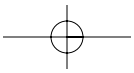
# PART I

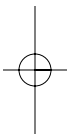
# Understanding and Planning

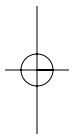


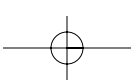
**This part of the book focuses on concepts and features of Microsoft's Exchange 2000 Server client/server electronic messaging system. It is designed to provide you with the underlying knowledge that you'll need when you tackle Exchange 2000 Server installation, administration, and management later in this book.**

**Chapter 1, "Introducing Exchange 2000 Server," presents some basic information about Exchange 2000 Server products, helping you optimize the value of these products in your organization. Chapter 2, "2000: A Magic Number?" looks in some detail at the similarities and differences between Windows 2000 Server and Windows NT Server 4; this chapter also examines the differences between Exchange 2000 Server and Exchange Server 5.5, assisting you in focusing your learning efforts if you're an old hand at NT 4 and Exchange 5.5. Chapter 3, "Two Key Architectural Components of Windows 2000 Server," attempts to break through the hype and vagaries promulgated by early technical treatises on Windows 2000, giving you a head start on the road to Windows 2000/Exchange 2000 competency. Chapter 4, "Exchange 2000 Server Architecture," focuses on the architecture of Exchange 2000, an understanding of which is essential to successful Exchange 2000 implementation. Chapter 5, "Designing a New Exchange 2000 System," covers Windows 2000 and Exchange 2000 system planning and design, facilitating your initial use of these complex products in your organization. Chapter 6, "Upgrading to Windows 2000 Server and Exchange 20000 Server," looks at the planning and design issues involved in bringing Windows 2000 and Exchange 2000 to existing Windows NT/Exchange 5.5 environments, again easing the introduction of these products in your organization.**









CHAPTER 1

# Introducing Exchange 2000 Server

***FEATURING:***

Microsoft's Exchange 2000 client/server electronic messaging system is a major player in what I call the "electronic messaging" decade. Exchange 2000 lets people work together in a variety of productivity-enhancing ways. It is one of the most exciting, innovative, and promising software products that I've ever seen.

Unlike its predecessor, Exchange Server 5.5, Exchange 2000 Server is tightly integrated into the Windows 2000 Server environment: You can't talk about Exchange 2000 Server without talking about Windows 2000 Server. This chapter concentrates on Exchange 2000 Server, but when we leave the safe confines of this introductory chapter, hardly a paragraph will go by without mention of Windows 2000 Server.

### A Confusing Array of Terms

Before we move on, let me clarify some of the terms that I'll be using. I'll use *Windows 2000 Server* to refer to the entire line of Windows 2000 server products. I'll use the names of the individual Windows 2000 Server products when referring specifically to one of them—for example, *Windows 2000 Advanced Server*. I'll follow the same conventions for Windows NT Server 4.

When I use the word *Exchange* or the words *Exchange system*, I'm talking about the whole Exchange 2000 Server client/server system.

*Exchange Server* refers to just the Exchange 2000 Server product (Server or Enterprise Edition), and an *Exchange server* is any computer running the Exchange 2000 Server product.

Got that? Okay, explain it to me.

## Exchange 2000 Server and the Electronic Messaging Decade

Electronic messaging is more than e-mail. It involves the use of an underlying messaging infrastructure (addresses, routing, and so on) to build applications that are based on cooperative tasking, whether by humans or computers. We can expect the years 1996 to 2005 to be the decade of electronic messaging *(electronic messaging),* when store-and-forward–based messaging systems and real-time interactive technolo-

gies will complement each other to produce wildly imaginative business, entertainment, and educational applications with high pay-off potential.

Microsoft's Exchange Server products have played and will continue to play a key role in electronic messaging. Exchange 2000 Server is one of the most powerful, extensible, scalable, easy-to-use, and manageable electronic messaging back ends currently on the market. Combined with Microsoft's excellent Outlook clients, Internet-based clients from Microsoft and other vendors, and third-party or home-grown applications, Exchange 2000 Server can help your organization move smoothly and productively into the electronic messaging decade.

In writing this book, I was guided by three goals:

- To share the excitement that I feel about both the promise of electronic messaging and the Exchange 2000 client/server system
- To help you decide if there's a place for Exchange 2000 Server in your organization
- To provide information and teach you skills that you'll need to plan for and implement Exchange 2000 Server systems of any size and shape

The rest of this chapter introduces you to the Exchange 2000 client/server system. We start with a quick look at several of the neat ways that you can use Exchange for e-mail and more, and then we focus on some of Exchange's key characteristics and capabilities. This is just an introduction, so don't worry if you don't understand everything completely by the end of this chapter. Everything that we discuss here we also will cover in more detail later in the book.

# Exchange 2000 Server Applications

I dare you not to get excited about electronic messaging and Exchange 2000 Server as you read this section. Just look at what's possible, and imagine what you could do with all this potential.

Exchange supports a range of e-mail protocols, including Microsoft's own proprietary Mail Application Program Interface (MAPI), as well as the Internet standard protocols Post Office Protocol version 3 (POP3) and the often overlooked Internet Message Access Protocol version 4 (IMAP4). But that's just the tip of the iceberg. Exchange servers can host user and organizational calendars, e-mail–enabled contact lists, to-do lists, notes, and other data. Users can access all this data using standard personal computer-based e-mail clients, Web browsers, and even those tiny personal digital assistants (PDAs) that are all the rage today.

Exchange servers are also great places to build and support custom applications. You can build simple applications using existing products such as Microsoft Word or Excel. If your application needs are more complex, you can turn to Exchange-based forms. And, if you've got the need, time, and skills, you can build applications using programming languages such as Java, Visual Basic, or C++. Finally, you can use the built-in sorting and searching capabilities of Exchange public folders to build some pretty powerful applications.

## E-Mail Is Only the Beginning

Together, Exchange 2000 Server and its clients perform a variety of messaging-based functions. These include e-mail, message routing, scheduling, and support for several types of custom applications. Certainly, e-mail is a key feature of any messaging system, and the Outlook Calendar is by far better than previous versions of Microsoft's appointment and meeting-scheduling software. (Figures 1.1 and 1.2 show the Outlook 2000 client Inbox and Calendar for Windows in action.) Take a look at Figures 1.3, 1.4, and 1.5 for a glimpse of the Internet-based POP3, IMAP4, and Web browser clients that you can use with Exchange 2000 Server.

