS design called split-brain DNS where you essentially keep two sets of books, DNS-wise—a DNS server that the outside Internet sees, which holds the address information for your Web, mail, and FTP servers, and a separate DNS server (or a set of DNS servers) inside your intranet that serves AD's needs.

Split-brain DNS works by bypassing the normal process whereby a DNS server converts DNS names like `www.bigfirm.biz` to an IP address. And it works fine, except when joined with a very useful feature of Windows called Active Directory-integrated zones. You'll learn more about this in Chapter 7, but basically AD-integrated zones let you secure a zone for a DNS domain (like `bigfirm.biz`) with one limitation: the DNS servers for `bigfirm.biz` must be domain controllers (DCs) for an Active Directory domain whose name is *also* `bigfirm.biz`.

Where that presents a problem is the case wherein you want to run more than one Active Directory domain in your intranet. Each AD requires a DNS zone to back it up (and, again, if you're not sure about what these things are, don't worry, I'll cover them in detail in Chapter 7, starting from the basics). If you want to use AD-integrated zones, however, then you'll have to have a separate set of DNS servers for each domain... and that's where the problem lies. It's easy to keep a separate set of books on just one DNS domain, as you divide the world up into two areas: folks on the outside of your network, who only see your external DNS server's information, and folks on your intranet, who see your internal server's DNS information and incidentally can also see DNS information on the outside world—so even though the folks inside your intranet are being deceived, so to speak, about the contents of your internal Active Directory's associated DNS data (`bigfirm.biz` in my example), they get the unfiltered DNS information about other DNS, like `microsoft.com`, `whitehouse.gov`, and the like.

Now add that second internal domain; let's call it `acme.com`. To make the `bigfirm.biz` folks see the correct separate set of books, you point all of their servers and workstations to the internal DNS servers that contain the internal-only version of the `bigfirm.biz` information. Recall that these servers must be Active Directory domain controllers for the `bigfirm.biz` AD domain. To support the people in `acme.com`, you'd set up a different set of DNS servers for your internal-only information for `acme.com` and point all of `acme.com`'s servers and workstations to those `acme.com` DNS servers.

People in `bigfirm.biz` can, then, get the internal-only DNS information about `bigfirm.biz`, as well as the public DNS information for any other domain. People in `acme.com` can get the internal-only DNS information about `acme.com`, as well as the public DNS information for any other domain.

Here's the problem: if a bigfirm.biz member wants to log onto some resource on acme.com, then that bigfirm.biz-ite will have to find a domain controller for acme.com, as DCs handle logons. But you find DCs in Active Directory via DNS. A bigfirm.biz user, however, uses DNS servers that know the internal-only information about bigfirm.biz, not acme.com. So if someone in bigfirm.biz tries to look up a DC in his local DNS server, that local DNS server will end up asking the public DNS server for acme.com, "Where are your DCs?" The answer will be a puzzled look from the public DNS server for acme.com, as it has no clue what a DC is.

There are workarounds for this, but Windows Server 2003 offers a terrific one: conditional DNS forwarding. It lets me set up the bigfirm.biz DNS servers by saying, "OK, you already know the internal-only information about bigfirm.biz. And you know that if you have to find out DNS information for someone else, like www.google.com or www.cnn.com, or the like, then you go search the public Internet. But here's a new bit of information: on the off-chance that you ever need to find out information about a zone called acme.com, then go straight over to that server over there (pointing to the internal-only acme.com DNS servers) and it'll have the answer." A great new feature for folks rolling out Active Directory forests with more than one domain. You'll see it at work in Chapters 7 and 8.

Active Directory Improvements

For a first try, Windows 2000's Active Directory was pretty good... not bad for a 1.0, Microsoft. (Of course, they *did* have Banyan and Novell's directory services to learn from, but let's ignore that for this discussion.) In Windows Server 2003, Microsoft dishes up a 1.1 version of AD that solves several irritating problems, makes running branch offices easier, and expands AD's flexibility.

