

chapter I

Introducing Exchange 2003 and Exchange Administration

74 percent of the business people surveyed recently believed that losing e-mail service presents more of a hardship than losing telephone service.

— META Group survey (www.metagroup.com)

WHEN COMPARED TO WINDOWS NT and Exchange 5.5, Windows 2000 and Exchange 2000 were revolutionary products. Everything from architecture and functionality to management interfaces changed drastically. The learning curve from the earlier to the later Microsoft operating and messaging systems was quite steep. On the other hand, compared to Windows and Exchange 2000, the 2003 versions are much more evolutionary than revolutionary products. If you know Windows and Exchange 2000, you will have little difficulty adapting to the 2003 flavors. You're going to welcome the evolutionary changes in Windows and Exchange 2003 with open arms. These changes will improve your end users' experiences and will make your administrative tasks easier. I will look at some of these changes in this chapter.

In order to manage Exchange Server 2003 successfully, you need to understand its various components. You need to know what executables run the components, what the components do, and to some extent how they do what they do. Finally, you need to know how components depend on each other and on various Windows services. Understanding these concepts will make it easier for you to perform day-to-day management tasks and put you in a much better position to troubleshoot problems that arise. A major portion of this chapter is devoted to Exchange components.

Exchange Server 2003 comes in two editions: Standard and Enterprise. The Enterprise Edition offers greater capacity, clustering, and more protocol support. I will try to help you better understand the two editions and the features of each so you can make cost-efficient decisions about the software that supports your Exchange and Windows 2003 systems.

If you have older Windows and Exchange installations, you have to decide how to get to Windows and Exchange 2003. You can upgrade or do a fresh install of the two products on new hardware and then move Exchange objects to the new server or servers. If your first thought is to upgrade existing servers, you've come to the right place. I'm going to try very hard in this chapter to talk you out of that approach. First and foremost, Exchange 5.5 cannot be directly upgraded to Exchange 2003.

Once your Windows and Exchange 2003 systems are up and running and you've eliminated earlier OS and Exchange versions, you have the option of switching them to native mode. Native mode offers a number of very useful enhancements, but sometimes for a variety of reasons you can't zap those old servers and "go native." I will try to help you deal with this dilemma later in this chapter.

A successful Exchange 2003 deployment hinges on many elements; these include a strong dependency on Windows 2003 Active Directory (AD), Windows 2003 Internet Information Server (IIS), a properly configured DNS (Domain Name Service) infrastructure, sufficient and reliable hardware, and good operational practices. Your Exchange 2003 installation will have serious problems unless you have a good understanding of not only Exchange 2003, but also Windows 2003, AD, and DNS. Like Exchange 2000, Exchange 2003's destiny is much more intertwined with Windows 2003 than versions 5.5 and earlier of Exchange. A basic understanding of the Exchange 2003 architecture and deploying Exchange 2003 on the proper hardware platform will also be crucial to your success.

One of the most important parts of deploying any Exchange system is making design decisions that relate to supporting your organization. This includes choosing the right edition of Exchange 2003 Server, deciding how to best store your data, maintaining time synchronization, setting reasonable standards (Active Directory, Exchange performance, user space allocation, etc.), and picking the right hardware. Placing your Exchange 2003 system on appropriately sized and configured hardware will also help to keep you happy and safe from end-user lynch mobs.

Finally, providing your user community with good documentation, notification, and training will help to minimize your administration woes. Most experienced Exchange administrators will tell you that educating their users, keeping them informed, and managing their expectations are some of the most powerful tools in their operations arsenal.

Yet perhaps first and foremost, essential tools to have in your bag of tricks are solid operational practices that will help reduce the likelihood of downtime and improve the recoverability from disasters—and help keep you sane. One particularly wise Exchange guru once said his secret to Exchange success was the following:

- ◆ Perform daily backups of Exchange.
- ◆ Check the event logs.
- ◆ Make sure the server does not run out of disk space.
- ◆ Check the queues.
- ◆ Then leave Exchange alone.

Although I elaborate on this in a *lot* more detail in Chapter 6, "Daily and Long-Term Operations," successful Exchange server administration and management strategies have not changed since Exchange was first released.

So, is that all there is to say about Exchange administration? If so, why have volumes of information been written about it, and why am I writing more? The answer is simple: We all benefit from shared experiences. Combine that with the fact that software documentation and training do not always make matters crystal clear, and you have good reasons for a book about skillfully maintaining Exchange.

THE MAKINGS OF A GOOD EXCHANGE ADMINISTRATOR

I have tended to a number of Exchange “disasters” where the clients were running Exchange 5.5, Exchange 2000, or Exchange 2003. These were situations in which I was called in to fix a pretty serious problem. I classify these as disasters because in each case the user community was without e-mail services for more than half of a business day. In one case, the user community was without e-mail for more than a week before I was called. One of the strengths I look for in system administrators is the ability to know when they are in over their heads and when to call for help. This includes not being afraid to call Microsoft Product Support Services.

With a few exceptions, the aforementioned disasters were either caused or compounded by administrators who were not prepared for the disaster, did not know what they were doing, or did not call for help when they should have. The administrators did not have a clear understanding of Exchange, Active Directory, and the steps to successfully manage an Exchange system, nor had they documented or practiced disaster recovery beforehand.

Disaster prevention involves two major steps. The first is recognizing that you cannot solve every problem in the world (and not being afraid to admit it). The second step—and the one you are taking now, by reading this book—is to do everything you can to improve your knowledge of Exchange 2003 (and Windows 2003).

What's New in Windows and Exchange 2003?

Windows 2003 includes improvements in Active Directory: easier deployment and management, increased security, and better performance and dependability. Additionally, overall security has been strengthened and support for applications that run on Windows 2003 has been significantly updated. Security improvements are a two-edged sword. Although they better protect everything in your Windows and Exchange environment, you and your users' first encounter with them is likely to come as a bit of a shock. For example, by default, Windows 2003 implements strong password requirements. Passwords must be of a specific length and must include uppercase and lowercase letters as well as numbers. All those three- and four-letter passwords won't cut it any more—at least if you don't change the defaults, which isn't all that easy.

Improvements on the Windows 2003 storage side, so important to smooth and reliable Exchange Server operations, include snapshot backups of disk volumes, system-level open-file backup and much easier Storage Area Network (SAN) management. On the networking side, Windows Server 2003 supports IPv6 for increased security and a solution to the rapid depletion of Internet Protocol (IP) addresses.

Together Windows and Exchange 2003 include a great new way to connect MAPI clients such as Outlook 2003 to Exchange servers over Internet-based connections. Until Exchange 2003, such connections required the use of the Windows RPC protocol either directly over TCP/IP or RPC-TCP/IP encapsulated in virtual private network packets. Use of direct RPC-TCP/IP became a major problem as many corporations and ISPs closed off port 135, the port that supports RPC, to protect against a variety of RPC-based attacks on Microsoft servers. Exchange 2003 supports RPC encapsulated in HTTP. This approach uses the same port 80 that is used for browsing the Web. You need Windows 2003, Exchange 2003, and Outlook 2003 running on Windows XP clients to pull all of this off, but RPC-over-HTTP solves a problem that has plagued Outlook-to-Exchange public network connectivity since the two products came on the market.

Exchange 2003 also includes something for wireless clients. The wireless client-server synchronization functionality of Microsoft Mobile Information Server, an add-on to Exchange 2000 Server, is now part of Exchange 2003. This allows all those wireless Pocket PC PDAs running around out there to seamlessly sync with Exchange inboxes, calendars, and contacts without user intervention.

Exchange 2000 used storage groups to hold both private and public e-mail and other items. This has not changed with Exchange 2003. However, 2003 brings a new kind of storage group to the table, recovery storage groups. You can use these to restore all or part of an Exchange mailbox or public folder store, without having to overwrite an existing store or create a new regular mailbox or public folder store.

