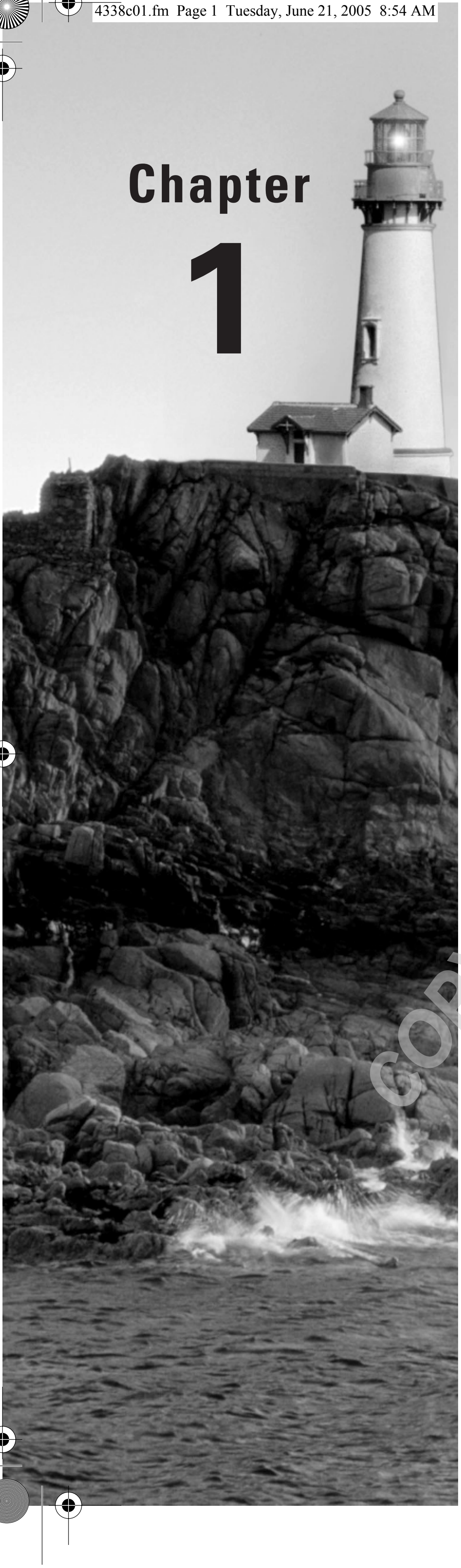


Chapter 1

Introduction to Microsoft Exchange

Microsoft Exchange Server 2003 is the server portion of a rather powerful client/server enterprise messaging system. “Great,” you say, “but what does that mean?” This chapter introduces the capabilities of Exchange Server 2003 and sets the stage for the rest of the book by dissecting the phrase “client/server enterprise messaging system.” First, this chapter introduces messaging systems—what they are and what they are used for on modern networks. Second, it examines several different computing models and shows how Exchange Server fits into the client/server model. Third, it answers the question “what does it take to be an enterprise-level system?” Finally, this chapter discusses the major industry standards on which Microsoft Exchange is based.

Messaging Systems

An enterprise needs information in order to get work done. For this reason, electronic messaging has become a mission-critical function in most organizations. While electronic mail (*e-mail*) is still the core ingredient of any messaging system, other applications are becoming more popular. Messaging can be divided into the following categories:

- E-mail
- Groupware
- Other messaging applications, such as real-time chat applications

Each of these categories, and how Exchange addresses them, is briefly discussed in the following text.

NOTE

Due to the multiple functionality of some of the client programs, a single program could fit into more than one of the categories listed. For example, Microsoft Outlook includes e-mail functions and groupware functions such as group scheduling.

E-mail

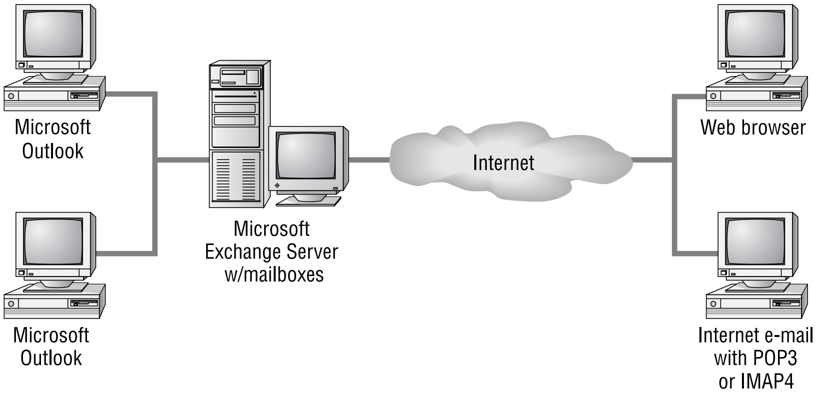
An e-mail program allows a user to create, send, read, store, and manipulate electronic messages and attachments. E-mail is an example of *push-style* communication, meaning that the sender initiates the communication. Because of the importance of e-mail in the overall communication of organizations, e-mail client programs have evolved from merely creating and sending text messages into multifeatured programs.

Microsoft also has server components that enable standard Internet clients to be Exchange e-mail clients. Those Internet clients include the following:

- Web browsers through the Outlook Web Access component
- Internet e-mail programs with the Post Office Protocol, version 3 (POP3)
- Internet e-mail programs with Internet Message Access Protocol, version 4 (IMAP4) support

Figure 1.1 illustrates these e-mail client applications.

FIGURE 1.1 E-mail clients to Exchange Server


Microsoft Outlook
Microsoft Outlook
Microsoft Exchange Server w/mailboxes
Internet
Web browser
Internet e-mail with POP3 or IMAP4

Microsoft Office Outlook 2003

Outlook 2003 is an e-mail client that ships with Exchange Server 2003 and Microsoft Office 2003 as a stand-alone product. Outlook is referred to as a desktop information manager because it is more than just an e-mail client. It also performs such tasks as calendaring, scheduling, and task and contact management. Outlook is intended to be a central program for management of data of all types.

Microsoft Outlook includes a vast feature set, some of which are listed here:

Universal inbox (mailbox) This central storage area can hold not only e-mail messages but other data such as word processing documents, spreadsheet files, faxes, electronic forms, and even voicemail files.

NOTE

Two terms refer to a user's mailbox: mailbox and inbox. The most common usage in this book will be mailbox. This is our primary term for two reasons. One, Microsoft divides a mailbox into folders, one of which is labeled the Inbox. Using the term inbox for only the folder helps prevent confusion. The other reason is that the server-based storage area for a user's messages is also called a mailbox.

Hierarchical data storage Outlook organizes the client's *mailbox* into several default folders: Deleted Items, Drafts, Inbox, Junk E-mail, Outbox, and Sent Items. Users can also create their own folders, thereby personalizing the organization of their data.

Customized views Users have the ability to determine what and how data is presented to them on their screens. Messages can be ordered by sender, date, priority, subject, and other properties.

Search tool Users can search and retrieve messages in their mailboxes using a variety of search criteria, such as sender, date, and subject.

Rich-text message content Historically, most e-mail content was simple text. Outlook enables the creation of rich-text message content that can include multiple fonts, sizes, colors, alignments, and other formatting controls. In addition, Outlook provides for the safe display of HTML mail if desired.

Microsoft Word as message editor Even though Outlook includes a rich message editor, it can also be configured to use Microsoft Word as its message editor. This ability provides access to many of the standard Word features, such as tables, embedded pictures, and linked objects, right inside the e-mail message.

Compound messages and drag-and-drop editing Outlook is *OLE 2 (Object Linking and Embedding)* compliant and, therefore, allows the creation of compound documents. For example, a user could drag and drop a group of cells from a spreadsheet into Outlook.

Secure messages *Digital signatures* and message *encryption* are advanced security features built into Outlook.

Cache mode A new feature in Outlook 2003 allows users to operate in cache mode, whereby they have the benefits of both online mode and offline mode simultaneously. When Outlook first connects to the user's Exchange mailbox during a session, the contents of the mailbox are synchronized to an offline folder file on the local computer. Outlook then connects to the Exchange server periodically to check for updates and synchronizes the local offline folder and sends any outgoing messages. Should the client lose connectivity to the Exchange server, the user can continue to work normally until connectivity has been restored.

Offline mode Because more and more employees spend part of their workday outside the office, special features allow users to create folders on their local computers that synchronize with folders on an Exchange server when the local computer is connected to a network. When the computer is not connected, the folders that have been synchronized are available and users can create, work with, and send messages. Sent messages are placed in the Outbox and are actually delivered the next time the offline folders are synchronized.

Delegated access Some users need to allow other users to access their mailbox. For example, a manager might want a secretary to read meeting request messages in order to handle the manager's schedule. In many mail systems, this would be accomplished by having the secretary log on as the manager. This creates an obvious security problem. Exchange and Outlook solve this problem by allowing the manager to grant the secretary limited permission to access the manager's mailbox. This permission can be restricted to certain folders. The secretary can also be granted permission to send messages on behalf of the manager or even send messages as the manager, using the Send As feature.

