

PART I

Windows 2000 Professional Basics

Learn To:

- *Install Windows 2000 Professional*
- *Install and organize remove programs*
- *Add a local and a network printer*

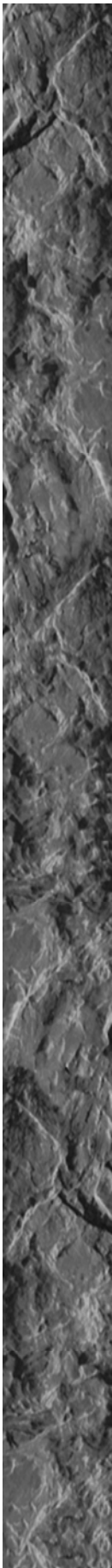


CHAPTER **1**

Introducing Windows 2000 Professional

FEATURING :

Windows 2000 Professional, Windows 2000 Server: Which is which?	4
What's new in Windows 2000 Professional?	8
What's not in Windows 2000 Professional?	16
Why choose Windows 2000 Professional over Windows 98?	17
How is Windows 2000 Professional different from other operating systems?	25
What are the features of Windows 2000 Professional architecture?	27
Weaknesses of Windows 2000 Professional	31



Since the advent of the personal computer, users have wanted three things in an operating system: power, stability, and usability. Windows 2000 Professional has all these and more in abundance. Since its inception in 1993, Windows NT has featured an indefatigable 32-bit architecture, complete with built-in networking and the capability to run almost every piece of Windows software on the market. Windows 2000 Professional now includes an interface, which has evolved from the NT 4 Explorer shell and is even easier to use.



NOTE Windows 2000 Professional is actually version 5 of Windows NT Workstation. Throughout the book, I make many references to NT in describing Windows 2000. The name change was a sign of the convergence of the Windows product lines.

What does all this mean to the consumer, the person sitting at a desk 52 weeks a year who just wants to be a productive computer user? It means a lot. It means a fast, up-to-date operating system with a slew of advanced features that ensure the computer will almost never crash. It means an operating system that can host the emerging multitude of 32-bit software applications, some of which offer amazing facilities for professional audio and video recording, editing, and broadcasting. It means a new interface, which gives users tremendous control over how the system performs its tasks and how it maintains its connections to peripherals and to other computers.

In this chapter, I'll give you an overview of Windows 2000 Professional: what's new, what's different, and what's under the hood. I'll compare Windows 2000 Professional to other operating systems you may be familiar with and emphasize comparisons between Windows 2000 Professional and previous versions of Windows. In later chapters, I'll expand on most of the topics I'll introduce in this chapter, but you'll find a lot here to whet your appetite.

Windows 2000 Professional, Windows 2000 Server: Which Is Which?

The two basic "flavors" of Windows 2000 are *Professional* and *Server*. It might be easier to think of Windows 2000 Server as the big brother of Windows 2000 Professional; they're from the same family but a little different in size and purpose. Both versions of Windows 2000 are solid, 32-bit operating systems, with C2 Level security and built-in networking. Both feature the Windows 2000 interface. Windows 2000 Professional is an operating system designed for corporate and high-end users who want a robust and

powerful operating environment. Windows 2000 Server does everything that Windows 2000 Professional does and adds a comprehensive set of tools for managing and administering a network. It is designed to run on a network file server or application server. There are also additional versions of Windows 2000 Server intended for more specialized environments, such as cluster servers and data warehousing. For more information on these versions, see my book *Mastering Windows 2000 Server* (Sybex).

Everything that I discuss in this book is relevant for both flavors of Windows 2000, but I don't discuss the advanced networking administration features found in *Windows 2000 Server*. (Not to worry, though, networkers: in Part III of this book I *do* cover the administration features that let you manage the built-in networking capabilities of Windows 2000 *Professional*.)

The Road to Windows 2000 Professional

For many years people believed that microcomputer operating systems and mainframe operating systems were totally different entities, as different as bicycles and airliners. In the case of bikes and airliners, most of us would say, "Sure, they're both forms of transportation, but the similarity ends there." It's true. Microcomputer operating systems mainly ended up being launching pads for personal productivity tools such as word processors, drawing packages, e-mail, spreadsheets, and simple databases, as well as games. Mainframes, which make truly awful platforms for word processing (ask anyone who's ever tried to run the Unix version of WordPerfect or IBM's DisplayWrite/370), mainly ended up being the "data warehouses" of large corporations.

Because mainframe systems generally operate in well-managed, centrally administered, and very fault-tolerant environments, huge mission-critical database programs have found a home on them. The IRS keeps its "master file" on every U.S. citizen in a building full of huge mainframes in Martinsburg, West Virginia. The IRS must run a program to examine the master file every day—all 300+ million records of it—and that file is so valuable that parts of it are backed up *continuously*. Running that application on a few dozen Pentium Pros under Windows is hard to imagine, isn't it? (Try explaining to the president that we weren't able to issue refunds this year because the master file GPFed.)

But think about it: Why *doesn't* the IRS do its work on PCs? PCs are slower than mainframes, certainly, but by a smaller and smaller margin with every machine generation. No, the main reason is that corporate information technology (IT) managers don't trust PC software, and with good reason. PC users assume and take for granted the existence of bugs in the software we buy. It's just a fact of computer life. Our acceptance is somewhat odd, though, when you think about it. Consistent low quality is almost unheard of in the rest of the manufacturing world. If one in a thousand Big Macs gave you food poisoning, would you still frequent McDonald's? Probably not. If

automobiles have manufacturing defects, do you meekly accept the answer, “Well, yes, it *is* a problem, but it’s not a big one. We’ll fix it in the next model.” Nope. In fact, in most areas of commerce you usually don’t need to worry about the issue of quality because laws protect consumers from shoddy products in most fields.

Why don’t PC vendors work harder to produce high-quality, never-failing software? Well, some have; IBM’s Personal Editor was one example of an essentially flawless piece of software. Operating systems haven’t been quite as lucky, though; you’d pretty much need a CD-ROM to hold the list of known Windows 3.x bugs.

The Beginnings

The top people at Microsoft knew that they were vulnerable on a very key point: all their products ran on Intel x86 hardware, meaning the 8088, 80286, 80386, 80486, and eventually Pentium and Pentium Pro processors and beyond. Microsoft’s fortunes, then, rode on the Intel horse, and at that time the Intel horse seemed the right one to back. But what about the future? Who could know what processor would be dominant in 10 years? Microsoft wanted to know that it would be on top no matter what processor the marketplace chose, so it needed an *architecture-independent* operating system.

Building an Architecture-Independent Operating System

Architecture-independent doesn’t mean that the product works as well in a Victorian house as it does in a Bauhaus office. Architecture-independent means that it’s built so that it can be modified fairly easily to work on a different processor. All of Microsoft’s operating systems up to 1989 were firmly Intel entrenched, and porting any of them to another processor would have been difficult. (For example, IBM was forced to agree that portability was a problem when it announced that it had stopped work on a version of OS/2 for its PowerPC processor, a CPU of non-Intel architecture.)

A piece of software can easily become dependent on a particular processor because the writers of the operating system must become intimately familiar with the processor’s strengths and weaknesses. Consequently, those strengths and weaknesses inevitably color a developer’s sense of what’s necessary and unnecessary for an operating system. For example, if you build an operating system on top of a 16-bit processor, you’re probably going to make the operating system a 16-bit operating system. If the processor handles memory in chunks (*pages* is the exact word) of 4KB, as the Intel processors do, the operating system’s pages will probably turn out to be 4KB as well. So converting the operating system to a machine with 1MB pages involves massive code rewriting unless the operating system is designed with flexible page sizes in mind.

Building an architecture-independent operating system might have been too difficult a task for Microsoft's development staff, but as luck would have it, a team of top-flight mainframe operating system designers became available in 1989. A group of former Digital Equipment Corporation programmers, led by a man named Dave Cutler, set up shop at Microsoft, promising to build the operating system of Microsoft's future. Influenced partly by an operating system called MACH, a Unix implementation created at Carnegie Mellon University, Cutler's team used a design approach that attempted to isolate all the processor-specific parts of the operating system into a single area called the *microkernel*. Most of the machine-specific parts of the operating system would live in a file called `hal.dll`, or the *Hardware Abstraction Layer*. The operating system would be stable, *carefully* debugged, and *scalable*—that is, able to grow to meet the requirements of the largest PCs imaginable (or, at least, imaginable in 1989).

To produce a stable and scalable operating system, the NT team would require a PC with at least 16MB of RAM and at least a 33MHz processor, quite an expensive machine for the time. Microsoft risked spending millions of dollars on an operating system that would be seen as a sluggish memory hog if in the long run memory prices didn't come down and CPU speeds didn't go up. Memory prices *have* come down, and processors *have* become faster; 400MHz is about the minimum speed you will find on a new computer nowadays.

