Chapter 1

Introduction to Messaging Administration

Congratulations! You may have just stepped into the most important job in your entire organization. No, not the CEO, the number one sales person, the janitor, the person that makes sure there is fresh coffee made, or even the one who prints out the monthly TPS reports. I'm talking about the person that keeps the e-mail system running.

Now, do I honestly believe the e-mail system is the most important component of an organization's information technology services? On a dollar-per-dollar basis, an organization's line-of-business applications (such as order entry, accounting, customer relationship management, shipping, billing, and others) are probably the most important types of applications when it comes to the actual value provided. However, e-mail is often one of the most visible services (if not the single most visible service) that IT professionals provide; most organizations have become dependent on ''soft'' information to run their business. As a result, users have in turn developed an attachment to e-mail that goes beyond the hard value of the information it contains. If there's a problem with e-mail, that affects users' confidence in their ability to do their jobs and their confidence in IT.

There is not much in this chapter that is specific to Exchange Server 2007, so an experienced e-mail administrator may want to proceed directly to Chapter 2, ''Designing a New Exchange 2007 System'' or Chapter 3, ''Introducing Exchange Server 2007.'' However, everyone needs to start somewhere and if you are new to the job or need a refresher, this chapter is for you!

In this chapter, I attempt to provide a primer on some of the issues that you'll need to know in order to maximize the coverage of Exchange provided in the rest of the book. I hope that this chapter will serve as a good introduction to e-mail administration and prepare you to put Exchange Server 2007 in to the proper context.

In this chapter you learn about:

- ◆ The basics of e-mail
- ◆ Things every e-mail administrator should know
- ◆ A day in the life of the e-mail administrator
- ◆ E-mail protocols and services
- ◆ What Exchange Server is

Introducing E-mail

Okay, I agree that ''Introducing E-mail'' does seem like a pretty silly header, because most everyone within 50 miles of an Internet connection has an e-mail address. Does a simple concept such as sending text and attachments from one person to another really need an explanation? Well, no, but ''What Does E-mail Do For Your Users, Or For Your Organization For That Matter?'' is too long.

Sure, sending simple text e-mail and file attachments is the most basic function, but e-mail systems (the client and/or the server) may also perform the following important functions:

- Act as a personal information manager, providing storage for and access to personal calendars, personal contacts, to-do and task lists, personal journals, and chat histories.
- Provide the user with a single ''point of entry'' for multiple types of information such as voicemail, faxes, and electronic forms.
- Provide shared calendars, departmental contacts, and other shared information.
- Enable users to send faxes from their desktop to outside of their organization.
- Receive notifications of ''work flow'' processes such as finance/accounting activities, IT events (server status information), and more.
- Allow users to access their ''e-mail data'' through a variety of means including clients running on Windows computers, Apple computers, Unix systems, web browsers, and mobile phones.
- Perform records management and enable long-term storage of important information or information that must be archived.

These are just a few of the types of things that an e-mail system may provide to the end user either via the client interface or as a result of some function running on the server.

A Brief History of E-mail

If you're currently responsible for electronic messaging in your organization, no one has to tell you about the steadily expanding use of e-messaging. You know it's happening every time you check the storage space on your disk drives or need an additional tape to complete the backup of your mail server. This section discusses some of the aspects of electronic mail and the ever-changing nature of e-mail. Even experienced Exchange Server administrators may want to review this section to better understand how your users and requirements are evolving.

Over the past ten years, the number of e-mail addresses has grown significantly. The technology research company International Data Corporation (IDC) estimated that in 2002, the number of e-mailboxes worldwide was more than 500 million. As of 2006, the Radicati Group estimates that there are now more than 1.5 billion e-mail accounts worldwide, accounting for more than 135 billion e-mail messages per day.

Slow Initial Adoption

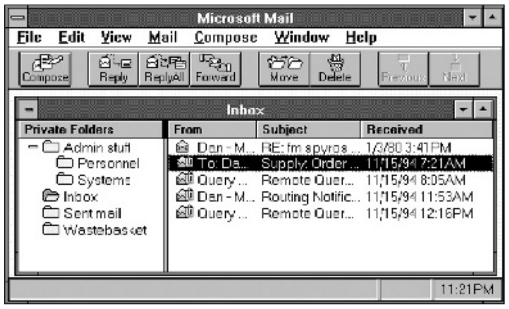
I had my first e-mail box on a Digital Equipment Corporation (DEC) VAX/VMS system in 1980 and used e-mail continually through the 1980s. When I said ''e-mail'' my friends all said ''e-what?'' Early e-mail systems were looked at by organizations as a luxury or an option rather than an important part of an organization's daily work processes.

E-mail systems really began to take hold in the corporate world around 1988. More rapid adoption started in the mid 1990s. What happened to change these organizations' minds? Well, a few things:

- More and more organizations moved from mainframe-based systems to PC-based systems.
- Electronic information processing for even small and medium-sized businesses became more affordable.
- E-mail clients and e-mail servers included more and more features and capabilities that were attractive to users and management.
- The Internet served as an ideal way to link organizations together, and as more organizations became connected to the Internet people had more options for who they communicated with.

Improving the Interface

Certainly e-mail systems have come a long, long way since the first mainframe and mini-computer systems from more than 30 years ago. Even the primitive text-based systems like cc:Mail, Microsoft Mail, WordPerfect Office, and Da Vinci eMail that first appeared on local area networks in the late 1980s are almost unrecognizable ancestors when compared with a modern system based on Exchange Server 2007 and Outlook 2007. Early e-mail clients were text-based and usually did not have any features other than the ability to read and send e-mail (no personal information management). Looking at an early version of Microsoft Mail (see Figure 1.1), it seems downright sparse.



