

Part 1

Windows Server 2003 Service Pack 1

In this part:

- ◆ **Chapter 1: Getting and Installing SP1**
- ◆ **Chapter 2: Hiding Folders from Prying Eyes: Access-Based Enumeration (ABE)**
- ◆ **Chapter 3: De-Worming Windows with Data Execution Prevention (DEP)**
- ◆ **Chapter 4: Stacking the Deck Against Bad Guys...The IP Stack, That Is**
- ◆ **Chapter 5: Solving SP1/R2 Incompatibilities: Understanding the “De-anonymizers”**
- ◆ **Chapter 6: Fine-Tuned Security Monitoring: Per-User Auditing**
- ◆ **Chapter 7: Stopping Spyware: Controlling ActiveX and Browser Helper Objects**
- ◆ **Chapter 8: Locking Up the Ports: Windows Firewall**
- ◆ **Chapter 9: Thwarting Mobile Thieves: Blocking USB Memory Sticks**
- ◆ **Chapter 10: Supporting Clients with Windows Terminal Services**
- ◆ **Chapter 11: Tightening Security with the Security Configuration Wizard**



Chapter 1

Getting and Installing SP1

As you've already read, you can get an awful lot of the features ascribed to R2 free of charge; just install SP1. And while I'm sure that many of you have already installed SP1, I *also* know that unfortunately some of you have been waiting to take the plunge. (And yes, I *additionally* know that it's more than a year since SP1's release—but I'm constantly amazed when I visit clients that so many of them are still leery of SP1. Install it, I say!)

In this chapter, I'll try to eliminate one of the reasons for the holdouts not to install SP1: the fear that installing SP1 will be difficult. You'll learn where to get SP1, how to install it, how to pre-install when creating new 2003 servers, and—just in case!—how to uninstall it. (Not that I've ever needed to.)

Do I Have SP1 Already?

As you're about to read, Microsoft's got a couple of systems in place that may have installed SP1 on your server so quietly that you may not have noticed that you *have* SP1. So here's a pretty reliable way to find out whether you're running the original year-2003 version of Windows Server 2003—the Release To Manufacturing or RTM version—or 2003 with SP1 installed.

Whenever Microsoft ships an operating system, they set the time and date of almost all of that OS's files to some particular date. They time-and-date stamped the RTM files as March 25, 2003, and the SP1 files as March 25, 2005. So find out your system files' dates like so:

1. Open a command prompt.
2. Type `dir %windir%\notepad.exe` and press Enter.
3. The date on the Notepad file will indicate whether you're RTM or SP1.

There's another just-as-easy way: right-click the My Computer icon, and choose Properties. Under the text System in the resulting page, you'll see the name of your operating system. If you see the line "Service Pack 1" under the OS's name, then you've got SP1; if you don't see any references to service packs, then you're probably RTM. I say "probably" and gave you two ways to figure out your SP level—there are others, like looking at Help/About on most Windows utilities—but occasionally I need more than one "corroborating witness," as sometimes I'll be working at a client's machine trying to fix something, and the client doesn't tell me that the reason the server is messed up is that the client started installing SP1, decided to reboot in mid-stream, and now things aren't working out the way the client would have preferred.

Getting SP1

Assuming that you need SP1, you can get SP1 in a number of ways:

- ◆ Automatically via Microsoft Update
- ◆ Automatically through Windows Server Update Services
- ◆ Order it on CD from Microsoft
- ◆ Download it from Microsoft

Anyone who's used a computer in the past few years knows by now that (1) Microsoft code has bugs, (2) some of those bugs can enable bad guys to write programs that enable those bad guys to control our computers, and (3) sadly, this isn't just a remote possibility, it happens a few times a year. Worms with names like Code Red, Nimda, Spida, SQL Slammer, Blaster, Sasser, and Zotob have, at various times between early 2000 and now, caused havoc amongst Microsoft users and the Internet as a whole. So most of us know by now that Security Rule Number One is "patch!" The second Tuesday of every month, Microsoft releases announcements of discoveries of various security bugs and patches to fix those bugs so that the worms stay away. This monthly event, known as "Patch Tuesday" to us already overworked administrator types, means that we've got to get the latest patches and get them on our systems.