Exchange 2003 is set for maximum security on installation, rather than depending on administrators to figure out what they need to do to secure Exchange. Outlook Web Access (Internet browser access to Exchange mailboxes) is not only more secure, but it looks and feels more like Outlook 2003. The Exchange 2003 antivirus application programming interface (API) supports more virus and spam catching options. Wireless access to Exchange Server 2003 is greatly improved when compared to Exchange 2000 options. Migration to Exchange Server 2003 from Exchange 5.5, or Exchange 2000 for that matter, has been greatly simplified with the addition of the Exchange Deployment Tools.

A few features of Exchange 2000 Server were removed from Exchange Server 2003. These include real-time collaboration features, automatic mapping of the M: drive, and Key Management Services.

Real-time collaboration features such as chat, Instant Messaging, Exchange Conferencing Server, and OWA Multimedia Messaging are gone. Some of these features will work if you upgrade from Exchange 2000 Server, but if you need these features for new installations of Exchange 2003, you'll have to install Microsoft's new real-time communications and collaboration server called the Microsoft Live Communications Server 2003 (formerly known as Greenwich or the Real Time Communications Server).

The default M: drive mapping gave you file system-based access to the Exchange Information Store. By and large, it was more trouble than it was worth, leading to corruption of the mailbox store when file-based operations such as backup were performed. You can still access the Information Store through the file system, but you have to enable it in the Registry.

Key Management Services supported sending secure messages through Exchange Server by allowing certificate issuance to Exchange users and key escrow. That feature is now fully supported by Windows Server 2003's Public Key Infrastructure. Therefore, Key Management Services are no longer required.

Major Exchange 2003 Components

If you want to understand how Exchange 2003 works, you first need to get comfortable with the major Exchange components and the executable files that support them. It also helps to understand how the Exchange executables depend on each other and on key Windows 2003 components.

When you know the executables and dependencies, you can more easily manage the components and undertake certain kinds of troubleshooting. Here are a few examples:

- ◆ Exchange components cannot function if the Exchange System Attendant is not running.
- ◆ Many Exchange components are dependent on the Exchange Information Store.
- ◆ The Windows SMTP and NNTP services must be running before Exchange components are able to start up.

An Overview of Exchange Components

Table 1.1 presents a list of major Exchange components and the executable files associated with them. The table is designed to provide you with a brief introduction to the components. The sections that follow fill in the details for each of the components.

TABLE 1.1: MAJOR EXCHANGE COMPONENTS AND ASSOCIATED EXECUTABLE FILES

EXCHANGE 2003 COMPONENT	ASSOCIATED EXECUTABLE FILE
Microsoft Exchange System Attendant	mad.exe
Microsoft Exchange Information Store	store.exe
Message Transport System Components:	
Exchange Interprocess Communications (ExIPC)	part of inetinfo.exe
Advanced Queuing Engine	part of inetinfo.exe
Message Categorizer	part of inetinfo.exe
Microsoft Exchange Routing Engine	part of inetinfo.exe
SMTP Service	part of inetinfo.exe
Microsoft Exchange MTA Stacks	emsmta.exe
Microsoft Exchange Event	events.exe
Microsoft Exchange Management:	
Server Component	exmgmt.exe
Workstation Management console	"exchange system manager.msc"
Internet Information Service	inetinfo.exe
Microsoft Search	mssearch.exe

The System Attendant

The Exchange System Attendant is essentially the general manager of the Exchange server. It is the first Exchange service that starts and the last one that shuts down. Although a novice might actually think that

this service performs few, if any, useful functions, it actually is responsible for a lot of odd yet important jobs. Some of the tasks that the System Attendant runs include:

- ◆ Performing offline address book generation.
- ◆ Running the DS2MB (Directory Service to Metabase) update process to keep the IIS Metabase in sync with the information in Active Directory.
- ◆ Generating proxy addresses for X.400, SMTP, and other address types based on the defined Exchange 2003 recipient policies.
- ◆ Emulating the Exchange 5.5 directory service through a process called DSProxy for MAPI clients prior to Outlook 2000 that cannot receive referrals.
- ◆ Passing referrals for Outlook 2000 and later clients that need to be referred to a Global Catalog server for querying address information.
- ◆ Running the Recipient Update Service to make sure that AD objects are included in the appropriate address lists.
- ◆ Running the DSAccess cache, which caches information about AD objects. This cache is available for Exchange 2003 to query rather than querying the AD directly for each lookup request.
- ◆ Inserting data into and managing the message tracking logs.

NOTE *The System Attendant's process name is MAD.EXE; this is short for either Mailer Administrative Daemon or the Monitoring and Administration Daemon, depending on who you ask.*

The Information Store

The current Information Store database engine is called ESE98 (Extensible Storage Engine). More information on the database engine and storage technology is in Chapter 4, "Understanding Exchange 2003 Data Storage."

Exchange breaks down storage into either public or private Information Stores. All mailbox data is stored in a mailbox store (private Information Store), while all public folder data is stored in a public folder store. These stores each have two separate components, an EDB file and an STM file.

NOTE *A MAPI (Messaging Application Programming Interface) client is any client that sends and reads messages where the message properties are defined as MAPI properties. MAPI clients include the original Exchange client, Outlook 97/98/2000/2002 and Outlook 2003.*

The EDB file is called the MAPI store. This is a rich, hierarchical property store; messages sent by MAPI clients are stored here. Therefore, all messages stored here have MAPI properties associated with them.

The STM file is not nearly as structured as the EDB file. Messages are not converted to MAPI messages when they arrive, but instead are stored in their native format (typically MIME). This includes messages sent by SMTP clients. However, the EDB file does contain a list of *all* messages stored in each folder, so certain MAPI properties for messages in the STM file are promoted to the EDB file. To improve performance, the STM file data is accessed through a kernel-mode device

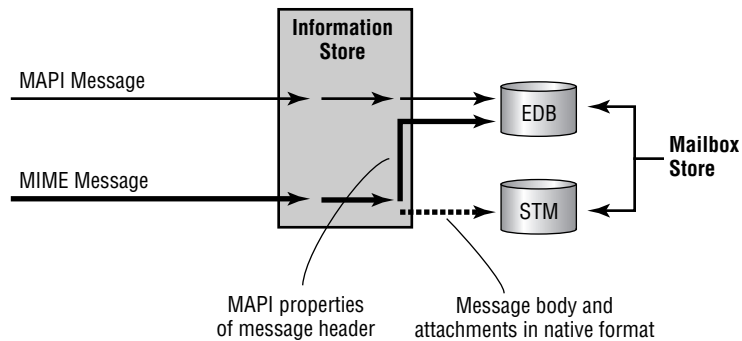
driver called ExIFS; a Windows Explorer extension that uses this device driver also allows the entire store to be accessible through the file system.

NOTE By now, you have probably seen the term “web store” or “web storage system” used (or overused) in technology media. The web store is not actually a single database but a technology for providing access to data through HTTP/DAV or the ExIFS.

Figure 1.1 illustrates two examples of message storage, a MAPI message and a MIME message.

FIGURE 1.1

Messages arriving in a mailbox store



The first message shown in Figure 1.1 is sent by a MAPI client such as Outlook 2003. The client designates most of the message’s MAPI properties, and the client sends the message to the Exchange server; the message transport and the store may also set some of the message properties. The Information Store saves the entire message in the EDB file.

The second message is formatted by a client such as Outlook Express as a MIME message. The Internet Mail Service in Exchange 5.5 would have converted this message, but the Advanced Queuing Engine in Exchange 2003 simply passes it along to the Information Store in its native format. The Information Store determines that the message is in native format, and then “promotes” certain properties of the message header (such as the To, From, Subject, and Date information) to MAPI properties, and finally stores this information in the EDB file along with a “pointer” that points to the message body and attachments in the STM file. Technically, there are three separate phases of property promotion: The initial properties are promoted when the message is sent to the server by the client, the second set when the messages is accessed, and finally if the content is changed by a MAPI client.