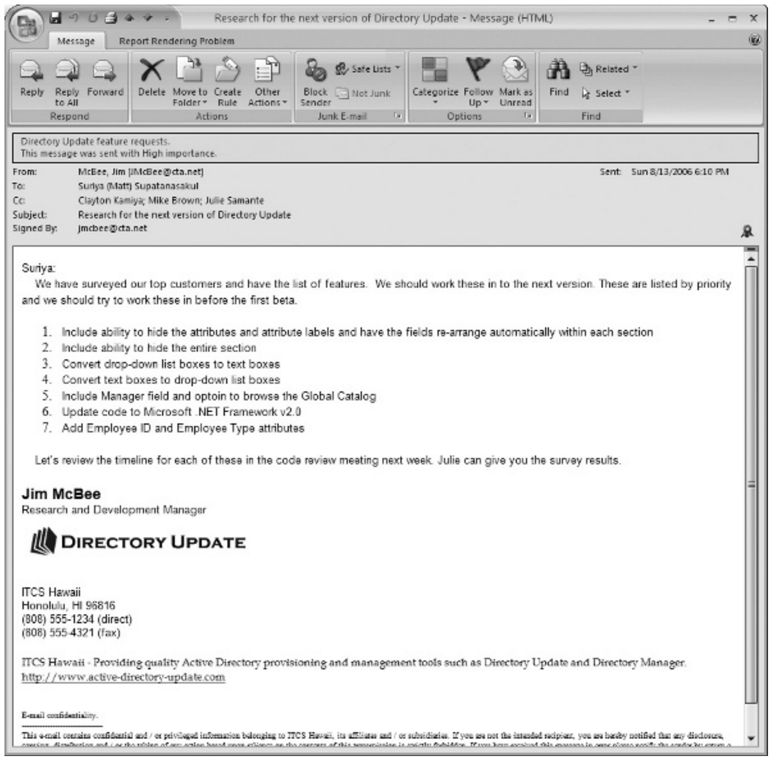
From	Subject	Received
Dan - M...	RE: fm spyros ...	1/3/80 3:41PM
To: Da...	Supply Order ...	11/15/94 7:21AM
Query ...	Remote Quer...	11/15/94 8:05AM
Dan - M...	Routing Notific...	11/15/94 11:53AM
Query ...	Remote Quer...	11/15/94 12:16PM

FIGURE 1.1
Early e-mail clients had few features.

Modern e-mail clients are much more feature-rich than their predecessors. One interesting evolution is the features available when creating a message. New clients allow for the creation of much more complex e-mail messages than in the past. What does ''more complex'' mean? Well, take as an example the message shown in Figure 1.2. This Outlook 2007 message is formatted with fonts, a numbered list, a substantial message signature/disclaimer, and a corporate logo in the signature; all of this is formatted using HTML or rich text so that the message is viewable by any web-based mail system or HTML-compatible client. Finally, the message is digitally signed and authenticated with a digital signature.

Messages formatted with rich text or HTML, containing disclaimers and digital signatures, can help us to communicate more effectively. Therefore, organizations depend far more on e-mail today than they did even five years ago, and their users send even more mail than in the past.

In addition to regular e-mail messages, users are sending scheduling requests, contact items, forms-enabled e-mail messages, and more. Each of these increases the complexity of the messaging system and also an organization's dependency on it.



Directory Update feature requests.
This message was sent with High importance.

From: McBee, Jim [JMcBee@cta.net] Sent: Sun 8/13/2006 6:10 PM
To: Suriya (Matt) Supatanasakul
Cc: Clayton Kamiya; Mike Brown; Julie Samante
Subject: Research for the next version of Directory Update
Signed By: jmcbee@cta.net

Suriya:
We have surveyed our top customers and have the list of features. We should work these in to the next version. These are listed by priority and we should try to work these in before the first beta.

1. Include ability to hide the attributes and attribute labels and have the fields re-arrange automatically within each section
2. Include ability to hide the entire section
3. Convert drop-down list boxes to text boxes
4. Convert text boxes to drop-down list boxes
5. Include Manager field and optoin to browse the Global Catalog
6. Update code to Microsoft .NET Framework v2.0
7. Add Employee ID and Employee Type attributes

Let's review the timeline for each of these in the code review meeting next week. Julie can give you the survey results.

Jim McBee
Research and Development Manager

DIRECTORY UPDATE

ITCS Hawaii
Honolulu, HI 96816
(808) 555-1234 (direct)
(808) 555-4321 (fax)

ITCS Hawaii - Providing quality Active Directory provisioning and management tools such as Directory Update and Directory Manager.
http://www.active-directory-update.com

E-mail confidentiality.

This e-mail contains confidential and / or privileged information belonging to ITCS Hawaii, its affiliates and / or subsidiaries. If you are not the intended recipient, you are hereby notified that any disclosure,

FIGURE 1.2
A typical modern e-mail message

The message shown in Figure 1.2 it is 30KB in size, but has only a few hundred bytes of actual message content, including the recipient information.

Attachments

As if that weren't complex enough, many e-mail messages contain *attachments* — word-processing, spreadsheet, and other files that you can attach to messages. Using attachments is a simple way to move files to the people who need to see them. They also tend to gobble up disk space extremely fast!

Sure, you could send your files on disk or tell people where on the network they can find and download them. But e-mail attachments let you make the files available to others with a click of their mouse buttons. Recipients just double-click an icon and the attachment opens in the original application that produced it (always assuming your correspondent has access to an application or software compatible with the attachment). Using attachments offers the added advantage of putting the files and accompanying messages right in the faces of those who need to see them. This leaves less room for excuses such as ''I couldn't find/open that network folder'' or ''The dog ate the disk.''

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

Shared Files

Office 2003 and Office 2007 offer two neat ways to avoid this problem. First, they let you insert a link to a file. When you open the file, you're opening the file the link points to. If the file is changed, you see the changed file. Second, Office lets you attach a file to a message and set a shared folder where an updatable version of the file is stored. When the copy attached to the user's e-mail is updated, these updates can be incorporated into the shared copy of the file. This option allows broader access to the file than a link.

The use of portals such as Microsoft Office SharePoint Server is becoming increasingly commonplace in organizations as they look for better ways to store, find, and manage the data that their users are producing. As new versions of Outlook and Outlook Web Access offer better integration with SharePoint and provide an alternate to using e-mail for attachments, messaging administrators may start being able to reduce or even remove the impact of attachments in e-mail.