While I don't want this to sound negative, it's a fact that Active Directory still suffers from most of its inflexibility—there is no simple way to rearrange the structure of an existing forest, to merge forests into one forest, or to break off a piece of a forest and make it a forest of its own. Don't think that those scenarios are marginal or unusual ones—they're not. The reorganizations that most organizations undergo every year or so will often require rearranging a forest. Two firms merging need to be able to merge their forests as well. And a firm divesting itself of a subsidiary would want to be able to detach one or more domains or trees from a forest. But perhaps that will appear in a future version of Server; let's hope so.

Meanwhile, the 2003 edition of AD has, again, some very good news. Here's a look at its high points.

Forest-to-Forest Trusts

Combining a bunch of AD domains into a forest offers two main benefits: first, those domains all automatically trust each other, and, second, the domains share a set of "super" domain controllers called global catalog (GC) servers, which are domain controllers that contain a subset of information not just about their own domains but about every single domain in the forest. Doing away with the unreliability of NT 4 trusts for the convenience and dependability of AD's automatic trusts is a big win for AD users.

But, as I suggested a few paragraphs back, AD forests were and are still pretty inflexible. So suppose you're an organization that finds itself with more than one forest, and you need to get those forests to share things? Well, there's always been the hard way—get a migration tool and copy all of the user accounts, machine accounts, and other objects from Forest 1 to Forest 2, then just plain delete Forest 1. The problem with that answer is that while migration tools are pretty nice, they don't do the whole job and they're a lot of work to get working.

With a Windows Server 2003–based forest, however, you have a new answer: forest root trusts. With these, you just build one new trust relationship between Forest 1 and Forest 2 and instantly every domain in Forest 1 trusts every domain in Forest 2 and vice versa. Cool; thank you, Redmond.

But I said that forests had two main features—complete trust and a centralized database of forest information called the global catalog. A forest-to-forest trust gives us back the first benefits of a single forest; what about the second? Unfortunately, two forests that trust each other do not share a global catalog. That means that forest trusts will not let applications that are GC-dependent see the trusting forests as one single overall directory. What apps are GC-dependent? Well, the most prominent one is Exchange 2000: it really wants to see your organization as one big forest. Forest trusts don't solve that problem.

I was surprised to learn of another limitation to forest trusts: they're not transitive. Interestingly enough, if Forest 1 trusts Forest 2 and Forest 2 trusts Forest 3, then Forest 1 does not trust Forest 3. Bummer. And *none* of this forest trust stuff works at all until you've upgraded every single DC in every single domain of both forests. So, overall the forest trusts are a good step forward... but not the whole story.

Group Replication Problem Solved

It's always been ironic that while Active Directory can support a far larger user list than could NT 4 domains, AD couldn't support *groups* as large as NT 4. You can create literally millions of users in an AD, but because of a quirk in AD's method of keeping domain controllers' information consistent ("AD replication") in combination with the way that group membership is stored in AD, you can't put more than about 5,000 users into a group.

In 2003's AD, Microsoft restructured the way they store group membership, and now the sky's the limit. It also solves another problem wherein it is possible in 2000's AD that you and I work in the same world-wide company and you change a group's membership while sitting in the Edenton office while I change that same group's membership while sitting in the Port Angeles office, and one of our changes overwrites the other person's changes. With 2003, that's fixed.

To get this benefit, you must upgrade all of the DCs in all of the domains in your forest.

Good News for Branch Offices

Branch offices have always presented a problem for IT folks. Many firms have one or two large centralized locations and dozens (or hundreds!) of small offices housing a dozen or two employees. These small branch offices are important but expensive to run, as a firm typically has to install some kind of persistent connectivity—frame relay, DSL, T1, cable modem, or the like—to the branch office so that the employees there have access to the corporate intranet and potentially the Internet.

As branch offices are typically served by only one WAN link and WAN links aren't always so reliable, companies have to make some tough choices: do we put a domain controller on every site? Does each site need a DNS, WINS, DHCP, etc. server? If we put servers on a branch office site, will they do so much chattering over the WAN link with the servers in the central office that they'll chew up a significant proportion of that link's bandwidth? And most importantly, when the WAN link is down, how do we ensure that the employees in the branch office can still get logged in and remain productive?