NOTE In a pure MAPI environment with no SMTP connectivity to the outside world, your STM files will hardly grow in size at all. In an environment with all POP3 and IMAP4 clients, the STM file will grow significantly while the EDB file will hardly increase in size.

CONTENT CONVERSION ON DEMAND

The obvious question now is “What happens if a MAPI client reads a message that was sent to the Exchange server via SMTP and is formatted as a MIME message?” Simple. The Information Store retrieves the message into memory on the Exchange server and performs an “on-the-fly” conversion. The message is *not* converted in the STM file, merely in the copy in memory. The message is saved

as a MAPI message only if a MAPI client modifies the message. If the message contains an attachment but the attachment is not modified, then it is not moved into the EDB data file.

The same holds true of a message that was sent by a MAPI client but is now being retrieved by a non-MAPI client such as Outlook Express. The Information Store converts the message on-the-fly to a MIME or non-MIME message and passes it on to the client.

So why all the conversion? Why not just store all messages in a common format? In Exchange 5.5, all inbound SMTP message content was converted to MDBEF message format by the Information Store's IMAIL process. If the message was retrieved by a POP3 or IMAP4 client, it was once again converted by the IMAIL process. If your environment is a pure environment of one type or another (all MAPI or all MIME clients), converting to another format would be too much overhead compared to keeping the content in its native format.

Microsoft's developers recognized the changing nature of the messaging world and that in the future we will have more mixed-client environments. By converting message content on demand, they achieved better performance than if they simply stored the message in its native format and converted the message only when necessary. OWA and IMAP4 clients are becoming increasingly popular, and future versions of Outlook will more than likely provide the ability to access data using HTTP/DAV rather than MAPI. With a steady turn toward an emphasis on XML, HTTP/DAV, and other "Internet" clients, it makes sense to figure out how to keep data stored in its native format without content conversion. Further, the streaming store provides much higher performance access for message attachments. Messages stored in the EDB file are written in 4KB page reads, whereas the STM file is accessed using kernel-level I/O in 64KB streamed chunks. This method is much more efficient.

STORAGE GROUPS AND MULTIPLE STORES

In Exchange 5.5, you are limited to a single private Information Store and a single public Information Store. If you are running Exchange Server 2003 Enterprise Edition, you can create up to 20 separate mailbox or public folder stores (maximum EDB size of 16TB). Storage groups are used to organize these mailbox and public folder stores. Exchange Server 2003 Standard Edition allows for only a single mailbox store (maximum EDB size of 16GB) and one public folder store.

Both editions of Exchange 2003 support a Recovery Storage Group (RSG). You use RSGs to restore from backups entire storage groups, databases or mailboxes. With RSGs, you no longer need to set up a separate Exchange server and jump through the tricky process of recovering data to the new server and then moving it to your production server.

Storage Groups

Storage groups are the building blocks for multiple stores. Exchange 2003 Enterprise Server allows you to create up to four separate storage groups, each of which can contain up to five mailbox stores or public folder stores and has its own set of transaction logs. Circular logging can be turned on for some storage groups depending on the requirements of the data stored in the storage group.

TIP For optimal performance, each storage group's transaction log files should be placed on a separate physical hard disk. The transaction logs should not share this hard disk with any other application or data.

When the first database in a storage group is mounted, a new instance of the ESE database engine is started. All instances of ESE run as part of the `store.exe` process.

Multiple Stores

What possible uses can there be for additional mailbox stores? Here is a list of possible advantages to using more than one mailbox store:

- ◆ Company executives or VIPs can be placed in a separate mailbox store to allow for quicker backup and restoration times.
- ◆ The overall size of any specific mailbox store can be reduced by splitting up the storage load between two stores.
- ◆ You can specify separately which stores need to be full-text indexed and which do not.
- ◆ Additional public folder stores can be used to store data that is accessed exclusively via OWA or the ExIFS driver.

However, be cautioned that if you choose to have more than one private mailbox store on a single server, there are a few things you should consider:

- ◆ Each additional store that you mount consumes at least another 10MB of RAM.
- ◆ Single-instance storage is preserved only within a single store. Recipients across multiple stores will cause multiple copies of a message to be created.
- ◆ Backup and recovery scenarios require more diligence and testing in this more complicated environment.

The Message Transport System

In Exchange 2003 all message transfer is the responsibility of the Message Transport System. One of the design goals for the Exchange 2003 message transport system was to ensure that all messages were processed exactly the same. To that end, all messages are delivered through the Advanced Queuing Engine—even those that are destined for local delivery.

To do this without affecting performance and scalability is something of a monumental task. Further, all message transport in a native Exchange 2003 organization is via SMTP rather than RPC, so all Exchange 2003 servers must have the capability to transfer SMTP messages between servers in the same routing group.

In 1996, when Exchange 5.5 was released, Microsoft had three separate teams of developers working with SMTP: the Exchange team, the IIS team, and the Microsoft Commercial Internet System team. When Windows 2000 was being developed, Microsoft decided to combine these three teams into one group that would develop a single SMTP transport system to be used by all Microsoft components requiring SMTP transport.

NOTE *All messages including those destined for local delivery are handled by the Advanced Queuing Engine.*

Windows and Exchange 2003 carry forward the concept of a single SMTP-based message transport system. This system depends extensively on IIS and its SMTP service. The Exchange MTA Stacks service is brought into play when messages must be delivered or received from Exchange 5.5 or X.400 systems.

EXCHANGE SERVER SMTP ENHANCEMENTS

The IIS SMTP component is required prior to the installation of Exchange 2003. When Exchange 2003 is installed, it enhances (not replaces) several of the existing SMTP components so that they can work with Exchange 2003 more effectively. The SMTP transport components include:

Exchange Interprocess Communication Layer (ExIPC) Provides the queuing layer that transfers message header information quickly and efficiently between IIS and the Information Store.

Advanced Queuing Engine Creates and manages the queues through which a message passes when it is being delivered. These queues include per-domain queues, the Pre-Categorizer queue, the Post-Categorizer queue, and the local delivery queue.

Message Categorizer Provides features specific to Exchange, such as checking recipient home servers, checking recipient limits, checking sender limits, and expanding distribution lists. This is an enhancement to the Advanced Queuing Engine. The IIS SMTP component has a basic message categorizer (`cat.d11`) that is disabled by default. When Exchange 2003 is installed, the Exchange categorizer (`phatcat.d11`) replaces the IIS categorizer.

Routing Engine The Routing Engine maintains the Link State Table, which is used by the Advanced Queuing Engine to determine the “next hop” through which a message needs to be routed. The Routing Engine also maintains information about whether or not a link is currently available.

SMTP Service Handles transmission of messages between hosts using the SMTP protocol.