Voting Outlook supports the ability to add voting buttons in the header of a mail message and to collect the responses. This allows surveys to be conducted through e-mail.

Auto Create Outlook can automatically convert one Outlook item into another. For example, a mail message may contain an action item that the user can simply drag and drop into the Task folder. Outlook would automatically convert the mail message into a task.

Recover deleted items Users of Outlook can recover deleted items in a mailbox or public folder for a certain amount of time determined by the Exchange administrator.

Block junk e-mail Outlook now provides an enhanced junk e-mail filtering system that can effectively keep junk e-mail out of your Inbox.

Block unwanted attachments Users of Outlook can prevent unwanted, and often bulky, attachments from taking up mailbox space. E-mail messages and files are blocked from people who have not been placed on a user's Safe Senders List, allowing the user to examine the message and determine whether they wish to receive the message and any attached files.

These are just some of the e-mail features of Outlook. This client program and others are discussed further in Chapter 7, "Configuring Client Access."

Web Browsers

Exchange Server 2003 includes a component named *Outlook Web Access (OWA)* that runs in conjunction with Microsoft *Internet Information Server (IIS)*. OWA enables web browsers to access Exchange resources such as mailboxes and public folders. Any standard web browser can be used, such as Microsoft Internet Explorer or Netscape Navigator, though only Internet Explorer 5 or later supports some of the advanced features that OWA provides. This Exchange functionality permits users of other operating system platforms, such as Unix or IBM's OS/2, to also be Exchange clients. Chapter 7 covers the Exchange components required for web browser clients.

NOTE

One of the new improvements in Outlook Web Access in Exchange Server 2003 is the ability to choose between the basic experience or the premium experience, which is supported only by Internet Explorer version 5 and later.

Internet E-mail Programs with POP3

Exchange has built-in support for the *Post Office Protocol, version 3 (POP3)*. POP enables mail clients to retrieve mail messages stored on a remote mail server. Exchange's support for this protocol allows Internet e-mail programs that support POP3, such as Outlook Express, to access their Exchange mailbox and download their messages. Chapter 7 covers POP3 in the Exchange environment.

Exchange also has built-in support for the *Internet Message Access Protocol, version 4* (*IMAP4*). IMAP is similar to POP in that it is a mail retrieval protocol. But IMAP has more features than POP, such as the ability to select the messages to download rather than having to download all new messages. Outlook Express is also an IMAP client that can be used with Exchange Server 2003. Chapter 7 provides further details on IMAP4.

Groupware

A simple definition of *groupware* is any application (the *ware* in *groupware*) that allows groups to store and share information. That is a very broad definition and one that includes applications such as e-mail and electronic forms. And indeed, as you will see, these applications are important ingredients in groupware. However, the emphasis in groupware is on collaboration—not merely sending an item, but enabling many people to cooperatively use that item. Outlook incorporates many groupware functions, such as the ability to share a calendar, schedule, task list, and contact list.

Another example of groupware is folder-based applications. These applications utilize public folders. A *public folder* is a special storage area for group access. Various types of information can be contained in a public folder, such as documents, spreadsheets, graphics, e-mail messages, forms, and many other types of information. Along with storing information, a public folder can be assigned security, so that only selected users or groups can access the public folder. Other features such as views and rules can also be assigned to a public folder. Using a simple folder-based application, a Sales department could place all of their sales letters in a specified public folder for the department. Only the employees in the Sales department would be given permission to access this public folder.

Folder-based applications can also utilize electronic forms. A specific electronic form or set of forms can be associated with a public folder. Users can fill out and post the form to the public folder. Other users can then access the public folder and view the posted information. An example of this type of application is a discussion-and-response application. A product manager could create a public folder for discussion about a product under development. That manager could also create customized electronic forms that people could use to enter their comments and that they could then send, through e-mail, to the public folder. The product manager and product developers could then access the public folder to read the comments. It is even possible to set up customized views of the content in the public folder in order to view only data on a specific topic. A marketing person, for example, might want to see only comments related to the possible market for the product.

Folder-based applications are examples of *pull-style* communication, because users go to the information and decide what is relevant to them. See Chapter 6, “Using Public Folders,” for more on the topic of public folders. Figure 1.2 illustrates folder-based applications.

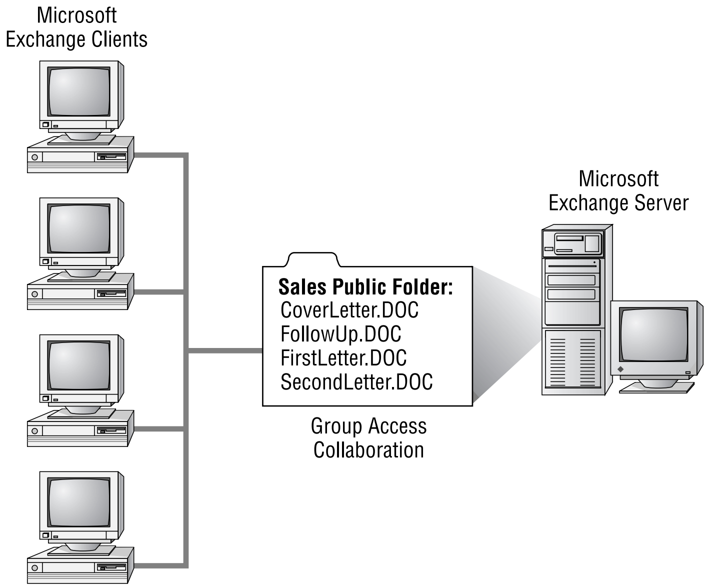
Other Messaging Applications

Along with e-mail, electronic forms, and groupware, there are many other types of messaging applications. Exchange provides an open platform that encourages the integration of other types of applications, including the following:

Workflow While Exchange includes some basic workflow capabilities, some third-party workflow solutions that work with Exchange are also available. An example of a workflow application would be Streem Center from Streem Communications.

Fax Fax software can be integrated with the Exchange clients so that e-mail and faxes can be sent and received from the same location, as well as share the same address book. An example of a fax application would be FAXMaker from GFI Software.

FIGURE 1.2 Folder-based applications



Voicemail There are various voicemail products that integrate with Exchange and store their messages in an Exchange mailbox. An example of a voicemail application would be Avaya Unified Messenger from Avaya.

Custom applications Through its installable file system, public folders, wide support of Internet protocols, and support of scripting, Exchange Server 2003 also serves as a wonderful platform for the creation of custom business applications.

Computing Models

Microsoft Exchange uses a client/server computing model to implement its messaging system. To better understand the client/server model, two other models are discussed briefly to provide a context for the client/server model. In all, the three models discussed in this section are:

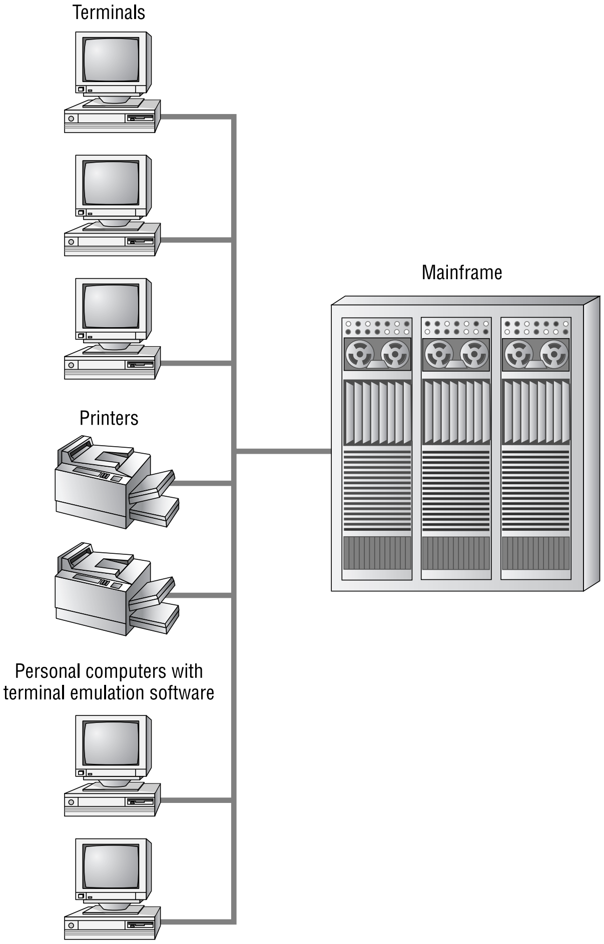
- Mainframe computing
- Shared-file computing
- Client/server computing

Mainframe Computing

Mainframe computing consists of a powerful host computer, such as a mainframe computer or minicomputer, and numerous input-output devices attached to the host, such as terminals, printers, and personal computers running terminal emulation software. The advantages of this architecture are its powerful, centralized processing, administration, and backup. These features permit a large number of users on systems built according to this model. The disadvantages are that these features incur high costs, that personal computing power and applications are not leveraged, and that most of these systems have a proprietary architecture. Examples of messaging systems that use this type of model are IBM PROFS (Professional Office System) and OfficeVision.