**FIGURE 1.1**

*The Outlook 2000 client for Windows Inbox*

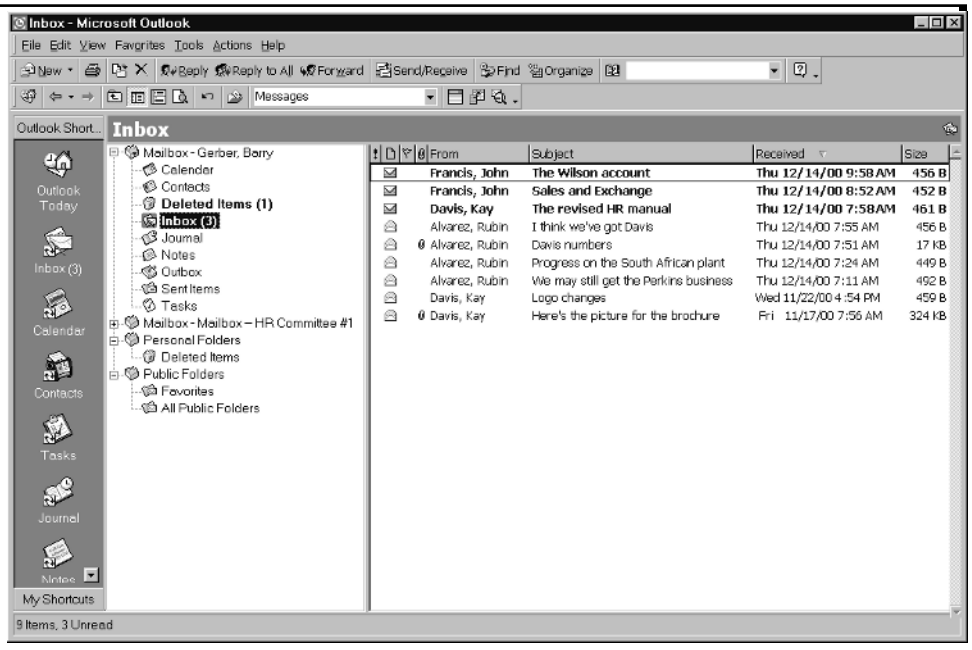


**FIGURE 1.2**

*The Outlook 2000 client Windows Calendar*

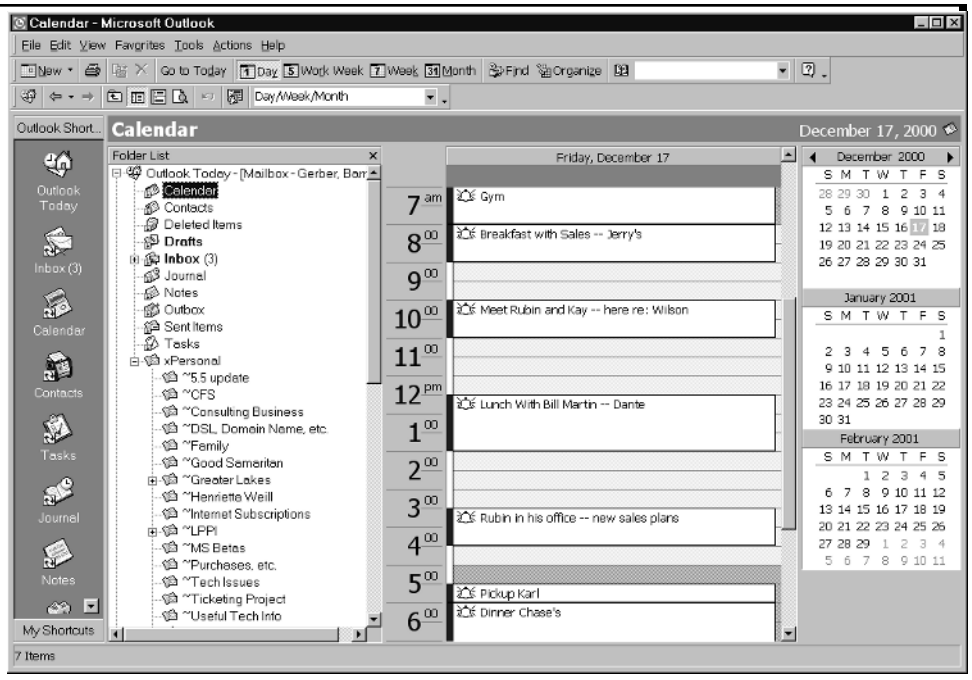



**FIGURE 1.3**

*Qualcomm's Eudora Pro 3.0 POP3-compliant client accesses mail stored on an Exchange server.*

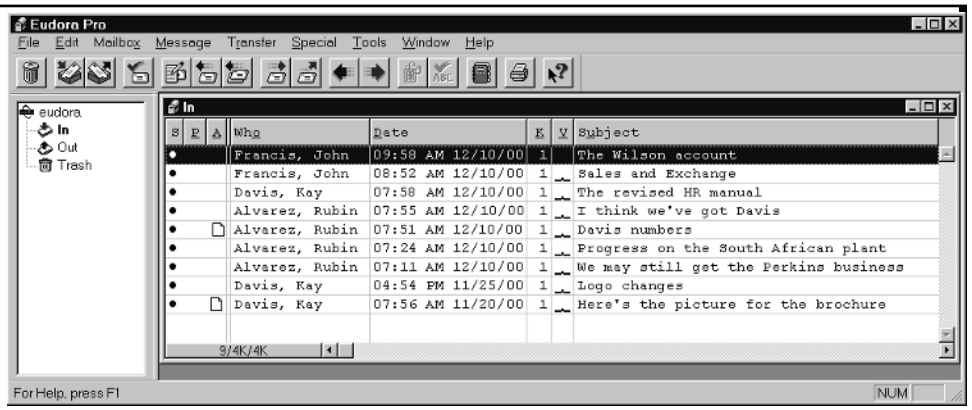


**FIGURE 1.4**

*Microsoft's Outlook Express IMAP4 client function accesses messages and folders on an Exchange server.*

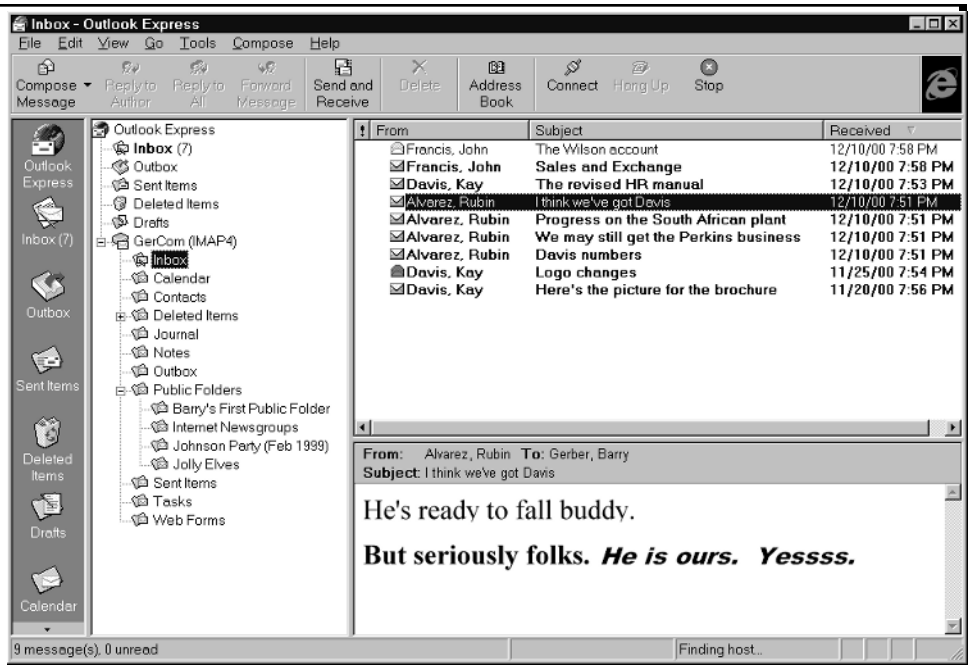
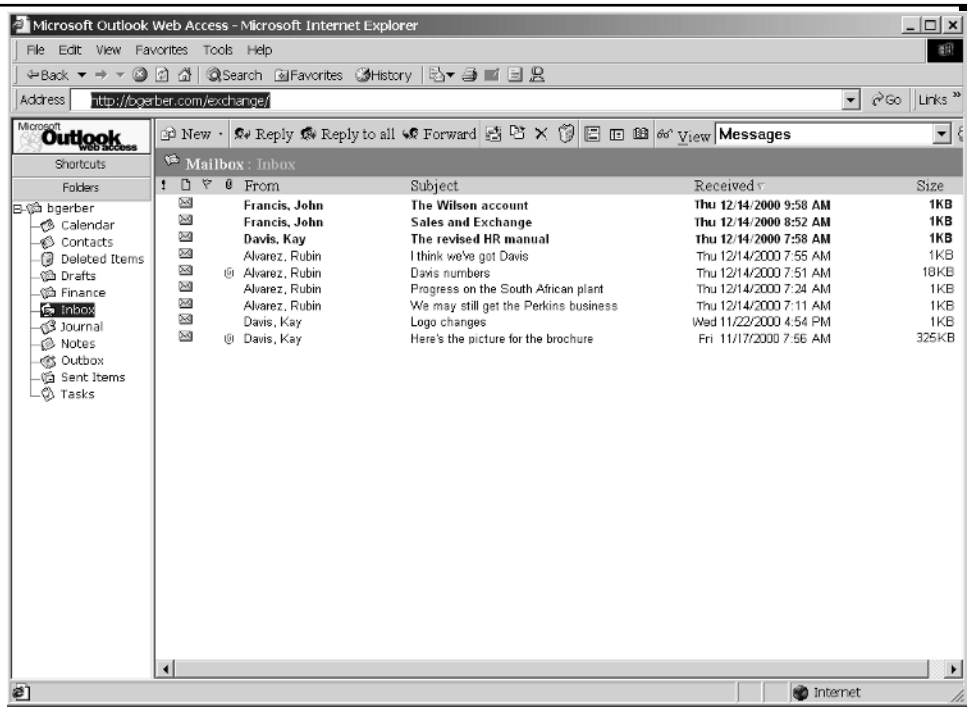



**FIGURE 1.5**

*Microsoft's Internet Explorer Web browser accesses mail stored on an Exchange 2000 server.*



E-mail clients are exciting and sexy, but to get the most out of Exchange 2000 Server, you need to throw away any preconceptions you have that messaging packages are only for e-mail and scheduling. The really exciting applications are not those that use simple e-mail or scheduling, but those that are based on the routing capabilities of messaging systems. These applications bring people and computers together for cooperative work.

So what do these hot apps look like? Let's start with the simplest and move toward the more complex.