About Messaging Services

Electronic messaging is now far more than e-mail. Together, Exchange Server 2007 and its clients perform a variety of messaging-based functions. These include e-mail, message routing, scheduling, and support for several types of custom applications. Together these features are called messaging services.

How Messaging Services Are Used

Certainly, e-mail is a key feature of any messaging system, and the Outlook Calendar is far better than previous versions of Microsoft's appointment and meeting-scheduling software. Outlook 2007 together with Exchange 2007 introduces even more improvements. Figures 1.3 and 1.4 show the Outlook 2007 client Inbox and Calendar in action.

Figure 1.5 shows the new Outlook Web Access 2007 web browser client that you can use with Exchange Server 2007.

E-mail clients are exciting and sexy, but to get the most out of Exchange Server 2007, you need to throw away any preconceptions you have that messaging systems are only for e-mail and scheduling. The really exciting applications are not those that use simple e-mail or scheduling, but those that are based on the routing capabilities of messaging systems. These applications bring people and computers together for improved collaboration.

How Messaging Servers Work

At the core of any messaging system, you will find a common set of basic functions. These functions may be implemented in wildly different ways depending on the vendor or even the version of the product. Exchange Server has evolved dramatically over the past 13 years and its current architecture is almost nothing like the Exchange Server from 1996. Common components of most messaging systems include:

- A message transport system that moves messages from one place to another. Examples include the Simple Mail Transport Protocol (SMTP) or Remote Procedure Calls (RPCs).
- A message storage system that stores messages until a user can read or retrieve them. Messages may be stored in a client/server database, a shared file database, or even in individual files.

- A directory service that allows a user to look up information about the mail system's users such as a user's e-mail address.
- A client access interface on the server that allows the clients to get to their stored messages. This might include a Web interface, a client/server interface (such as RPCs), or the Post Office Protocol (POP).
- The client program that allows users to read their mail, send mail, and access the directory. This may include Outlook, Outlook Web Access, or iPhones.

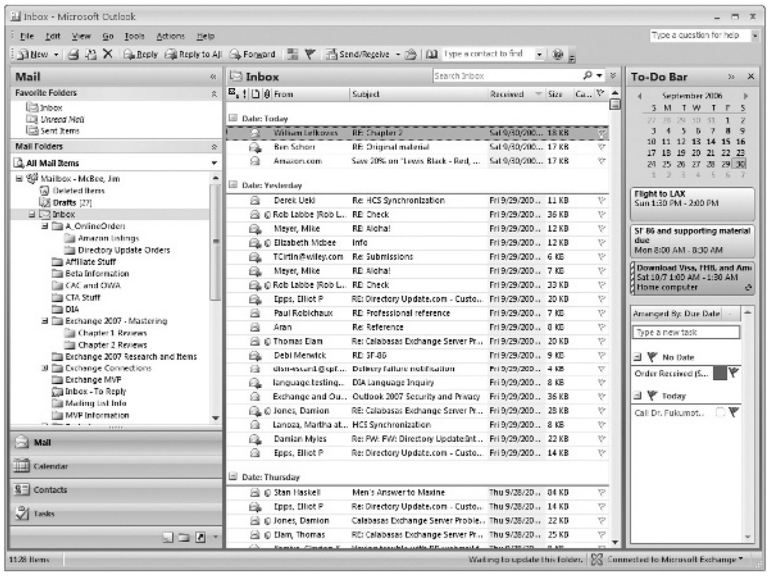


FIGURE 1.3
The Outlook 2007 client Inbox

Date: Today

From	Subject	Received	Size
William Lefkovics	RE: Chapter 2	Sat 9/30/200...	18 KB
Ben Schorr	RE: Original material	Sat 9/30/200...	17 KB
Amazon.com	Save 20% on "Lewis Black - Red, ...	Sat 9/30/200...	17 KB

Date: Yesterday

From	Subject	Received	Size
Derek Ueki	Re: HCS Synchronization	Fri 9/29/200...	11 KB
Rob Labbe [Rob L...	RE: Check	Fri 9/29/200...	36 KB
Meyer, Mike	RE: Aloha!	Fri 9/29/200...	12 KB
Elizabeth Mcbee	Info	Fri 9/29/200...	12 KB
TCirtin@wiley.com	Re: Submissions	Fri 9/29/200...	6 KB
Meyer, Mike	RE: Aloha!	Fri 9/29/200...	7 KB
Rob Labbe [Rob L...	RE: Check	Fri 9/29/200...	33 KB
Epps, Elliot P	RE: Directory Update.com - Custo...	Fri 9/29/200...	20 KB
Paul Robichaux	RE: Professional reference	Fri 9/29/200...	7 KB
Aran	Re: Reference	Fri 9/29/200...	8 KB
Thomas Elam	Re: Calabasas Exchange Server Pr...	Fri 9/29/200...	20 KB
Debi Merwick	RE: SF-86	Fri 9/29/200...	9 KB
dlsn-escard1@spf....	Delivery failure notification	Fri 9/29/200...	4 KB
language.testing...	DIA Language Inquiry	Fri 9/29/200...	8 KB
Exchange and Ou...	Outlook 2007 Security and Privacy	Fri 9/29/200...	36 KB
Jones, Damion	RE: Calabasas Exchange Server Pr...	Fri 9/29/200...	28 KB
Lanoza, Martha at...	HCS Synchronization	Fri 9/29/200...	8 KB
Damian Myles	Re: FW: FW: Directory Update&nt...	Fri 9/29/200...	22 KB
Epps, Elliot P	Re: Directory Update.com - Custo...	Fri 9/29/200...	14 KB

Date: Thursday

From	Subject	Received	Size
Stan Haskell	Men's Answer to Maxine	Thu 9/28/20...	84 KB
Epps, Elliot P	Re: Directory Update.com - Custo...	Thu 9/28/20...	14 KB
Jones, Damion	Calabasas Exchange Server Proble...	Thu 9/28/20...	22 KB
Elam, Thomas	RE: Calabasas Exchange Server Pr...	Thu 9/28/20...	25 KB

Flight to LAX
Sun 1:30 PM - 2:00 PM

SF 86 and supporting material due
Mon 8:00 AM - 8:30 AM

Download Visa, FHB, and Am
Sat 10/7 1:00 AM - 1:30 AM
Home computer

No Date
Order Received (S...