Figure 1.3 illustrates the host-based computing model.

FIGURE 1.3 Host-based computing model



Shared-file Computing

This network computing model works in a local area network (LAN) context. At least one computer is used as a server computer to store files. Users, working on their own networked personal computers, access and share the files on the server computer. Microsoft Mail is a messaging system that uses this type of architecture.

Using this model, a *shared-file messaging system* has active clients and passive servers. Each mail user is assigned a mailbox. A mailbox is actually a directory on the server where mail messages are placed. The server software is passive in that its main task is to store mail messages. The client software is said to be active because it performs almost all mail activities. Along with the normal mail activities of creating and reading mail, the client software is also responsible for depositing mail in the correct recipient mailboxes and checking its own mailbox for new mail (this is referred to as *polling*).

This model could be compared to a postal system where people must take their outgoing mail to the post office and place it in the respective recipients' mail slots and also visit the post office to check their mail slots for any new mail. The primary duty of the post office is to store the mail. This is analogous to the shared-file messaging system in that the people (clients) are active and the post office (server) is passive.

The advantages of shared-file messaging systems include the following:

Minimal server requirements Because the server has a passive role, it does not need to run on a high-end hardware platform.

Minimal server configuration in a single-server environment Because the server is mainly a storage location, it does not need a lot of configuration.

The disadvantages of shared-file messaging systems include the following:

Limited security Because the client software is responsible for sending mail to a recipient's mailbox, each client must have write permissions on each mail directory. Each client must also have read permissions on the entire mail directory structure in order to read forwarded or copied messages. From a security standpoint, this is considered an excessive level of permissions.

Increased network traffic The periodic client polling of mailboxes for new mail increases network traffic.

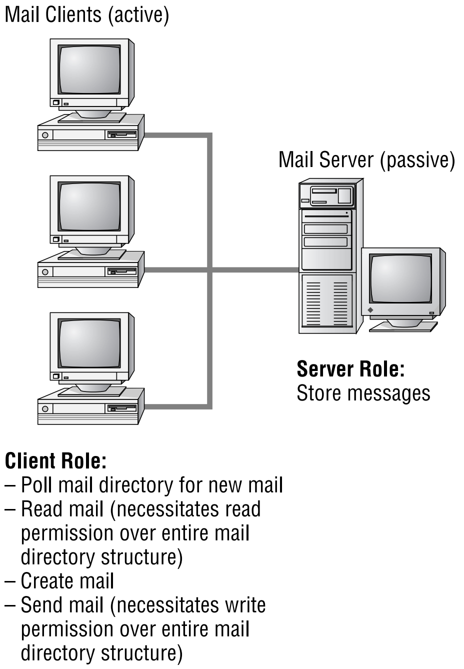
Increased client load The active clients do almost all of the processing work.

Limited scalability These systems cannot accommodate large numbers of users because of the shared-file model. Users must access common files that can be opened by only one process at a time.

Single point of failure A single mail server in this type of arrangement becomes a single point of failure for clients seeking to access its services.

Figure 1.4 illustrates a shared-file messaging system.

FIGURE 1.4 Shared-file messaging system


Mail Clients (active)
Mail Server (passive)
Server Role:
Store messages
Client Role:
– Poll mail directory for new mail
– Read mail (necessitates read permission over entire mail directory structure)
– Create mail
– Send mail (necessitates write permission over entire mail directory structure)

Client/Server Computing

In *client/server messaging*, a task is divided between the client processes and server processes. Each side works to accomplish specific parts of the task. The two processes are usually running on separate computers and are communicating over a network. The communication is in the form of requests and replies passed back and forth through messages.

The client side includes a user's personal computer or workstation and client software. The client software provides the interface for the user to employ when manipulating data and making requests to and receiving replies from the server. The processing power to carry out those tasks is provided by the client's computer.

The server side includes the server computer and server software. The server software receives and processes client requests, provides storage capabilities, implements security, provides for administrative functions, and performs many more duties. The server's processor, or processors, powers these functions.

When this model is applied to a mail system, both the client side and the server side are active participants. Mail activities are divided between the two sides in a way that takes advantage of both parties. The client software enables users to initiate mail activities such as creating, sending, reading, storing, and forwarding mail and attachments.

The server software also has an active role. Some of its tasks are implementing security, placing messages in mailboxes (as opposed to the client software doing it), notifying clients of new mail (which eliminates the need for clients to poll their mailboxes), and performing specified actions on mail, such as applying rules, rerouting messages, and many other tasks. Many of the mail activities that are initiated by the client software are actually implemented on the server. For example, when a client initiates the reading of a message, the client software sends a read request to the server where the message physically resides. The server software receives this request, processes it (for example, checks security to see if this user is permitted to read this message), and then sends the message to the client. The user can then use the client software and processor to manipulate the message (edit the message, for example). This illustrates how both sides are active.

In this model, the software running on the client machine is frequently referred to as the front-end program, while the software running on the server is referred to as the back-end program.

NOTE

Exchange Server 2003 now supports front-end and back-end servers, a designation that allows an Exchange administrator to balance the various loads placed on Exchange servers among multiple computers. This concept is discussed in detail in Chapter 2, "Microsoft Exchange Architecture," and Chapter 7. Do not confuse front- and back-end programs with front- and back-end servers.

The advantages of the client/server model include the following:

Distributed computer processing The computer processing power of both the client and server machines is utilized. The client processor handles the end-user mail activities, such as creating, reading, and manipulating mail, while the server processor (or processors) handles the security, routing, and special handling of mail. This spreads the processing load over a multitude of client processors, while still utilizing the powerful processing of the server machine.

Tight security The server software is responsible for the security of the mail system. The server software is the entity that actually places messages in mailboxes. The clients therefore do not need permissions to all mailboxes. This creates a much more secure mail system.

Reduced network traffic Because the server software informs clients of new mail, the client software does not have to poll the server, thus reducing network traffic.

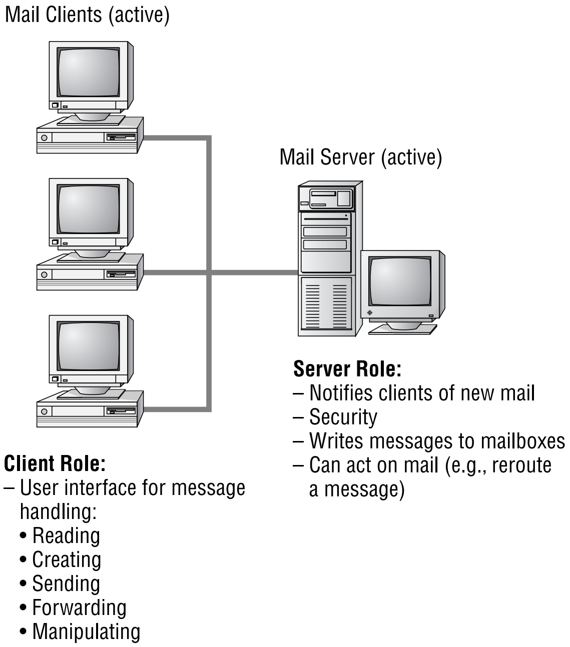
Scalable The term *scalable* relates to the ability to grow easily. A client/server mail system can scale to any size organization.

The primary disadvantage of the client/server model is the following:

Increased server hardware requirements Because the server has an active role in the messaging environment, there are greater requirements for the server hardware platform. This should not be seen as much of a disadvantage in light of scalability, central administration, backup, and other advantages.

Figure 1.5 illustrates the client/server mail system.

FIGURE 1.5 Client/server mail system


Mail Clients (active)
Mail Server (active)
Server Role:
– Notifies clients of new mail
– Security
– Writes messages to mailboxes
– Can act on mail (e.g., reroute a message)
Client Role:
– User interface for message handling:
• Reading
• Creating
• Sending
• Forwarding
• Manipulating

Exchange is a client/server messaging system. The Exchange Server 2003 software runs as a series of services on a Windows Server 2003 or Windows 2000 Server computer. It provides server-side messaging functions for the client applications.

So far, you have learned what features make up Exchange Server 2003 and how the system is implemented, namely the client/server model. Now we need to turn our attention to the context or scale in which Exchange can be implemented.

Enterprise-Quality Features

Microsoft Exchange Server 2003 actually comes in two editions. The first is the Standard Edition (or, simply, Microsoft Exchange Server 2003), which is targeted for use by small- to medium-sized businesses. The other edition is Microsoft Server 2003 Enterprise Edition. It is designed to be an enterprise messaging system, meaning one that is more scalable and includes features meant for larger organizations. Table 1.1 details the key specifications of both Microsoft Exchange Server 2003 Standard Edition and Microsoft Exchange Server 2003 Enterprise Edition.