Server 2003 can't solve all of those problems because, well, unreliable WAN connections aren't Microsoft's fault. But 2003 offers some changes that will make setting up and maintaining branch offices easier.

SIMPLIFIED BRANCH OFFICE DC INSTALLATION

I've helped a number of firms get AD up and running. Sometimes, however, they call me back to help out with a particularly difficult part. In one case, it was the Case of the Dial-Up Office.

This company had a branch office that did not have a persistent connection either to the Internet or to the head office; instead, they dialed up when necessary. And they were having trouble getting a domain controller set up in that branch office. Now, you see, to create a domain controller, you start from a regular old vanilla Windows Server, either vintage 2000 or 2003, and run a program called DCPROMO, a wizard that will convert a member server into a domain controller or will decommission a DC back to a member server. In order to create a new DC, you must have a live connection back to the main office, so before trying to set up the DC I dialed out to the Internet and from there established a connection to the “mothership” back at HQ.

DCPROMO started out fine, accepting my credentials and okaying the idea of promoting this member server. But a new DC needs a copy of the Active Directory, so DCPROMO's last act is to hook up with another DC and download the latest version of AD. This firm had a few thousand employees, so their AD was actually not too large—under 10MB.

Did I mention that their phone line was a bit noisy? That it only connected at about 26 kilobits? And that it tended to disconnect at inconvenient times?

Anyway, DCPROMO would try to start replicating and get partway through... and then the line would hang up. Sometimes a reboot and another DCPROMO would get us back to member server, where we could start all over again; in a couple of cases, I had to reinstall Win2K Server from scratch. After only about a day of trying, though, I found that the phone lines were quiet and clean enough around midnight to allow the initial replication to complete. Grrrr.

I really would have welcomed Windows Server 2003 in that case. With Server 2003 you can take a backup of your AD domain database with you to the remote site, and DCPROMO then lets you start a new DC out from the backup of the AD, rather than forcing a complete initial replication over the WAN. From there, you connect the new DC up to that unreliable phone line, and all the DC must do is to replicate whatever's changed in AD between when the backup occurred and now, which usually isn't much.

This feature does *not* require you to upgrade every DC in creation; in fact, this works fine if the very first Server 2003-based DC in your network is the one that you're installing in that branch office.

BRANCH OFFICE REPLICATION CONTROL

Should you put a DC in a branch office or not? It's not an easy question. On the one hand, having a local DC in a branch office means that when the WAN link is down the local users can still log on. On the other hand, having a local DC means that DC must keep a complete copy of the entire domain's Active Directory database. So if there are 15 users in the branch office and 50,000 members of the domain, every time those 50,000 people change their passwords those changes must be replicated across the WAN link to your branch office's DC. (That's an example of what I meant when I said earlier that server communications can seriously burden the WAN links to branch offices.)

AD has always tried to limit its effect on branch offices in a couple of ways. First, it uses a routing algorithm that is designed to enable it to get data from a DC in one office to a DC in another office in the least-cost way. Second, it compresses the data before moving it between DCs. Those both sound like good features, but Server 2003 improves upon them.

First, there is a large body of literature about optimal routing algorithms... but the Microsoft programmers working on AD in Windows 2000 didn't employ them. Instead, they made up an algorithm all their own. (Why? I don't know. But I do know that many firms, Microsoft included, are sometimes struck by what's called the "NIH syndrome"—short for Not Invented Here. It refers to the fact that it's more fun to sit down and reinvent your own wheel than it is to merely reimplement someone else's wheel.) Microsoft found that AD bogs down when faced with more than a few hundred sites; implementing industry-standard algorithms shot that up into the multithousand-site range.

Second, odd as it sounds, apparently some branch offices found that the CPU power required to compress and uncompress data outweighed any benefits gained from bandwidth recovery. So in Server 2003, Microsoft lets you choose to shut off intersite compression.

Both of these features require that you upgrade every DC in every domain in your forest.