The Early Releases

The first version of NT (which stands for New Technology) wasn't named NT version 1, probably out of some marketing notion that people are scared of any software labeled version 1. (Actually, going immediately to 3.1 was to coordinate with Windows 3.1.) The first release, called NT 3.1, was supposed to ship in late 1991 but was delayed—first until early 1992 and then until late 1992; it finally shipped in July 1993. This version of NT was dubbed NT Workstation 3.1 to distinguish it from its server sibling NT Advanced Server. NT Workstation 3.1 didn't sell all that well—mainly because it required a fairly expensive PC to run it and it lacked software. Despite the weak sales, NT got high marks from its users. I don't think I *ever* remember any of its detractors pointing to its bugs—because there were hardly any. I recall skeptically loading NT onto a few of my workstations and being pleasantly surprised at its stability—very few bugs and very stable multitasking.

About 14 months later, NT version 3.5 appeared in server and workstation configurations. NT Advanced Server had been renamed NT Server without the *Advanced* because people looked at its \$1,500 price tag and said, "Hey, that's a lot of money, and I don't need that much power; where do I find the *basic* Server?" Again in version 3.5, the NT team showed its commitment to high-quality, bug-free software. Very few features had been added—just some improvements in the TCP/IP software and in NetWare connectivity tools. The NT development team had worked for a little more than

a year, and what had it done? Simple: Instead of adding pointless features, the programmers focused on fixing bugs and tightening up the operating system. It got a little leaner, a little faster, and a bit more stable.

In the early summer of 1995, yet another version of NT appeared—version 3.51. Version 3.51 again produced very few new features. PCMCIA (Personal Computer Memory Card International Association) support, a client license management system, and file compression were the main new items. NT version 3.51 seemed a bit less stable, as evidenced by Microsoft's release of *three* Service Packs (that is, groups of files that patch up the operating system and repair bugs—in other words, bug fixes) for it by the end of 1995.

After 3.51's introduction, another Microsoft operating system appeared, Windows 95. Like NT, it claimed to be a 32-bit OS and sported a shiny new user interface and some new network management tools. Users soon wanted to know when NT would reach parity with Windows 95's features.

The answer to that question was, of course, "Soon," and NT 4 was the down payment on that promise. Windows 2000 Professional has carried on with that promise to provide many of the features formerly the exclusive domain of Windows 95/98.

What's New in Windows 2000 Professional?

Windows 2000 Professional is a mix of old and new. It carries on the legacy of Windows NT including most of NT's strengths and architecture, but it adds many new features. Windows 2000 Professional is much more than just Windows NT 5. It represents the convergence of technologies from Windows 95/98 and Windows NT Workstation. Once again, Microsoft has developed an operating system designed for ease of use, fantastic hardware support, and technology that sets new standards.

Nearly all the functionality of NT 4 Workstation has been preserved, though many of the tools have new locations and new looks in Windows 2000 Professional. The most obvious changes lie in the consolidation of the administration tools into the Microsoft Management Console (MMC), a one-stop shopping place for your management needs. The MMC lets you manage your entire computer from one toolbar. You will also notice changes in the appearance of Control Panel and Explorer. Even the Desktop has changed. Don't worry if the new interface and its new tools seem unfamiliar now; I'll be covering them in detail throughout this book.

Plug-and-Play Support

If you've ever installed a board in a PC, you know what a pain it can be. If you've ever installed a *complex* board, such as a sound card, you know that making it work can be a weekend project. That's why Microsoft and Compaq developed Plug and Play.

Plug and Play is an ambitious project, but here's a brief description of how it works. Every device inside your PC uses a combination of some or all of these components, known generically as *hardware resources*:

- Memory addresses
- Input/output (I/O) addresses
- Interrupt request (IRQ) levels
- Direct memory access (DMA) channels

If you've never come across any of these items, don't be intimidated by the terms; they only *sound* technical. All you have to know is this: Your computer has a limited number of memory addresses, I/O addresses, IRQ levels, and DMA channels. In general, you cannot assign two devices to use the same hardware resource. For example, if you try to put both your sound card and your local area network on IRQ5, neither the sound card nor the local area network card works, or one will work, but not the other.



NOTE The essence of PC expansion board configuration is to first find out what hardware resources a board needs. Some need just an I/O address, and others need an I/O address and an IRQ level; a few demand an I/O address, an IRQ level, and a DMA channel. The design of a board determines its requirements. The next step is to reconfigure the board so that it doesn't conflict with any other board's resources.

If I wanted to install a sound card into a system, I'd start by finding out that it required three I/O address ranges, an IRQ level, and two DMA channels. (I'm not making this up; your basic 16-bit sound card is a nightmare to configure.) Then I'd determine which resources (memory, I/O, IRQ, and DMA) were free and available on my computer before installing the sound card. After I know what resources my computer currently *uses*, I can then figure out what resources are *available*, and I'd tell the sound board to use those unused resources.

Telling the sound board to use those resources sometimes means physically taking the board in hand and moving small electronic contacts called *jumper*s or tiny switches called *dual in-line package (DIP) switches*. Or it may mean running a program that comes with the sound board, or a combination of the two. And I'm still not done. Next, I'll have to load drivers for the sound board, and the drivers will not work until they know which I/O addresses, IRQ level, and DMA channels I selected for the sound board. Reboot, and *then* I'm done, assuming that I did each step correctly.

In contrast, all Plug-and-Play devices contain *smart* boards, boards that can be interrogated by the system about their resource needs. At startup, a program in the BIOS called the Configuration Manager asks all Plug-and-Play hardware—Plug-and-Play hardware resides on the PC's motherboard as well as on expansion boards—to announce its existence. The Configuration Manager then asks each of the boards,

“What resources do you need?” They respond, and the Configuration Manager figures out what configuration will make all the boards happy. It then sets the memory addresses, I/O addresses, IRQ levels, and DMA channels on all the Plug-and-Play hardware and commences the boot process. This procedure sounds time-consuming but requires only a second.

Notice that so far, the operating system is irrelevant; a Plug-and-Play-compliant PC has a Configuration Manager in its BIOS ROM, which runs before the operating system even gets started. But recall that every Plug-and-Play computer completely reconfigures itself whenever you turn it on; this means, in theory, that if you insert a new sound card into your computer, it could set off a chain reaction that would cause the Configuration Manager to reassign all the resources to all the boards.

Reassigning resources would be a problem because all the driver programs that are such an important part of Windows 2000 Professional (and, for that matter, *any* operating system) must figure out what resources each board uses. Also, any Plug-and-Play-compliant operating system must be able to take the self-identification information provided by each Plug-and-Play expansion board and use it to choose drivers automatically. For example, suppose you install a 3Com 3C590 Ethernet card, which is a Plug-and-Play card, into your PC. The Configuration Manager sets its resources automatically, but then how do you tell the operating system to load the driver for a 3Com 3C590? Well, in a perfect world, you don’t want to have to tell the operating system *anything*; it should figure it out for itself—and a Plug-and-Play-compliant operating system does just that.

Windows 2000 Professional includes Plug-and-Play support. For longtime users of Windows NT Workstation, that statement bears repeating: Windows 2000 Professional has Plug-and-Play! No more suffering through the mysteries of manual configuration when installing a new piece of hardware in your computer. After installing the hardware, all you have to do is boot the computer, and Plug-and-Play should take care of the rest.

Partial Plug-and-Play Systems

So, to review: If you have a non-Plug-and-Play system, you configure all the boards by hand, make notes on which memory addresses, I/O addresses, IRQ levels, and DMA channels each board uses, and relay that information to each of the drivers that you must load. If you have a Plug-and-Play system, the system configures itself. Each Plug-and-Play board can communicate its configuration to the corresponding driver.

But what about this fairly common configuration: a Plug-and-Play system with a number of non-Plug-and-Play boards? If your computer has both PCI (Peripheral Component Interconnect) and ISA (Industry Standard Architecture) slots—which is the case for most PCs nowadays—virtually all your PCI boards are Plug and Play, and virtually all your ISA boards are *not* Plug and Play. Thus, you must configure the ISA boards

by hand. In other words, the PCI boards configure themselves without even asking for your opinion, and the ISA boards sit mutely waiting for your instructions.



NOTE Not all ISA boards are Plug-and-Play unaware. In fact, a number of ISA boards provide Plug-and-Play configuration. Examples include the Intel and 3Com ISA Ethernet cards and the Creative Laboratories Plug-and-Play 16-bit sound card. But the vast majority of ISA boards have no Plug-and-Play capability and must be configured by hand.

Suppose you configure one of your ISA boards to use IRQ10. The Configuration Manager, which knows nothing of the ISA boards, happens to assign one of the PCI boards to IRQ10. Result: instant, automatic conflict. What to do?