TABLE 1.1 Exchange Server 2003 Version Comparison

Feature	Standard Edition	Enterprise Edition
Maximum database size	Limited to 16 GB	Limited only by hardware
Number of regular storage groups supported	One	Four
Supports the recovery storage group (above the regular storage groups)	Yes	Yes
Number of databases per storage group	Two	Five
Supports clustering	No	Yes
Supports front-end and back-end server configuration	Yes	Yes
Supports Volume Shadow Copy	Yes	Yes
X.400 connector	Not included	Included

For Exchange to be an enterprise messaging system, a large number of technologies had to be included or leveraged from other products (such as Microsoft Windows Server 2003). This section briefly discusses the technologies that make Exchange a true enterprise messaging system. Those technologies fall into six categories:

- Enterprise-quality application platform
- Scalability
- Interoperability
- Performance
- Administration
- Reliability

Enterprise-Quality Application Platform

Before a determination can be made as to whether or not a product can scale to the size an organization needs, it must be determined that the product can do the things it needs to do. Exchange provides the necessary application platform to meet the requirements of almost any organization. The following are some of the elements of the Exchange application platform:

Supports a large number of messaging services E-mail, electronic forms, groupware, and add-on products for faxing, paging, videoconferencing, voicemailing, and many other services are supported.

Supports a large number of client platforms There is client software that runs on MS-DOS, Windows 3.*x,* Windows 95, Windows 98, Windows NT, Windows 2000, Windows XP, Windows Server 2003, Apple Macintosh, Unix, and IBM OS/2, as well as virtually every other operating system available today.

Provides open architecture/extensibility Exchange is based on an open architecture, meaning that the specifications of many of its protocols are available in the public domain. Examples of published protocols include the *Messaging Application Programming Interface* (*MAPI*), Internet protocols, and various Comite Consultatif International Telegraphique et Telephonique (CCITT) protocols. Developers can use this openness to create additional applications and programs that work with or extend Exchange. That is what is meant by extensible. One example of the way Microsoft encourages this is by including a single-user version of the Microsoft Visual InterDev product with Exchange Server. Developers can use Visual InterDev to create web-based applications that enable web clients to access Exchange resources.

Based on industry standards The Exchange protocols, along with being open and extensible, are based on industry standards (protocols can be open and extensible but not based on industry standards). The MAPI protocol is considered an industry standard. Some of the industry standard Internet and CCITT protocols used in Exchange are as follows:

> **Internet mail** *Simple Mail Transfer Protocol* (*SMTP*), Post Office Protocol, version 3 (POP3), and Internet Message Access Protocol, version 4 (IMAP4). See Chapter 7.
>
> **Internet directory access** *Lightweight Directory Access Protocol* (*LDAP*). See Chapter 7.
>
> **Internet news services** *Network News Transfer Protocol* (*NNTP*). See Chapter 7.
>
> **Internet management** *Simple Network Management Protocol* (*SNMP*). See Chapter 10, “Administration and Maintenance.”
>
> **Internet security** *Secure MIME (S/MIME)*, Secure Sockets Layer, version 3 (SSL), and Simple Authentication and Security Layer (SASL).
>
> **Internet web protocols** *HyperText Transfer Protocol* (*HTTP*) and HyperText Markup Language (HTML). See Chapter 7.
>
> **CCITT message transfer** Comit Consultatif International Telegraphique et Telephonique (International Telegraph and Telephone Consultative Committee): X.400. See the section “Industry Standards” later in this chapter.
>
> **CCITT directory** X.500. See the section “Industry Standards” later in this chapter.

Security features Using the Internet security protocols listed above, along with other protocols, Exchange can provide advanced security features. For example, messages can be sent with a digital signature to confirm the identity of the sender, and message content can be encrypted to prevent unauthorized viewing. Exchange Server 2003 supports many of the most popular encryption algorithms today, including 3DES and SHA. Chapter 15, "Securing Exchange Server 2003," discusses the protocols and administration of advanced security in Exchange. Further security features, and ones that are leveraged from Microsoft Windows Server 2003, include:

Mandatory logon A user must have a domain account and password to log on to a Windows Server 2003 or a Windows 2000 Server domain.

Discretionary access control An Exchange administrator can use Windows Server 2003 or Windows 2000 Server security to control access to Exchange resources. For example, one administrator could have permission to manage particular Exchange servers or features but not others.

Auditing Windows Server 2003 or Windows 2000 Server can be configured to monitor and record certain events. This can help diagnose security events. The audit information is written to the Windows Event Log.

Scalability

Once a product has been determined to accomplish the types of things you need to get done, then you must find out if it can do them on the scale you need. Exchange is extremely scalable due to the following features:

Software scalable Exchange can be implemented with a single Exchange server, or dozens of servers, depending on the messaging requirements. Even with multiple Exchange servers, a single enterprise messaging system exists. This is due to the Exchange features that enable communication between servers. This functionality permits Exchange to scale from single-server to multiple-server implementations. Microsoft itself uses Exchange for its worldwide messaging system.

Hardware scalable Scalability is also evidenced by the maximum hardware specifications that Exchange can utilize.

CPUs Scalable from 1 to 64 processors, depending on the operating system in use.

RAM Maximum addressable by Exchange is 4 GB.

Disk storage Storage is limited only by hardware capacity. The Standard Edition of Microsoft Exchange Server 2003 has a 16-GB storage limit on each of the Exchange databases. The Enterprise edition has no limit on the databases.

Interoperability

For a product to fit into an enterprise, it might need to work with an existing messaging system. This is called interoperability or *coexistence*. An organization might need to move all of its existing messaging data to a new messaging product. This is called a *migration*. Exchange addresses both of these issues.

To interoperate with various non-Exchange systems, referred to as foreign systems, Microsoft had to write special software programs called *connectors*. Connectors are similar to translators that understand both Exchange and the foreign system and translate between them. Third-party companies have also written similar programs. Microsoft refers to these programs as gateways. Messaging systems that Exchange can interoperate with include the following:

- Internet mail
- X.400 mail systems
- Lotus Notes
- Novell GroupWise
- Digital Equipment Corporation (DEC) All-IN-1
- Verimation MEMO

Exchange Server 2003 ships with connectors for Internet mail, Lotus Notes, and Novell GroupWise systems. Connectivity to other systems, such as DEC ALL-IN-1 or Verimation MEMO, is provided through third-party gateway products.

NOTE

For Exchange to interoperate with some of the previous systems, third-party software is required. Chapter 13, “Connecting with Other Messaging Systems,” discusses interoperability in more detail.

Exchange Server 2003 can perform a migration from the following messaging systems:

- Microsoft Mail
- Microsoft Exchange
- Lotus cc:Mail
- Lotus Notes
- GroupWise 4.*x*
- GroupWise 5.*x*
- Internet Directory LDAP via ADSI
- Internet IMAP4 mail

NOTE

If you have other messaging systems in place, such as Verimation MEMO, you will need to migrate them to Exchange 2000 Server first and then upgrade them to Exchange Server 2003. Alternatively, you may find a third-party conversion utility that can be used for this task.

Performance

A messaging system requires adequate performance to be used on an enterprise scale. Exchange meets that requirement by being a 32-bit, multithreaded program running on a high-performance operating system, Microsoft Windows Server 2003 or Windows 2000 Server. Many features are built into the Exchange System Manager to help optimize server performance.

Administration

An important element of any enterprise application is the ability to effectively and efficiently administer it. Exchange meets this need by including powerful administration programs, one of which is the *Exchange System Manager* snap-in for the Microsoft Management Console (MMC). This program provides a single point of administration for an entire Exchange organization. Exchange servers anywhere in the enterprise can be managed from this program, as well as such activities as configuring a server, managing connections to foreign systems, and monitoring services centrally.

Along with its own administrative utilities, Exchange can leverage the administrative capabilities of the Windows Server 2003 or Windows 2000 Server operating system. Exchange integrates with Windows utilities such as Performance Monitor and Event Viewer. Another powerful administration feature in Exchange Server 2003 involves Active Directory. Exchange-related user features (such as mailbox properties) are now managed using the Active Directory Computers and Users utility—the same tool used by Windows administrators to manage users and groups.

Exchange Server 2003 also supports the Simple Network Management Protocol (SNMP). This enables third-party SNMP monitor programs to collect various management information about an Exchange server, such as the performance information gathered by Performance Monitor. The topic of Exchange Server administration is covered in Chapter 10.

Reliability

Because of the importance of a messaging system to an enterprise, it must be reliable. Exchange provides reliability through the following ways:

Transaction log files Data that is to be written to an Exchange database is first written to these log files (which can be done very fast). The data is later written to the appropriate database, which takes longer because of the structured nature of a database. If, for whatever reason, a server has an unintended shutdown, data that has not been written to the database is not lost; it can be automatically reconstructed from the transaction log files. Chapter 2 and Chapter 14, “Backup and Recovery,” discuss this topic further.