BRANCH OFFICE LOGON INFO CACHEABLE

When the WAN goes down, does everyone get a day off? Well, that's essentially true if they need the WAN to do a logon. Windows 2000 and later systems require several ingredients in order to log on. First, of course, a workstation must be able to find a domain controller; that's always been true. Second, Active Directory member machines need to be able to find a global catalog server in order to log a user on.

It is, then, possible that you might have a local DC but not a local GC. In that case, a WAN failure means that you'd only be halfway to logon, so you're logged on with "cached credentials." One answer is to put a GC on every site, but that can be very expensive in terms of WAN bandwidth: GCs not only replicate from other DCs in their same domain, GCs also replicate from every other domain in the forest!

AD 2003 offers a nice workaround: Server 2003-based DCs will locally cache the information that they need from a GC. So if you logged on yesterday from your branch office, your local DC collected enough information over the WAN from your GC that it was satisfied to let you log on. If the WAN's down today then your local DC remembers that it logged you on yesterday, and logs you on today.

The best part of this news is that it requires no other upgrades—the DC in your branch office can be the first Windows Server 2003 introduced into your enterprise and this will still work fine.

Domains Can Be Renamed

One of 2000's most annoying AD limitations was that it prevented you from renaming a domain; if Bell Atlantic had had an AD forest when it merged with GTE and was renamed Verizon, there would have been no way to rename an AD domain named bellatlantic.com to verizon.com. Now you can rename a domain, but it's not a simple matter, even now.

First, you will have to be completely Server 2003ed in the domain: every DC in the domain to be renamed (not all DCs in the forest, just the ones in the domain) must be running Windows Server 2003. And second, there are... well, I was going to write "... a few steps to perform in order to complete the domain renaming," but the truth is that Microsoft has a white paper online explaining how to do it.

The paper is *60 pages long*. So it's *possible*, just not easy, at least not yet.

AD Can Selectively Replicate

Active Directory is a database, and domain controllers are database servers, just like systems running Access, Oracle, MySQL, or SQL Server and holding some other kind of database. (Well, not *just* like...

DCs do not respond to SQL queries. Instead, their query language is LDAP.) While the AD database was originally designed for storing user accounts, machine accounts, and the like, there's no reason application designers can't take advantage of AD's built-in database engine to store other information.

Microsoft's own programmers did just that when designing 2000's DNS server. As you may know, 2000 introduced you to the option to create a DNS zone that was an Active Directory–integrated zone. A zone of that type stores the DNS info for your systems in the AD itself and replicates it along with the normal domain information from DC to DC. But *only* DCs get copies of the database, so if you choose AD-integrated DNS, all of your DNS servers must be DCs.

But now consider: what if you had a lot of DCs, but only a few of them were DNS servers? Wouldn't that be a bit wasteful? You'd use precious bandwidth to replicate DNS info to every DC, whether it used it or not. Server 2003 solves that problem with the notion of an *application partition*. Partitions are subsets of the AD that only replicate to a subset of DCs. Microsoft then applied that notion to their DNS servers, so in a network using AD-integrated zones only the DCs running DNS will get the DNS info. This feature doesn't require any preparation; you get its benefit on any DC running Windows Server 2003.

Remote Administration Upgrades

For years, remote administration and control of Microsoft operating systems drove me nuts. It seemed only Microsoft OSes required you to be physically sitting down at a computer in order to control the software running on it. Sure, there were third-party alternative tools like PCAnywhere or VNC, but remote control/admin always seemed like something that really needed to be “in the box,” integrated into the OS.

Windows 2000, then, was a great advance, incorporating remote Telnet sessions and a remote control tool called Terminal Services that was a cut-down version of a program from a company named Citrix. Terminal Services only ran on Server, though, so remote control of 2000 Pro boxes was dicey. But then came XP and now Windows Server 2003.

First, the workstation/desktop version of Windows Server 2003, Windows XP Professional, includes Microsoft's adaptation of Citrix's remote control product. It and the server version of Terminal Services are built around a tool called the Remote Desktop Protocol (RDP). Microsoft has improved RDP to make it run on slower connections, and I'm not exaggerating when I say that remote control over a 40-kilobit dial-up connection works very well, almost as well as sitting at the computer.