### Change Is the Name of the Game

Some of the marvelous user interfaces that you see in Figures 1.1 through 1.5 may look very different by the time you read this book. Software development and marketing is running at hyperspeed, especially in the world of electronic communications. Updates and even major revisions hit the market at a breakneck pace. The Internet makes it even easier for vendors to market and deliver their wares. New pieces and parts of applications appear almost daily for manual or totally automatic download and installation.

The basic architecture of Exchange 2000 Server and its clients is unlikely to change much over the next year or so, but the outward appearance of user interfaces is much more likely to change. As far as Exchange goes, plan for change as a way of life. Keep an open mind and at least one eye on Microsoft's Exchange-oriented Web pages.

I will admit that I sometimes long for the days of yearly or less-frequent updates on low-density 5¼-inch floppies. In the long run, however, all of this hyperactivity will prove a good thing. Our requirements will find their way *into* products faster, and bugs will find their way *out* of products faster.

## Just a Step beyond Mail

You're probably familiar with e-mail *attachments*—those word processing, spreadsheet, and other work files that you can drop into messages. Attachments are a simple way to move work files to the people who need to see them.

Sure, you could send your files on diskette or tell people where on the network they can find and download the files. But e-mail attachments let you make the files available to others with a click of their mouse buttons: Recipients just double-click on an icon, and the attachment opens in the original application that produced it (if your correspondent has access to the application, of course).

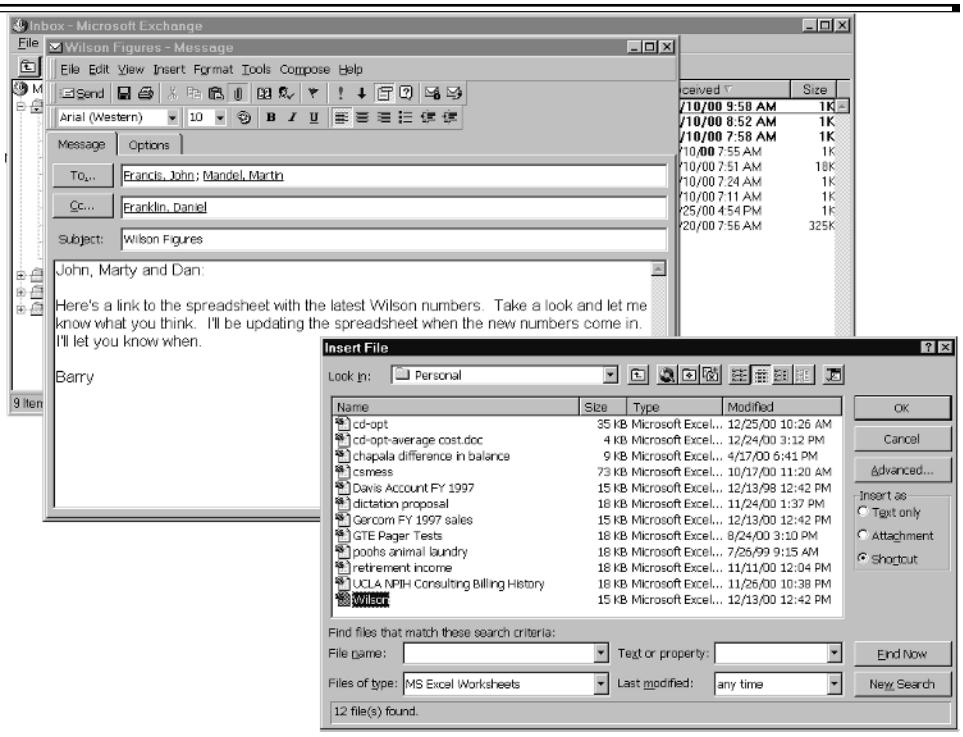
Using attachments offers the added advantage of putting the files and accompanying messages right in the faces of those who need to see them. This leaves less room for excuses such as "Oh, I forgot" or "The dog ate the diskette."

As great as attachments can be, they have one real weakness: The minute that an attachment leaves your Outbox, it's out of date. If you do further work on the original file, that work is not reflected in the copy that you sent to others. If someone then edits a copy of the attached file, it's totally out of sync with the original and all other copies. Getting everything synchronized again can involve tedious hours or days of manually comparing different versions and cutting and pasting them to create one master document.

Exchange offers several ways to avoid this problem. One of the simplest is the *attachment link* or *shortcut:* Instead of putting the actual file into a message, you put in a link to the file (see Figure 1.6), which can be stored anywhere on the network. The real kicker is that the file can also be stored in Exchange public folders (more about these in the section "Applications Using Exchange Public Folders," later in this chapter). When someone double-clicks on an attachment link icon, the linked file opens. Everyone who receives the message works with the same linked attachment, so everyone reads and can modify the same file. Of course, your users will have to learn to live with the fact that only one person can edit an application file at a time. Most modern end user apps warn the user of this fact and allow the user to open a read-only copy of the file. Third-party applications offer tighter document checkout control (see Appendix A "Cool Third-Party Applications for Exchange Server and Outlook Clients.")

**FIGURE 1.6**

*Exchange shortcuts keep attachments alive.*



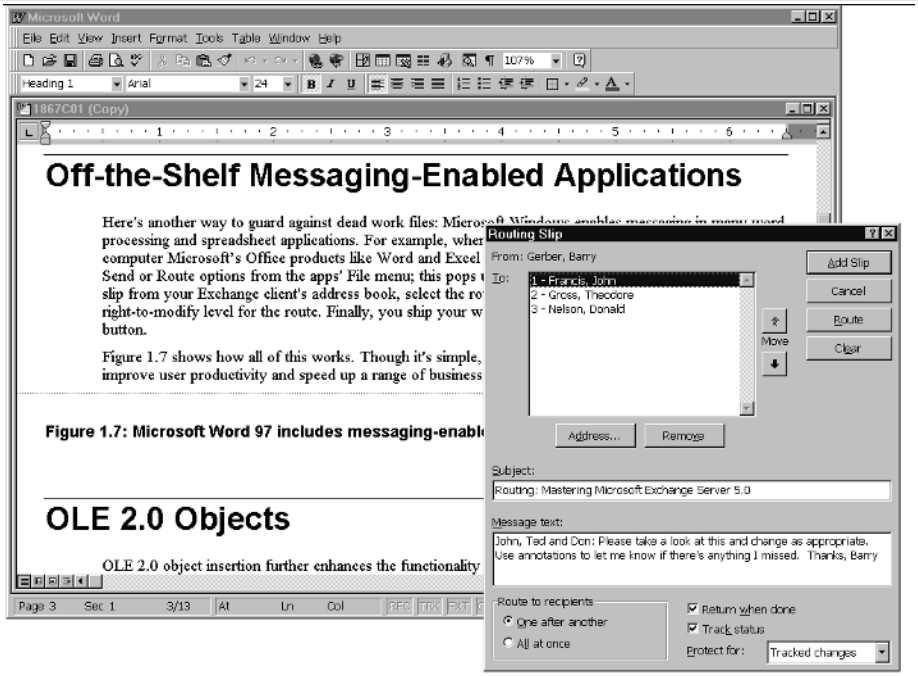
## Off-the-Shelf Messaging-Enabled Applications

Here's another way to guard against dead work files: Microsoft Windows enables messaging in many word processing and spreadsheet applications. For example, when you install the Outlook client on your computer, Microsoft's Office products such as Word and Excel are electronic messaging-enabled. You can select Send or Route options from the app's File menu; this pops up a routing slip. You then add addresses to the slip from your Outlook client's address book, select the routing method that you want to use, and assign a right-to-modify level for the route. Finally, you ship your work off to others with just a click of the Route button.

Figure 1.7 shows how all of this works. Although it's simple, application-based messaging can significantly improve user productivity and speed up a range of business processes.

**FIGURE 1.7**

*Microsoft Word 97 includes messaging-enabled functions for sending and routing.*




## OLE 2.0 Objects

OLE 2.0 object insertion further enhances the functionality of the Exchange messaging system. Take a close look at Figure 1.8. Yes, the message includes an Excel spreadsheet and chart. The person who sent the message simply selected Object from the Insert menu that appears on every Exchange message. The Outlook client then inserted a blank Excel spreadsheet into the message as an OLE 2.0 object. Having

received the message, we can see the spreadsheet as an item in the message, as shown in the figure. When we double-click on the spreadsheet, Excel is launched and Excel's menus and toolbars replace those of the message (see Figure 1.9). In essence, the message becomes Excel.

The Excel spreadsheet is fully editable. Although Excel must be available to your recipients, they don't have to launch it to read and work on the spreadsheet. Even if your recipients don't have Excel, they can still view the contents of the spreadsheet, although they won't be able to work on it. (That is, even if they don't have the app, they can still view the object when they open the message.)