Windows Backup utility When Exchange is installed, it adds extensions to the Windows Backup utility, allowing that program to back up Exchange information.

Replicas Exchange can be configured to have multiple copies, called replicas, of a single public folder on different servers. This prevents a single point of failure in terms of data access and provides quicker access by putting folders on servers closer to the users in an organization.

Intelligent message routing This feature allows multiple routes to a destination, thereby preventing a single point of failure for message delivery.

Windows Server fault tolerance Exchange takes advantage of the many fault-tolerant features of the Windows Server 2003 and Windows 2000 Server operating systems, such as disk mirroring and disk striping with parity. Exchange Server 2003 Enterprise Edition also supports Active/Passive and Active/Active clustering, which provides fault tolerance in the event of a server malfunction. If one server fails, another server can take its place, thereby providing uninterrupted service to users.

Industry Standards

Microsoft Exchange is based on industry standard technologies, ensuring an open architecture and, therefore, extensibility (i.e., the ability to easily add on to the product). An adequate understanding of the standards used in Exchange will help in utilizing it. This section presents a brief explanation of the following standards:

- Messaging Application Programming Interface (MAPI)
- The Remote Procedure Call (RPC) protocol
- X.400
- X.500

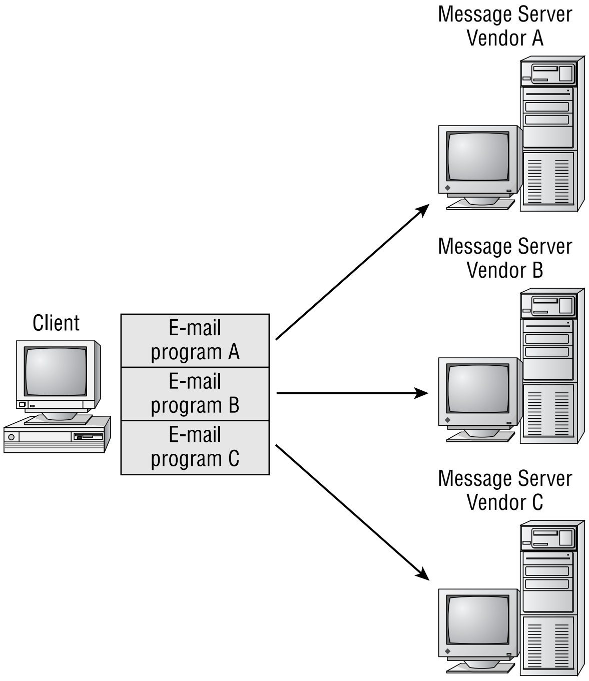
NOTE The Internet standards are also very important in Exchange, and they will be discussed in Chapter 2 and Chapter 7.

Messaging Application Programming Interface (MAPI)

To understand MAPI, you must first understand what an application programming interface is. At the code level, a program's functions are invoked through specific instructions. The collection of those instructions is referred to as an *application programming interface* (*API*). That phrase is appropriate because the API allows a programmer to interface with the functions of a program. For example, if a program has the ability to read a message, there is a specific API instruction, also called a function call, that can invoke that ability. If two programs need to interact, they must do so with an API they both understand. For example, if program A sends the instruction `Read_Message 4` to program B, but program B understands only the instruction `Message_4_Read`, then the instruction will not be understood. Humans can use slightly different grammar and still understand one another, but computers are not that forgiving.

In the past, many client/server messaging products had their own APIs for the client/server interaction. If someone wrote a client program, it would work only with the messaging system whose API it used. If a user needed to connect to multiple messaging systems, multiple client programs were needed. See Figure 1.6.

FIGURE 1.6 Multiple messaging APIs require multiple programs.

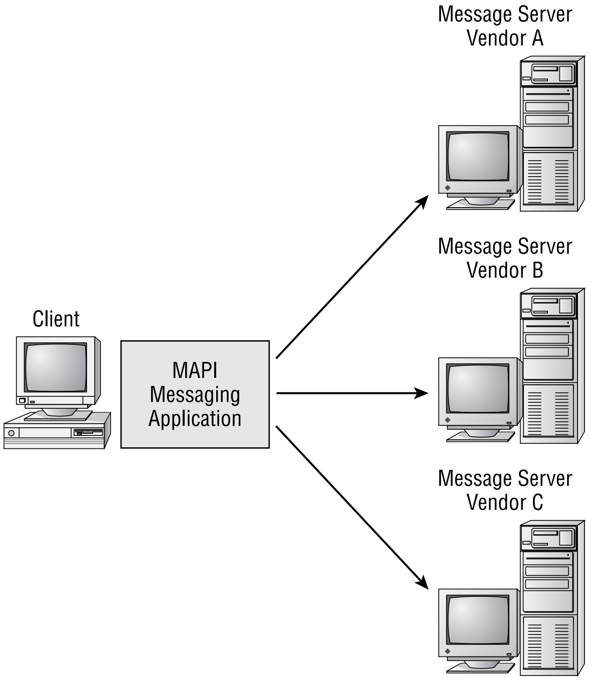


Microsoft decided to remedy that situation by creating a standard messaging architecture, referred to as the Messaging API (MAPI). MAPI accomplishes two broad goals. One, it provides a standard API for client/server messaging interaction. This role makes MAPI a type of middleware, meaning that it stands in the middle between clients and servers. Some authors refer to middleware as the slash (/) between the words *client* and *server*. MAPI makes it possible for a single-client application to access different messaging servers. See Figure 1.7 for an illustration.

The second broad goal of MAPI is to provide a standard set of services to client messaging applications. These services include address books, message storage, and transport mechanisms. Even when using different types of MAPI applications, such as e-mail, fax, and voicemail, a user can access a single address book (a universal address book) and store different data types in the same folder (a universal inbox). The transport mechanisms relate to a single client application that can connect to different messaging systems. A single MAPI e-mail application can access an Exchange server, a Microsoft Mail post office, an Internet mail server, and others.

Although MAPI includes individual API instructions, it most often communicates those instructions in an object-oriented manner. An object is a container; in this context, it functions as a container of API instructions. The Microsoft specification for object-oriented programming is called the Component Object Model (COM). MAPI, OLE, ActiveX, and other technologies are part of the COM standard.

FIGURE 1.7 Accessing different messaging servers through MAPI



The original version of MAPI (called Simple MAPI) was developed by Microsoft. But in the subsequent version (MAPI 1), Microsoft worked with over 100 different vendors to develop an industry standard. Microsoft has also turned over the vast majority of the MAPI specification to standards organizations, while still taking a leadership role by including the core MAPI component with its Windows operating systems.

While MAPI deals with instructions, the next section discusses the protocols that enable those instructions to be passed between clients and servers.

Procedure Calls

You now know the instruction standard used by the Exchange client/server messaging applications, namely MAPI. But client/server applications are divided across physical machines. When a client issues a read instruction for a message, that message could be on the server. The server could understand that instruction and could send the message, but the instruction has to get to the server and the message has to get back to the client. MAPI does not handle those procedures. From MAPI's perspective, the physical distinction of the client and the server does not exist; it is transparent. Microsoft uses the Remote Procedure Call (RPC) protocol to pass instructions and data between machines. Before discussing the RPC protocol, we will first define what a procedure call is and then discuss the two types of procedure calls, local and remote.

NOTE In Exchange Server 5.5 and earlier, servers in the same Exchange site relied on RPCs to transfer messages and directory information between them. Exchange Server 2003 and Exchange 2000 Server use SMTP to exchange this information between servers. You'll learn more about this in Chapter 2.

Procedure calls handle the transfer of instructions and data between a program and a processor or processors. When a program issues an instruction, that instruction is passed to the processor for execution, and the results of the execution are passed back to the program. Now, let's look at the two main types of procedure calls.

Local Procedure Calls

When a program issues an instruction that is executed on the same computer as the program executing the instruction, the procedure is referred to as a *local procedure call*. When Exchange Server components perform activities on that server, they issue instructions that are executed by that server's CPU or CPUs. That is an example of a local procedure call. Exchange uses a Microsoft protocol called the Local Procedure Call (LPC) to implement this mechanism.

Remote Procedure Calls

A *remote procedure call* is similar to a local procedure call in that it relates to the transfer of instructions and data between a program and processor. But unlike a local procedure call, a remote procedure call enables an instruction issued on one computer to be sent over the network to another computer for execution, with the results being sent back to the first computer. The computer making the instruction and the computer performing the execution are remote from each other. The transfer of instructions and data between the computers is totally transparent to the original program and to the user. To the program issuing the instruction, all of its instructions appear to be locally executed. Remote procedure calls are a key ingredient in distributed processing and client/server computing.