RDP also matures in that it automatically gives your remote control session access to your local printers and drives, something that Terminal Services for Windows 2000 couldn't do. It supports colors beyond the simple 8-bit, 256-color of Windows 2000's RDP, and transports sound as well.

Perhaps even better, Windows Server 2003 and XP repackage RDP in two forms: *remote desktop support* and *remote assistance*. These are ways to provide remote control or offer remote assistance but are nothing more than new user interfaces placed atop Terminal Services. If you've not used them yet for XP, I think you're really going to like them on Windows Server 2003.

Finally, Windows Server 2003 offers a completely new set of remote control tools in the form of Web pages. You can install a bunch of modules on your server that will let someone do approximately 80 percent of the administrative functions you'll ever need, all through a secure Web connection. The bottom line is that we don't have to put up with those Unix guys kicking sand in our faces telling us that their OS is more manageable!

Command-Line Heaven

Okay, I admit it, the command line is harder than the GUI. GUI-based administrative tools walk you through a process and offer tons of online help and wizards while they're at it. The command line is definitely an acquired taste. But may I offer a very heartfelt bit of advice?

Acquire the taste. You'll be glad you did.

Take a common problem that I hear about a lot: a private DNS root. Through a process that I'll cover in Chapter 7, it's possible to set up a DNS server that lives in its "own private Idaho," and is unable to resolve names on the rest of the Internet. It happens through a common bit of misconfiguration. And it can be fixed from the GUI, with about two paragraphs of explanation. Or you can just open up a command line and type

```
dnscmd /zonedelete /f .
```

Then press Enter and it's done. (Most of the time, but I'm keeping this simple.) Command lines let you type a few dozen characters and accomplish amazing things. Just a few keystrokes can often accomplish quite a lot.

But how's that different from saying, "Use the GUI, and in a few dozen mouse clicks you can get a lot done?" Well, that's true, you can. But the command line offers two more things:

- ◆ First, simply opening a Telnet session lets you run one of those powerful command-line commands on a remote computer, so it's a great way to do remote administration. "Wait a minute, Mark," you say, "didn't you just tell me a page or two back how well Terminal Services runs in low bandwidth?" Sure, but command-line sessions run in even *lower* bandwidth. Imagine administering your computer remotely with nothing more than your cell phone and either a wireless keyboard or a bit of patience and the phone dialing keyboard. It's possible with command lines.
- ◆ Second, suppose you have some repetitive administrative job, something that needs doing pretty regularly or, worse, regularly at some inconvenient time, like 3 A.M. daily. It's a task so simple that you could train a monkey to do it... if they'd only let you hire monkeys and give them administrator accounts. Instead, you can create an "e-monkey." Figure out how to do the task from the command line. Then type those commands into an ASCII text file with Notepad. Give the file the extension `.CMD`. And whammo: you've just written a batch file that you can schedule to run at 3 A.M. Try writing a batch file that stores *mouse clicks* and you'll see how neat the command line can be!

Windows 2000 made some great strides in offering better command-line tools, but didn't go all the way. With Windows Server 2003, it's actually possible to do about 98 percent of your administration from the command line.

Desktop Support Improvements

Most of you don't use Server as a Desktop operating system, so you wouldn't expect much in the way of improvements to Desktop control, but recall that Windows Server 2003 incorporates all of the new things that came to XP. If keeping Desktops up and running is part of your job, then you'll like what Windows Server 2003 brings, although in most cases you need XP on the Desktop to see Server 2003's improvements.

Profiles and Policies

When they first arrived, roaming profiles seemed like a great idea. . . but then we tried them. Slow, prone to breaking. . . auugh. But Windows 2000 made them more palatable, and so has Windows Server 2003.

First of all, there's a new group policy that you can apply to a machine (or machines) that says, "Ignore all roaming profiles." This is terrific—now I can ensure that just my laptop and desktop get my roaming profile, by setting up all of the public access/shared systems and the servers to "ignore roamers."