Some well-designed computers (Compaq is the premier example) have a user-configurable Configuration Manager that you can just plain tell to “stay away from IRQ10.” Unfortunately, most computers aren’t that well built, so here’s my advice.

First, remove the ISA boards. Leave in the PCI boards, and boot the system under DOS or Windows 95/98. Let the Configuration Manager configure the PCI cards. You can then find out which resources the PCI cards used by examining the Windows 95/98 Device Manager. (On the Windows 95/98 Desktop, right-click My Computer to open the System Properties dialog box, and click the Device Manager tab.)

You can also run a DOS-based Plug-and-Play manager, which you can get from Intel or from the manufacturer of your PC. After you know where the PCI boards have settled, configure the ISA boards in the remaining resources. In effect, you let the PCI boards do whatever they want and then make the ISA boards tiptoe around them.

As you can see, configuring a Plug-and-Play system without a Plug-and-Play operating system makes the task more like “plug and pray” than Plug and Play.

Hot Plug-and-Play Support

Another neat feature of Windows 95/98 is *Hot Plug and Play*, which is useful in laptops with PCMCIA/PC Card slots.

If I turn on my laptop computer without a network card in it, Windows 95/98 does not load any network software. But if I insert a PCMCIA/PC network card into my system after it’s booted up, the Configuration Manager recognizes that the network card needs network drivers. Then all the network software loads without the need for a reboot.

Hot Plug and Play is a neat feature for Windows 95/98 and is a feature that is shared by Windows 2000 Professional. When I insert a PCMCIA adapter into my laptop, Windows 2000 Professional recognizes the device, installs the driver (if needed), and starts using the device. Be careful, though. Some devices really don’t like hot swapping. For that reason, there is an icon in your status area to prepare your PCMCIA cards for ejection.



NOTE Although Windows 2000 Professional doesn't require you to reboot or even ask if you want to do so, in many cases you'll need to reboot for a device to work properly.

Offline Files

Using Offline Files, you can work with network files and folders even when you aren't connected to your network. For example, if you lose your network connection or you undock your portable computer, anything that you've made available offline continues to be available to you, with the same access and permissions you'd have if you were connected to the network. To use Offline Files, use the options in My Computer.

Windows Management Instrumentation

Almost all hardware functions are now built around Windows Management Instrumentation (WMI), a software interface that lets an administrator view and modify any hardware settings on the computer you're sitting at as well as on any machine that you can see on the network. In addition you can format and partition disks locally or over the network.

Advanced Power Management Support

If you ran NT 4 on a notebook computer, you'll be glad to know that using Windows 2000 Professional on a notebook computer will double the life of a battery. Advanced Configuration and Power Interface (ACPI) is an industry standard, implemented in Windows 2000 Professional Advanced Power Management, that allows the operating system to conserve power by routing power away from unused hardware components.

Encrypting File System

Using encrypting file system (EFS), you can secure your documents and folders so that only you—or the people to whom you give the key—can view the documents, if you are using the NTFS file system. Encryption is a handy way to keep even shared documents private or to protect files on a machine that could easily be stolen, such as a laptop.

Support for Windows 2000 Server Deployment Tools

Windows 2000 Professional includes support for features of Windows 2000 Server that simplify getting an operating system on a new computer and then simplify installing applications on that computer. If your computer is part of a Windows 2000 Server/Professional network, the administrator can use Remote Installations Services (RIS) to

configure a workstation and then install the operating system, necessary drivers, and so on. An administrator can then use the Software Installation (SI) tool to automatically install an application company-wide.

Microsoft Management Console

Microsoft Management Console (MMC) is a framework for management applications, providing a unified interface for Microsoft and third-party management tools. MMC doesn't replace management applications; it integrates them into one single interface. There are no inherent management functions in MMC at all. It uses component tools called snap-ins, which do all the work. MMC provides a user interface; it doesn't change how the snap-ins function.

Our old familiar administrative tools—such as the User Manager and User Manager for Domains, Server Manager, Event Viewer, and even Disk Administrator—have been assimilated into things called Microsoft Management Console (MMC) tools.

USB Support

Universal Serial Bus (USB) is an electronic pathway that supports Plug and Play and that allows you to connect and disconnect devices without shutting down or restarting your computer. The Windows 2000 Professional USB supports everything from game controllers to digital cameras.

Digital Camera Support

Setting up a digital camera in Windows 2000 Professional is a cinch using the Scanners And Camera applet in Control Panel. If your camera is Plug and Play, Windows 2000 Professional will detect it and install it automatically.

Multiple Monitor Support

Unlike NT 4 and like Windows 98, Windows 2000 Professional supports the use of multiple monitors with a single CPU. All you need is an additional PCI video card that supports multimonitor. If you're thinking of getting a bigger monitor, you might want to consider getting an additional monitor instead (if you have space on your physical desktop). For example, two 17-inch monitors have more display area than a single 21-inch monitor.

DirectX 7

DirectX is Microsoft's architecture for running multimedia applications using any hardware platform, and it is a tool that developers use to create multimedia content.

Windows 2000 Professional supports the entire suite of DirectX 7 components. DirectX 7 is composed of the following components:

DirectDraw Controls the graphics and video functions on your computer (the way that images are drawn on your screen).

DirectSound Controls the way that sound is played on your computer.

DirectInput Controls joystick commands on your computer.

Direct3D Controls the manipulation of 3-D images on your screen.

DirectPlay Provides network access for programs written to take advantage of DirectX 7.

For starters, all this means that your favorite Windows 95/98 games will work in Windows 2000 Professional exactly as they did in Windows 95/98.

Advanced VPN Security

A Virtual Private Network (VPN) is a tunnel through the Internet that connects your computer to a network. When you're on the road, you can dial up almost any ISP and set up a VPN session to your network over the Internet. VPN supports the following protocols:

Point-to-Point Tunneling Protocol (PPTP) Part of the TCP/IP family of protocols used to transmit data over serial lines and point-to-point dial-up telephone connection.

Layer-2 Tunneling Protocol (L-2TP) Based on a Cisco protocol, allows you to establish a tunnel from an ISP to a corporate site and transmit data through it.

IP Security (IP-Sec) A suite of communications protocols that add security to the Internet Protocol (IP).



NOTE For all the details about VPN and Windows 2000 Professional, see Chapter 21.

Recovery Console

Recovery Console is a program that you run from a command prompt to make changes to a Windows 2000 Professional system that refuses to start—if you are a knowledgeable system administrator. You can also use the Recovery Console to start or stop system services and to format disks.

Hibernation

Hibernation is a feature of power management in Windows 2000 Professional that saves everything in memory to the hard drive, turns off your monitor and hard disk,

and then turns off your computer. You typically use hibernation when you are going to be away from your computer for an extended period or overnight. When you restart your computer, the Desktop is restored exactly as it was when the computer shut down. Hibernation is primarily for use on portable computers, but the computer's hardware must be set up by the manufacturer to support this feature.

Remote Desktop Sharing

Remote desktop sharing allows the user of a remote computer to take over another computer just as though they were in front of it. This is a new feature of NetMeeting that you cannot replicate using VPN or any other remote access service. For all the details, see Chapter 11.

Internet Connection Sharing

Internet Connection Sharing (ICS) is a feature that allows multiple computers to share a connection to the Internet whether that connection is a modem, ISDN, cable modem, DSL, or whatever. You run ICS on the computer that's dialed in (or cable-modemed or DSLed) to the Internet. When you enable Shared Access, the ICS machine acts as a DHCP server and as a router. For details about sharing an Internet connection, see Chapter 10.

Fax Services

Windows 2000 Professional includes a fax service, which sends and receives faxes through a separate fax queue (just like a printer) or through various fax-aware applications such as your favorite e-mail client software.

Internet Explorer 5

Internet Explorer (IE) is Microsoft's all-purpose Web browser. Much as in Windows 98, IE is integrated into the Desktop environment. IE functions are integrated into the entire operating system to make it easier to reach documents either on your local network or anywhere on the Internet.

Internet Information Services

Windows 2000 Professional includes the same Internet Information Services (IIS) as its big brother, Windows 2000 Server. IIS provides the ability to act as a Web server, File Transfer Protocol (FTP) server, Simple Mail Transfer Protocol (SMTP), and Post Office Protocol version 3 (POP3) server for e-mail.

Accessibility Options

Accessibility options are features that you can enable to help people with physical challenges use the computer easier. Features include Narrator, which reads items that are displayed on the screen, Magnifier, which displays magnified portions of your screen in a separate window, and an onscreen keyboard that you type on with the mouse.

Telnet Server

The Telnet Server service enables remote computers to connect to a Telnet server via the Telnet protocol. Telnet Server service in Windows 2000 Professional supports two simultaneous connections.