Today
Call Dr. Fukumot...

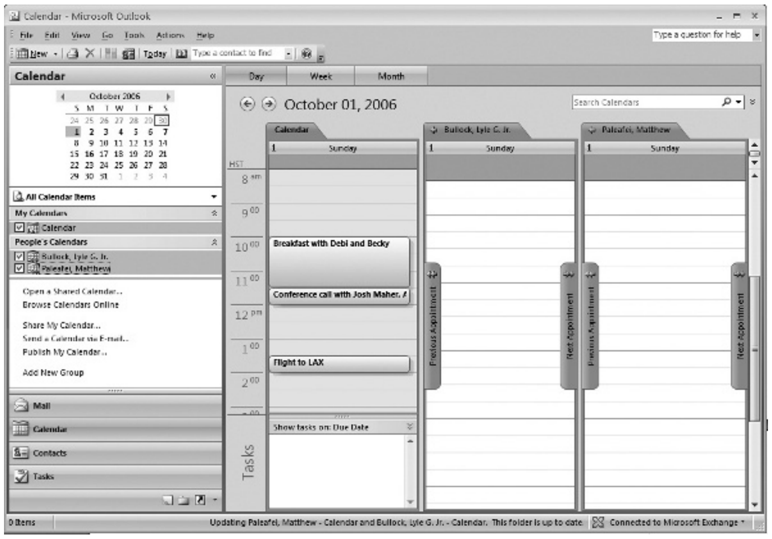


FIGURE 1.4
The Outlook 2007 client Calendar

October 01, 2006

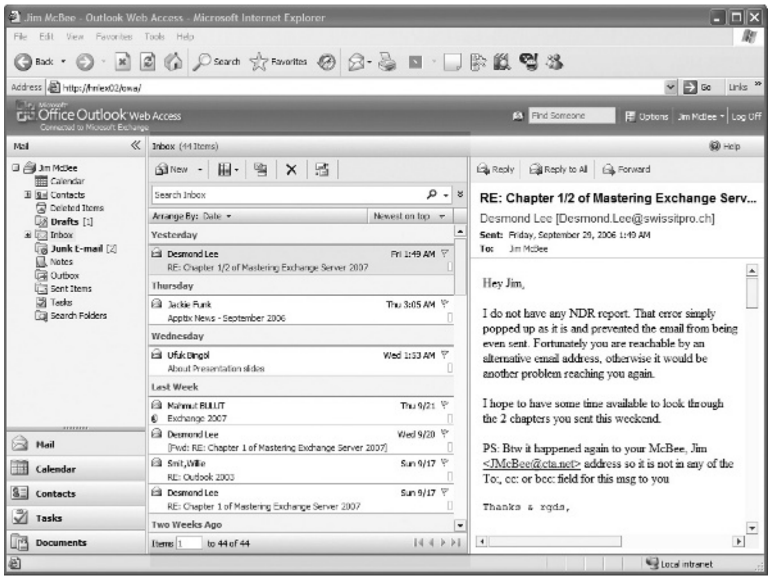
Breakfast with Debi and Becky

Conference call with Josh Maher. A

Flight to LAX

Updating Palcafel, Matthew - Calendar and Bullock, Lyle G. Jr. - Calendar. This folder is up to date.

Figure 1.5
Outlook Web Access web browser accesses mail stored on an Exchange Server 2007



Yesterday
Desmond Lee Fri 1:49 AM
RE: Chapter 1/2 of Mastering Exchange Server 2007
Thursday
Jackie Funk Thu 3:05 AM
Apptix News - September 2006
Wednesday
Ufuk Bingol Wed 1:53 AM
About Presentation slides
Last Week
Mahmut BULUT Thu 9/21
Exchange 2007
Desmond Lee Wed 9/20
[Fwd: RE: Chapter 1 of Mastering Exchange Server 2007]
Smit,Willie Sun 9/17
RE: Outlook 2003
Desmond Lee Sun 9/17
RE: Chapter 1 of Mastering Exchange Server 2007
Two Weeks Ago

RE: Chapter 1/2 of Mastering Exchange Serv...
Desmond Lee [Desmond.Lee@swissitpro.ch]
Sent: Friday, September 29, 2006 1:49 AM
To: Jim McBee

Hey Jim,

I do not have any NDR report. That error simply popped up as it is and prevented the email from being even sent. Fortunately you are reachable by an alternative email address, otherwise it would be another problem reaching you again.

I hope to have some time available to look through the 2 chapters you sent this weekend.

PS: Btw it happened again to your McBee, Jim <JMcBee@cta.net> address so it is not in any of the To:, cc: or bcc: field for this msg to you

Thanks & rgds,

Working in tandem with real-time interactive technologies, electronic messaging systems have already produced a set of wildly imaginative business, entertainment, and educational applications with high payoff potential. All of this action, of course, accelerates the demand for electronic messaging capabilities and services.

Most organizations that deploy an e-mail system usually deploy additional components from their e-mail software vendor or third parties that extend the capabilities of the e-mail system or provide required services. These include:

- Message hygiene systems that help reduce the likelihood of a malicious or inappropriate message being delivered to a user
- Backup and disaster recovery solutions
- Message archival software to allow for the long-term retention and indexing of e-mail data
- Electronic forms routing software that may integrate with accounting, order entry, or other line-of-business applications
- Mail gateways to allow differing types of mobile phones to access the mail server such as BlackBerry devices or Palm-based mobile phones
- E-mail security systems that improve the security of e-mail data either while being transferred or while sitting in the user's mailbox

Application Networking Models

The technology industry has overused the term *client/server* to the point where it is almost meaningless. To put it simply, there are two kinds of networked applications: shared-file and client/server. The typical Exchange Server and Outlook deployment is a client/server messaging system and always has been. However, for people just getting involved in Exchange Server deployments, these concepts should be reviewed. It is also helpful to note that Exchange Server and Outlook are completely separate components. Although Outlook is the most popular (and feature-rich) client for Exchange Server, some organizations deploy Exchange Server entirely for Web or POP3 clients

Shared-File Applications

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

Microsoft's first e-mail product, Mail for PC Networks, was a shared-file application. You ran a Windows, OS/2, DOS, or Macintosh client application, which sent and received messages by accessing files on a Microsoft Mail for PC Networks post office that resided on a network file server. The front-end application and your PC did all the work; the server was passive. Figure 1.6 shows a typical Microsoft Mail for PC Networks setup.