The RPC mechanism permits the optimization of performance by assigning different computers to do specific tasks. For example, some programs require lots of processor power, memory, or storage or all three. It would be impractical to give every computer running these applications the necessary levels of resources. But one specialized computer could be given, for example, four processors, 1 GB of RAM, and 200 GB of storage. Clients could use those resources through the RPC mechanism.

Because the request/reply aspect of RPC is intended to be transparent to the client program and user, the speed of network communication is a factor. The computers involved in an RPC session need to have a high-speed permanent link between them, such as a local area network (LAN) or a high-speed wide area network (WAN).

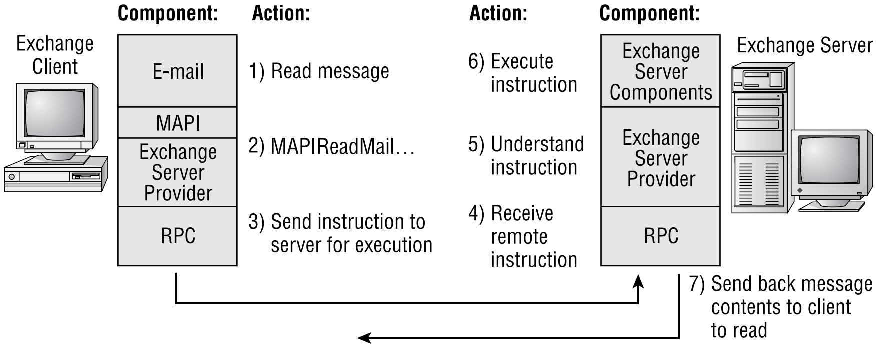
Exchange uses remote procedure calls in many of its communications. The protocol that Exchange uses to implement remote procedure calls is called the Remote Procedure Call (RPC) protocol. This protocol is discussed in the following section.

Remote Procedure Call (RPC) Protocol

The Remote Procedure Call protocol is based on a protocol created by the standards group Open Software Foundation (OSF) and is part of the OSF's Distributed Computing Environment (DCE) protocol suite. Microsoft includes the RPC protocol with their Windows Server 2003 and Windows 2000 Server operating systems. In older versions of Exchange (Exchange 5.5 and earlier), servers within an Exchange site transferred messages between themselves using RPCs. In Exchange Server 2003 (and Exchange 2000 Server), this functionality has been largely taken over by SMTP. RPCs are still used to communicate with Exchange 5.5 servers, but SMTP is now used for all communications between newer Exchange servers inside and outside the boundaries of a routing group.

When a user chooses to read a message, the client program issues a MAPI instruction (`MAPIReadMail`). The RPC protocol on the client transfers this instruction to the Exchange server where the message physically resides. This is called a request. The RPC protocol on the server receives this request, has it executed, and sends the message back to the client's screen. This is called a reply. RPC clients make requests, and RPC servers make replies. RPC is sometimes referred to as a request/reply protocol. RPCs are also used in some Exchange server-to-server communications. Figure 1.8 illustrates the RPC mechanism.

FIGURE 1.8 The Remote Procedure Call protocol



Note that being an RPC client or server doesn't really have anything to do with being a messaging client or server. In the example in Figure 1.8, the messaging client is also the RPC client, but an RPC client is really just the computer that issued the RPC. Exchange servers often communicate information between themselves using RPCs. The computer that initiates the connection request is the RPC client, and the computer that receives the request is the RPC server.

CCITT X.400

For most of the history of electronic messaging in the private sector, there were no widely accepted messaging standards. Different messaging products used vastly different messaging protocols. This made interoperability between different systems difficult and costly, sometimes impossible. To address this situation, different standards organizations began to develop what they hoped would become internationally recognized messaging standards. One of those standards organizations was the Comite Consultatif International Telegraphique et Telephonique (CCITT). This is translated in English as the International Telegraph and Telephone Consultative Committee. One of the standards they developed was the *X.400* Message Handling System (MHS) standard. Exchange uses some of the technologies of the X.400 standard.

The CCITT is now a subdelegation of the International Telegraph Union (ITU), which is an agency of the United Nations. The State Department is the voting member from the United States.

The different versions of the X.400 standard are referred to by the year they were officially published and by a specified color. Versions to date are as follows:

- 1984 “Red Book”
- 1988 “Blue Book”
- 1992 “White Book”

The Message Handling System (MHS) discussed in this section is not the same standard as the Novell-related Message Handling System (MHS).

X.400 is a set of standards that relates to the exchange of electronic messages (messages can be e-mail, fax, voicemail, telex, etc.). The goal of X.400 is to enable the creation of a global electronic messaging network. Just as you can make a telephone call from almost anywhere in the world to almost anywhere in the world, X.400 hopes to make that a reality for electronic messaging. X.400 defines only application-level protocols and relies on other standards for the physical transportation of data (e.g., X.25 and others).

X.400 Addressing: Originator/Recipient Address

Try to imagine what the American telephone system would be like if different parts of the country used different numbering schemes: different number lengths, different placement of the area code, etc. Obviously that would lead to a lot of complexity and problems; hence a standard numbering scheme exists. Electronic messaging also needs a standard addressing scheme to avoid the same sort of chaos.

You might think that you could simply list people’s names in alphabetical order. But there are many problems with that scheme. The addressing scheme needs to be able to potentially scale to the entire world’s population. An alphabetical list would be quite long. There is also the problem of what constitutes a last name; different countries have different methods (e.g., Anwar el-Sadat, Willem de Kooning). A truly global addressing scheme needs to be totally unambiguous.

The addressing scheme that X.400 uses is called the Originator/Recipient Address (O/R Address). It is similar to a postal address in that it uses a hierarchical format. While a postal address hierarchy is country, zip code, state, city, street, and recipient's name, the O/R Address hierarchy consists of countries, communication providers (such as AT&T), companies or organizations, and other categories. Figure 1.9 and Table 1.2 present some of these categories, called fields.

The O/R Address specifies an unambiguous path to where the recipient is located in the X.400 network (it does not specify a path the message might take, only the path to where the recipient is located).

NOTE

In actual practice, this addressing scheme is not as standardized as Table 1.2 makes it seem, nor is it used in the standardized way. Although the address fields have always been specified, the order in which to write them was not specified until 1993. Consequently, you will see them written in different ways. Some X.400 implementations have modified the standard.

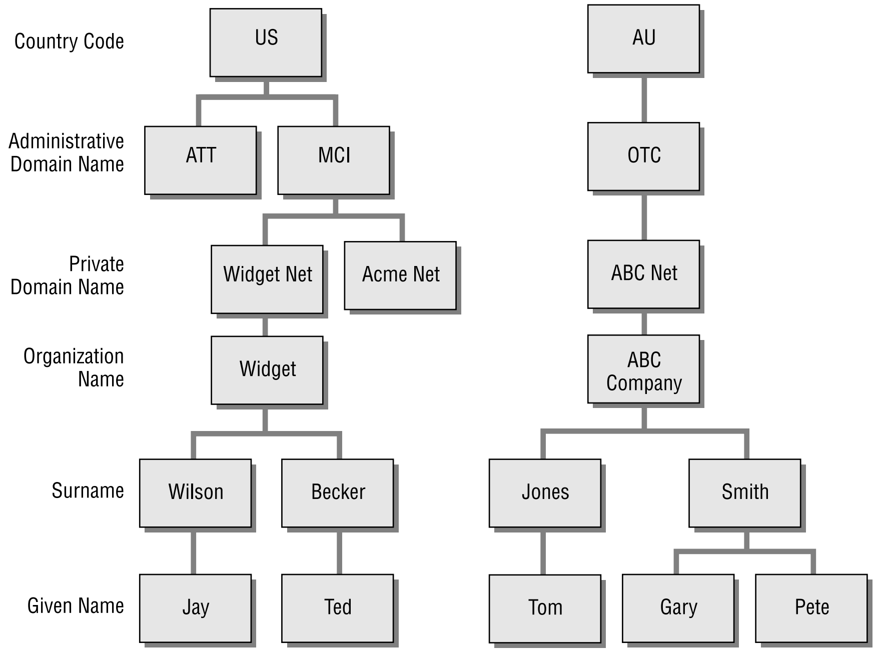
X.400 Message Format: Interpersonal Messaging (IPM)

X.400 also specifies the protocols for formatting messages. The most common one is called Interpersonal Messaging (IPM) and is used for e-mail messages. There are other protocols for other types of messaging, such as Electronic Data Interchange (EDI).