Personalized Menus

In Windows 2000 Professional, menus contain only the items you most frequently use. Those items that you don't use don't appear until you click the More button. Opinions vary as to whether this is a bug or a feature. But if you find this "feature" annoying, you can always revert to the standard menu, using the options in the Taskbar And Start Menu Properties dialog box.

What's Not in Windows 2000 Professional?

The following is a very short list of features not available in Windows 2000 Professional.

MSN Support

The much-talked-about Microsoft Network (MSN) software that was a key part of Windows 95/98 is not available in Windows 2000 Professional. However, you can still use MSN as an Internet service provider (ISP), for use with Microsoft IE or the Web browser of your choice. Look for an MSN client to be released for Windows 2000 Professional in the near future.

Support for MIPS or PowerPC Chips

Windows 2000 Professional does not support the MIPS (millions of instructions per second) and PowerPC processors. These platforms haven't been very well represented in the marketplace, and the development effort put into the approximately 36 million lines of code for Windows 2000 Professional demands that only the most prevalent systems be supported.

Why Choose Windows 2000 Professional over Windows 98?

Given that their interfaces are so similar, it would be a fair question to ask why you should choose one over the other. The answer is straightforward. Windows 2000 Professional may look like Windows 98, but it does many things that Windows 98 can't. For starters, Windows 2000 Professional is a secure operating system. Just try to log on to a Windows 2000 Professional computer without the correct password. You can't do it. In Windows 98? Just press the Esc key and you're in! The lesson here is, if you have data that absolutely, positively needs to be kept secret, Windows 2000 Professional is the operating system for you.

Windows 2000 Professional is also the most complete 32-bit environment you can buy. If you need an operating system that won't crash and won't shut down on you, Windows 2000 Professional is the one for you.

How different is Windows 2000 Professional from Windows 98? Both use the same style interface; both are 32-bit operating systems. But after that, the similarity ends. Although they are both 32-bit operating systems, they implement their 32-bitness in much different ways. Table 1.1 lists the differences, which are explained below in greater detail.

TABLE 1.1: COMPARISON OF WINDOWS 2000 PROFESSIONAL AND WINDOWS 98

Windows 2000 Professional	Windows 98
Hardware Abstraction Layer (HAL) with Plug and Play	Plug and Play
Multiprocessor support (with either x86 or Alpha)	x86 support only
Runs entirely in protected mode	Runs in protected mode with some real-mode support
Separate 16-bit memory spaces	Shared 16-bit memory spaces
Multithreaded operating system	Single-threaded operating system
NT File System (NTFS), File Allocation Table (FAT), File Allocation Table 32 (FAT32)	File Allocation Table (FAT), File Allocation Table 32 (FAT32)
File and resource rights assignments	No rights
Complete user-level security	Limited password security

More Solid, Crashless

Ever since his days at DEC and before that at DuPont, Dave Cutler had a simple formula for making sure that a piece of code was as bug-free as possible: He forced the people who wrote the code to *use* the code. His phrase was that programmers should “have to eat their own dog food.” From as early as possible, it was Alpo for breakfast, Ken-I-Ration for lunch, and Mighty Dog for dinner with the NT team. Putting a bug into the latest “build” of NT brought tons of peer disapproval and, worse, a *communication* from Dave.

Additionally, Cutler wrote some of the toughest pieces of the operating system himself. No one on the team just talked about what the system should do; everyone participated in building it. And Cutler carried all of NT in his head. Maybe this centralization was a *disadvantage*—after all, what would happen when Cutler left the project? But at that time, having at least one person with the grand overview—something that’s rumored not to have existed for Windows 95/98—was a great strength.



NOTE Looking at some of the bug lists that accompany the Service Pack updates for NT 4 is a worthwhile exercise. A Service Pack may fix hundreds of bugs, but those bugs could be so minor that not a single one applies to your installation. And consider that Service Packs were never released for DOS or Windows 3.1. One appeared for Windows 95, but it fixed only two bugs, leaving the dozens of shell bugs untouched. We can expect to see Service Packs for Windows 2000 Professional too, so keep an eye on Microsoft’s Web site from time to time.

When Windows 2000 Professional *does* die, it does a pretty good job of telling you who killed it. If you ever see Windows 2000 Professional crash, you’ll see a distinctive sight. The screen clears and goes to a 50-line text screen with a blue background and bright white letters. When Windows 2000 Professional crashes, it fills up this blue screen—Windows 2000 Professional experts call this event *bluescreening*—with memory dump information and an error message or two. In my experience, what kills Windows 2000 Professional is a bad video driver, a bad disk driver, or a bad RAM chip, and the blue screen helps you identify which one is the culprit.

Furthermore, when Windows 2000 Professional crashes, it writes an entire image of the physical memory to disk in what is called a *crash dump*. You can use the crash dump analysis program `dumpexam.exe` (included on the Windows 2000 Professional installation CD) to look through the crash and narrow down the causes. You can also send the crash dumps to software and driver vendors, meaning that you have some leverage when you ask them to fix their software.

Complete Protected Mode over Protected Mode with Real-Mode Support

One of the selling points for Windows 95 was its emphasis on backward compatibility, which meant that it wasn't *completely* a 32-bit operating system. If you're running Windows 95 and you still have some old, real-mode, 16-bit drivers for your favorite peripheral, you can still use them with Windows 95. You can't use them with Windows 2000 Professional, though. That's because everything about Windows 2000 Professional must be able to run in protected mode.

What are these modes, and what difference do they make? Simply put, *real-mode* drivers can talk directly to your PC hardware. Although this is a direct and efficient way for things to work, it also causes crashes and other errors that often beset computer users. *Protected-mode* drivers, on the other hand, cannot talk directly to computer hardware; they must go through the operating system. In Windows 2000 Professional, all protected-mode drivers must talk to the HAL (Hardware Abstraction Layer) to get to the hardware, which makes for a more stable operating environment.

Faster

One thing that can't be denied about NT is that it runs slower on a given system than Windows 95/98. At first glance, that would suggest that NT and speed aren't subjects you'd often put in the same sentence. And if you're talking about the standard Pentium workstation with 16MB of RAM and a 100MHz processor, it's true that you'd be better off running most applications under Windows 95/98 than under NT 4.

But now consider the more powerful systems. The top end of the x86 world for a long time was the Pentium Pro processor. And believe it or not, the Pro runs NT faster than it runs 95/98.

Why? Well, when Intel started designing the Pentium Pro in the late '80s, it predicted that by 1995 most people would have converted from 16-bit to 32-bit operating systems. Consequently, Intel designed a CPU that ran fastest in 32-bit mode and not very fast in 16-bit mode; in fact, the Pentium Pro is usually slower than a Pentium in executing 16-bit code. Unfortunately, as we now know, Intel was a little optimistic in its beliefs about people's willingness to upgrade their operating systems. (The fact that memory prices didn't fall significantly until early 1996 certainly didn't help their willingness.) Hence, 16-bit operating systems, such as DOS or Windows 3.1, won't run very well on a Pentium Pro.

But what about Windows 95/98? Well, even though Windows 95/98 is advertised as being a 32-bit operating system, it's not, really. It combines a lot of the old 16-bit code with some new 32-bit code. In fact, a lot of the 32-bit code in Windows 95/98 is just 16-bit code with some programming fixes built in, a process called *thunking*. You won't

find any such thunks in Windows 2000 Professional, because it's an actual 32-bit operating system. The upshot is, because Windows 95/98 is partially 16-bit, it takes a performance hit when running on a Pentium Pro.

Windows 2000 Professional, in contrast, gets *faster* on a Pentium Pro, because both use true 32-bit architecture. But the Windows 2000 Professional speed boost doesn't end with the Pentium Pro. The new Pentium IIIs have that 32-bit architecture as well, and at the time of this writing they're running at 800MHz and faster!



NOTE Although Windows 2000 Professional can support as many as 32 processors, the Hardware Abstraction Layer (HAL) that ships with Windows 2000 Professional supports only two processors. To use more, you'll need a custom HAL from your computer manufacturer. The HAL that ships with Windows 2000 Server supports four processors, and the higher versions of Windows 2000 Server support as many as 32 processors.

The first desktop PC to run at 1000MHz won't be running Windows 95/98—it'll most likely be running Windows 2000 Professional. So, while the lower-end workstations run fastest with one of the old 16-bit or hybrid 16/32-bit operating systems, the more powerful machines will find Windows 2000 Professional a better fit.

Home Directories

A home directory is the location where a user stores their files. The user can then access these files from any computer that they can log on to. Home directories supply another level of data protection that is not available in Windows 95/98.