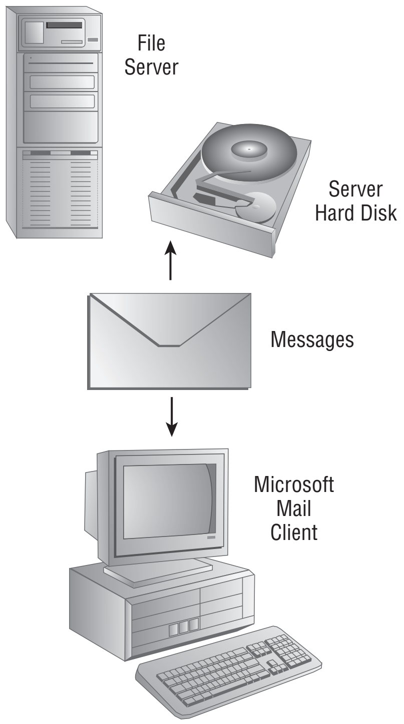


Figure 1.6
Microsoft Mail for PC Networks is a typical shared-file electronic messaging system.

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

- Changing the underlying structure of the server file system is difficult because you have to change both the server and the client.
- System security is always compromised because users must have read and write permissions for the whole server file system, which includes all other users' message files. Things are so bad that in some cases a naive or malicious user can actually destroy shared-file system databases.

- Network traffic is high because the client application must constantly access indexes and hunt around the server's file system for user messages.
- Because the user workstation writes directly to shared files, the server-based files can be destroyed if workstation hardware or software stops functioning for some unexpected reason.
- Often the client program would open these shared files and lock them for use. This frequently prevented important data files from being backed up.

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

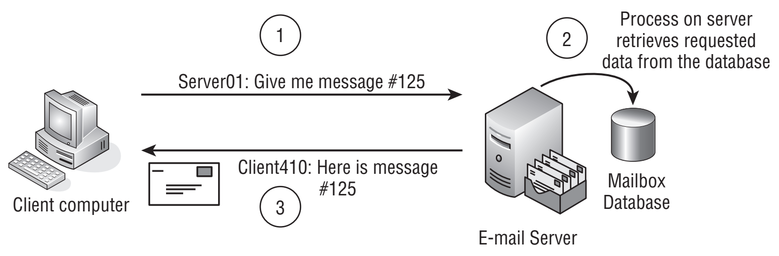
Client/Server Applications

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

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

A very simplified example of this is shown in Figure 1.7. In step 1, the client requests a specific e-mail message from the server. Then, in step 2, the server responds by opening the appropriate database, searching through the database, retrieving the message, and sending it back to the client in step 3. Although the database might be hundreds of gigabytes in size, this operation usually takes less than a second.

Figure 1.7
A simple client/server messaging system


1
2
Process on server retrieves requested data from the database
Server01: Give me message #125
Client410: Here is message #125
3
Client computer
Mailbox Database
E-mail Server

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

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

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

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

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

The good news is that Microsoft has done a lot to reduce the management load and to make it easier for someone who isn't a computer scientist to administer an Exchange system. I've looked at many client/server messaging systems, and I can say without any doubt that Exchange is absolutely the easiest to administer, even in its slightly more complex 2007 implementation. Exchange Server 2007 includes both a graphical user interface (GUI) and a management shell that organizes the processes of management very nicely. With these interfaces, you can do everything from adding users to assessing the health of your messaging system.

Things Every E-mail Administrator Should Know

The information in this section is something that I often find even my own e-mail administrators and help desk personnel are not aware of. Sometimes the most important skill any technology administrator has is not a specific knowledge of something, but generic knowledge that they can use to quickly find the right answer.

Finding Answers

This topic deserves special attention. One of my jobs is working in Tier 3 support for a large organization. The thing I respect the most about the administrators that actually run the system and handle the trouble tickets is if they have done their homework prior to coming to me with a problem.

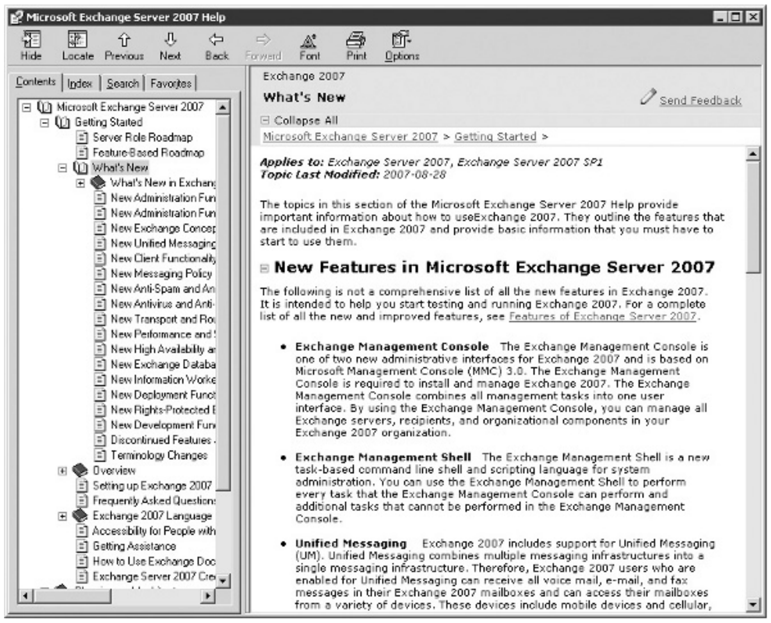
Too often techies tend to make up an answer when they are not sure about something. Don't do that! When you are asked a question that you don't know the answer to, it is okay to say you don't know the answer. But follow that up by indicating that you will find the answer. Knowing the right resources (where to get answers) is therefore just as important as the technical knowledge it takes to implement the answer.