**FIGURE 1.8**

*With OLE 2.0 objects, sophisticated messaging-enabled applications are easy to build.*

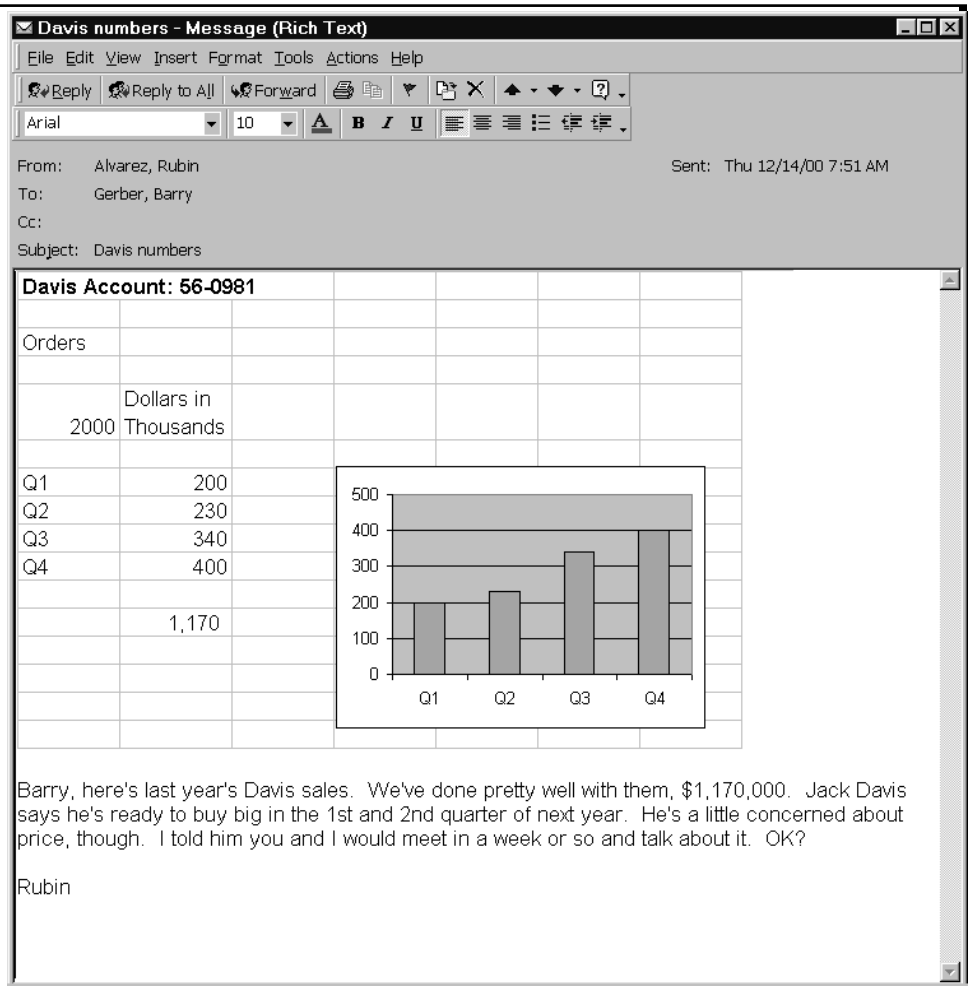
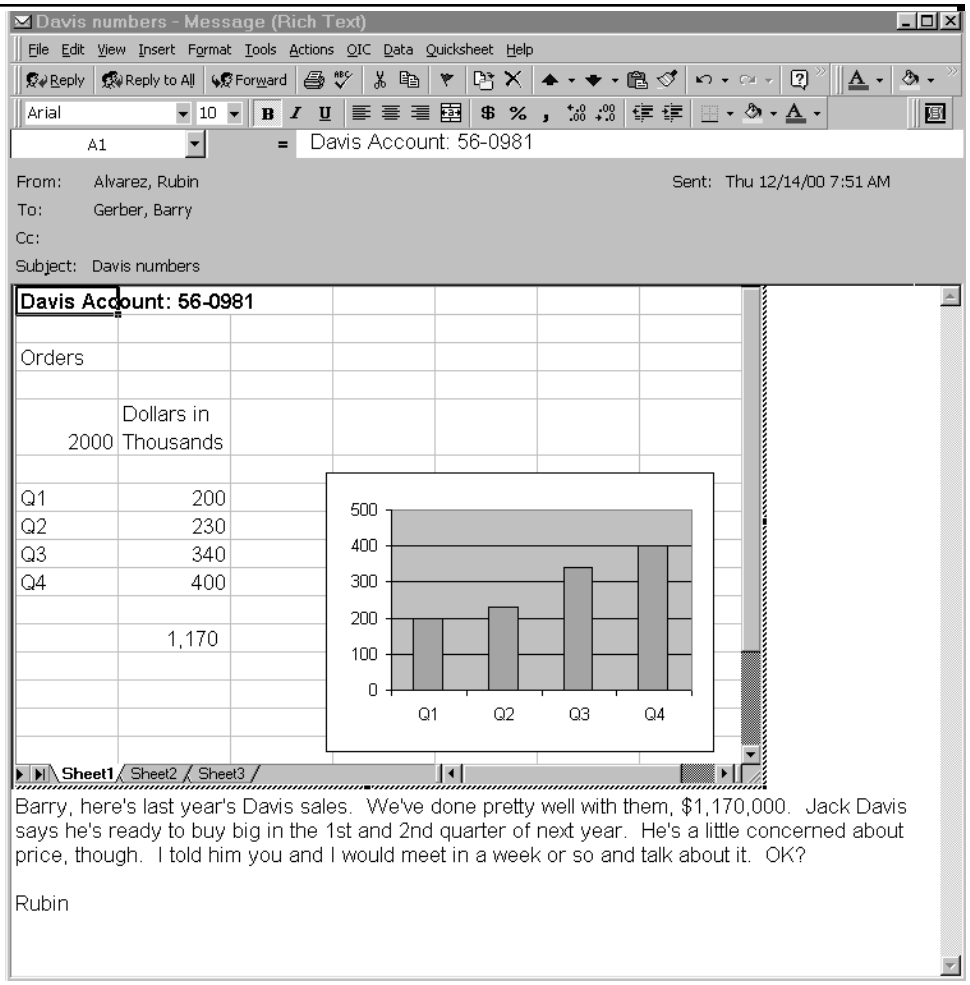


**FIGURE 1.9**

*Double-clicking on an OLE 2.0-embedded Excel spreadsheet in a message enables Excel menus and toolbars.*




## Electronic Forms

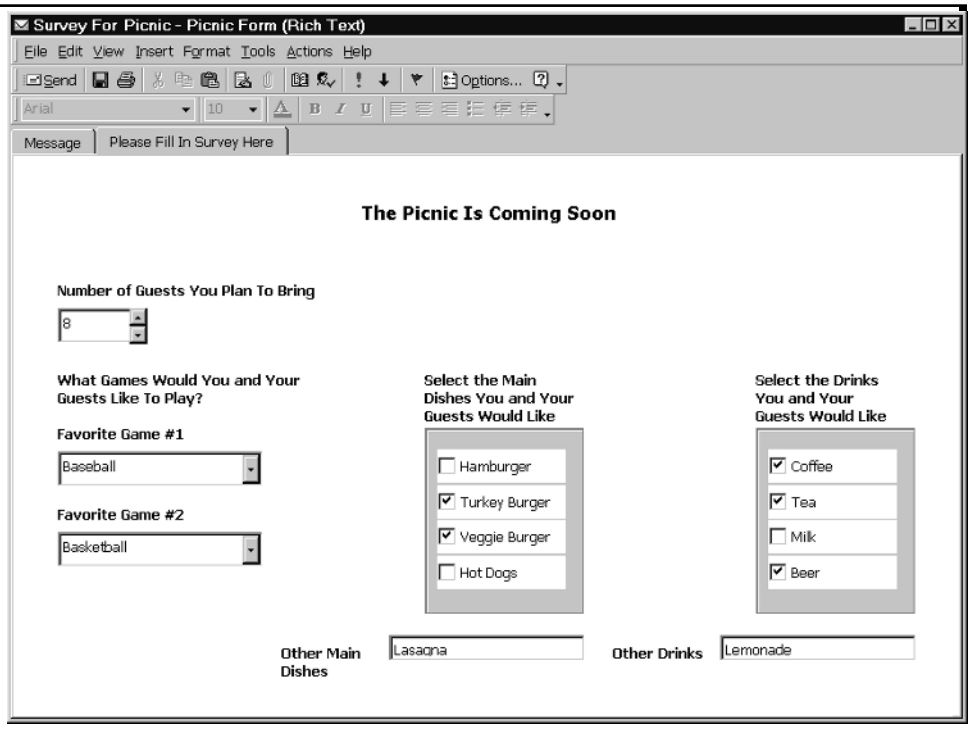
Exchange 2000 Server supports Outlook Forms Designer (OFD). You can use OFD to build information-gathering forms containing a number of the bells and whistles that you're accustomed to in Windows applications. These include drop-down list boxes, check boxes, fill-in text forms, tab dialog controls, and radio buttons (see Figure 1.10).