Error Recording

Ever had a message appear on the screen while you booted a DOS or Windows 95/98 workstation, only to have the message scroll off the screen before you could read it? You won't have to endure this particular frustration under Windows 2000 Professional, because all important system messages are written to one of three logs: the System Log, the Security Log, or the Applications Log. You view all three with an application called the Event Viewer, which is in the Administrative Tools applet.

Event Viewer messages are not always the most useful things in the whole world, but you'll sometimes be able to get a lot of information out of them. Well-written applications (you *did* catch the weasel words *well-written*, didn't you?) write useful information into the Applications Log, so put "uses the Applications Log" on your list of things to look for when buying 32-bit applications for Windows 2000 Professional. One of the new features that Microsoft touted for Windows 2000 Professional is new, clearly written error messages. So far, the messages I've seen have been remarkably easy to work with.

Multiprocessor Support over x86-Only Support

Windows 2000 Professional is built to run on the Intel x86 and IA64 families of computer processors. It can also run on computers running more than one processor. As I mentioned earlier, Windows 2000 Professional can support up to two processors. What's the advantage to all this? Well, it means you can use Windows 2000 Professional on some very fast machines; you're not limited to the x86 series. On the other hand, Windows 95/98 can run only on an x86 processor and on only one of them at a time.

Separate 16-Bit Memory Spaces over Shared 16-Bit Memory Spaces

Despite all the advances brought about by 32-bit software, sooner or later you'll probably find yourself in a situation where you need to run 16-bit applications. Both Windows 2000 Professional and Windows 95/98 run 16-bit programs just fine, but they run them differently, and this affects performance and system reliability.

In the world of Windows 95/98, 16-bit Windows programs share the same memory space, or, if you prefer, the same room. If one 16-bit Windows application goes down, it will also crash any other 16-bit applications you may be running. Windows 95/98 runs 32-bit programs much the same way that Windows 2000 Professional does—it runs them in separate memory spaces. Windows 95/98 also runs MS-DOS programs in separate memory spaces because the operating system first must create the illusion of a computer running MS-DOS. This simulation is called the Virtual DOS Machine (VDM).

Windows 2000 Professional also uses a VDM to run MS-DOS applications, giving the appearance of a 486 computer with all the memory the program needs. Each VDM exists in its own memory space and runs as a 32-bit program. Windows 2000 Professional uses the VDM to create the Win16 on Win32 (WOW) environment too. The WOW is a simulation of Windows 3.1 on top of the simulation of MS-DOS. It runs just like Windows 3.1 with all the good points and the bad. By default, Windows 2000 Professional shares the same WOW environment for all 16-bit Windows applications, which means that if one crashes, they all crash. But Windows 2000 Professional differs from 95/98 in that you can start your 16-bit applications in separate memory spaces so they run just like 32-bit programs. That means you have the choice of using the default model of running the 16-bit applications in a shared memory space where they can crash one another or running them in a separate 32-bit memory space where you can take advantage of preemptive multitasking and multiprocessing.

Running 16-bit programs in a separate memory space may not always work. Some applications require a shared memory environment in order to exchange information or draw upon services provided by another program. Unfortunately, you may need to experiment some to find out if your 16-bit applications will run correctly on Windows 2000 Professional.

NT File System over FAT File System

DOS, Windows 3.x, and Windows 95/98 use the same type of file system, called FAT, for File Allocation Table. Here, when I mention a file system, I'm not just talking about the order in which your files are stored on your hard drive. Rather, I'm talking about the entire set of organizing principles that allow your operating system to communicate with your hard drive. The FAT approach to organizing this activity is to create a database (the FAT) at the boot sector of your hard drive. The boot sector is an area at the "beginning" of your hard drive. It's the place where the hard drive keeps the files that your operating system needs to talk to the drive, and the FAT is the most crucial of these.



NOTE The disk error messages you may have seen when running a disk utility under other operating systems are informing you of problems in the FAT!

NTFS (New Technology File System) is a high-performance file system used by Windows NT and Windows 2000 Professional. It incorporates more powerful features than the FAT system. For example, not only can NTFS understand long filenames, up to 255 characters (actually, Windows 95/98 can also do this), NTFS also creates backup records of its Master File Table (MFT), which is the NTFS version of the FAT. If the boot sector of your hard drive (the place where file table information is stored) happens to be damaged through some unfortunate incident, the information can be replaced from one of the MFT backup records. As a result, it's much less likely that you'll lose data because of disk problems under NTFS than under FAT.



NOTE Windows 2000 Professional includes the newer version of NTFS that was first released with Service Pack 4 on NT 4. If you want to multiboot Windows 2000 Professional and NT 4, be sure to install the latest Service Pack.

Moreover, it is much more difficult for viruses to infect NTFS volumes than it is for them to infect FAT volumes. This is because the data on the NTFS volume is written differently than it is under FAT. Most viruses are looking for FAT file types, and they don't know what to do when they encounter NTFS.

NTFS also allows security controls for individual files, programs, and folders—controllable by the Windows 2000 Professional Administrator or anyone with administrator privileges. FAT has no security features at all.

Neither DOS, nor Windows 3.x, nor Windows 95/98 understands NTFS. If you have two hard drives in your computer, and one of them is formatted as NTFS and the other as FAT, DOS, Windows 3.x, and Windows 95/98 systems will see only the FAT drive.

Similarly, if you make multiple partitions of a single hard drive and format each of them with a different file system, the non-NT/Windows 2000 Professional operating systems won't see the NTFS-formatted partitions. Table 1.2 lists the main differences between NTFS and FAT.



WARNING Earlier versions of Windows NT supported the High Performance File System (HPFS) from OS/2. NT 4 and Windows 2000 Professional do not support HPFS at all. If you have any partitions formatted with HPFS, you must convert them to NTFS before installing Windows 2000 Professional if you want the data on those partitions to be accessible.



NOTE Windows 2000 Professional does support the FAT32 file system, so multibooting with Windows 98 is now possible.

TABLE 1.2: A COMPARISON OF NTFS AND FAT

NTFS	FAT
Backup records	No backup records
Long filenames	Long filenames with Windows 95/98 8.3 filenames with earlier Windows and MS-DOS
Full user security support	No security
Limit of 16EB	2GB under Windows 95/98 (1EB = 1,073,741,824GB) 4GBs under Windows NT
Windows NT only	MS-DOS, Windows 3.x, Windows 95/98, Windows NT

Security in Windows 2000 Professional

Security is an A-number-1 *raison d'être* for Windows 2000 Professional. NT was designed to fulfill the needs of people requiring high levels of security, and NT is one of the only microcomputer operating systems that has been reviewed for the U.S. government's C2 security standard. Windows 2000 Professional also incorporates this deep-seated need for security.

Here are some of the ways in which Windows 2000 Professional implements security:

- Requires users to log on to a workstation.
- Defines user rights that control a user's actions.
- Uses file and printer permissions to control access.
- Encrypts security information over networks.

Windows 2000 Professional Requires Users to Log On to a Workstation

Windows 95/98 (for comparison with Windows 2000 Professional) has a login screen that appears when you turn it on, but the login procedure is not really secure, for a couple of reasons. First, you can bypass it by pressing Esc. Second, you can always boot a Windows 95/98 computer with a floppy and run the `regedit.exe` program to export the Registry, a set of files that controls the configuration of Windows 95/98.

Windows 2000 Professional, in contrast, doesn't let you do anything on your computer until you log on, identifying yourself by name, password, and whether you want to be acknowledged by a local Windows 2000 Professional domain. (Windows 2000 Professional domains are methods of centrally storing user account information.)

Windows 2000 Professional Defines User Rights That Control a User's Actions

Have you ever loaned your machine to someone, only to get it back with all your settings changed? Or, worse yet, to find it loaded with a new piece of software that messed up your DLL files or program groups?

The good news is that Windows 2000 Professional divides the range of things that you can do with a computer into *user rights*. Local Users And Groups enables you to restrict what someone can do on your system. Giving someone else an account on your system doesn't necessarily mean giving them the keys to your system; it may mean giving them access to only a subset of your computer's capabilities.

The bad news is that you don't have the same fine control of user rights in Windows 2000 Professional that you do in Windows 2000 Server. The good news is that Windows 2000 Professional lets you define a local user account that can use the computer and run common programs but cannot install new programs. Members of the Users group can run programs and save data to their own folders, but they can't install programs or modify system files.

Windows 2000 Professional Uses File and Printer Permissions to Control Access

Like its bigger brother Windows 2000 Server, Windows 2000 Professional lets you point to any given folder and remove a user's access to that folder. Believe it or not, you can create a local account on a Windows 2000 Professional machine that, at the

same time that it lets someone get on your computer and run a particular program, also prevents that user from doing anything else.

When you examine a file or a folder with Explorer, you can right-click a folder or file, choose Properties from the shortcut menu, select the Security tab, and see the permissions attached to that file or folder.