Helpful Resources

Exchange has to be one of the most documented and discussed products (short of maybe Windows) that Microsoft produces. This means that most of the questions that I have about Exchange Server I can usually answer with the right Google search or looking in the right place. The most obvious place to start when you have a problem or a question is to perform an Internet search, but there are many other resources.

Exchange Server documentation There is a world of information available to you out on the Internet, but let's start right on the local hard disk of your Exchange Server or anyplace you have installed the admin tools. Microsoft has done an excellent job of providing better and better documentation for Exchange Server over the past few years. The Exchange Server 2007 documentation is comprehensive and so readable you will wonder if it is really from Microsoft. Figure 1.8 shows an example of the Exchange 2007 documentation. Look for the following file: `C:\Program Files\Microsoft\Exchange Server\Bin\ExchHelp.chm` or run it from the Microsoft Exchange Server 2007 folder on the Start menu.

FIGURE 1.8
Viewing the Exchange 2007 documentation



Exchange 2007

What's New

Microsoft Exchange Server 2007 > Getting Started >

Applies to: *Exchange Server 2007, Exchange Server 2007 SP1*
Topic Last Modified: *2007-08-28*

The topics in this section of the Microsoft Exchange Server 2007 Help provide important information about how to useExchange 2007. They outline the features that are included in Exchange 2007 and provide basic information that you must have to start to use them.

New Features in Microsoft Exchange Server 2007

The following is not a comprehensive list of all the new features in Exchange 2007. It is intended to help you start testing and running Exchange 2007. For a complete list of all the new and improved features, see Features of Exchange Server 2007.

- **Exchange Management Console** The Exchange Management Console is one of two new administrative interfaces for Exchange 2007 and is based on Microsoft Management Console (MMC) 3.0. The Exchange Management Console is required to install and manage Exchange 2007. The Exchange Management Console combines all management tasks into one user interface. By using the Exchange Management Console, you can manage all Exchange servers, recipients, and organizational components in your Exchange 2007 organization.
- **Exchange Management Shell** The Exchange Management Shell is a new task-based command line shell and scripting language for system administration. You can use the Exchange Management Shell to perform every task that the Exchange Management Console can perform and additional tasks that cannot be performed in the Exchange Management Console.
- **Unified Messaging** Exchange 2007 includes support for Unified Messaging (UM). Unified Messaging combines multiple messaging infrastructures into a single messaging infrastructure. Therefore, Exchange 2007 users who are enabled for Unified Messaging can receive all voice mail, e-mail, and fax messages in their Exchange 2007 mailboxes and can access their mailboxes from a variety of devices. These devices include mobile devices and cellular,

You can also download updated versions of the `ExchHelp.chm` from the following URL:

`http://technet.microsoft.com/en-us/exchange/bb330843.aspx?wt.svl=2007resources`

Exchange Server release notes Another good resource for ''I wish I had known that'' types of things is the release notes. You should be able to find a link to the release notes here:

`C:\Program Files\Microsoft\Exchange Server\RelNotes.htm`

Exchange Server TechNet forums If you have a question on which you have done your due diligence in searching and researching the problem, but you don't have an answer, it is time to ask the world. A good place to start is the Microsoft TechNet forums. You can find the Exchange server section here:

`http://forums.microsoft.com/technet/default.aspx?siteid=17`

When you post your question, please do not post a question like ''Exchange is giving me an error.'' Post the exact error message and any error codes you are seeing. Also indicate, at minimum, what version of the software you are using (including service pack), the role of the server, and what operating system.

MSExchange.Org website One of the best sites on the Internet for free, easy-to-access content about Exchange Server is `www.msexchange.org`. The articles are written by Exchange gurus from all over the world and are usually in the form of easy-to-read and easy-to-follow tutorials. There is also a forums section where you can post questions or read other people's questions.

Calling for Support

If your system is down or your operations are seriously hindered and you don't have a clue what to do next, it is time to call in the big guns. Sure, you should do some Internet searches to try and resolve your problem, but Internet newsgroups and forums are not the place to get support for ''business critical'' issues.

Microsoft Product Support Services (PSS) is Microsoft's technical support organization. Its home page is `http://support.microsoft.com`. Professional support options (ranging from peer-to-peer support to telephone support) can be found at the following URL:

```
http://support.microsoft.com/default.aspx?scid=fh;en-us;Prodoffer11a
```

Telephone support is currently US$245 per incident for supporting during normal business hours; though this may seem expensive, believe me, when an Exchange server is down and the users are burning you in effigy in the company parking lot, $245 for business hours support is cheap.

When you call and get a support technician on the phone, don't be surprised or offended if he or she starts at the beginning and asks you a lot of elementary questions. They have to double-check everything you have done before they can look into more advanced problems. Once or twice, one of these basic questions has helped me locate a problem that I was convinced was more complicated than it really was.

I always encourage people to call PSS if they truly need assistance. But PSS engineers are not mind-readers, nor do they know every bit of Exchange code. You will do both yourself and the PSS engineer a big favor if you have all of your ducks in a row before you call. Do the following before you call:

- Attempt a graceful shutdown and restart of the server in question, if applicable.
- Perform a complete online backup if possible; if not, do a complete offline backup.
- Have a complete, documented history of everything you have done to solve the problem. At the first sign of trouble, you should start keeping a chronological log of the things you did to fix the problem.
- Find out if you are allowed to initiate support sessions with remote support personnel through a tool like Live Meeting or WebEx.
- Be at a telephone that is physically at the server's console, or be in a place where you can access the server remotely via the Remote Desktop Client. Your support call will be very brief if you cannot immediately begin checking things for the PSS engineer.
- Have the usernames and passwords that will provide you the right level of administrative access. If you don't have those, have someone nearby who can log you in.
- Save copies of the System and Application event logs. Be prepared to send these to PSS if requested.
- Download a copy of MPSReports (for more information see `http://exchangeorg.net/archive/2004/06/30/mpsreports-exchange-edition.aspx`) and run it on your Exchange server. Be prepared to send the resulting report to PSS if requested.
- Know the location of your most recent backup and how to access it when needed.