THE MESSAGE TRANSFER AGENT STACKS

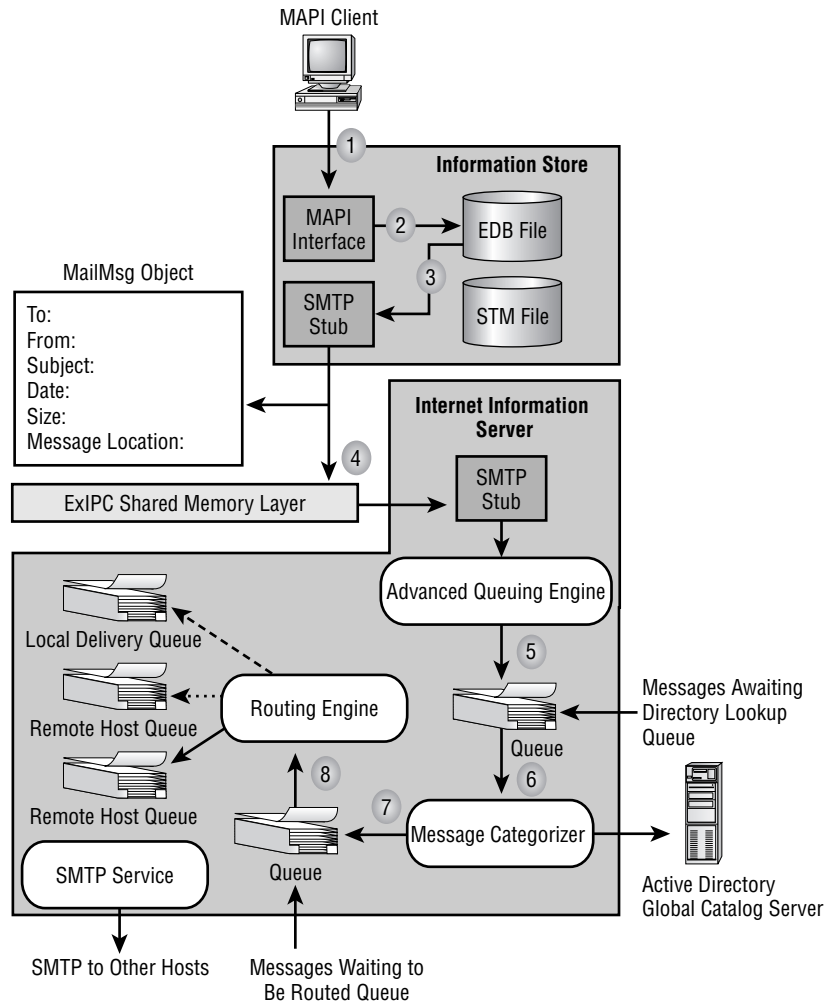
The Message Transfer Agent Stacks service (MTA) performs two functions. It allows for backward compatibility with Exchange 5.5 servers, and it performs X.400 message delivery functions. All messages between Exchange 2003 and Exchange 5.5 servers are delivered by the MTA using the Microsoft RPC protocol. Additionally, the MTA supports delivery of messages to foreign X.400 systems as well as internal message routing when the X.400 connector is used.

The MTA will be an important component if you are managing an Exchange-based U.S. Department of Defense (DOD) Defense Messaging System (DMS). For more information on DMS, visit www.1mcdms.com.

HOW MESSAGES ARE ROUTED IN EXCHANGE 2003

The Advanced Queuing Engine is central to message routing in Exchange 2003. When a message is transferred to the Advanced Queuing Engine, each component has specific functions that it performs to move the message to its next hop. Figure 1.2 shows a basic diagram of the Advanced Queuing Engine.

FIGURE 1.2
SMTP Advanced
Queuing Engine



If we follow the message through its path as it travels through the Information Store and Advanced Queuing Engine, it looks something like this:

1. A MAPI client submits a message through the Information Store's MAPI interface.
2. The Information Store determines that the message is a MAPI message and stores the entire message in the EDB portion of the mailbox store.
3. The Information Store creates an object that represents the message called the *MailMsg* object (also called the *IMsg* or *IMailMsg* object). This object is merely a small chunk of memory that

identifies information such as the To, From, Subject, Date, Size, and other message properties, as well as where the actual message content is stored. In this case, the message content is stored in the EDB portion of the mailbox store. Only the MailMsg object, not the entire message content, is passed to the SMTP memory stub in the Information Store. The SMTP memory stub is a queuing location provided by the ExIPC queuing layer between IIS and the Information Store.

4. The Information Store's SMTP stub passes the MailMsg object through the ExIPC shared memory layer to the SMTP stub in IIS.
5. The MailMsg object is passed to the Advanced Queuing Engine, which stores the message in the Messages Awaiting Directory Lookup queue (Microsoft also refers to this queue as the Pre-Categorizer queue). You can see messages in this queue by using the Exchange System Manager and viewing the queues in the SMTP virtual server.
6. The MailMsg object proceeds to the Message Categorizer component, which takes message information—such as the sender, recipient, and size—and performs AD queries (to a Global Catalog server) to determine if the message exceeds the sender's or recipient's limits. Also at this point, gateway and routing restrictions are determined. If the message is sent to a distribution list, the Message Categorizer also expands the distribution list. If the message is being sent to both external and internal recipients, the Message Categorizer performs a bifurcation of the message (two or more copies are created) so that an RTF copy is sent to internal recipients and a MIME copy is sent to external recipients.

TIP When the Message Categorizer component is performing directory lookups, connectivity to the Global Catalog server is critical. Any location that contains an Exchange 2003 server should also have a local Global Catalog server.

7. The MailMsg object is placed in the Messages Waiting To Be Routed queue (I also refer to this queue as the Pre-Routing queue).
8. The MailMsg object is handed off to the Routing Engine, which examines the destination domain or server and compares the destination with routes that are available in the Link State Table. If the message is for a local recipient, the MailMsg object is placed in the local delivery queue and the object is passed back to ExIPC. If the message is for remote delivery, the message is placed in the appropriate outgoing queue, and the SMTP service delivers the message off of the server. If the message is to be delivered by the message transfer agent (MTA) to Exchange 5.5 server or to an X.400 connection, the message is routed back to the local store and placed in the MTA mailbox's MTS-OUT folder. Only when the message delivery to a remote host begins are the actual contents of the message moved out of the Information Store.

NOTE Only the MailMsg object—not the entire message—is passed through the Advanced Queuing Engine. The message content is moved only when the message is ready to be delivered to another server or store.

There are slight variations on this message routing process for inbound messages, but the process is essentially the same.

EVENT SINKS

If all messages, regardless of whether or not they are destined for local delivery, are routed through the same message routing components, you might think this would be a good place to handle other types of message processing needs. If you thought this, you are not alone. The Exchange developers introduced the concept of event sinks to Exchange 2000 and carried it through to Exchange 2003. An *event sink* is a small program that runs when a specific type of event occurs, such as a message arrival or the completion of categorization. In fact, the Exchange 2003 extensions to the IIS SMTP service are implemented as event sinks.

Types of Event Sinks

There are three major categories of event sinks: Information Store events, transport events, and protocol events. When dealing with the Message Transport system, we are concerned mostly with the protocol and transport events.

Protocol events are used to extend SMTP functionality by enhancing or providing additional SMTP command verbs. Possible uses include rejecting all messages from domains that do not have a reverse lookup record, changing the behavior of an existing SMTP command verb, or adding a custom SMTP command verb.

Transport events can be used when the message is passing through the Message Transport system. Uses include content inspection, adding message disclaimers, antispam features, and message compression. Another use might be a virus-scanning service, but the Exchange 2003 antivirus API provides much better virus-scanning performance. Transport events can be fired at the following points:

- ◆ When a message is submitted to the Message Transport system (inbound from the Information Store or from the SMTP service)
- ◆ When a message is placed in the Pre-Categorizer queue
- ◆ When a message is in the categorizer
- ◆ When a message is in the Pre-Routing (Post-Categorizer) queue
- ◆ When a message is being processed by the Routing Engine

Writing an Event Sink

The keys to writing a successful event sink are speed and accuracy. The accuracy part comes with comprehensive testing. However, the speed part comes with your choice of a programming language (and of course, writing efficient code). Development platforms include any programming language that is compatible with the Component Object Model (COM), including VBScript, JavaScript, Visual Basic, C, and C++.

If the event sink you want to develop will fire only for a select few messages an hour, then you can use Visual Basic, VBScript, or JavaScript. However, if your event sink will fire for all messages being processed, then you should use C or C++ to ensure maximum performance.