Most of us get and deliver those patches in one of two ways. First, we can have every one of our servers and workstations connect daily to a special website run by Microsoft called Microsoft Update.

TIP Despite the fact that the website in question is probably one of the ten most visited and well known websites in the universe, let me offer its URL for the sake of completeness: <http://update.microsoft.com>. (And I do mean "universe"—I've heard rumors that Microsoft's web server logs indicate some patch downloads occurring to clients using IPv35!)

Any system running Windows 2000 (Server or Pro) with SP3 or later, Windows XP with SP1 or later, and any copy of Server 2003 have built into them some web tools that can be configured to automatically hook up to Microsoft Update. It's a nice, convenient way to get patches without thinking about it. But it *can* be a bit of a pain in the neck, as there have been a few Patch Tuesdays when I left a file unsaved on my Desktop only to stumble into my office the next morning with a hot mug of Earl Grey (Twinings or Stash, of course) in my hand and see the Windows "Please press Ctrl-Alt-Del" Welcome screen on my monitor. (I'm such a Boy Scout that I configured Microsoft Update to reboot my system automatically if it deems it necessary, so it's really my fault, not Microsoft's.)

Microsoft Update's not a bad answer for patching, but if you've got a bunch of systems, then it can be horribly inefficient in terms of download bandwidth. For example, a while back Microsoft released a service pack for Internet Explorer 6 that ran around 10MB. Now, that may not seem like much, but consider what happened on the evening of Patch Tuesday at a company with 1000 workstations. In total, that firm would have downloaded that same 10MB 1000 times, turning a small 10MB download into a bandwidth-squandering 10 GB. Ugh.

So Microsoft offers Windows Server Update Services (WSUS), an application that you can run on a 2000 or 2003 server. WSUS acts as a kind of "local Microsoft Update." Whatever server you've installed WSUS on sucks down the patches from Microsoft's Microsoft Update servers, and then you configure your internal systems to no longer check with Microsoft's servers for patches but instead to look to your WSUS server.

TIP WSUS is a big topic and not one we're covering in this book, but you can find more information and download links at www.microsoft.com/windowsserver/system/updateservices/default.aspx. If you don't currently have a tool to make patching easier, consider WSUS. It's free and pretty good.

If you've set your system up to automatically draw patches from Microsoft Update, or if you've got it configured to get patches from your WSUS server, then you've probably got SP1 already. But if not, then you may want to either order a CD with SP1 on it from Microsoft, or just download it. Why not just download it? Well, SP1 is over 300MB in size, and if your Internet connection is at a low speed, then you might not get the download done before, say, the release of 2003 SP2. If you do want to order SP1 on a CD, Microsoft charges \$5.25 per CD, and you can find the links to order it at www.microsoft.com/technet/downloads/winsrvr/servicepacks/sp1/default.aspx.

To download SP1 instead of getting it shipped to you, just go to the above URL and click the Downloading it from Download Center hyperlink, or go to www.microsoft.com/downloads and search on "Microsoft windows server 2003 service pack 1"—2003 SP1 will be one of the hits the search engine will return. (Note that if you're using Small Business Server 2003, SBS 2003 has its own separate set of SP1 files; look down the list of files offered by Microsoft's web server and you'll see them.) Start it downloading, and it'll be done in no time, geologically speaking. The file you'll get will have a name representing the language that the service pack is built to support. As I'm doing this for the U.S. English version of SP1, the file I ended up downloading was named `WindowsServer2003-KB889101-SP1-x86-ENU.exe`, where the ENU means "United States English"; you'll see a different code if you're downloading a version localized to another place.

Before we go any further, do yourself a favor: rename the file to something short like `sp1.exe`. For brevity's sake, that's how I'll refer to it for the rest of this chapter. Next, let's install it.

Installing SP1

Assuming that you're not letting Microsoft Update or WSUS install SP1 for you, you've got a few options at this point to get SP1 on one or more of your systems:

- ◆ Just put SP1 on a CD, walk around to your 2003 servers and run `sp1.exe` on each server from the GUI.
- ◆ Put `sp1.exe` somewhere on the network where all of your 2003 servers can access it, and use the command-line options to kick off the SP1 install.
- ◆ If you have an Active Directory, then extract `sp1.exe` to its component files and deliver it as a group policy.