- Keep copies of all error messages. Don't paraphrase the message. Screen captures work great in this case. Pressing Alt+Print Scrn and pasting into a WordPad document works great, too. I usually create a document with screen captures along with notes of what I was doing when I saw each message.

Be patient; telephone support is a terribly difficult job. A little kindness, patience, and understanding on your part will most certainly be returned by the PSS engineer.

A Day in the Life of the E-mail Administrator

I know and work with a *lot* of e-mail administrators and I can honestly say that no two people have the same set of tasks required of them. Your CEO, Director of Information Technology, or even your supervisor is going to ask you to pull rabbits out of your hat, so don't expect each day to be the same as the last one. (And invest in some rabbits.) Keep up with your technology and supporting products so that you can be ready with answers or at the very least intelligent responses to questions.

Daily Administrative Tasks

So, what are some typical tasks that you may perform as part of your duties as an e-mail administrator? These tasks will really depend on the size of your organization, the number of administrators you have running your Exchange organization, and how administrative tasks are divided up.

- **Recipient management** is certainly the biggest day-to-day task involved with Exchange for medium and large organizations where there is lots of turn-over. Recipient management tasks may include:
  - Assigning a mailbox to a user account
  - Creating mail-enabled contacts
  - Creating and managing mail groups
  - Managing mail-enabled object properties such as users' phone numbers, assigning more e-mail addresses to a user, or adding/removing group members.
- **Basic monitoring tasks** to ensure that your Exchange servers are healthy and functioning properly:
  - Checking queues for stalled messages
  - Verifying that there is sufficient disk space for the databases and logs
  - Making sure that the message hygiene system is functioning and up-to-date
  - Running and verifying daily backups
  - Reviewing the System and Application event logs for unusual activity, errors, or warnings
- **Daily troubleshooting tasks** include the following:
  - Reviewing non-delivery report messages and figuring out why some mail your users are sending might not have been delivered.
  - Looking up errors and warnings that show up in the Application and System event logs to determine if they are serious and warrant corrective action.

- **Security related tasks** are sometimes performed daily but others are performed only weekly or monthly:
  - Looking at server and service up-times to ensure that servers are not rebooting unexpectedly.
  - Reviewing the event logs for warnings that may indicate users are inappropriately accessing other users' data.
  - Saving the Web and SMTP and connectivity logs.
- **E-mail client administration tasks** include the following:
  - Helping users get Outlook connected and configured properly.
  - Diagnosing problems Windows Mobile devices, BlackBerry devices, or iPhones.

Communicating with Your Users

Communicating with your users is probably one of the most important things you do. Keeping your users informed and delivering good customer service is almost as important as delivering the IT service itself. Keeping users informed of full or partial service outages such as mobile or BlackBerry support or Web connectivity may not score any immediate points, but users appreciate honest, forthright information. Remember how you felt the last time you were waiting for an airplane to arrive that kept on being delayed and delayed and all the airline could do was be evasive?

Preparing Reports

Maybe I have just worked in a large IT environment for too long now, but it seems to me that Information Technology is more and more about reports and metrics and less about delivering technological capabilities to the users. I am frequently asked to provide reports, statistics, and information on usage — not necessarily information on performance (how well the system performed for the users), but other types of metrics. Depending on your management, you may be asked to provide:

- Total number of mailboxes and mailbox sizes
- Top system users and top source/destination domains
- Anti-spam and message hygiene statistics
- Disk space usage and growth
- System availability reports indicating how much unscheduled downtime may have been experienced during a certain reporting period.

Exchange does not provide you with a way to easily access most of this data. The mailbox statistics can be generated using the Exchange Management Shell, but many of these will actually require a third-party reporting product.

Something that you can do to prepare for a reporting requirement is to ensure that you are keeping two to four weeks' worth of message tracking and protocol logs.

Scheduled Downtime, Patches, and Service Packs

No one likes downtime, whether it is scheduled or not. Management may actually be holding you to a specific service level agreement (SLA) that requires you to provide so many hours of up-time per month or to provide e-mail services during certain hours. Unscheduled downtime is anything that happens during your stated hours of operation that keeps users from accessing their e-mail.

Even a small organization can provide very good availability for its mail services, and without large investments in hardware. Good availability begins with the following:

- Server hardware should always be from a reputable vendor.
- Server hardware should be installed using the vendor recommended procedures.
- Once the server is in production it should not be used as a test-bed for other software.

Don't underestimate the importance of training and documentation. In general, the industry formula for providing better availability for any system is to spend more money to purchase redundant servers and build fail-over clusters. But often better training for IT personnel and a simple investment in system documentation as well as system policies and procedures can improve availability as well, and for less money. I once became involved in a system outage where an untrained operator accidentally brought down a 15,000 mailbox cluster simply because he had been asked to do a task he had never done before and there was no documentation on how to proceed. So keep in mind that documentation, training, and procedures are very important in improving up time.

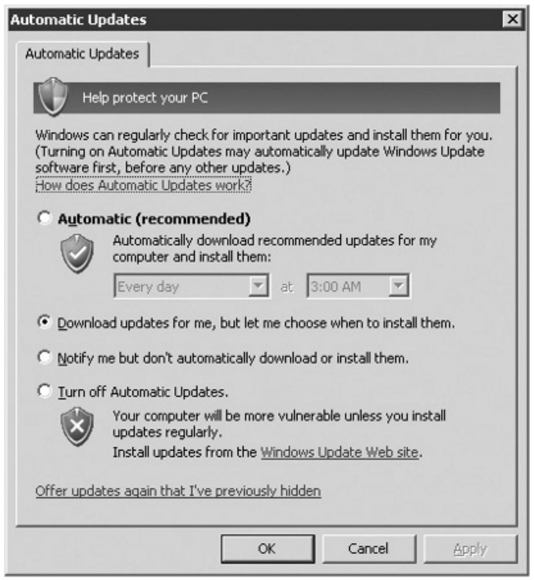
Even the biggest fail-over clusters and most highly available systems need some scheduled downtime. Even if it is scheduled in the wee hours of the morning, undoubtedly someone somewhere somehow will need access when you are working on the system. Therefore, your scheduled downtime should be documented as part of your operational plans and your user community should know about these plans. The specific time window for maintenance should always be the same; for some organizations, this might be 6:30PM–10:30PM on Thursday once per month, whereas other organizations might schedule downtime 11:00PM until 4:00AM every Sunday.