Exchange Management

You perform most Exchange 2003 management tasks using the Exchange System Manager (ESM), which you can run standalone or as a Microsoft Management Console (MMC) snap-in. The snap-in, which is an

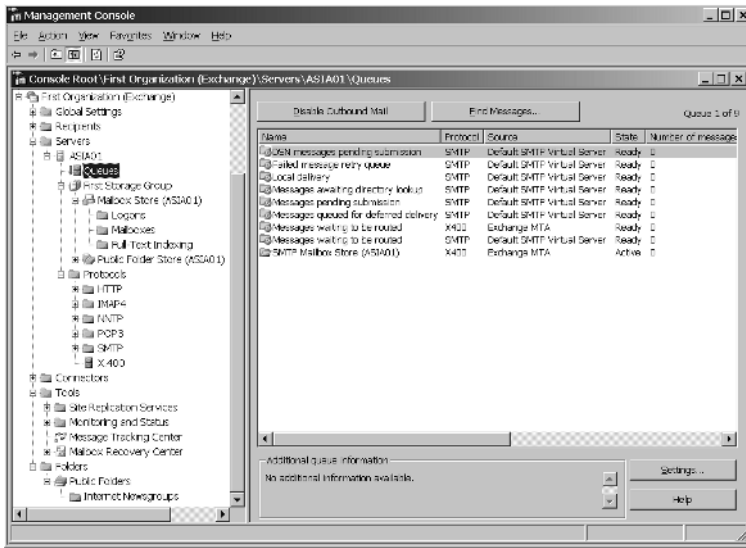
executable file, is called Exchange System Manager .msc. ESM is client software that talks to Exchange servers using a variety of protocols and services, including MAPI for mailboxes in Exchange mailbox stores, the Windows Management Instrumentation system (WMI) to display simple monitoring data such as data relating to Exchange queues, and HTTP/DAV for Exchange public folders. ESM also communicates with Active Directory using Active Directory Service Interfaces (ADSI). A server-based program called `exmgmt.exe` supports Exchange message tracking and access to Active Directory both directly and through certain WMI providers.

To run ESM standalone, select Start > All Programs > Microsoft Exchange > System Manager.

To create a new MMC and Add ESM to it, select Start > Run and enter `MMC` in the Open field. When MMC opens, select File > Add/Remove Snap-in. Then click Add on the Add/Remove Snap-in dialog box and select System Manager from the Add Standalone Snap-in dialog box. OK your way out of the dialog boxes. Be sure to save your new MMC.

Figure 1.3 shows an MMC with the Exchange 2003 system manager in place. As you can see, you can manage everything in the ESM tree from enterprise wide settings to server-based storage groups to messaging protocols to server and network link monitoring to mailbox recovery.

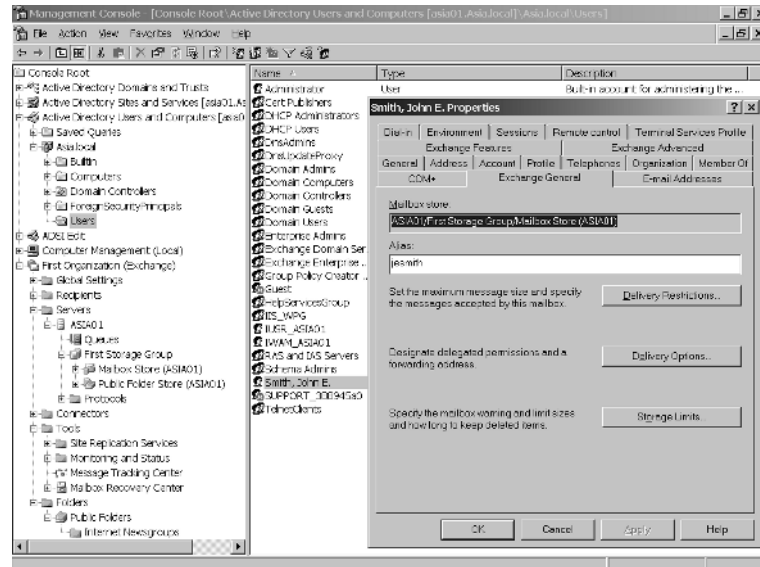
FIGURE 1.3
An MMC with the Exchange System Manager snap-in



You don't manage users and key attributes of their Exchange mailboxes, contacts, and distribution groups in ESM. You manage them using the Active Directory Users and Computers plug-in shown in Figure 1.4. Throughout this book, I will show you how to use both ESM and Active Directory Users and Computers to perform various management tasks. My goal here is just to show you how comprehensive EMS is and how nicely MMC brings together the tools you need to manage both your Windows and Exchange 2003 environments.

WARNING You need different Windows 2003 privileges to manage Exchange 2003 and Windows 2003. If you work in a large shop where Windows and Exchange management tasks are the responsibility of different groups, you may have to rely on others with appropriate permissions to create users and the mailboxes associated with those users. You may also have to rely on others to create other Exchange recipients, such as mail-enabled users, contacts, and distribution groups.

FIGURE 1.4
An MMC with the Active Directory Users and Computers snap-in



Internet Information Server

Internet Information Server (IIS) 6 plays an important role in accessing and storing data in Exchange 2003. All Internet protocol support is handled by IIS. Figure 1.5 shows a basic architectural diagram of IIS and the Exchange 2003 Information Store.

All communication for POP3, IMAP4, SMTP, HTTP, and NNTP is now handled by IIS rather than being integrated into other Exchange components. IIS receives Internet protocol requests and messages, and passes these on to the Information Store. In order to achieve optimal performance, the Exchange developers implemented a shared memory layer between IIS and the Information Store called the Exchange Inter-Process Communication (ExIPC) layer, which I discussed earlier in this chapter. The ExIPC layer is also referred to as EPOXY because it's the glue that holds the Information Store and IIS together, and thus the ExIPC DLL name is EXPOXY.DLL.

Essentially, ExIPC is nothing more than an area of memory that the two processes share for queuing data and requests between them. Because it is shared memory, data and requests are transferred quickly and efficiently.

FIGURE 1.5
IIS and Exchange
2003 Information
Store interaction

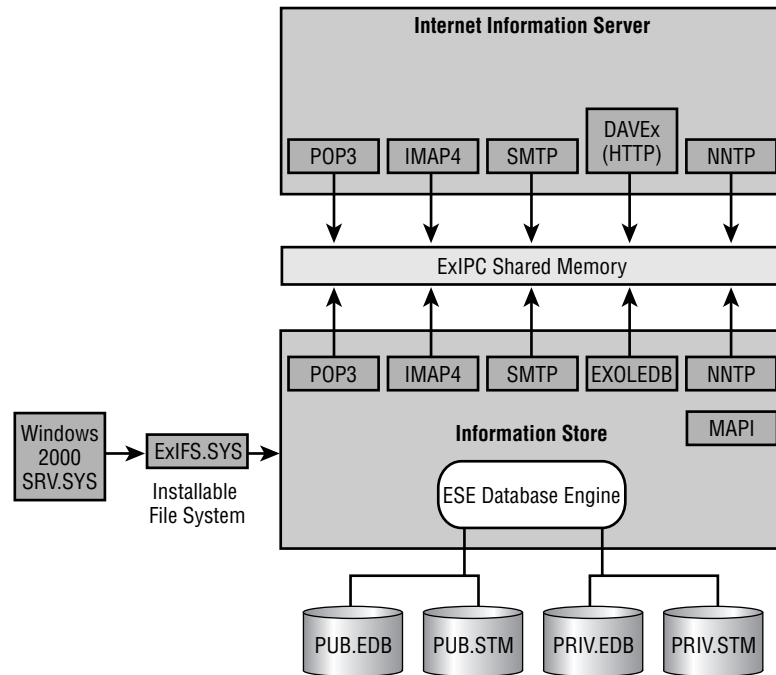


Figure 1.5 does not include Outlook Web Access (OWA), end-user web browser access to an Exchange mailbox and a public folder store. OWA architecture is similar to the architectures shown in Figure 1.5. ExIPC acts as the intermediary between IIS and the Information Store. On the IIS side the W3 service talks to DAVEx. DAVEx then talks to ExOLEDB, which is on the Information Store side, through ExIPC. Finally, ExOLEDB talks to the Information Store.