OFD, which is easy enough for nontechnical types to use, includes a variety of messaging-oriented fields and actions. For example, you can choose to include a preaddressed To field in a form so that users of the form can easily mail it off to the appropriate recipient. (The preaddressed To field for the form shown in Figure 1.10 is

on the page with the tab marked Message.) When you've designed a form, you can make it available to all users or select users, who can access the completed form simply by selecting it while in an Outlook client.

**FIGURE 1.10**

*Electronic forms turn messages into structured information-gathering tools.*




I discuss OFD further in Chapter 18, "Building, Using, and Managing Outlook Forms Designer Applications." That chapter also includes a nice hands-on exercise using OFD.

## Applications Built on APIs

If all this functionality isn't enough, you can go to the heart of Exchange Server and use its *application programming interface (API)*. Exchange Server supports both the Simple and Extended versions of Microsoft's Windows-based Mail Application Program Interface (MAPI). It also supports the X.400-oriented, platform-independent Common Mail Call (CMC) APIs, which have functions similar to those of Simple MAPI. Using Simple MAPI or CMC, you can build applications that use electronic messaging addresses behind the scenes to route data between users and programs. Extended

MAPI lets you get deeper into Exchange's storage and electronic messaging address books to create virtually any messaging-enabled application that you can imagine.

These custom-built applications may involve some level of automation, such as performing regular updates of your company's price lists for trading partners or sending a weekly multimedia message from the president to employees at your organization. Building apps based on MAPI or CMC requires someone with programming skills in languages such as Visual Basic or C++, and this is beyond the scope of this book.

## Applications Using Exchange Public Folders

As you'll discover later in this chapter and in chapters to come, Exchange Server supports mailboxes and private and public folders. All of these can hold messages and any kind of computer application or data file. Mailboxes and private folders are the places where Exchange users store and manage their messages and files. Public folders are for common access to messages and files. Files can be dragged from file access interfaces, such as Explorer in Microsoft's Windows 98, NT 4, and Windows 2000, and can be dropped into mailboxes or private or public folders. If you begin thinking of mailboxes and private and public folders as a messaging-enabled extension of Explorer, you'll have a fairly clear picture of Microsoft's vision of the future as to how an operating system organizes and displays stored information.

You can set up sorting rules for a mailbox or a private or public folder so that items in the folder are organized by a range of attributes, such as the name of the sender or creator of the item, or the date that the item arrived or was placed in the folder. Items in a mailbox or private or public folder can be sorted by conversation threads. You can also put applications built on existing products such as Word or Excel, or with Exchange or Outlook Forms Designer, server scripting, or the API set into mailboxes and private or public folders. In mailboxes and private folders, these applications are fun for one, but in public folders, where they are accessed by many people, they can replace the tons of maddening paper-based processes that abound in every organization.

If all this isn't already enough, Exchange is very much Internet-aware. With Exchange 2000 Server, you can publish all or selected public folders on the Internet, where they become accessible with a simple Internet browser. You can limit Internet access to public folders to only users who have access under Windows 2000 Server's security system, or you can open public folders to anyone on the Internet. Just think about it: Internet-enabled public folders let you put information on the Internet without the fuss and bother of Web site design and development. Any item can be placed on the Internet by simply adding a message to a public folder.

Before we leave public folder applications, I want to mention one more option: Exchange 2000 Server enables you to bring any or all of those Usenet Internet newsgroups to your public folder environment. With their Outlook clients, users then can read and reply to newsgroup items just as though they were using a standard newsgroup reader application. Exchange Server comes with all the tools that you need to do this. All you need is an Internet connection, access to a host computer that can provide you with a feed of newsgroup messages, and a set of rules about which groups to exclude. (Remember, this is where the infamous alt.sex newsgroups live.)

### A New Era For Exchange-Oriented Web Application Developers

Though it's beyond the scope of this book I must say something about the fantastic new programming options enabled by Microsoft's exposing Exchange 2000 Server's Information Store through the Windows 2000 file system and the Web. Using a variety of built-in and custom file system, HTML, and other commands, it's possible to program sophisticated custom applications with third-party products and Microsoft proprietary products ranging from Word to Visual Basic and C++. For an excellent introduction to this exciting new development opportunity, see Mindy Martin, *Programming Collaborative Web Applications with Microsoft Exchange 2000 Server* (Microsoft Press, 2000).

## Some Exchange 2000 Server Basics

It's important to get a handle on some of Exchange's key characteristics and capabilities. When you do, you'll better appreciate the depth and breadth of Microsoft's efforts in developing Exchange, and you'll be better prepared for the rest of this book. In this section, we'll take a look at these topics:

- Exchange as a client/server system
- The Outlook client
- Exchange Server's dependency on Microsoft's Windows 2000 Server
- Exchange Server's object orientation
- Exchange Server scalability
- Exchange Server security
- Exchange Server and other electronic messaging systems
- Third-party applications for Exchange Server

Taken together, Exchange 2000 Server's attributes make it a powerful, flexible, and extensible platform, capable of meeting the needs of small and large enterprises alike.

## Exchange 2000 Server as a Client/Server System

The term *client/server* has been overused and overworked. To put it simply, there are two kinds of networked applications: shared-file and client/server.

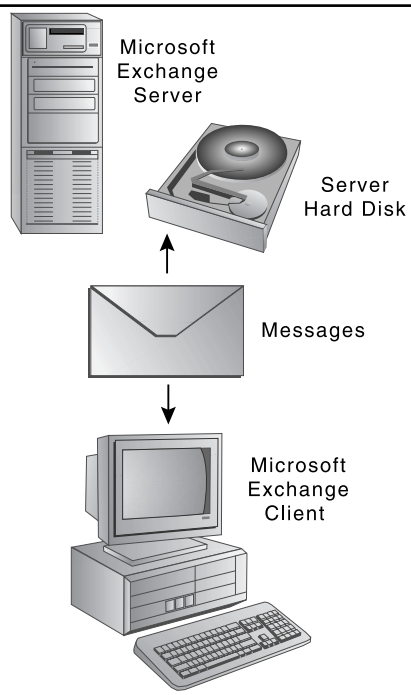
### Shared-File Applications

Early networked applications were all based on *shared-file* systems. The network shell that let you load your word processor from a network server also allowed you to read from and write to files stored on a server. At the time, this was the easiest and most natural way to grow networked applications.

Microsoft Mail for PC Networks is a shared-file application. You run Windows, OS/2, DOS, or Macintosh front ends, which send and receive messages by accessing files on a Microsoft Mail for PC Networks post office that resides on a network file server. The front end and your PC do all the work; the server is passive. Figure 1.11 shows a typical Microsoft Mail for PC Networks setup.

**FIGURE 1.11**

*Microsoft Mail for PC Networks is a typical shared-file electronic messaging system.*




Easy as it was to develop, this architecture leads to some serious problems in today's networked computing world:

- Changing the underlying structure of the server file system is difficult because you have to change both the server and the client.
- System security is always compromised because users must have read and write permissions for the whole server file system, which includes all other users' message files. Things are so bad that a naive or malicious user can actually destroy shared-file system databases in some cases.
- Network traffic is high because the front end must constantly access indexes and hunt around the server's file system for user messages.
- Because the user workstation writes directly to shared files, the server-based files can be destroyed if workstation hardware or software stops functioning for some unexpected reason.

Shared-file applications are in decline. Sure, plenty of *legacy* (that is, out-of-date) apps will probably live on for the data processing equivalent of eternity, but client/server systems have quickly supplanted the shared-file model. This is especially true in the world of electronic messaging.

## Client/Server Applications

Though they have some limitations of their own, client/server applications overcome the shortcomings of shared-file apps. So, today, networked applications increasingly are based on the client/server model. The server is an active partner in client/server applications. Clients tell servers what they want done, and if security requirements are met, servers do what they are asked.