SP1 from the GUI

Assuming that you want to install SP1 by just double-clicking the `WindowsServer2003-KB889101-SP1-x86-ENU.exe` file (which, recall, I suggested that you rename `sp1.exe`), then you'll first see a dialog box labeled Extracting Files like Figure 1.1.

`Sp1.exe` creates a temporary file with a random name like `b0bda746128` or the like and unpacks SP1's heft to about 413MB of files. After a bit, you'll see a typical wizard greeting page like Figure 1.2.

Click Next and you get to agree to the SP1 software license as in Figure 1.3.

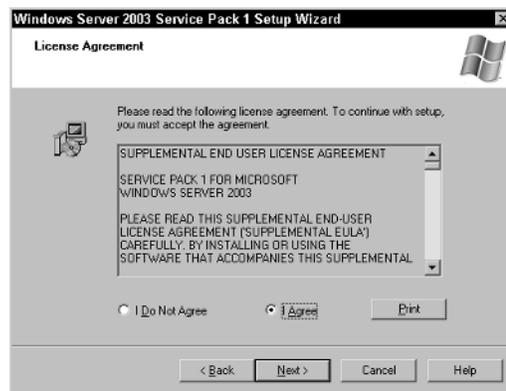
FIGURE 1.1
Unpacking SP1



FIGURE 1.2
“Hello!” from SP1



FIGURE 1.3
SP1's software license



I find agreeing to software licenses of all kinds sort of liberating, you know? I mean, when I go to install a piece of software, I'm usually a little anxious about whether it'll break something and cause me trouble. But once I've read the license and clicked I Agree, then I've pretty much surrendered all of my goods to that software company. (Read a license some time and you'll see what I mean.) That causes a wonderful Zen-like feeling of freedom from material wants to gently wash over me. Anyway, click the I Agree radio button and click Next to see the “Where shall we backup before installing?” page, as in Figure 1.4.

The idea here is that in the unlikely event that you find yourself suffering from “updater's remorse” and want to restore your server to its pre-SP1 state, then you can: as you'll see in a bit, `sp1.exe` will let you uninstall SP1. This step tucks away the RTM files so that the uninstaller can function if necessary.

Now, personally I have not experienced a single problem with 2003 SP1. Yes, I've run into trouble with some NT 4 service packs, but not with any of the 2000, XP, or 2003 service packs, so I'm inclined to trust SP1 and not bother with the backups. Notice, however, that the SP1 install wizard doesn't offer you the option not to back up, which is one reason why I don't install SP1 with the GUI much. Many of my clients like to create a small C: partition so that they can devote the rest of their storage to separate drives for data, databases, logs, and the like. As a result, I've found that many people have installed Server 2003 on C: drives as small as 4 GB. Now, understand please that I don't *recommend* it, but people have done it because of reasons that made sense back in the NT 4 days. Those folks soon find that 2003 has a much bigger appetite for disk space than 2000 or NT 4 did, and so their systems can run into out-of-space problems, and skipping the pre-SP1-install backup saves about 400MB of space. So if you find yourself applying SP1 to a system with less than, say, 2 GB or so of free space on whatever drive contains the operating system, then I recommend that you skip the GUI and look to the next section, where I'll show you how to install SP1 without making `sp1.exe` back up a lot of files and thereby put your system in a free space squeeze. But if you've got plenty of free space on your operating system's partition and want to finish running the wizard, then click Next and you'll see something like Figure 1.5.

After a bit, SP1's installed and the wizard offers you a Finish button. Click it, reboot your system, and you've got SP1 installed.

FIGURE 1.4
Where to backup
for an uninstall?