TABLE 1.2 X.400 Originator/Recipient Address Example

Field	Abbreviation/Example	Description
Country code	c=US	Country
Administrative Management Domain (ADMD)	a=MCI	The third-party networking system used (e.g., AT&T, MCI, Sprint, etc.)
Private Management Domain (PRMD)	p=WidgetNet	Subscriber to the ADMD (company name)
Organization	o=Widget	Name of company or organization
Surname	s=Wilson	Last name
Given name	g=Jay	First name

FIGURE 1.9 X.400 Originator/Recipient Address example



X.400 Message Routing: Message Transfer Agent (MTA)

Another very important X.400 protocol is Message Transfer Agent (MTA). MTA is the protocol that runs in the message routing machines (i.e., routers). MTA is like a local post office in that it receives and routes messages to their ultimate destinations. And just like a postal system (a snail-mail system), electronic messages can go through several MTAs before they arrive at their ultimate destinations. This type of delivery method is called store and forward. An MTA machine receives a message, stores it so it can calculate its next route, and then forwards it to either another MTA machine or its ultimate destination. This method eliminates the need for the sender's application and the recipient's application to perform any simultaneous actions in order to exchange data. A sender's message is simply packaged with all the necessary addressing information and is sent to the next store-and-forward MTA machine (i.e., router). That MTA can route it to the next MTA, and so on, until it reaches its final destination.

Other X.400 Information

While the X.400 standard does not define the protocols for the physical transportation of messages, it does specify what other standards it can use. They include the following OSI (Open Systems Interconnection) protocols:

- TP0/X.25

- TP4 (CLNP)
- TP0/RPC 1006 to TCP/IP

NOTE TP stands for Transport Protocol.

Third-party X.400 networks that can be subscribed to include AT&T Mail, AT&T EasyLink, MCI Mail, Sprintmail, Atlas 400 (France), Envoy 100 (Canada), Telebox 400 (Germany), and Telecom Australia. Microsoft Exchange is an X.400 messaging product.

NOTE Exchange Server 2003 does not support the TP4 protocol, unlike previous versions of Exchange.

CCITT X.500

The CCITT X.500 standard defines the protocols for a global directory service. A directory service is a database of information on resources. Resources can be user accounts, user groups, mailboxes, printers, fax machines, and many other items. These resources are officially referred to as objects. The information about an object, such as a mailbox, can include the owner of the mailbox and the owner's title, phone number, fax number, as well as many other types of information. The information about an object is referred to as its properties or attributes. A directory enables objects and their properties to be made available to users and administrators.

The directory's importance cannot be overstated. To use a telephone analogy, imagine the current global telephone system without telephone directories. The technology to make a call would be in place, but you would have a hard time locating a person's number to call. The creation of global electronic yellow pages could go a long way toward solving the "I know it's out there, I just can't find it" problem.

To create a directory service, X.500 addresses two main areas:

Directory structure How resources should be organized

Directory access How one is able to read, query, and modify a directory

X.500 Directory Structure

The X.500 directory structure is hierarchical, which facilitates a logical organization of information. Figure 1.10 illustrates the X.500 directory structure, and Table 1.3 explains it.

NOTE The X.500 terminology for the structure of a directory is the Directory Information Tree (DIT). The term for the information in the directory is Directory Information Base (DIB).

FIGURE 1.10 X.500 directory structure example

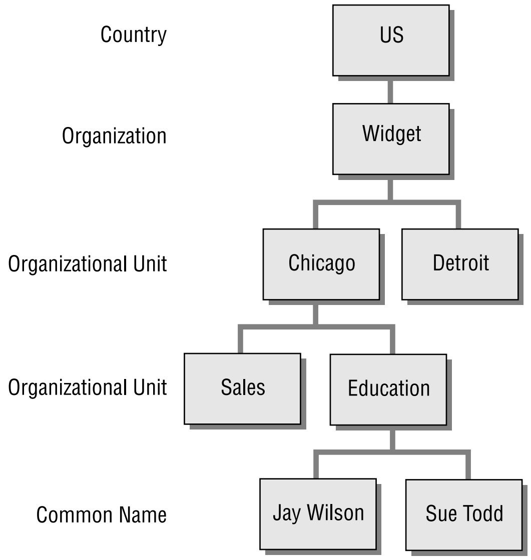


TABLE 1.3 Descriptions of X.500 Objects

X.500 Object	Abbreviation/Example	Description
Country	c=US	Country of the organization
Organization	o=Widget	Name of the organization
Organizational unit	ou=Chicago	Subcategory of the organization
	ou=Detroit	
Organizational unit	ou=Sales	Subcategories under the ou=Chicago
	ou=Education	
Common name	cn=JayWilson	Name of a specific resource (username, fax name, printer name, etc.)

To communicate the location of an object in the directory hierarchy, list the path to that object, starting at the top and moving down. This is called a Distinguished Name (DN). The DN of the example in Figure 1.10 is as follows:

c=US; o=Widget; ou=Chicago; ou=Education; cn=JayWilson

The differences between an X.500 address, the Distinguished Name (DN), and an X.400 address are due to their different purposes. A DN is the location of an object in the directory, whereas the X.400 address is the location of an object in a messaging system. Getting back to the telephone analogy, a DN is the location of a person in the phone book, and an X.400 address is where they are in the physical telephone system. This is illustrated by the fact that an X.400 address can include information about third-party messaging networks that are used to physically deliver a message, some examples being AT&T, MCI, and Sprint.

The 1988 release of X.400 incorporated the use of a DN address instead of, or along with, an O/R address. Some implementations of X.400 also incorporated some of the X.500 fields, such as ou=*x* and cn=*x*.

A directory also puts a more natural interface on network resources. Many communication objects have long numeric identifiers that are hard to remember. A directory allows objects to be presented to users by a natural descriptive term. The directory then maps the descriptive term to the numeric identifier.

X.500 has a 1988 version and a 1993 version.

Directory Access

Having a directory is only half the equation. Users and administrators must also be able to access it to read, query, and write to it. A user might query the directory for a printer on the fourth floor in the Sales department, and the directory could respond with the needed information about the printer. Other issues that must be addressed are security (e.g., who can access an object and modify its properties) and directory replication (a true global directory would need to be on more than one machine). These issues are addressed by directory access protocols.

The standard access protocol in the X.500 recommendations is the Directory Access Protocol (DAP). DAP is considered more of a model than a real-world protocol. This is because DAP is very computer-resource intense (i.e., heavy) on client machines, and the few implementations of it are proprietary. But a newer access protocol that is getting a lot of attention today is the Lightweight Directory Access Protocol (LDAP). LDAP is an Internet protocol derived from the X.500 DAP. One of the reasons LDAP is called lightweight is because it requires fewer computer resources on the client. While LDAP is an Internet protocol, it is designed to enable access to an X.500-type directory. Almost every major software vendor has pledged support for LDAP.

Summary

The Exchange product is a powerful client/server enterprise messaging product.

The types of applications in an Exchange environment are as follows:

- Electronic mail (e-mail)
- Groupware
- Other applications, such as fax, paging, and voicemail

The network computing model that Exchange uses to implement its messaging system is the client/server model. This model utilizes the computing power of both client computers and server computers.

Exchange was designed for enterprise-wide implementations and, consequently, meets the following requirements:

- Enterprise-quality application services
- Scalability
- Interoperability
- Performance
- Administration
- Reliability

The following industry standards are used by Exchange:

- Messaging Application Programming Interface (MAPI)
- Internet protocols (e.g., SMTP, POP3, IMAP4, LDAP, SNMP, and others)
- Remote Procedure Call (RPC)
- X.400
- X.500

Exam Essentials

Know the features and limitations of Exchange Server 2003. Remember that the two different versions of Exchange Server 2003 have very different capabilities and limitations. Exchange Server 2003 Standard Edition is limited to one storage group plus the recovery storage group with a limit of two databases (16 GB maximum in size) in that storage group. Clustering is not supported in Standard Edition, although a front-end/back-end configuration is. Exchange Server 2003 Enterprise Edition allows four regular storage groups plus the recovery storage group on each server. Each storage group can have up to five databases of unlimited size (limited only by hardware configuration). The Enterprise Edition does provide support for clustering, the X.400 connector and front-end/back-end configurations.

Know the available migration paths to Exchange Server 2003. As with most things, the migration paths to Exchange Server 2003 have changed from those in Exchange 2000 Server. Keep in mind these available migration paths using the Exchange Migration Wizard: Microsoft Mail, Microsoft Exchange, Lotus cc:Mail, Lotus Notes, GroupWise 4.*x*, GroupWise 5.*x*, Internet Directory LDAP via ADSI, and Internet IMAP4 mail.

Don't forget about X.500. As you will see the more you work with Windows Server 2003 and Exchange Server 2003, everything in Active Directory is based on the X.500 standard. If you can't already break down a distinguished name into its component parts by looking at it, you should look again at what makes up X.500.

RPC is key. In previous versions of Exchange, message transport between local Exchange servers occurred via RPC. Exchange 2000 Server and Exchange Server 2003 use SMTP for local message transport but still provide RPC for backward compatibility with Exchange 5.5 servers. Of course, the Outlook client still uses RPCs to interact with the Exchange servers. Keep an open eye later in the book, in Chapter 7, when we discuss the new RPC over HTTP feature of Exchange Server 2003 and Office Outlook 2003.