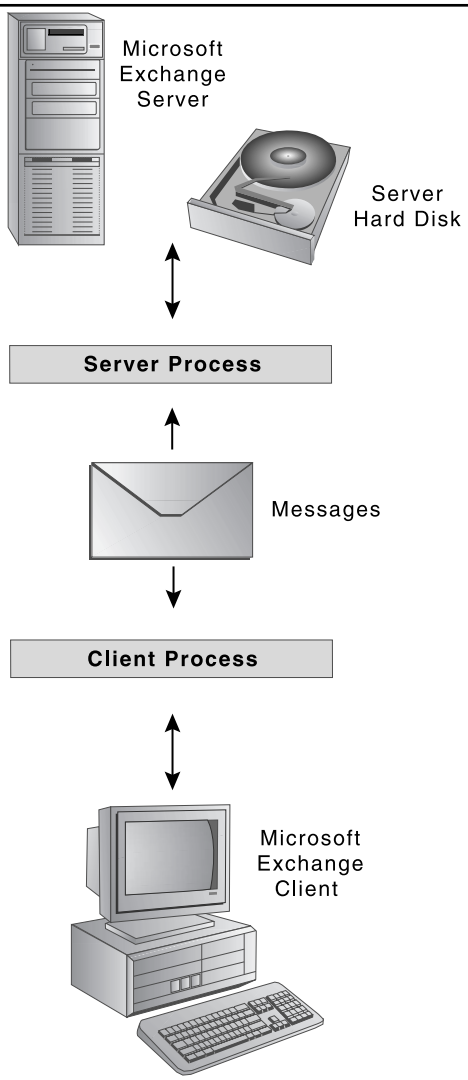
Processes running on a server find and ship data to processes running on a client. When a client process sends data, a server receives it and writes it to server-based files. Server processes can do more than simply interact with client processes. For example, they can compact data files on the server or—as they do on Exchange Server—automatically reply to incoming messages to let people know, for instance, that you're going to be out of the office for a period of time. Figure 1.12 shows how Exchange implements the client/server model.

Client/server applications are strong in all the areas in which shared-file apps are weak:

- Changing the underlying structure of the server file system is easier than with shared-file systems because only the server processes access the file system.
- System security can be much tighter, again because only the server processes access the file system.
- Network traffic is lighter because all the work of file access is done by the server, on the server.

- Because server processes are the only ones that access server data, breakdowns of user workstation hardware or software are less likely to spoil data. With appropriate transaction logging features, client/server systems can even protect against server hardware or software malfunctions.

**FIGURE 1.12**
*Microsoft Exchange is based on the client/server model.*



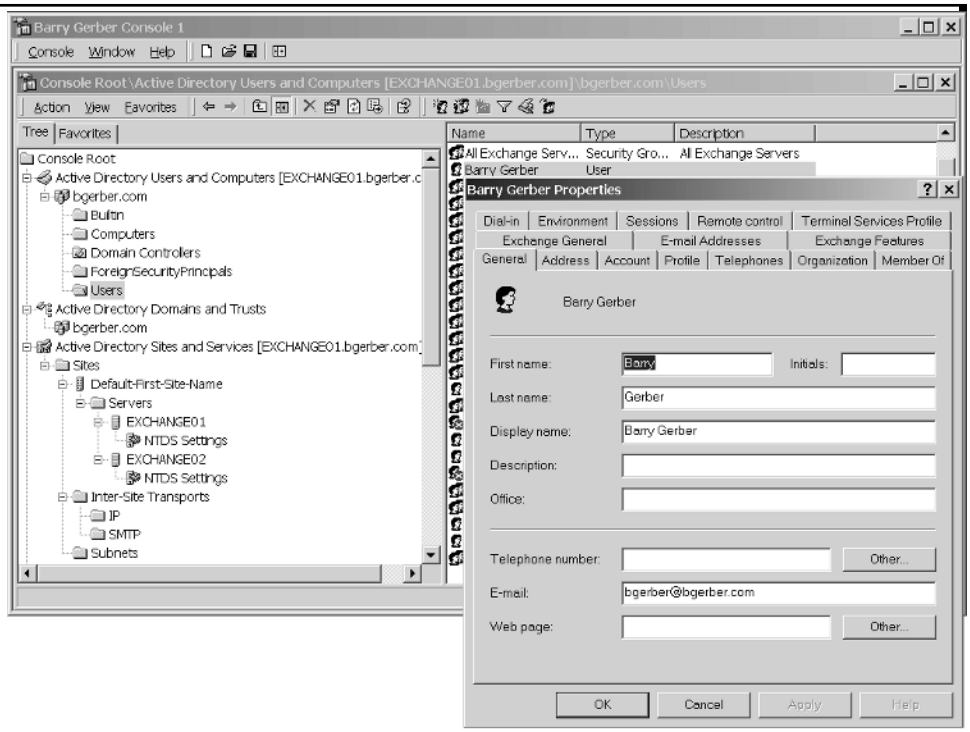
As good as the client/server model is, it does have some general drawbacks. Client/server apps require more computing horsepower, especially on the server side. With Exchange, you should plan to start with very fast Pentium machines, lots of RAM, and plenty of hard disk capacity—and expect to grow from there.

Client/server applications are more complex than shared-file apps. This is partly because of the nature of the client/server model and partly because of the tendency of client/server apps to be newer and thus filled with all kinds of great capabilities that you won't find in shared-file applications. Generally, you're safe in assuming that you'll need to devote more and more sophisticated human resources to managing a client/server application than to tending to a similar one based on shared files.

The good news is that Microsoft has done a lot to reduce the management load and to make it easier for someone who isn't a computer scientist to administer an Exchange system. I've looked at many client/server messaging systems, and I can say without any doubt that Exchange is absolutely the easiest to administer, even in its slightly more complex 2000 implementation. Exchange 2000 Server includes a set of graphical user interfaces (GUIs) that organize the processes of management very nicely. With these interfaces, you can do everything from adding users to assessing the health of your messaging system. In Figure 1.13, I'm using the Windows 2000 Server Active Directory Users and Computers interface to modify an Exchange user's mailbox.

**FIGURE 1.13**

*Managing an Exchange user's mailbox in the Active Directory Users and Computers interface*



## A Quick Take on the Outlook Client

As should be clear from our look at some of its applications earlier in this chapter, the Outlook client is the sexy part of Exchange. This is where the action is, the view screen for the backroom bits and bytes of Exchange Server. Although this book is mostly about Exchange Server, you can't implement an Exchange system without the clients. So, we'll spend some time on the Outlook client in various places in this book. Meanwhile, let's discuss some client basics.

### Information Storage

The client stores information in one of two places on an Exchange 2000 server: a mailbox store or a public store. Each has a different purpose and function. Furthermore, an Outlook client can have personal folders, which reside outside the Exchange Server environment.

#### Mailbox Stores and Mailboxes

Mailbox stores contain individual Exchange 2000 Server user mailboxes. Mailboxes can send and receive messages. Although you can share their contents with others, mailboxes generally hold items to which you alone have access. You access mailboxes using an Exchange client or Internet-based clients such as the POP3 and IMAP4 clients built into Microsoft's Outlook Express.

You can add folders to a mailbox to help you organize your messages. If you have the rights to other mailboxes, you can open them in your Exchange client as well.

#### Public Stores and Public Folders

Public folder stores stores contain, you guessed it, public folders, which hold items that you want others to see. Users whom you authorize can create public folders and then drag and drop anything that they want into them. Public folders can also be nested, and rules can be applied to them.

Public folders are key to the organization-wide implementation of Exchange. Some, all, or none of an Exchange server's public folders can be automatically replicated to other Exchange servers. This lets you post items to public folders on one Exchange server and have them quickly and painlessly appear on any combination of the Exchange servers in your system. Even without replication, users all over your organization can access public folders.

#### Personal Folders: Another Place for Clients to Store Information

Outlook has personal folders that reside outside the Exchange Server on local or network hard disks. Personal folders may or may not have the send and receive capabilities of mailboxes. You can create as many personal folders as you want, and a personal folder can hold as many subfolders as you want. Like the folders that you add to mailboxes, personal folders help you organize information. You can drag and drop

messages between folders. Using *rules* (discussed in the section "Rules," later in this chapter) you can direct incoming mail into any of your personal folders.

## Sharing Information

You can share information with others by sending it to them or placing it in public folders for them to retrieve on their own. You can drop messages, word processing documents, other work files, and even whole applications into public folders. You can use public folders to implement many of the kinds of applications that I mentioned at the beginning of this chapter.

For example, instead of electronically routing a draft word processing document to a bunch of colleagues, you can just drop it into a public folder. Then you can send e-mail to your colleagues asking them to look at the document and even to edit it right there in the public folder.

## Organizing Information

Creating a set of personal and public folders and then dropping messages in them is a simple way to organize information. More sophisticated approaches include the use of rules, views, and the Exchange client's Finder.