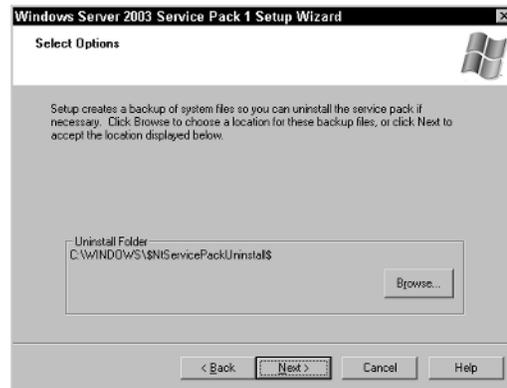
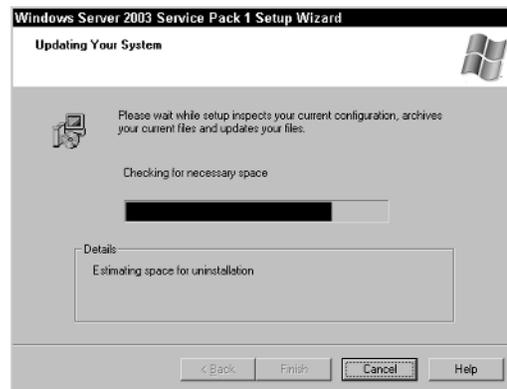


FIGURE 1.5
SP1 installation
under way



SP1 from the Command Line

As I've already said, many folks will choose to start up `sp1.exe` and run the SP1 installation from the GUI because it's simple. But if you have some special needs, then consider running SP1 from the command line with one or more of its options. Those options fall into a few rough categories:

- ◆ `/quiet`, `/passive`, and `/o` reduce the amount of information that the SP1 installer shows and causes it to ask you fewer questions.
- ◆ `/norestart`, `/forcerestart`, and `/f` let you control whether or not your system immediately reboots after SP1's installed, and how.
- ◆ `/n` and `/d`: let you control the amount of space that the SP1 installation process takes up on your disk by letting you skip the backup part of the SP1 install (`/n`), or telling `sp1.exe` to store the backup on another drive (`/d`).
- ◆ `/x` lets you pre-extract the files from `sp1.exe` to a folder.
- ◆ `/uninstall` uninstalls SP1.
- ◆ `/integrate` preinstalls SP1 onto an I386 directory so that you can do fresh installs of 2003 with SP1 already included.
- ◆ `/l` tells `sp1.exe` to list the hotfixes that you've got on this system.

And note that `sp1.exe`, the SP1 installer, doesn't care about uppercase or lowercase; `/forcerestart` and `/FORCERESTART` get the same results.

MAKE THE SP1 INSTALLER QUIETER

Let's consider these options in the order that I've already listed them. First, `/quiet` tells `sp1.exe` not to show you any input, *including* error messages. That's good if you want to kick off `sp1.exe` and have it just do its job without splashing things on the screen *and* if you're sure that you've got SP1's syntax down correctly. It's *bad*, in contrast, if you're *not* so sure about the `sp1.exe` syntax, as it can be pretty puzzling to try to start up SP1 with some options, leave the service pack to install itself, and then return a bit later only to find that nothing's installed, and SP1's not produced an error message that offers any clues to why it didn't work.

If you'd like SP1 to install without asking you any questions, as `/quiet` does, but to have it still provide you some feedback, as `/quiet` *doesn't*, then use `/passive`. `/passive` installs SP1 in a hands-off manner but shows SP1's installation progress bar. For example, to tell `sp1.exe` to just install with all of the default settings—back up the RTM files, install SP1, and do not reboot until told to do so—but to still show progress on the screen, you could open up a command prompt and type

```
sp1 /passive
```

WARNING This will only work if you've put `sp1.exe` somewhere on your system path, or if you have changed your current directory to wherever `sp1.exe` is using the `CD` command. That advice applies to any of the examples in the rest of this chapter. (And, of course, as with any command-line command, you've got to press the Enter key to activate the command.)