Review Questions

1. In Exchange Server 2003 Enterprise Edition, how many total storage groups can each server hold?

A. One

B. Two

C. Three

D. Four

E. Five

2. You are evaluating client applications for use with Exchange Server 2003. Many of your users need to be able to have offline access to all their mail folders and address books. Which of the following clients provide this? (Choose all that apply.)

A. Outlook 2003

B. POP3 clients

C. IMAP4 clients

D. Web browsers via Outlook Web Access

3. What is the name of the mechanism used when two Exchange components on the same machine pass instructions and data?

A. Remote procedure call

B. Remote instruction call

C. Local instruction call

D. Local procedure call

4. Which of the following operating systems can be used to run client applications that let users access Exchange Server 2003 computers? (Choose all that apply.)

A. Windows 3.*x*

B. Windows 98

C. Windows 2000

D. Macintosh

E. Unix

F. OS/2

5. What new feature of Exchange Server 2003 and Outlook 2003 allows clients to work from a local copy of their Exchange mailbox while still periodically connecting to the Exchange server?

   A. Cache mode

   B. Online mode

   C. Connected mode

   D. Offline mode

6. You are preparing a report on the features of Exchange Server 2003 for your manager. Which of the following would you list as features that are available in Exchange Server 2003 Enterprise Edition that are not available in the Standard Edition? (Choose all that apply.)

   A. Multiple storage groups per server

   B. Volume Shadow Copy support

   C. The X.400 connector

   D. Clustering support

7. What does Exchange Server 2003 use to speed up the writing of data into the database?

   A. Hyper-threading

   B. Symmetric multi-processing (SMP)

   C. Multithreading

   D. Transaction logs

8. Which of the following protocols can be used to *retrieve* messages from Exchange Server 2003? (Choose all that apply.)

   A. LDAP

   B. PNP

   C. PPP

   D. POP3

9. In the X.500 Distinguished Name c=US; o=Widget; ou=Chicago; ou=Education; cn=JayWilson, what is the name of the organization?

   A. US

   B. Widget

   C. Chicago

   D. Education

   E. JayWilson

10. Which Outlook feature enables e-mail content to include multiple format types, such as fonts, sizes, and colors?

A. WordPerfect

B. This software cannot do this.

C. Richman message content

D. Rich-text message content

11. Which of the following components work together to allow web browsers to access an Exchange server? (Choose two.)

A. Internet Information Services

B. Outlook Browser Access

C. Outlook Web Access

D. Exchange Web Server

12. You are the manager of a large Exchange organization. Recently, you began to suspect that someone was attempting to log on to resources without permission. You enabled auditing on your servers to keep track of suspicious activity. Which utility would you use to view the audited information?

A. Exchange System Manager

B. Active Directory Users and Computers

C. Computer Management

D. Security Manager

E. Event Viewer

13. You are a consultant who has been hired by the Arbor Shoes Company. Some time ago, their network administrator set up a messaging system. The administrator has left the company, and the company now wants you to help them decide whether to keep the existing system or move to Exchange Server 2003. In either case, you will need to train someone how to manage the system. A person at the company describes the messaging system to you in the following way: There is one server on their network that functions as the mail server. Whenever a new person needs a mailbox, the administrator creates a new folder and assigns it permissions. Then, the client application must be pointed to that folder. What type of messaging system do you suspect the company is running?

A. Client/server messaging system

B. Shared-file messaging system

C. Mainframe messaging system

D. Host-based messaging system

14. You have created a set of public folders on one of six Exchange servers. Since their creation, the public folders have become an important resource in your organization, and you are concerned that having all of the folders on one server creates a single point of failure; should that server fail, no one will have access to the folders. What is the simplest way to remedy this situation?

    A. Back the public folders up hourly so that you can quickly restore them to another server if necessary.

    B. Configure two of the servers as a cluster.

    C. Configure replicas of the public folder on another Exchange server.

15. You are currently making recommendations for the purchase of Exchange Server 2003 software. You expect that the size of your databases on several servers will run around 25 GB. Which edition of Exchange 2003 would you need for these servers?

    A. Exchange Server 2003

    B. Exchange Server 2003 Advanced Edition

    C. Exchange Server 2003 Enterprise Edition

    D. Exchange Server 2003 Datacenter Edition

16. In X.500, which of the following constructs describes the location of an object in a directory?

    A. Distinguished Name

    B. X.500 Address

    C. Organizational unit

    D. Common name

17. Your company is currently running Microsoft Mail, a shared-file messaging system. You have been trying to convince your manager to move to Exchange Server 2003. In telling your manager about the features of client/server messaging systems, which of the following features would you *not* include?

    A. Distributed processing

    B. Tight security

    C. Passive client application

    D. Reduced network traffic

18. Which of the following is an Internet management protocol supported by Exchange Server 2003?

    A. SNMP

    B. SMTP

    C. MMX

    D. MID

19. Which of the following protocols are used by web browsers? (Choose all that apply.)

A. HTTP

B. HTML

C. SNMP

D. SMTP

20. Which of the following is the X.400 component whose primary responsibility is to receive and route messages to their ultimate destination?

A. Message Routing Agent

B. Message Transfer Agent

C. Message Handling Agent

D. Message Delivery Agent

Answers to Review Questions

1. E. Exchange Server 2003 Enterprise Edition can support four production storage groups and a fifth storage group, called the recovery storage group, which can be used to perform quicker restorations of databases to the server. Exchange Server 2003 Standard Edition can support only one production storage group and also the recovery storage group, for a total of two storage groups.

2. A. Only Outlook 2003 and some previous versions of Outlook provide support for using folders and address books offline.

3. D. A local procedure call is an instruction passed between two components on the same computer. A remote procedure call is passed between two components on different computers that are linked via a permanent high-speed network.

4. A, B, C, D, E, F. Any operating system that can run a web browser (and there is one available for every OS out there) or a POP3 or IMAP4 client can access Exchange Server 2003 as long as Exchange is configured correctly.

5. A. Outlook 2003 allows users to operate in cache mode, whereby they have the benefits of both online mode and offline mode simultaneously. When Outlook first connects to the user's Exchange mailbox during a session, the contents of the mailbox are synchronized to an offline folder file on the local computer. Outlook then connects to the Exchange server periodically to check for updates and synchronizes the local offline folder and sends any outgoing messages. Should the client lose connectivity to the Exchange server, the user can continue to work normally until connectivity has been restored.

6. A, C, D. Volume Shadow Copy support is available in the Standard Edition of Exchange Server 2003. All of the other features listed require Exchange Server 2003 Enterprise Edition.

7. D. Data that is to be written to an Exchange database is first written to these log files (which can be done very fast). The data is later written to the appropriate database, which takes longer because of the structured nature of a database. If, for whatever reason, a server has an unintended shutdown, data that has not been written to the database is not lost; it can be automatically reconstructed from the transaction log files.

8. D. POP3 is a message retrieval protocol. LDAP is a directory access protocol. PPP is a remote access protocol.

9. B. In the X.500 Distinguished Name c=US; o=Widget; ou=Chicago; ou=Education; cn=JayWilson, Widget is the name of the organization. The country is US, the organizational units are Chicago and Education, and the common name is JayWilson.

10. D. Outlook enables the creation of rich-text message content that can include multiple fonts, sizes, colors, alignments, and other formatting controls.

11. A, C. Internet Information Services is a web server service built into Windows Server 2003 and provides the Internet protocol support for Exchange Server 2003. Outlook Web Access is a component that allows IIS to access the Web Store and then present information to web browsers in HTML format.

12. E. Windows Server 2003 and Windows 2000 Server can be configured to monitor and record certain events. This can help diagnose security events. The audit information is written to the Windows Event Log.

13. B. A shared-file messaging system is one in which a passive server is basically configured with a set of shared folders. Client applications are configured to regularly poll the shared folders to see if new mail has been deposited there.

14. C. Exchange Server 2003 allows you to configure replicas of public folders on multiple servers.

15. C. Exchange Server 2003 Standard Edition has a 16-GB limit on the size that a database can reach. Exchange Server 2003 Enterprise Edition does not have this limit. There is no such edition as Exchange Server 2003 Advanced Edition or Datacenter Edition.

16. A. In X.500, the Distinguished Name (DN) describes the location of an object in the X.500 directory.

17. C. In a shared-file messaging system, servers are relatively passive, and clients perform almost all active messaging functions. Even though the server in a client/server system plays a much more active role, the client is still by no means passive.

18. A. Simple Network Management Protocol (SNMP) is a TCP/IP-based management protocol supported by Exchange Server 2003 in the form of the MADMAN MIB, an information base of manageable Exchange components.

19. A. HTTP is the protocol used to define how messages are sent between a web browser and a web server.

20. B. The Message Transfer Agent (MTA) is the protocol that runs in the message routing machines (i.e., routers). An MTA machine receives a message, stores it so it can calculate its next route, and then forwards it to either another MTA machine or its ultimate destination.