Microsoft Search

Microsoft Search is a Windows service that supports text-based searches. Various Microsoft products, such as SQL Server and Exchange, use it to build full-text indexes that make finding objects containing specific text easier. For detail on the architecture of Exchange full-text indexing functionality, see Chapter 4.

You will appreciate full-text indexing because it reduces the load that nonindexed text searches can put on a server. Once the index has been created, you can choose to update it on whatever schedule works for your organization. A daily update is sufficient in many cases, though you can schedule updates at much shorter intervals.

With an updated index, your users will be very happy. Searches that took forever in Exchange 5.5, which lacked a text-indexing system, are often finished in seconds.

Exchange 2003 Service Dependencies

There are a number of Windows 2003 services that must be started before you can start the first Exchange service. There are still other services that Exchange 2003 requires to be installed prior to

Exchange installation (IIS services including the web service, SMTP, and NNTP.) Table 1.2 lists the Windows 2003 and Exchange services that must be started in order to start the principal services.

TABLE 1.2: EXCHANGE 2003 DEPENDENCIES

EXCHANGE 2003 SERVICE	WINDOWS 2003 SERVICES (DEPENDENCIES)
Microsoft Exchange System Attendant (mad.exe)	Event log NT LM Security Support Provider Remote Procedure Call (RPC) Server Workstation
Microsoft Exchange Information Store (store.exe)	ExIFS Microsoft Exchange System Attendant
Microsoft Exchange MTA Stacks (emsmta.exe)	Microsoft Exchange System Attendant
Microsoft Exchange IMAP4 (part of inetinfo.exe)	IIS Admin Service
Microsoft Exchange POP3 (part of inetinfo.exe)	IIS Admin Service
Simple Mail Transport Protocol (SMTP) (part of inetinfo.exe)	IIS Admin Service Event Log Service (The SMTP service is actually part of Windows 2003, but is enhanced during the installation of Exchange 2003.)
Network News Transport Protocol (NNTP) (part of inetinfo.exe)	IIS Admin Service Event Log (The NNTP service is actually part of Windows 2003, but is enhanced during the installation of Exchange 2003.)
Microsoft Exchange Event (events.exe)	Microsoft Exchange Information Store
Microsoft Exchange Routing Engine (part of inetinfo.exe)	IIS Admin Service
Microsoft Search (mssearch.exe)	NT LM Security Support Provider Remote Procedure Call (RPC)
Microsoft Exchange Site Replication Service (srsmain.exe)	Event log NT LM Security Support Provider Remote Procedure Call (RPC) Remote Procedure Call (RPC) Locator

TABLE 1.2: EXCHANGE 2003 DEPENDENCIES (*continued*)

EXCHANGE 2003 SERVICE	WINDOWS 2003 SERVICES (DEPENDENCIES)
	Server
	Workstation
Microsoft Exchange Connectivity Controller (lscntrl.exe)	Event log
	Microsoft Exchange System Attendant
Microsoft Exchange Connector for Lotus cc:Mail (ccmc.exe)	Event log
	Microsoft Exchange Information Store
Microsoft Exchange Connector for Lotus Notes (dispatch.exe)	Event log
	Microsoft Exchange Connectivity Controller
	Microsoft Exchange Information Store
Microsoft Exchange Directory Synchronization (dxa.exe)	Microsoft Exchange MTA Stacks
Microsoft Exchange Router for Novell GroupWise (gwrouter.exe)	Event log
MS Mail Connector Interchange (mt.exe)	Event log
	Microsoft Exchange MTA Stacks
MS Schedule Plus Free-Busy Connector (msfbconn.exe)	Event log
	Microsoft Exchange Information Store
Microsoft Exchange Connector for Novell GroupWise (dispatch.exe)	Event log
	Microsoft Exchange Connectivity Controller
	Microsoft Exchange Information Store
	Microsoft Exchange Router for Novell GroupWise
Microsoft Active Directory Connector (adc.exe)	Event log
	NT LM Security Support Provider
	Remote Procedure Call (RPC)
	Remote Procedure Call (RPC) Locator
	Server
	Workstation

If you worked with Exchange 2000, you may notice that the Exchange service dependencies have been “flattened” out a little more. Essentially, all Exchange services depend on the system attendant to be started; this improves startup time on clusters.

Other network services that must be available in order for Exchange 2003 to function properly include:

- ◆ Windows 2000 or 2003 domain controller
- ◆ Windows 2000 or 2003 Global Catalog server
- ◆ DNS server that will resolve service location records (SRV) for the Windows Active Directory forest, MX (mail exchanger) records, and A (address or host) records

Getting the Right Edition

Now that you understand some of the basics of Exchange 2003, it’s important that you understand the differences between the two editions of Exchange 2003: Exchange Server 2003 Standard Edition and Exchange Server 2003 Enterprise Edition. You must pick the right version to meet the needs of your organization. Table 1.3 lists available features and which edition of Exchange 2003 provides them.

TABLE 1.3: FEATURES WITH EXCHANGE SERVER 2003 AND EXCHANGE ENTERPRISE SERVER 2003

FEATURE	EXCHANGE SERVER 2003 STANDARD EDITION?	EXCHANGE SERVER 2003 ENTERPRISE EDITION?
Active/Active clustering (Clustering can be either Active/Active or Active/Passive depending on the number of nodes.)		√
Active Directory integration	√	√
Content indexing and searching	√	√
Database size larger than 16GB		√
Exchange Installable File System (ExIFS)	√	√
Exchange policies	√	√
Front-end/back-end configuration	√	√
Multiple mailbox stores		√
Multiple storage groups		√
Routing Group Connectors	√	√
SMTP Connector	√	√
Web storage system	√	√
Windows 2003 security	√	√
Workflow Designer for Exchange 2003	√	√
X.400 Connector		√

Upgrading between Editions

You can easily upgrade from Exchange 2003 Standard Edition to Enterprise Edition by simply running the Exchange 2003 Enterprise Server Setup program and choosing the Reinstall option.

However, you cannot “downgrade” from the Exchange 2003 Enterprise Server version to Exchange 2003 Standard Edition. If you must do this, consider installing an additional server using Exchange 2003 Standard Edition and moving the mailboxes over to that new server.

CLUSTERING

Microsoft server clustering is supported by Windows Server 2003. Applications, such as Exchange 2003, that are compatible with this clustering technology are able to benefit from clustering when they are installed on Windows 2003 platforms.

The Enterprise and Datacenter editions of Windows Server 2003 include clustering capabilities. Inter-server redundancy clustering is supported by the Microsoft Cluster Service (MSCS). MSCS supports clusters using up to eight physical servers or nodes; Exchange runs as a “virtual server” on one or more of the physical servers. Each virtual servers present itself to clients as a standalone server. Each virtual server can be “failed over” to another node if there are problems with the hardware on which the virtual server is running. Microsoft clustering services do not provide “load balancing” between the clustered nodes; clustering services provide higher availability by moving services from a failed node to an active node.

A server in a cluster uses ultra-high-speed inter-node connections and very fast, hardware-based algorithms to determine if a fellow server has failed. If a server fails, another server in the cluster can take over for it with minimal interruption in user access. It takes between one and two minutes for a high-capacity Exchange server cluster with a heavy load, around 5,000 users, to recover from a failure. With resilient e-mail clients such as Outlook 2003, client-server reconnections are transparent to users.

Clusters share disk storage, ideally SAN disk storage. More basic, standalone, sharable RAID boxes work fine too, as long as they can be connected on high-bandwidth links to multiple servers. It’s important to note that clusters alone do not provide any protection for data stored on disks. Such protection comes from the redundancy built into disk storage components.