LETTING THE SP1 INSTALLER OVERWRITE NON-MICROSOFT FILES

The `/o` option tells you to automatically overwrite any existing files that *didn't* come with 2003 RTM. Wait a minute—why would a 2003 service pack, which should only contain updated files for Windows Server 2003 RTM, have to overwrite a file that *didn't* come with Windows Server 2003 RTM? Because your hardware manufacturer may have created their own customized versions of some file or files to replace some of the files in the RTM version. For example, suppose HP shipped 2003 Server on some of their server computers but with a rewritten version of some SCSI driver, replacing Microsoft's version. (Or consider the Help files—every computer manufacturer seems to tweak them a bit from the original Microsoft files.) In that case, when 2003 SP1 tried to overwrite that SCSI driver, it would notice from the driver's digital certificate—or lack of one—that this copy of 2003 Server has a file on it that's not a Microsoft-created file. By default, `sp1.exe` *won't* overwrite the non-Microsoft file with a Microsoft file because Microsoft figures that if your hardware manufacturer—your “OEM” in computer sales lingo—replaced Microsoft's SCSI driver with one of their own, then that OEM probably had a pretty good reason, and so if `sp1.exe` were to overwrite that customized-to-HP's-hardware driver, then things *could* turn out badly, as in “Auugggh! I installed that stupid service pack and now my system won't boot!!!!” If, however, you have reason to believe that SP1 contains files that will work fine on your server hardware, then you might not want to have to click OK every time `sp1.exe` stumbles across an OEM file and asks, “Is it okay to overwrite this file?” particularly as the SP1 installer might have to do that oh, say, 364 times. In *that* case, add `/o` to the `sp1.exe` command line, and you won't wear out your mouse button and give yourself carpal tunnel by having to click Yes 3,811 times.

CONTROLLING THE POST-SP1 REBOOT

`/norestart` and `/forcerestart` are options that tell the SP1 installer `sp1.exe` to do just what the options sound like—either finish installing SP1 and then do *not* automatically restart, or finish installing SP1 and then automatically restart. (The default behavior is to not restart.) `/f` works in conjunction with `/forcerestart` and says “when restarting, force any open programs to close.” That would mean that *if* while you were waiting for SP1 to install you were running Word on the server's Desktop, *and* perhaps if you were using Word to write the Great American Novel, *and* if you hadn't saved your work in a while, then when SP1 finished installing, it'd close Word without so much as an, “Are you sure?” and reboot, dropping your literary brilliance in a black hole. (And no, I'm not suggesting that you should use your server's Desktop to run personal productivity applications, it's just an example. And while I'm at it, I guess I'd also suggest not bothering to write the Great American Novel anyway, as when last I'd heard Samuel Clemens had already accomplished it.) If, on the other hand, you'd started up `sp1.exe` without the `/f` option, then SP1 wouldn't reboot until you'd responded to Word's Save changes to document? dialog box.

NOTE On a slightly more serious note, I should point out that some applications are programmed to care more about the operating system's wants (that is, 2003 saying, “I want to reboot and you're getting in my way by refusing to close, Wordpad!”) than the user's wants (that is, you saying, “Oh no, don't close, just let me save thi... auugh!”). Such applications will close with files open even *if* you've omitted the `/f` option. Two examples of apps like that are Notepad and WordPad. I've never actually installed Word on a server to find out if it cares more about me or about the server OS. (As they say, There Are Things That Man Is Not Meant To Know.)

Here's another command-line example. To tell `sp1.exe` to install SP1 with the defaults (back up RTM files) and to automatically restart the system, but not to offer any kind of feedback about the progress of the installation and not to reboot if any applications reported unsaved files, you'd type

```
sp1 /quiet /forcerestart
```

TELLING THE SP1 INSTALLER NOT TO BACKUP RTM FILES

As I mentioned before, I tend to skip backing up the RTM files, as it saves disk space and speeds up the install process. You can tell `sp1.exe` to skip backups with the `/n` option.

WARNING I said that I don't do RTM backups, but that doesn't necessarily mean that it's a good idea for you. Let me reiterate this: if you forgo backups then there's no way to return your server to its pre-SP1 state without a complete wipe and reinstall.

Skipping backups also means that you'll save just under 400MB worth of space on the operating system's partition, which might be important to those working on a system with a small amount of space on that OS partition drive. (Here I'm again referring to the folks who I run into now and then who've unadvisedly installed Server 2003 on a 4 GB C: drive.) And as long as I'm discussing saving space on a fairly full C: drive, what about the unpacking process? When `sp1.exe` unpacks its 413MB of files, where does that go? Well, I guessed that if I ran `sp1.exe` from the C: drive then `sp1.exe` would automatically put the unpacked files into a folder on the C: drive—but I was wrong. A bit of experimentation with `sp1.exe` shows that it apparently tries to unpack itself onto a drive that is *not* the one holding the operating system—a nice touch on Microsoft's part.