Another group policy makes roaming profiles better for laptop users. Sometimes I'll check into a hotel and find that it offers Ethernet connections to the Internet (yippee! I will sleep on a *stone floor* if it means I get high-speed Internet access), so I plug my laptop into the Ethernet and boot it up, only to realize that my stupid laptop is trying to suck my roaming profile over the Internet. A half-hour later, it gives up.

Or at least that's what *used* to happen. Now I just set the group policy on my laptop that stops and asks, "Do you want to download your roaming profile?" I say no and log on in seconds. (Of course, the laptop must be running XP.)

Those are just two examples of the new things you can do to control profiles; there is a ton more, as a look at the Group Policy Editor (which you'll meet in several places in the book) shows.

Software Restriction Group Policies

Every help and support desk person has a little list of things she'd like to see. One is almost always, "I'd really like to keep users from running particular programs on the system." (If you're having trouble thinking of examples, then see if the names Morpheus or Kazaa ring any bells.) With XP desktops, you can do that.

XP and Windows Server 2003 include a whole new set of group policies called software restriction policies. With them, you can tell a Desktop, "Nothing runs except Word, Internet Explorer, Outlook, and the Palm Desktop." It's pretty neat and pretty powerful, and you can learn more about it in Chapter 9.

The Group Policy Management Console (GPMC)

After reading the last page, you may be shaking your head saying, "Yeah, that's nice and all, but you're talking about group policies? Those guys are a nightmare." Yes, they can be, particularly when a group policy refuses to run—"Let's see, I just created this policy that keeps Access from running on Ronnie's desk and he can *still* run Access!" Several things might keep your new policy from running—Ronnie's Desktop might not have refreshed policies, or it might have refreshed policies but your policy might have been overridden by another policy. You look and see that there are only 24 other policies that apply to Ronnie and his Desktop, so time to start sifting through policies. . . or not.

Microsoft has been working on a really terrific group policy troubleshooting tool called Group Policy Management Console. It *didn't* ship with Windows Server 2003, but as of this writing Microsoft expects to give it away free on their Web site by March/April 2003. You'll learn more about it in Chapter 9.

Tightened Security

Sometime in late 2001, two things occurred to Bill Gates: first, network security is important and, second, Microsoft software is buggy as heck when it comes to security (among other things), so a lot of Microsoft security is lacking a bit. So he derailed virtually all of Microsoft's coding efforts for two months as Microsoft trained nearly everyone about security.

In the end, this was a good thing. NT has always had a reputation of being an insecure operating system, but it's an inaccurate reputation. NT (3.1–4, and Windows 2000) is an extremely secure OS in that it provides the option to lock many things; a properly tweaked NT server is a secure server indeed. NT's reputation comes, however, from the fact that a default installation leaves the vast majority of those locks unlocked. For Windows Server 2003, that changes.

For example, NT 4 and Windows 2000 installed an unsecured Web server by default on every server you ever installed. Not a good idea, as we learned in June 2001 when a worm called Code Red infected millions of servers—*though the Web server*. (As I write this in late 2002, there are still thousands of servers out there infected with the Nimda virus, a year after Nimda's arrival.) With Windows Server 2003, in contrast, you don't get IIS unless you ask for it. And even then, it's a pretty locked-down version of IIS. (You'll learn how to set up IIS in Chapter 17.)

To see another example, look at the NTFS permissions on the C: drive of any Windows Server 2003. Where the default permission for every previous version of NT was Everyone/Full Control—"C'mon in, y'all, we're all friends here!"—Windows Server 2003 gives Everyone only Read and Execute permission on the root of C:. The Users group has more power, as it can read files and create folders on C:, but it cannot create new files on the root of C:. You can change all of this, of course, but by default Windows Server 2003 is a bit tighter security-wise than its predecessors.

That's a good thing. But it won't be an unmixed blessing. I'm sure that at least once in your Windows Server 2003 career you will be sitting at the server trying to get something done but getting nowhere. You've got Help open, or a book at your side—this one, I hope!—clicking where the book says to click and dragging where the book says to drag, but it's not working. In that case, you may be doing the right thing but lack the permissions to do it. So Windows Server 2003 offers you one more impediment to getting our jobs done: you'll have to wend a maze of security to do some things.