In addition to providing a level of redundancy, clusters can also be used to implement network load balancing (NLB) strategies. NLB requires the installation of supporting Microsoft software. NLB is especially useful in Exchange environments with lots of incoming POP3, IMAP4, OWA, HTTP over RPC, and LDAP traffic. Network load balancing is available on all editions of Windows 2003 server.

A full discussion of MSCS clusters is beyond the scope of this book. Chapter 11, “Clustering and Other High Availability Stories” discusses MSCS from perspective of Exchange 2003. For more information on planning and deploying MSCS clusters, check out Mark Minasi’s *Mastering Microsoft Windows Server 2003* (Sybex, 2003) and Microsoft’s Windows Server 2003 website.

WHICH VERSION AM I RUNNING?

How can you tell which edition of Exchange 2003 you have? Check under the server’s Protocols container and see if you have an X.400 container. If so, that server is an Enterprise Edition server. You can also review your event logs and look for event 1217 from the MExchangeIS Mailbox Store. This indicates that the mailbox store has unlimited capacity; this is another indicator that you are running Exchange 2003 Enterprise Edition. This event is logged in the Application event log at Information Store startup.

Should I Do a Fresh Install or an Upgrade?

If you are running Exchange 5.5 or Exchange 2000 and need to be running Exchange 2003, you have to select an upgrade strategy. Although there are a number of upgrade options, your most important choice is whether to upgrade existing servers to Windows 2003 and Exchange 2003 or to install one or more new Windows 2003 servers and Exchange 2003 servers and then move mailboxes and public folders from the old to the new servers.

In my experience, a fresh install is the better choice. This is especially true with Windows 2003 and Exchange 2003. First, Windows and Exchange 2003 are much more secure on installation than earlier versions of these products. Upgrading to either product can leave a number of security holes that existed in earlier versions. Also, while you can upgrade Windows NT 4 to Windows 2003, you cannot do an in-place upgrade from Exchange 5.5 to Exchange 2003. You could do a 5.5-to-2000 upgrade and then upgrade to 2003, but that is a lot of work and fraught with a number of possibilities for failure. Better you should start fresh and then move Exchange objects from existing to new Windows/Exchange 2003 servers.

SECURE BY DEFAULT

Microsoft has received considerable criticism for the vulnerability of much of its software to security attacks. Much of this criticism was well deserved. The company's intentions were good: to make Microsoft products user-friendly and easier to manage. However, security breaches of Windows and other server and client products became so frequent and destructive that system managers and users alike were near willing to throw away all of their Microsoft software in favor of alternatives offered under operating systems such as Unix.

Under the banner of "Secure by Default," Microsoft promised that Windows 2003 and other server and client products would be much less vulnerable to security attacks. While security holes have been and will be found, Windows and Exchange 2003 are far more secure out of the box than their 2000 version brethren.

Although enhanced security is great, it can lead you to hair-tearing sessions as things that worked fine in earlier versions of Windows and Exchange refuse to work under the 2003 flavors of the products. This includes Windows passwords and certain Exchange-related functionality that is disabled by default.

Windows Passwords

Windows passwords control access to Windows resources, including Exchange server mailboxes and public folders. They play a very important role in overall Exchange system security. When establishing policies for passwords in your organization, you have to walk a fine line between the absurdly simple and the absurdly complex. Windows Server 2003 makes it very easy to see that fine line and the simple-to-complex continuum on either side of it.

Microsoft has graciously built a vision of password policies into Windows Server 2003. However, unless you're ready for them when you first encounter these policies, "gracious" is probably not the first word that will come to mind. When I created my first Windows 2003 user, it was all I could do to avoid swearing in a variety of languages. The password had to be three miles long, with numbers and uppercase and lowercase letters, and it expired almost before I created it.

You can make Windows 2003 password policies less absurdly complex. However, if you do so, don't revert to the almost-no-security-at-all default policies of pre-Windows 2003 operating systems. I

certainly endorse a strong password policy, but it is not the sort of thing that can be forced on a user community that is used to using “secret” as their password. Training on how to create and manage good passwords must be included in a stronger password policy.

Password policy modification requires the kind of extensive discussion that is beyond the scope of this book. You can find more on this subject and related issues in Mark Minasi’s book *Mastering Microsoft Windows Server 2003* (Sybex, 2003).

Services Disabled by Default

Table 1.4 presents a list of some of the Windows and Exchange 2003 services that are disabled by default that you might want to enable. Be careful here. Don’t enable a service unless you plan to use the functionality it supports.

TABLE 1.4: HOW TO ENABLE EXCHANGE-RELATED FUNCTIONS THAT ARE DISABLED BY DEFAULT

FUNCTIONALITY DISABLED	HOW TO ENABLE FUNCTIONALITY
POP3 Service	Change the Microsoft Exchange POP3 Service Startup Type from Disabled to Automatic
IMAP4 Service	Change the Microsoft Exchange IMAP Service Startup Type from Disabled to Automatic
Network News Transfer Protocol Service	Change the NNTP Service Startup Type from Disabled to Automatic

MORE REASONABLE DEFAULT SETTINGS

Overall, the default security settings for Windows and Exchange 2003 are far more reasonable from a management perspective than they were with earlier products. That doesn’t mean you won’t have to make changes. Just be sure you know what you’re doing before changing password restrictions or enabling a disabled service.

Other new defaults include message sizes and deleted item retention. A newly created Exchange 2003 organization will include a maximum incoming and outgoing message size of 10MB. New mailbox stores on servers automatically include a seven day deleted item retention time.

Going Native

Windows 2003 (Active Directory) and Exchange 2003 each have two modes in which the organization can operate: mixed mode and native mode. Windows 2003 native mode and Exchange 2003 native mode have no effect on each other; they are completely independent.

SWITCHING TO EXCHANGE NATIVE MODE

By default, the organization is in mixed mode, which allows Exchange 2003 to interoperate with Exchange 2000 and Exchange 5.5 servers. Several limitations are imposed on an Exchange 2003 organization that is operating in mixed mode with Exchange 5.5 servers, including:

- ◆ Windows 2003 Administrative groups are mapped directly to the Exchange 5.5 site architecture.
- ◆ Routing group membership can consist only of the servers that are in the administrative group containing that routing group.

- ◆ Exchange 2003 servers cannot be moved between routing groups.
- ◆ RPCs are used between Exchange 2003 servers and Exchange 5.5 servers.

Switching to Exchange 2003 native mode is easier if you have only Exchange 2000 and 2003 servers. You just need to be sure that all Exchange 2000 servers have been upgraded at least to Service Pack 3. Exchange 2003 native mode offers no additional features over Exchange 2000 native mode.

To switch the organization to native mode, check that all Exchange 5.5 and Exchange 2000 servers have been upgraded or removed from service, and remove all ADCs and site replication services (SRSs). (Make sure to remove the SRS from the Tools container in Exchange System Manager.) Then display the Exchange organization's properties using Exchange System Manager, and click the Change Mode button.

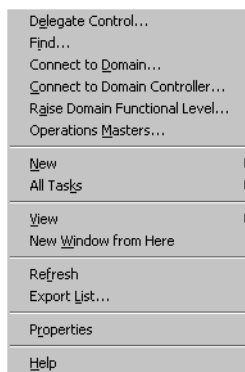
WARNING Changing to Exchange 2003 native mode cannot be reversed. So, be sure you will never need to introduce an earlier version of Exchange into your Exchange Organization.

Once you are in native mode, you will have a little more flexibility than you do in mixed mode. Some of its features include:

- ◆ Each administrative group can have multiple routing groups or no routing groups.
- ◆ A routing group can contain servers from any administrative group.
- ◆ Servers can be moved between routing groups.
- ◆ SMTP is used as the default message transport protocol between all Exchange 2003 servers.