So, for example, to install SP1 so that it does not back up the RTM files (`/n`), doesn't ask any questions but does show us progress (`/passive`), and reboots the system automatically when SP1's installed (`/forcerestart`) but holds off on the reboot if there are applications with open files (leave off the `/f`), you could open up a command prompt and type

```
sp1 /n /passive /forcerestart
```

As you can imagine, this is a nice "type this, press Enter, and walk away" method of installing SP1. But, again, SP1 might still not install automatically, as it might want to prompt you if it can overwrite OEM files, or it might stop because of an application that refuses to close itself. The truly "damn the torpedoes, full steam ahead!" command would be

```
sp1 /n /passive /forcerestart /o /f
```

SAVING SPACE BUT STILL DOING BACKUPS

But suppose you like the idea of backups but are short of space on your OS's drive? That's a toughie. It's quite improbable that you'll need those backups, but it's a complete certainty that the backup will suck up 400MB of disk space on your OS's drive. What to do? Use the `/d:` option.

Normally, the SP1 installer creates the backup in the Windows directory of your operating system, in a folder called `$ntservicepackuninstall$`. But if you add the option "`/d:path`" to an `sp1.exe` invocation, then the SP1 installer will instead do the backup to *path*. For example, if

your C: drive is low on free space but your E: drive has 400MB free, you could tell the SP1 installer to put the RTM backups in a folder named `e:\rtmbackups`, along with the other options that we used in the last example, like so:

```
sp1 /d:e:\rtmbackups /forcerestart /passive /o /f
```

You needn't even create the `e:\rtmbackups` directory; the SP1 installer will do it for you.

PREUNPACKING SP1 WITH THE /X OPTION

Consider this scenario: you need to put SP1 on a number of servers, but for some reason don't want to use WSUS or Microsoft Update. So you burn `sp1.exe` onto a CD-ROM disc and walk it around your servers. After installing SP1 on a couple of servers, you notice that every time you pop that disc into a server's drive and start up SP1, you have to spend 10 minutes twiddling your thumbs while `sp1.exe` unpacks itself. You could save time, you reason, if you could first unpack the whole thing, converting it from one big EXE file to a folder full of files, and *then* burn that folder to a CD-ROM. Then—somehow—you'd figure out how to get all of those SP1 files to install themselves. Once you figure that out, you could walk that CD-ROM around to your remaining servers, saving precious time on the SP1 install. But how to unpack `sp1.exe`? With the `-x` option. Just type

```
sp1 -x:path
```

Where *path* is a drive and folder name, like `sp1 -x:d:\unpackedsp1files`. That causes `sp1.exe` to create the `d:\unpackedsp1files` folder if it's not already created, and to put a folder *inside* that one called `i386`. (If you alternatively leave the `:path` off the option and just type a simple `-x`, then `sp1.exe` will prompt you for a location to save the unpacked files to.) The resulting folder contains about 413MB of files and folders. Inside `i386` is a folder called `update` that contains a few files, one of which is named `update.exe`, and it's the program that we'd use to tell this already-unpacked copy of SP1 to install itself. And controlling `update.exe` is easy—it uses the exact same options as `sp1.exe` does. So, for example, suppose I'd unpacked SP1 to some folder on my hard disk, and then took the resulting `i386` folder and burned it to a CD-ROM disc. Then suppose I pop that disc into the CD-ROM drive of one of my servers, and that the CD-ROM's drive letter is D:. I could then start up an SP1 install that skipped backups, showed the progress bar, and automatically rebooted despite apps with open files. To do this, I'd type

```
d:\i386\update\update.exe /n /passive /forcerestart
```

UNINSTALLING SP1

As I've said, I've never had the occasion to uninstall SP1. But if you do, then as far as I can see there's only one way to do it—with the SP1 installer program and the `/uninstall` option. Just type

```
sp1 /uninstall
```

And in a few minutes, you'll be back to RTM-ness. When uninstalling, SP1 will accept the `/forcerestart`, `/passive`, `/quiet`, and `/f` options. (And by the way, if you've been avoiding the command line so far, then in this case resistance is futile; there's no choice!)