### Rules

As a user, you can set up a range of *rules* to move mail from your Inbox into personal or public folders. For example, you might want to move all the messages from your boss into a folder marked "Urgent." Rules can be based on anything from the sender of a message to its contents. Depending on its type, a rule may run on the Exchange server or on the client. The Outlook client doesn't have to be running for server-based rules to execute.

### Views

Exchange messages can have numerous attributes. These include the obvious, such as sender, subject, and date received, as well as less common information, including the sender's company, the last author, and the number of words. You can build views of messages using almost any combination of attributes and a variety of sorting schemes. Then you can apply a particular view to a folder to specially organize the messages that it contains.

### The Finder

You can use the Outlook client Finder to search all folders or a single folder for messages from or to specific correspondents; messages with specific information in the subject field or message body, or attachments to messages; and even messages received between specific dates or of a specific size.

## Exchange 2000 Server's Dependency on Windows 2000 Server

Exchange 2000 Server runs only on specific versions of Windows 2000 Server. It won't run on Windows NT Server, Windows 2000 Professional, or Windows 98.

Among operating systems, Windows 2000 Server is the new kid on the block. As a longtime Windows NT Server user, I initially faced Windows 2000 Server with more than a little fear and foreboding. That was then, however. Now I am a confirmed Windows 2000 Server user and supporter. My personal workstation is a Windows 2000 Server-based machine, and all my servers but one run Windows 2000 Server. (The one holdout is a NetWare server that I use to ensure that Windows 2000 Server and Windows-based software work with Novell's IPX/SPX.)

It took me about two weeks to get comfortable with Windows 2000 Server, and it took a month or so to become totally productive with it. What sets Windows 2000 Server apart from all other operating systems for workstations and servers is Microsoft Windows. Windows 2000, whether the workstation or server version, *is* Microsoft Windows. If you can use Windows 98, you can get started using Windows 2000 Server in no time. You'll have to learn how to accomplish various server-related tasks, but once you figure out how to do a task, the Windows graphical user interface greatly simplifies performing almost any task. Networking with Windows 2000 Server is pretty much a breeze if you understand a few basic concepts, and running apps on top of Windows 2000 Server is a piece of cake. Figure 1.14 shows one of my Windows 2000 Server/Exchange Server desktops with some Windows 2000 Server and Exchange 2000 Server management applications running. This shouldn't be foreign territory for any Windows aficionado.

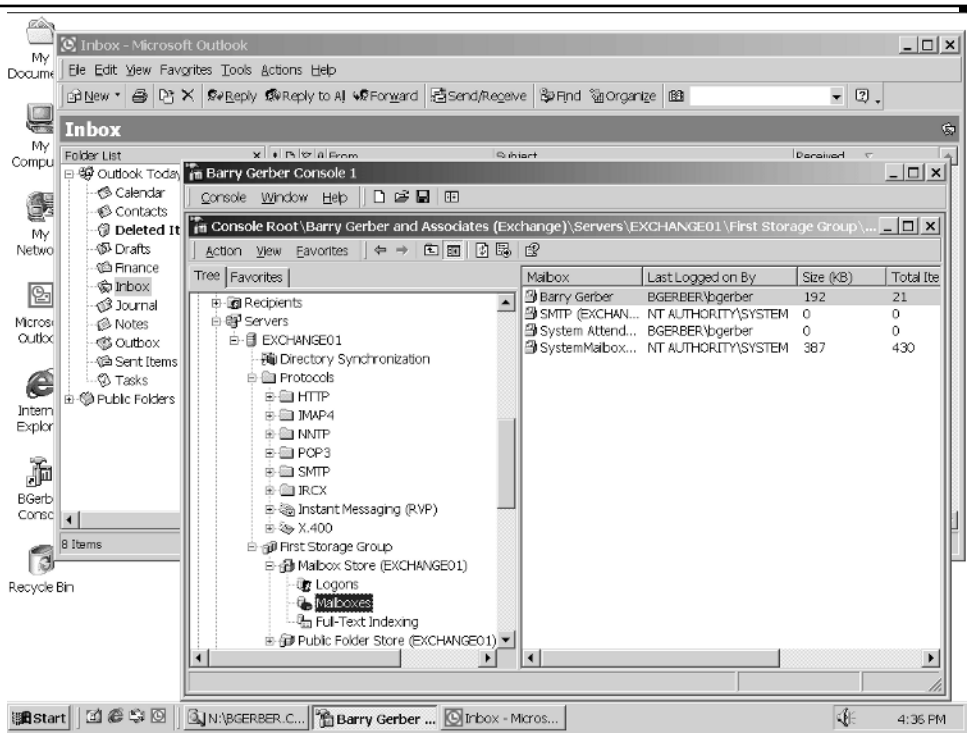
Windows 2000 Server is chock-full of features that make it an especially attractive operating system. One of these is its very usable and functional implementation of Microsoft's domain-based security system. Domains have names—mine is called bgerber.com—and include Windows 2000 servers, Windows 2000, and Windows NT workstations, and all flavors of other Windows- and DOS-based machines. Although there are a number of ways to approach domain structure and security, the general rule is that the members of a domain can use any resource that they have been given permission to use in the domain—disk files, printers, and so on—no matter where these resources reside. Exchange 2000 Server depends on Windows 2000 Server domain structure and security for its security.

In Chapter 2, "2000: A Magic Number"; Chapter 3, "Two Key Architectural Components of Windows 2000 Server"; and Chapter 6, "Upgrading to Windows 2000 Server and Exchange 2000 Server," you'll read a lot more about Window 2000 Server

and what you need to know about it to run Exchange Server. You'll install Windows 2000 Server in Chapter 7, "Installing Windows 2000 Server."

**FIGURE 1.14**

*On the surface, Windows 2000 Server is just plain old Microsoft Windows.*

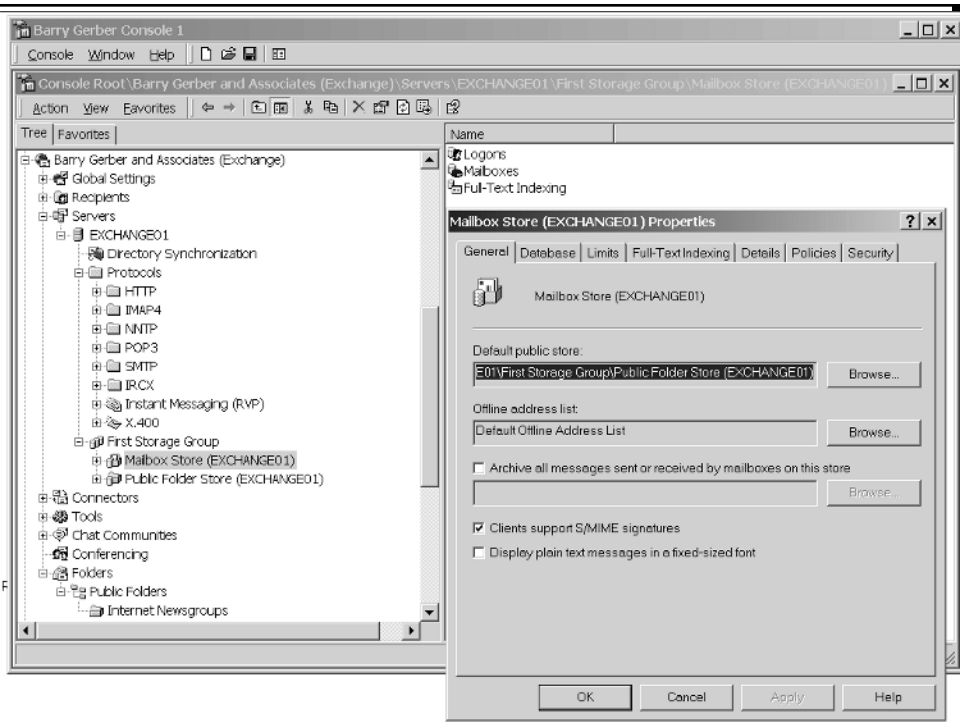



## Exchange 2000 Server's Object Orientation

Exchange is a classic example of an *object-oriented* system. Figure 1.15 shows the main tool for managing an Exchange 2000 Server organization, Exchange System Manager. Take a look at all those items on the tree on the left side of the tool, such as Barry Gerber and Associates (Exchange), Servers, EXCHANGE01, Protocols, and First Storage Group. Each of these is an *object*. Each object has attributes and can interact with other objects in specific ways. Exchange objects can hold other objects, serving as what Microsoft calls *containers*.