Printers, as well as folders and files, have permissions associated with them. You can prevent specific people from using your locally attached printers, and of course, domain network administrators can control which network printers those people can access.

Windows 2000 Professional Will Optionally Log All Security Events

Want to keep track of *everything* that happens on your computer? It's possible. Windows 2000 Professional includes a Security Log that will, at your option, log many events, including the following:

- Who logged on to your machine, when the person logged on, and when they logged off
- What files were used
- When others attempted and failed to log on to your system
- Whether the person who logged on to your system changed any user rights or permissions (if they had administrative privileges)

All that information goes in the Security Log, one of the three logs you can find in the Event Viewer.

How Is Windows 2000 Professional Different from Other Operating Systems?

I've discussed the differences and advantages of Windows 2000 Professional compared with Windows 95/98, but how does it stack up against other popular operating systems?

Windows 2000 Professional Compared with Windows 3.x

Windows NT, Windows 3.1, and Windows for Workgroups (Windows 3.11) all once had the same interface, and all were made by the same company, but that's where the similarities end. The Windows 3.x family of operating systems are 16-bit, completely real-mode operating systems and are also completely beholden to the DOS 640KB memory model. Although Windows for Workgroups has built-in networking capabilities and can coexist quite nicely with Windows 2000 Professional on a network, it offers no security features of its own, relying instead on the network for things like password

validation and rights authorization. Whereas Windows 2000 Professional can run any program that the others can run, neither 3.1 nor 3.11 can take advantage of the 32-bit Windows programs that have become available in the wake of Windows 95/98.

Windows 2000 Professional Compared with OS/2

Both Windows 2000 Professional and OS/2 are 32-bit operating systems that also run 16-bit Windows applications and DOS applications in separate memory addresses. In terms of robustness, Windows 2000 Professional and OS/2 are on about even footing. Past that, OS/2 gets left in the dust. OS/2 is not a secure environment, and it lacks native networking capability—you must buy additional software so OS/2 can talk to its network brethren. Furthermore, it won't run 32-bit Windows applications. Because only about three native OS/2 applications are available, this is a serious shortcoming.

Windows 2000 Professional Compared with Unix

Of all the operating systems under discussion here, Unix is probably the most similar to Windows 2000 Professional in terms of architecture. There are many different flavors of Unix, however. Each flavor has a different user interface, and not all of them are graphical. As a group, Unix operating systems are 32-bit, secure, and capable of running on Intel, RISC, and DEC Alpha processors.

In the past, artists and designers have used high-end Unix-based workstations to create special effects for films. Windows 2000 Professional, however, supports high-end 3-D protocols such as Raydream. Running Windows 2000 Professional on a high-powered x86 processor such as the Pentium Pro or Pentium II or III will give you equal processing power to those Unix workstations, with the added punch of Windows 2000 Professional, for a fraction of the price.

Windows 2000 Professional Compared to Macintosh OS

Like Windows 2000 Professional, the Macintosh operating system (OS) is a 32-bit environment with built-in networking capabilities. Despite its well-known and intuitive interface, the Macintosh OS lacks many of the powerful features found in Windows 2000 Professional. Object linking and embedding (OLE), MAPI, and TAPI are all unfamiliar to Apple users. The Mac OS supports only a limited sort of cooperative multitasking, and you can switch between programs only with the mouse; there's no equivalent to the mighty Alt+Tab key combination in Windows. There is also a relatively limited amount of software available to the Macintosh market as compared with the Windows market.

What Are the Features of Windows 2000 Professional Architecture?

I've already discussed some of the Windows 2000 Professional architecture. In this section, I'll analyze it a little more closely to see what makes Windows 2000 Professional tick and why it is different from other operating systems. I'll go into the Windows 2000 Professional architecture in much greater detail later in the book (see Chapter 14), but this will serve as a primer for those of you who can't wait.

From a programmer's perspective, Windows 2000 Professional is divided into two layers. These are actually two separate operating modes. The *kernel mode* is where Windows 2000 Professional performs its internal tasks and controls interaction between programs and the operating system, and between programs themselves. It is generally protected from end users fiddling with it. The *user mode*, which is usually described as sitting above the kernel mode because it is closer to the end user, is where your applications run. The area is considered nonprotected, because you have access to it through your applications.

Kernel Mode

Nothing happens in the Windows 2000 Professional universe without the operating system knowing about it and giving its blessing. The *kernel* is the core of Windows 2000 Professional. It acts as a "gofer" between the operating system and the computer's processor. As such, it is responsible for scheduling all the operating system's interactions with your computer. The kernel also manages all the interrelationships between the different kernel mode operations. It does this by means of *threads*. A thread is a series of instructions that are attached to a command that is executed by a program. Threads include memory addresses, scheduling for the amount of time the process will take, and anything else that describes the process.

The kernel mode is divided into three subsections: the HAL, the kernel, and Executive Services. Executive Services is subdivided further into its own component parts; more on those parts a little later in the chapter.

The HAL and the Kernel

The HAL controls the interaction between the kernel and the system hardware. One of the design goals of Windows 2000 Professional was the capability to be easily ported from one type of computer to another. To this end, the HAL *abstracts* the hardware from the kernel so that the kernel does not need to know what type of hardware is installed in the computer. This extends to running multiple processors in your computer. Windows 2000 Professional is one of the only PC-based operating systems that can take advantage of symmetric multiprocessing. The abstraction provided by the HAL gives Windows 2000 Professional greater stability.

Because of the abstraction provided by the HAL, software that attempts to directly access the hardware is not permitted to run. The HAL, working with the kernel, stops those programs dead in their tracks. For performance reasons, many DOS programs—particularly games—use direct hardware access. Programs written for Windows 2000 Professional don't need to worry about how to access the hardware; they just ask the operating system for support, and Windows 2000 Professional takes care of the rest.

Windows 2000 Professional does have some communication that bypasses the HAL and goes directly to the hardware. In each of these cases, the communication is between the Executive Services and the individual drivers for hardware devices such as the video card and network card. But the concept of hardware abstraction is still maintained.

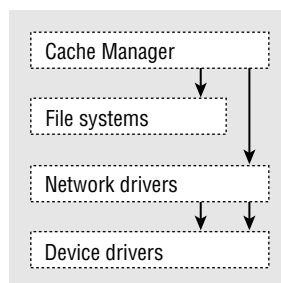
Executive Services

Executive Services is a set of separate components that complete the underpinnings of the Windows 2000 Professional kernel mode. Each of the Executive Services controls a specific function. The following is a brief overview of each of those functions.

I/O Manager

I/O Manager handles all communication between your applications and your hard disk. Additionally, it manages drivers for different file formats (FAT, NTFS) and keeps the Windows 2000 Professional kernel informed of the hard drive's status. I/O Manager also manages network cards and modems. In short, this manager controls any device that delivers data to or from the computer. Its components are illustrated in Figure 1.1.

FIGURE 1.1
I/O Manager is built to simultaneously control multiple devices and drivers.



I/O Manager is divided into the following sections:

Cache Manager Monitors your use of disk cache and disk pages.

File systems Manages the file systems you have installed on your computer.

Network drivers Controls your network cards and network protocols.

Device drivers Manages the hardware you have in your computer. This includes everything from your hard drive(s) and modem to your keyboard and mouse.

Object Manager

Object Manager manages all the system objects that are used by Windows 2000 Professional. An *object* is a piece of data used by the operating system to create system events. Objects are acted upon by processes. Every object has a handle (yes, that's the technical term) that a process connects to in order to complete its job. The Object Manager is extensible, meaning that it can be expanded and added to as new object types are developed.

Security Reference Monitor

Security Reference Monitor is the “watchdog” for Windows 2000 Professional. Security Reference Monitor assigns security tokens and authenticates users each time they execute a task.

Process Manager

Process Manager is the complement to Object Manager. Process Manager manages the creation and deletion of processes. A *process* is a set of threads combined with a memory address and the necessary objects needed to complete a system task. The Process Manager works in conjunction with the Security Reference Monitor to ensure that every process is assigned a security token.

Local Procedure Call Facility

Local Procedure Call facility acts as a negotiator between user mode and kernel mode. From an internal standpoint, Windows 2000 Professional uses a client-server model to administer itself. Just as Windows 2000 Professional connects to Windows 2000 Server and requests services from it, the Windows 2000 Professional user mode requests services from the Windows 2000 Professional kernel mode. Those requests are handled by local procedure calls. Local procedure calls are also used in standard client-server networks.

Virtual Memory Manager

Virtual Memory Manager oversees how Windows 2000 Professional uses virtual memory. To increase the amount of usable memory space, Windows 2000 Professional uses hard-disk space as memory when it runs out of memory (which can happen fairly quickly with some high-powered programs). The disk memory is known as *virtual* memory.