Rolling Out SP1 with a GPO

In *Mastering Windows Server 2003's* Chapter 12, we showed how to create a domain-based group policy object (GPO) that would deploy software to domain members. You can use software deployment GPOs to roll out SP1 as well, quite simply. There are just a few steps.

1. Extract the files in the SP1 installer with the `-x` option, as you've already read.
2. Create a share accessible by your 2003 servers, and put the extracted `i386` folder in that share.
3. Create a software-deploying group policy object. Use the *computer* Software Installation category, not the *user* Software Installation category.
4. Look in the `i386\update` folder and you'll see a file named `update.msi`. As its extension suggests, it is a Microsoft Installer package file. Configure the GPO to deploy this file to your 2003 servers. Assign the file, do not publish it, or it won't work.

How do you ensure that this only gets to your 2003 Servers, and not your 2000, XP, or Vista systems? Simple—2000 and later are smart about service packs. 2000, XP, and Vista will not even try to install 2003's SP1.

Preinstalling SP1: "Integrating"

As you probably know, the folder that contains the Server 2003 installation files on 2003's Setup CD is called `i386`, the same name as the folder in the unpacked SP1. You may also know that it's been possible in every version of NT since NT 4.0 to take a service pack and incorporate it into the `i386` folder of an existing Setup CD. When you do that, you end up with an `i386` folder that contains all of the files needed to install a brand-new copy of NT that starts out life with the service pack already installed. That's nice for two reasons: first, it saves you the time of first installing the OS and then installing the service pack, as now you need only install the OS and, second, you end up with a more secure system from the very beginning. And if the reason for *that's* not clear, try installing the RTM version of Windows XP on a computer directly connected to the Internet. RTM XP is vulnerable to a number of vicious worms, and it's a pretty good bet that between the time that you boot up this freshly installed RTM copy of XP and when you get the XP Service Pack 2 CD into the computer's drive, your new system will have already caught something nasty. That's why it's nice to do all new XP installs from a Setup disk that has SP2 incorporated into it; all of a sudden, XP setup isn't a footrace between the worms and the poor guy installing the XP patches. It's the same story with Server 2003.

How to take SP1 files and incorporate them into an `i386`? With a few simple steps.

1. Copy the `i386` folder from a Server 2003 Setup disk to your computer's hard disk. (After all, you can't modify files on a CD-ROM.) For the sake of example, I'll say that we've copied it to `e:\I386`. The folder must have the name `i386`, no matter how deeply buried it is in your disk's folder structure. In other words, `e:\i386` is fine, `e:\myfiles\i386` is fine, `e:\files\setup\project\i386` is fine—but `e:\2003setupfiles` would not be. You will not be able to "integrate"—SP1's new word for what it used to call "slipstreaming"—SP1 files into a Server 2003 `i386` unless the folder's name is `i386`.

2. Extract the SP1 files to a folder somewhere.
3. Tell the SP1 installer to integrate the new SP1 files in your i386 folder like so:

```
sp1.exe /integrate:folder-location
```

Be aware that *folder-location* is the name of the folder that contains the i386 folder. So, for example, if the i386 folder with the Setup programs is in `e:\i386`, then you'd type

```
sp1.exe /integrate:e:\
```

This is what I meant when I said that the folder must be named `i386`—the `/integrate` option expects it. You *cannot* convince the SP1 installer to integrate its files with RTM 2003 Setup files unless you call the folder `i386`, oddly enough.

Once the integration's done, you get a confirmation dialog box telling you that "Integrated install has completed successfully." Now you've got an up-to-date I386 setup folder ready to roll out servers sporting SP1 right out of the maternity ward!

Now that you've got SP1 on your system, let's see what it can do. Turn the page, and you'll learn about its first neat tool—Access-Based Enumeration.

Summary

SP1 may be the biggest patch in history (at least until 2003 SP2), but it's worth getting and installing. As you saw in this chapter, Microsoft's offered us a pile of options to both acquire and apply SP1. If you've not rolled it out yet, I'd recommend considering it.