But don't take that as a negative comment. It is simply a fact of life in the twenty-first century that there are tons of dirt bags out there and the Internet has now given them the chance to come knock at your door so we have no choice but to install locks on our doors. Yes, it was nice back in the days when we didn't have to lock our doors or carry keys, but those days are gone forever. NT 5.2 changed, yes, but it was just changing with the times.

Reliability

Continuing from the last section's topic, what makes an OS secure? In addition to the traditional security topics, like the ones that I just discussed, there's a more visceral sort of security—do you trust the thing not to crash on you?

In general I have always found NT to be sturdier than its compatriots; I think that no one would argue with me when I say that it's always been more reliable than Windows 3.x, 9x, and Me. I'd argue further that it was more reliable than the Mac, at least through OS 9.x. (OS/X is a completely different story; I think Apple did a great thing with OS/X—the result will be eventually be, I think, both Apple and Microsoft sometime in the future both offering OSes so reliable that we'll actually trust those OSes implicitly. Unfortunately we're not there yet. But I think it's possible.)

Windows 2000's System File Protection and Driver Verifier made great strides in making Windows 2000 far sturdier than its NT 4 predecessor; XP took that further with System Restore, Application Verifier, and Driver Rollback. As with some other Windows Server 2003 features, they're not exactly new, as they first appeared in XP, but they're new to Server. Unfortunately, one of the three,

System Restore, apparently doesn't come with Server, and that's puzzling: it's an XP tool that lets you roll back the entire state of a system to some time in the past, undoing the effects of installing some new unreliable program that's made your previously reliable system wobbly. I don't know why they left it out of Server; perhaps we'll see it return with a future version of Server.

Driver Verifier was—and is—a useful tool for checking up on new device drivers and other system-level programs. It was a great addition to 2000 and still is, with Windows Server 2003; smoking out problems with kernel-mode programs is far easier with its help. Application Verifier performs a similar service, but for user-mode programs.

Have a program that ran fine under NT 4 or Windows 9x but won't run under Windows Server 2003? Then run it under Application Verifier. When it fails, Application Verifier will tell you what caused it to fail and, even better, it can add information to the application that lets it run under Windows Server 2003.

Another source of operating system instability can be new drivers. You've got the system running fine, but the vendor of one of your pieces of hardware comes out with a new driver. As it looks like you're running smoothly, you're leery about chancing it with a new driver... there must be some subtle bug that someone found that this updated driver fixes, but this new driver could make your system unstable... what to do? Well, Driver Verifier is a great way to check out a new driver, as it was in Windows 2000. But now it's got a simple partner in Driver Rollback. You load a driver and decide that it's no good... now, where did you put the old driver? Just go to Device Manager, find the device with the new driver, right-click it and choose Properties... you'll see a new button, Rollback Driver. Like XP, Windows Server 2003 keeps the previous version of all drivers.

Storage News

XP and Windows Server 2003 brought some much-needed fixes to NTFS and one great new feature: volume shadowing.

In brief, volume shadowing lets you take snapshots of a file share. At predetermined times of the day, Windows Server 2003 will record the status of whatever it's shadowing and let you roll back to that quickly and easily. For example, suppose you keep your important documents in a share `\\serv01\documents`. You could tell Server 2003 to take snapshots—*shadow copies* is the Microsoft term—of the files in that share at 7 A.M., 10:30 A.M., noon, and 6 P.M.

A few days later, at 10:15 A.M., you realize that you've accidentally deleted an important document. But all's not lost; just fire up the shadow copy client software (included with Server 2003) and restore the 7 A.M. version of the document. A few hours' work lost, but that's all. And no need to go find the tape librarian and beg to get a tape with last night's backup mounted.

Volume shadowing lets you create a kind of imaginary copy of a file, with the state of that file frozen in time. That means that you can take shadow copies of open files and then back up the shadow copy! For example, suppose you have a SQL database that you need to back up every day, but there's never a good time to stop the database server. No problem: take a shadow copy at 3 A.M. That copy does not change on a second-by-second basis, unlike your real SQL database file, so you can back it up at your leisure.