Win32K and GDI

The Win32K and Graphics Device Interface (GDI) are the graphics subsystem of Windows 2000 Professional. The graphics functions in earlier versions of Windows NT were provided by the Win32 subsystem in user mode. Microsoft moved the graphics subsystem into the Executive Services for version 4 to give the graphics functions a performance boost. In fact, this resulted in the single greatest increase in the apparent speed of Windows NT 4.

User Mode

In the Windows 2000 Professional user mode, each of your programs runs in a separate memory space, an arrangement that protects each program from the others in case one should crash. This is true for 32-bit and 16-bit programs, both Windows and DOS.

The user mode is divided into subsystems. Each subsystem handles a different type of application and can report directly to the kernel mode. The subsystems are described briefly in Table 1.3.

TABLE 1.3: USER MODE SUBSYSTEMS

Subsystem	Description
Win32	Administers 32-bit and 16-bit Windows programs.
OS/2	Administers character-based OS/2 programs.
POSIX	Administers POSIX programs, which are Unix hybrids that can be run on any POSIX-compliant system, including Windows 2000 Professional. They are written to a series of application programming interfaces (APIs) that are platform independent. They control interaction with system components such as hard drives and memory.
Security	Administers system security and manages security tokens, monitors, and passwords. The user mode portion of the Security subsystem runs only during a user login. For more information on security, see Chapter 18.

In addition to the subsystems noted in Table 1.3, user mode may also contain VDMs. A VDM simulates a computer running MS-DOS 5, with 16MB of RAM and conventional, expanded, and extended memory. As stated earlier in this chapter, this simulation makes it possible to run MS-DOS programs on Windows 2000 Professional. It also enables you to run 16-bit Windows applications by simulating Windows 3.1 running on that MS-DOS computer with 16MB of memory. By default, Windows 2000 Professional starts all 16-bit Windows applications in the same WOW environment. This simulates exactly the environment the programs were written to operate in under Windows 3.1. However, Windows 2000 Professional gives you the ability to start the application in a separate memory space, which creates another WOW for each 16-bit Windows application.

Remember that the WOW environment imitates Windows 3.1 so well that it even hangs just like the old Windows did! That means that if one of your 16-bit Windows applications crashes, it will take all the other 16-bit applications with it—unless you have chosen to start them in their own memory spaces, in which case the other 16-bit programs will keep running without a problem.



NOTE If a 16-bit application crashes, it will also crash any other program running in the same memory space. If this happens, you can stop the WOW by using Task Manager and restart the 16-bit program. All 32-bit applications will be unaffected by the crash and continue to run normally.

Windows 2000 Professional Weaknesses

Ah, but Windows 2000 Professional is not all wine and roses. Windows 2000 Professional isn't right for everybody, at least not yet. The problems tend to fall into the following categories:

Problems related to scale Some problems stem from the fact that a lot fewer people are using Windows 2000 Professional than are using Windows 95/98 or Windows NT. As a result, fewer people are writing applications and drivers for Windows 2000 Professional.

Problems related to expectations Other problems come from people's prior knowledge of Windows. People expect Windows 2000 Professional to have all the features of whatever operating system they already know, and Windows 2000 Professional doesn't always comply with that expectation.

Problems related to Windows 2000 Professional security The middle name of Windows 2000 Professional, security, gets it into trouble. Deciding to secure an operating system means that you can lock yourself out of your own computer. Believe it or not, it's actually possible to lock Windows 2000 Professional out of itself!

Problems related to the legacy of Windows 2000 Professional and problems that aren't really problems It inherits some unpleasant characteristics from earlier operating systems. Also, some "problems" are really shortcomings, things that it *should* be doing, but hasn't gotten to yet.

Problems Related to Scale

In more than 20 years of experience with microcomputer operating systems, I've seen a number of really good products go down in flames through no fault of their own, save for one thing: They never saw enough acceptance. Although it's true that most Windows 95/98 applications run under Windows 2000 Professional, a system requires more than applications. For example, drivers must be available to run for every piece of hardware that the application needs.

Fewer Drivers Are Available

Running your video board, network card, mouse, SCSI host adapter, or scanner requires a piece of software called a *driver*. Someone must sit down and write a driver program for each piece of hardware in your system. Of course, Windows 2000 Professional includes drivers right in the box for major pieces of hardware. But what about not-so-major pieces of hardware? For example, although obtaining a driver for the 3Com Ethernet board is easy enough, can you get one for the SMC Ethernet cards? And if a major vendor, such as 3Com, comes out with a new card, will anyone write a Windows 2000 Professional driver for that board?

The relatively small number of available drivers is definitely one of NT's main weaknesses. Windows 2000 Professional uses the same drivers as Windows 98 for *most* of its hardware, but not all. It includes a new version of the Network Device Interface Specification (NDIS) drivers for network cards. Now this isn't as bad as it may sound. In this case, the driver for Windows 98 works fine, but the information (INF) file that tells Windows 2000 Professional how to install the driver is in a different format.

The things to check out before you commit to Windows 2000 Professional are usually the drivers for your video card and sound card. SCSI and network card drivers are more plentiful, but you can't be positive that the generic Sound Blaster 16 driver will work on your Sound Blaster clone. Unless you're using a Microsoft or Logitech mouse, check for Windows 2000 Professional mouse drivers as well. Or restrict your purchases to devices on the Microsoft Hardware Compatibility List.

Now, after all the gloom, it has to be said that driver support for Windows 2000 Professional is getting better all the time. If in doubt, check the Web site for the company that made your hardware. There were many cases in which the support staff for a company told me that a driver for NT wasn't available, and then I found one right on their Web site! The moral here is that everyone, including Microsoft, has been surprised by the rapid acceptance of NT as the operating system of choice by savvy users, and we can expect that to continue with Windows 2000 Professional.

Not All Windows 95/98 Apps Run

Windows NT 4 did not include much support for Microsoft's game API DirectX, so there were many games and other applications that would not run correctly or not run at all under NT. Windows 2000 Professional includes the full DirectX 7 API set, so this should no longer be a major issue for users of Windows 2000 Professional. But there will still be issues if applications are written to look for a specific version of Windows. For instance, if a program is written to only run on Windows 95 by name, it won't run on Windows 2000 Professional. It probably won't even run on Windows 98, for that matter.

Additionally, many application programmers just plain haven't figured out the correct way to write a 32-bit application, so they cobble together some of the 16-bit things that they've already written and understand and wrap a bit of 32-bit code around it,

calling the result a Windows 95 application. Before you know it, Windows 95 is out of memory and you end up with a Windows 95 application that doesn't run under Windows 2000 Professional.

Problems Caused by the Secure Nature of Windows 2000 Professional

Windows 2000 Professional can meet the U.S. government C2 security standards, as you read earlier. Security means more than requiring password encryption and forcing you to log on to your own computer. It also means that Windows 2000 Professional is engineered not to crash—after all, stability is a kind of security. Years ago, Bob Muglia, one of NT's product managers and a very influential force in the design of the first version of NT, said to me, "Our goal is for you to set up your NT machine and *never reboot it*. If it crashes, if an application brings it down, we haven't done our job." I have to admit that I smiled inwardly when Bob made that seemingly hubristic claim. But, as you've already read, NT has achieved much of that goal; it's very stable.

(And when Bob says something, he's serious. When I called him for the interview we had scheduled, he told me, "I'm a little distracted; my wife's in the hospital getting ready to have our baby." I told him that it wasn't *exactly* necessary to do the interview at that moment, but he told me he was just waiting around for the birth anyway. Now, *there's* dedication to a job....)

But the point I'm getting to is this: the stability of Windows 2000 Professional comes with a price. Think of it this way. You have a choice between Windows 95/98 and Windows 2000 Professional for your desktop operating system. Both are, as I've already observed, pretty good, particularly when stacked up against their predecessors.

The defining criterion boils down to this: You can buy Windows 95/98 and get speed and massive compatibility; virtually all your DOS and Windows 3.1 applications will run with little or no tweaking on your part. But when Windows 95/98 crashes, you're out of luck. Or you can buy Windows 2000 Professional, which almost never crashes. It examines everything that an application wants to do before it allows the application to do it. When Windows 2000 Professional *does* crash, it provides you with full disclosure of why the crash happened via the crash dump and the `dumpexam.exe` program. But the price of vigilance is performance; a computer running Excel under Windows 2000 Professional will not run as quickly as that same computer running the same application under Windows 95/98.

You make the choice: speed and compatibility versus stability. Personally, I choose stability, but that's probably because I've lost too many important files to unstable operating systems.