SWITCHING TO WINDOWS 2003 NATIVE MODE

Windows 2003 modes apply at the domain level. You must run Windows 2003 in mixed mode until all of your NT 4 BDCs are either removed or upgraded to Windows 2003. You can run in Windows 2000 mixed mode if all of your servers are running Windows 2000 and Windows 2003. If you only have Windows 2003 servers, you can switch to Windows Server 2003 mode, which gives you the most functionality. To switch to native mode, you need the MMC snap-in Active Directory Domains and Trusts. (Refer to the upper-left corner in Figure 1.4.) Find your domain, right-click it, and select Raise Domain Functional Level.



Once you have switched to 2003 native mode, the following Active Directory-related functionality is added beyond what was possible under Windows 2000 native mode:

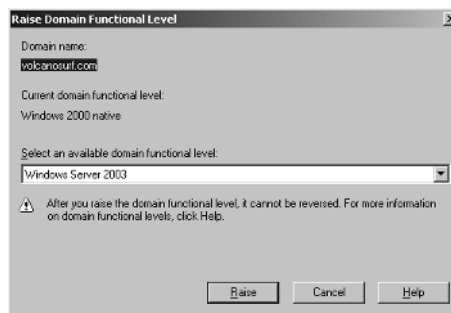
- ◆ Capability to rename DCs without a DCPROMO controller demotion and re-promotion
- ◆ Capability to deactivate schema classes or attributes that will never be needed
- ◆ Less network traffic because a member can be added to a group without the entire group membership being replicated across DCs
- ◆ Less network traffic with more efficient Global Catalog replication

You can also check what level the domain is currently set to by clicking the Raise Domain Functional Level selection on the domain's context menu, as shown in Figure 1.6.

WARNING *As with Exchange 2003, you cannot go to a lower mode level once you've switched to a higher level. So, be sure you are really ready to live with the level you choose, including being sure that you'll never again need to incorporate NT 4 or Windows 2000 servers in your domain.*

FIGURE 1.6

Setting or checking the domain functional level



Once all of your domains have been switched to Windows 2003 native mode, the next thing you should do is to switch the entire forest's functional level to Windows 2003. This is done on the properties of the console in Active Directory Domains and Trusts. Once the forest is at a Windows 2003 functional level, you will recognize additional benefits such as:

- ◆ Improved global catalog replication
- ◆ Linked value replication for group membership
- ◆ Tracking of last login date
- ◆ Capability to create a two way transitive trust that joins two forests (especially helpful if you need to support merged companies; works with Exchange 2003 management)

WHY YOU MAY NOT BE ABLE TO GO NATIVE

“Going native” has lots of advantages. However, sometimes circumstances may prevent your making the move. These include budgetary limitations, concern about having to support earlier versions of OS and Exchange software, and having to run third-party or internally developed software that isn’t ready for 2003.

Staff or software budget limitations might make it difficult for you to complete all of the changes required to move to native mode. The benefits of the additional functionality offered by native mode operation are often enough to break budget barriers. For example, if you’re dealing with a merger that includes two or more Exchange organizations, Windows 2003’s native mode cross-forest trusts could save your company a fortune.

As I mentioned in the last two warnings, you may be unsure about having to incorporate earlier versions of Windows or Exchange in a domain. Given the pace of corporate change, you have to remain flexible. In most cases, you shouldn’t have to worry about this issue. You can always set up another forest and domain for any new sub-2003 servers and move messages from the new servers to your Native 2003 environment.

Third-party or even home-grown software can also put the brakes on your moving to native mode. While your Windows and Exchange 2003 servers may be humming away just fine, you might still need the functionality in a certain piece of backup software or a particular gateway or management tool that isn’t yet 2003 compatible. A piece of in-house developed software might not work on the latest and greatest from Microsoft, because it uses hooks that are no longer available or it was developed with tools that no longer function on newer OS or Exchange systems. In such circumstances, “going native” has to wait.

Read Receipt

As I said earlier in this chapter, Exchange 2003 is really an evolutionary release of Exchange rather than an entirely new product. Administrators who are comfortable with Exchange 2000 will jump right on board with Exchange 2003. There are many enhancements that will make this upgrade worth while. They include:

- ◆ Dramatic improvements in Outlook Web Access
- ◆ Integrated mobile device support
- ◆ Improvements in scalability and larger clusters
- ◆ Exchange Server 2003 Standard Edition can now be a front-end server

Understanding the basic Exchange components is also an important skill for Exchange administrators. The basic components of Exchange 2003 includes:

- ◆ The system attendant
- ◆ The Information Store service
- ◆ The message transport components
- ◆ Exchange management components

One of the most dramatic changes from Exchange 5.5 was the message transport system. Now, all messages (local or remote) pass through the Advanced Queuing Engine which is a component of Internet Information Server (IIS).

TALKING ABOUT AN E-OLUTION

I got my first e-mail account from the University of Tennessee in the fall of 1981; it was on a DEC PDP-11. I only got the account because I was taking FORTRAN class for engineers. None of my professors had e-mail accounts; in fact most students had no clue what e-mail even was. Outages of this system were not uncommon and it was scheduled to be down fairly frequently. We took this in stride as neither the FORTRAN compiler nor the e-mail system were critical to my success. Nor were there very many people I could actually send an e-mail message. This did not change dramatically the next several years.

Today, all university students get a user account and mailbox when they are accepted for enrollment. Most high school students and even elementary school students have e-mail accounts. Both of my parents have e-mail accounts (a sure sign of the coming apocalypse) and my sister has just created a Yahoo! account for Gabrielle, my 3-year old niece.

My first job in the corporate world was a large California-based law firm. Our cc:Mail installation allowed us to communicate internally with anyone in the firm, but we still had no gateways to the outside world except for our vendor and other offices.

E-mail has exploded as a method of communication among friends, peers, managers, customers, and vendors. I spend far more time using e-mail today than I did sending e-mail than I ever did on the telephone. I can communicate with people all over the world almost instantly. I can send thoughts, documents, pictures and queries to someone in Budapest, Bakersfield, Bangkok, or Baghdad with the same ease.

Business people, executives, managers, and even administrative staff are checking their e-mail from home, Internet cafés, airport kiosks, and wireless devices. Military commanders deploy their Exchange servers into the field with them. One manager recently commented to me that e-mail allowed her business to be five times as agile and responsive as it had ever been in the past. 80 percent of 387 business people by the META Group (www.metagroup.com) believe that e-mail is more important than the telephone for communications with customers, coworkers, and business partners due to the ease of rapidly disseminating information to multiple parties and the written record of communication.

The Radicati Group (www.radicati.com) estimates that as of 2004 there are 900 million active mailboxes worldwide; of those 445 million of those are corporate mailboxes. In 2001, the Radicati Group estimated that 44 percent of the corporate messaging market used Microsoft Exchange. Radicati expects the growth to exceed 1.5 billion mailboxes by 2007. IDC estimates that as of 2002 Microsoft had sold 115 million Exchange seats; this is independent of product bundles such as Small Business Server and OEM sales of Exchange such as Lockheed Martin's DMS (Defense Messaging System) version.

continued on next page

TALKING BOUT AN E-VOLUTION *(continued)*

A November 2002 survey conducted by the Gartner Group (www.gartner.com) asked the question “Which technologies deployed in the last three years have given you the best payback?” E-mail was the number one choice.

What does all of this mean for us, the e-mail dudes and dudettes? E-mail has become a “business critical” for much of the world. The e-mail server is no longer an afterthought in most organizations. The e-mail administrator is now visible on both management and the IT department’s radar. Ensuring a stable e-mail environment, good functionality, and acceptable availability is a key part of our jobs because these things are now key to the success of our organizations.

Both Windows Server and Microsoft Exchange Server have evolved along with our own roles in an organization. Where Exchange 4.0 required continual “loving care” to keep it running, Exchange 2003 can reliably and efficiently support thousands of simultaneous users in any type of environment. I hope that this book will provide you the guidance necessary to meet the growing expectations of e-mail users.