I told you that NTFS got some other improvements; they include

- ◆ NTFS clusters can be any size, unlike Windows 2000, where their cluster size could not exceed 4KB or the volume could not be defragmented.

- ◆ A server can now host as many Dfs (Distributed File System) roots as you like; Windows 2000 only allowed each server to host just one root.
- ◆ Offline files can now cache encrypted files.
- ◆ You can set up encrypted files so that more than one person can view an encrypted file.
- ◆ You can now both compress and encrypt a file.
- ◆ EIDE drives can now run independently, meaning that you can run a small database server with two EIDE drives rather than SCSI drives—one drive for the database, the other for the transaction log. This was always possible in NT, but never made sense, as EIDE drives were limited to only run one at a time—if your SQL software said to the hardware, “Save these bytes to the database file and those bytes to the transaction log,” then in actuality the OS would make the EIDE drives take turns. It might first write the bytes to the drive holding the database file while the drive holding the transaction log cooled its heels, and then write to the transaction log while keeping the database idle. The techie term for this would be that EIDE drives are now *asynchronous*, at least when they are on different channels—for example, this works if one hard disk is on the primary EIDE channel and the other is on the secondary EIDE channel.

None of those are truly earth-shaking, but they’re all quite welcome improvements. Which brings me to my last point in this chapter...

Windows Server 2003: Not Yet or Good Bet?

Should you upgrade? Is it worthwhile to move up to Windows Server 2003, Standard Edition? That’s a really tough question.

On the one hand, it’s hard to point to any one feature that grabs you by the throat and says, “You gotta have me.” For some people it’ll be the new Active Directory stuff, either forest roots, domain renames, or the new branch office–friendly features. Or it might simply be that they’ve been waiting to go to a full-blown LDAP-based directory service like Active Directory for a while but were leery of the version 1 feel of Windows 2000’s AD. But are these reasons to toss out an already-existing infrastructure built on Windows 2000 Servers? Buying all of those server licenses might be a hard sell in a place with a lot of servers. For those with just a handful, then the upgrade might be simple, not too expensive, and the fact that you needn’t buy new client access licenses when upgrading to Windows Server 2003 has to make Server 2003 go down easier. But again 2003 seems to lack that one killer feature.

Furthermore, as I wrote this book I found time and time again that some section of Windows Server 2003 didn’t do anything that Windows 2000 Server didn’t do but that Microsoft had changed the user interface, wizards, syntax or the like. As a result, much of the time that I spent researching the book was time spent trying to figure out how to do something that I’d already figured out in 2000!

On the other hand, Server 2003 has a real preponderance of attractive features. Even the much-maligned (by *me*, to tell the truth) XP user interface has been toned down in Windows Server 2003 and is pretty nice—it’s convenient in the Active Directory tools to select a group of users and do one operation on them, or to just drag and drop them between organization units. The more I work with Windows Server 2003, the more I like it. This is always true, of course—features that you first think are kinda okay soon become “man, do I miss them” when running an earlier version of the operating system. Some people will find particular small aspects compelling, as in the case of conditional DNS forwarding.

I first met Windows Server 2003 in its beta 2 form in 2001, and I can't say that I was impressed. But from beta 3 onward it's grown on me and as I write this, just before its final release, I can say honestly that I will replace all of my Windows 2000 Servers with Windows Server 2003s, as soon as I can. That's not to say that I think that all of you should do that—read the rest of the book and decide for yourself.

As you can see, there's a lot of fun new stuff to play with and learn about in Windows Server 2003. But Windows Server 2003 is sort of the second chapter in the second book in a series—NT 3.1, 3.5, 3.51, and 4 were basically chapters in the first book, and Windows 2000 was the first chapter in the second book. Some of you have been following along with the Server story and you're ready for the new Server 2003 stuff; but for those of you just joining us, we've got the next chapter, which brings up to speed those who are new to Microsoft networking. So if you're already NT-savvy, skip ahead to Chapter 3. If you're new to the Microsoft networking game, or just want a short refresher, then turn the page and let's review The Story So Far.