**FIGURE 1.15**

*Exchange 2000 Server's object orientation is evident in the Exchange System Manager tool.*




Barry Gerber and Associates is the name of my consulting business; it is the equivalent of a company name such as IBM or TRW. This top-level object is called the *organization*. The Barry Gerber and Associates organization contains all the objects below it.

Servers is the container that holds the individual servers in the Barry Gerber and Associates enterprise hierarchy. Right now, there's only one server in the Servers container, EXCHANGE01, but just you wait.

The First Storage Group object contains mailbox and public stores; remember those from earlier in this chapter?

Take a look at the dialog box on the right side of Figure 1.15. It shows the properties or attributes of the mailbox store. Objects have attributes. The attributes of a mailbox store object are puny compared to those of a Windows 2000 Server/Exchange 2000 Server user: Compare this to the number of tabs on the user object back in Figure 1.13.

Object orientation makes it easy for Microsoft to distribute Exchange 2000 Server's functionality and management, and it makes it easy for you to administer an Exchange Server environment. For example, based on my security clearances, I can manage any set of Windows 2000 Server/Exchange 2000 Server users, from only a small group of users to all the users in my Windows 2000 Server domain.

## Exchange 2000 Server Scalability

Exchange 2000 Server scales very well both vertically and horizontally. Windows 2000 Server runs on top of computers based on single and multiple Intel processors, so it's very easy to scale an Exchange server upward to more powerful hardware when increased user loads make additional computing power necessary. You can also cluster Windows 2000 Server/Exchange 2000 Servers so that they mirror each other in a fault-tolerant way and share the load placed on them by users. This is another way to vertically scale Exchange 2000 Server systems.

If vertical scalability isn't what you need, horizontal scaling is also a breeze with Exchange Server. You can set up a new Exchange server and quickly get its directory and public folders in sync with all or some of your other servers. You can even move mailboxes between Exchange servers with a few clicks of your left mouse button.

How do you know whether it's time to scale up or out? Microsoft has an answer for this, too: You can use the load simulation tools that Microsoft provides to simulate a range of different user loads on your server hardware. By analyzing the results of your tests, you'll get some idea of the messaging loads that you can expect a server to handle in a production environment.

## Exchange 2000 Server Security

Exchange 2000 Server security starts with Windows 2000 Server's security system. Several different Windows 2000 Server security structure options are available; the one that's right for you depends mostly on the size and structure of your organization and the department that supports Exchange 2000 Server. In all cases, the idea is to select a security model that puts the lightest burden on users and system administrators while still appropriately barring unauthorized users from messaging and other system resources. (More on this in Chapters 7 and 8, "Installing Exchange 2000 Server.")

Windows 2000 Server also audits security. It can let you know when a user tries to add, delete, or access system resources.

The security of Exchange 2000 Server is enhanced in several ways beyond the Windows 2000 Server operating system's security. Access to Exchange Server objects such as public folders can be limited by the owner of the object. Data encryption on the server and client protects messages and other Exchange resources from eavesdropping

by those with server or workstation access. Digital signatures prove the authenticity of a message. Even traffic between servers can be encrypted.

## Exchange 2000 Server and Other Electronic Messaging Systems

The world of electronic messaging is far from a single-standard nirvana. A good electronic messaging system must connect to and communicate with a variety of other messaging systems. Microsoft has done a nice job of providing Exchange 2000 Server with key links, called *connectors,* to other systems, including Exchange 5.5 servers. The company has also built some cross-system message-content translators into Exchange 2000 Server that work automatically and very effectively. With these translators, you're less likely to send a message containing, say, a beautiful embedded image that can't be viewed by some or all of the message's recipients.

In the case of Microsoft's legacy messaging systems—Microsoft Mail for PC Networks and Microsoft Mail for AppleTalk Networks—you have an option beyond connectivity. You can choose to migrate users to Exchange. Migration utilities for other messaging systems such as Lotus cc:Mail are also provided with Exchange.

The following sections describe the most prominent messaging systems in use today.

### Exchange Server 5.5

Exchange 2000 Server wouldn't be much of a connectivity product if it couldn't link with its predecessor, Exchange Server 5.5. It can, as you'll see in Chapter 6.

### X.400

A fully standards-compatible X.400 service is built into Exchange Server. It can be used to access foreign X.400 messaging systems and to link groups of Exchange 2000 Servers. The 1984 and 1988 standards for X.400 are supported, as discussed in detail in Chapter 16, "Connecting to Other Foreign Messaging Systems."

### SMTP

In league with Windows 2000 Server, Exchange 2000 Server supports the Simple Message Transport Protocol (SMTP) service. Unlike the old Microsoft Mail for PC Networks SMTP gateway, this implementation is a full-fledged SMTP host system capable of relaying messages and resolving addresses, while supporting several Enhanced SMPT (ESMTP) commands. UUencode/UUdecode and Multipurpose Internet Mail Extensions (MIME) message-content standards are also supported. So, after you've moved your users from MS Mail for PCs to Exchange 2000 Server, you won't hear any more of those vexing complaints about the meaningless MIME-source attachments

that users get because the SMTP gateway was incapable of converting them back to their original binary format.

### Microsoft Mail for PC Networks

A built-in connector makes Microsoft Mail for PC Networks 3.x (MS Mail 3.x) post offices look like Exchange 2000 servers to Outlook clients, and vice versa. If connectivity isn't enough, you can transfer MS Mail 3.x users to Exchange with a supplied migration tool. If all this is too much, Exchange clients can directly access MS Mail 3.x post offices. Thus, you can keep your MS Mail 3.x post offices, at least until you've got Exchange 2000 Server running the way you want and have moved everyone off the legacy mail system. I'll talk about MS Mail for PC Networks connectivity in Chapter 16.

### Microsoft Mail for AppleTalk Networks

Connectivity for Microsoft Mail for AppleTalk Networks systems is also provided by a connector built into Exchange. When connectivity isn't enough, Mail for AppleTalk users can be migrated to Exchange Server. We'll look at MS Mail for AppleTalk Networks in Chapter 16.

### cc:Mail

If Lotus cc:Mail is running in your shop, you'll be happy to hear that Exchange 2000 Server comes with tools to connect and migrate users to Exchange. Never let it be said that Microsoft doesn't care about users of IBM/Lotus products. At least there's a way to pull them into the MS camp.

### Lotus Notes

Exchange 2000 Server also includes a connector for Lotus Notes. With this connector, Exchange and Notes clients can see each other's address directories and can exchange mail.

### Other Messaging Systems

Gateways are available for links to other messaging systems such as Notes, PROFS, SNADS, fax, and MCI Mail. Both Microsoft and third parties build and support these gateways. You can even extend the benefit of these gateways to your MS Mail users.

## Third-Party Applications for Exchange 2000 Server

Exchange Server has been around for some time now. This time has given third-party application providers time to develop an exciting range of add-on products. These

include sophisticated products that enhance Exchange in such areas as document management, work flow, system backup, system management, faxing, security, virus control, wireless access, and application development.

At various places in this book, I'll spend some time discussing one of these third-party applications. For example, in Chapters 7 and 8, I talk about Exchange 2000 Server-specific backup products; in Chapter 17, I focus on apps for controlling those pesky viruses right inside Exchange Server. Appendix A, "Cool Third-Party Applications for Exchange/Outlook Clients and Exchange Server," contains a fairly exhaustive list of third-party apps for Exchange Server.

## Summary

Microsoft Exchange 2000 Server represents a significant upgrade over previous versions, and contains features that position it as the premier messaging package for this age of electronic messaging. Even experienced Exchange Server 5.5 administrators will need to digest the information in this book to become proficient with Exchange 2000 Server.

For starters, Microsoft Exchange 2000 Server is closely tied to Windows 2000 Server—in fact, it won't run on Windows NT Server. Nonetheless, it is capable of messaging with a variety of legacy systems, as well as systems from other vendors. If all that isn't enough, you can use Exchange Server's APIs to build custom applications. Furthermore, Exchange 2000 Server scales both vertically and horizontally to grow with the needs of your enterprise.

Outlook 2000, Exchange's *native* e-mail client, provides a large variety of features for sending, receiving, and organizing mail messages using mailboxes, public and personal folders, and rules for automatically sorting and classifying e-mail. Furthermore, its e-mail functions are only part of the picture. Scheduling and contact management are big parts of Exchange 2000 Server's features. But perhaps most impressive is its capability of dynamically linking documents that are sent as e-mail attachments. Outlook can also import documents as OLE 2.0 objects into e-mail messages from other applications, while maintaining the formatting and editing features of the original application.

Now that you've gotten a taste of the features and functions of Exchange 2000 Server, it's time to learn about its constant companion, Windows 2000 Server. Read on to learn about this subject in Chapter 2.