Some DOS and Windows 3.x Apps Run Slowly

Because DOS and Windows 3.1 programs require watching (they're likely to do something that violates Windows 2000 Professional security), Windows 2000 Professional runs them at a noticeable speed disadvantage. If they're written "clean" (that is, if they attempt to use only the operating system to access the drive, video, mouse, and other devices), Windows 2000 Professional won't demonstrate much of a slowdown at all when compared to Windows 3.1 or DOS. But you'll have to test each program in turn.

Some DOS and Windows Apps Will Not Run

Virtually all DOS and many Windows 3.1 applications bypass the operating system and directly manipulate the PC hardware. Windows 2000 Professional can't allow this behavior, of course.

Rather than terminate those applications, Windows 2000 Professional has a sneaky way of making them *think* they're directly accessing hardware. Windows 2000 Professional uses a set of programs called VxDs, or *virtual device drivers*. Basically, a VxD exists for each device—the keyboard, the mouse, the video board, and so on—and it detects attempts by an application to access its companion hardware. The VxDs then swoop in and impersonate the hardware, leading the DOS or Windows 3.1 application to think that it actually *has* directly twiddled the hardware, making the application happy.

You do, of course, pay a price—performance. As I said earlier, this can significantly slow down execution for a DOS or Windows 3.1 application. But, even worse, it may keep the application from running in the first place.

Here's why: The designers of Windows 2000 Professional had to know what kinds of devices to build VxDs for in the first place. Most devices are easy; everybody has a keyboard, a mouse, and a video board. But what about the marginal stuff, such as a proprietary scanner interface board, fax board, or possible factory control interface card? Microsoft figured that those were small markets and didn't try to write VxDs for those boards.

Now consider what happens when you run a DOS or Windows 3.1 program that was built to work with one of these "unusual" boards. Windows 2000 Professional sees the program attempt to directly access some hardware for which Windows 2000 Professional has no VxD. What should Windows 2000 Professional do? The Windows 2000 Professional design says, "When in doubt, leave it out," and so Windows must terminate the DOS program. Is this design feature good or bad? Well, remember your choice: fast and compatible versus stable.

Problems Related to Legacy

Some of the things that Windows 2000 Professional does are just plain dumb because it's missing relatively simple and eminently possible features that Microsoft hasn't

taken the time to implement. Don't misunderstand this statement—most of the items in this section are not things that Microsoft has promised at any time. But I think they would make Windows 2000 Professional much more useful.

File Associations Rely on File Extensions

When you open a folder, you may see files with the PowerPoint, Excel, or Word icons attached to them. If you double-click the icon, Windows 2000 Professional automatically starts up the application and feeds the file to the application. But how does Windows 2000 Professional know that the file `household.xls` is an Excel spreadsheet? By its XLS file extension. This is a dumb idea for a number of reasons.

First, now that both Windows 95/98 and Windows 2000 Professional support long filenames—names that have as many as 255 characters with spaces and uppercase and lowercase letters—saving files with names such as `portfolio analysis done on 3 March 2000` makes sense; having to add `.xls` to the end of the filename is cumbersome.

Second, some file extensions could be in doubt. Whoever said that Microsoft Word should own the extension DOC? Lots of people that I know use the DOC extension for their WordPerfect documents. What, then, do they do if they want to load both Microsoft Word and WordPerfect on the same computer?

Windows 2000 Professional supports the notion of *file forking*, whereby a file contains two pieces: the data file and a small program, what Macintosh users call a *resource fork*. The resource fork is a little program that says, "To use me, run program XYZ." Thus, all Microsoft Word documents would get a resource fork that says, "To use me, run Microsoft Word." This convention would free us from file extensions and make the user interface considerably more user-intelligible. The PC world is one wherein we always seem to be hearing, "What's that? The tool we built for you doesn't fit your hand? Hmmm...have you considered surgery?"

Perhaps most embarrassing for Microsoft is (or *should* be) the fact that IBM included file forking with OS/2 way back in 1992.

Operating System Components Are Largely Static

Run Windows 2000 Professional for a while, and you'll be pleasantly surprised at how stable it is. So stable, in fact, that the only reason that I ever reboot my Windows 2000 Server machines is to add or remove a module. Some of the slightest modifications to your configuration cause Windows 2000 Professional to demand that you reboot the system.

There's no real reason to require rebooting, either. Today's mainframe and mini-computer operating systems go for months at a time without being rebooted. You can load new system software and keep on going. The big-iron folks abandoned the notion of a heavily static operating system more than a decade ago. Nevertheless, it persists in the microcomputer world as just another thing that seems to be acceptable.

Boot Windows 2000 Professional, and you can see why I feel that it's imperative for Microsoft to do something about how slowly their computers boot up. You can wait for 10 minutes for NT 4 to start a workstation with a whole bunch of drivers and services, and of course, demonstrating a slow startup on an NT Server machine is even easier. Now, Windows 2000 Professional *is* faster than NT 4 at booting, but some of us are never really satisfied when it comes to boot speed. All that is really required to reset Windows 2000 Professional during the installation of software is to stop and restart the devices and services involved in what you installed, so why doesn't someone write their program to stop and restart the services instead of requiring a reboot?

Shortcuts Don't Link Two Ways

One of the features of the Windows 2000 Professional graphical user interface is the *shortcut*. Shortcuts are small files with the extension LNK; they are usually smaller than 200 bytes. The value of a shortcut is that it provides a quick way to access some file or program.

For example, suppose you want to start up Microsoft Word. The path to the program is probably C:\Program Files\Microsoft Office\Office\WINWORD.exe. So to start it up, you might open My Computer, then the C drive, then the Program Files folder, then the Microsoft Office folder, then the Office folder, and then finally double-click the WINWORD icon, which represents WINWORD.EXE. But that's a lot of folder-diving. Instead, you could create a shortcut by putting an icon for Microsoft Word on the Desktop. Double-click the shortcut, and Word starts; so far, so good.

But this process falls down when you want to examine or change the properties of the WINWORD file. Right-click the Microsoft Word icon, and you'll get the option to examine the properties of that file; the tabs in the WINWORD Properties dialog box display information such as the file's version, size, and trademarks. But try right-clicking the shortcut, and all you'll get is an offer to find the original item. So, to modify the properties of Microsoft Word from a shortcut, you've got to open the Properties dialog box for the shortcut, which then offers you the opportunity to find the target of the original file. Right-click the original file, and choose Properties from the shortcut menu to open the Properties dialog for WINWORD an extra and annoying couple of steps.

In contrast, a truly object-oriented operating system would support shortcuts that act exactly like their original objects (objects such as Microsoft Word). Again, this feature *isn't* rocket science; OS/2 included it back in 1992.

No Bookmark Support Exists for Applications

Ever since Macintosh and Windows appeared, they—and subsequent graphical operating systems—have used the paradigm of a desktop. The basic idea has probably been to get across the notion that you get real work done with one of today's operating system, hence the desktop metaphor.

The problem is, computer desktops don't resemble my physical desktop at all. For one thing, when I leave for the day, my desktop pretty much stays the way it is. If I leave my stapler in the middle of the desktop, it doesn't spirit itself off to the upper-left corner automatically.

For another thing, my "data files" and "applications" return each day to the place where I left them. Books open to page 45 are still at page 45 the next day; folders are open to the same sheet of paper. Half-finished drawings and my drawing tool (a pen) are waiting for my return.

Compare this real-life scenario to how Windows 2000 Professional starts up in the morning. It remembers which folders you had open when you shut it down, and it reopens them, but that's it. You must restart all the applications by hand, tell them what files they were working on (or, of course, you can double-click the files themselves), and, after the files and apps are loaded, you still have to tell the application to go back to where you were yesterday.

Why not build a bookmark programming interface that application vendors could put into their applications, an interface that could remember where you left off when last running the application or, perhaps, the file? Then every data file could contain a bookmark pointing to where it was last edited. (After all, virtually every application keeps track of where you are *while* you're running it; why not just save that information?) The Desktop would then only have to remember which data files you had open when you shut down. When the Desktop started up again, it would open the files, and their file associations would enable them to tell the Desktop which applications to start. Then the bookmark pointer would restore the application to exactly the state that existed when the computer powered down. One of these days, I guess...

Summary

By now, you should have a good idea of what Windows 2000 Professional is and isn't. This may sound complicated (and it is), but it's also easy to learn. The new Explorer interface offers huge leaps in usability that make it a simple task to harness the power and complexity of this state-of-the-art operating system.

You may be saying to yourself at this point, however, "Who cares about C2 level security and flat memory models? I just want to get my work done and increase my productivity so I can get a raise." Well said, and rightly so. That's just what the rest of this book is about. Consider this, though: Now that you've begun to understand what the operating system does and how it differs from others that you may be more accustomed to, you're in a much better position to make the best use of the information you'll find in the following chapters.

So sit back, press Control+Alt+Delete to log on to your Windows 2000 Professional system, and let's go!