The number one reason for downtime is to apply updates and fixes to the operating system or to the applications running on the server. Microsoft releases monthly updates for the Windows operating system and Windows components. Every few months, Microsoft releases roll-up (RU) fixes for Exchange Server 2007 that fix bugs or that may even add functionality.

Microsoft's updates are usually downloaded to your servers shortly after they are released. The server can download them directly from Microsoft or they can be downloaded from a Windows Software Update Service (WSUS) server inside your network. It is important that you make sure the Automatic Updates component of Windows Server is configured correctly. Figure 1.9 shows the Automatic Updates properties.

Configure the server with the option Download Updates For Me, But Let Me Choose When To Install Them. This is an important setting because if you choose the Automatic (Recommended) option, the server will automatically apply any update within a day or so of downloading the update. This is not a desirable result. Instead, you want the server to download the updates and notify you via the updates icon in the system tray.

FIGURE 1.9
Configuring automatic updates



Windows can regularly check for important updates and install them for you. (Turning on Automatic Updates may automatically update Windows Update software first, before any other updates.)
How does Automatic Updates work?

- ( ) **Automatic (recommended)**
  Automatically download recommended updates for my computer and install them:
  Every day at 3:00 AM
- (•) Download updates for me, but let me choose when to install them.
- ( ) Notify me but don't automatically download or install them.
- ( ) Turn off Automatic Updates.
  Your computer will be more vulnerable unless you install updates regularly.
  Install updates from the Windows Update Web site.

Offer updates again that I've previously hidden

Tools You Should Know

Out of the box, Exchange Server is an excellent product, but sometimes the base software that you install can use some assistance. Some of these tools are actually installed with Exchange Server whereas other tools you may need to download.

PowerShell and the Exchange Management Shell Even here in the very first chapter, I am extolling the virtues of the new management shell (or command line and scripting interface) for Windows called PowerShell. PowerShell enables some basic Windows management functions, such as managing event logs and services, to be performed via a command-line interface. This interface is simple to use and easy to learn, even for a GUI guy like me. The Exchange team has built the entire Exchange Server 2007 management interface as an extension to PowerShell. It is called the Exchange Management Shell (or EMS).

Though almost every chapter in this book will include at least some information about using the EMS to perform Exchange management tasks, I have dedicated all of Chapter 8, ''Exchange Management Shell Primer,'' to helping you learn the EMS.

Exchange Management Shell test cmdlets The Exchange Management Shell (EMS) has a series of command-line tools that are very useful for testing and diagnosing problems. These include tools for testing Outlook Web Access connectivity, Unified Messaging connectivity, Outlook connectivity, and even mail flow. They are installed when you install the Exchange Server 2007 Management Tools. For more information, at the EMS prompt type **`Get-Help test*`**.

Exchange analyzers and troubleshooters In the Exchange Management Console under the Toolbox section (shown in Figure 1.10) you can find a series of troubleshooting wizards and analyzation tools including the Exchange Best Practices Analyzer, the database troubleshooting wizard, and the performance troubleshooting wizard.

One of the most essential things that you can learn is how to use the Exchange Best Practices Analyzer to run Health Check reports and determine if there is something that might not be configured properly within your organization.

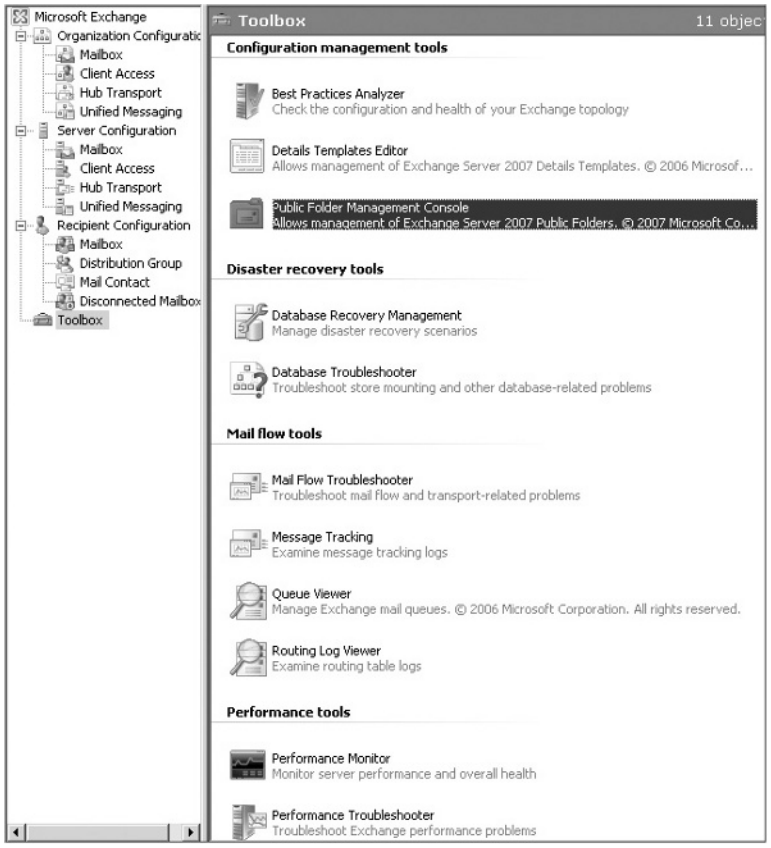


FIGURE 1.10
Viewing the Exchange Management Console Toolbox

Configuration management tools

Best Practices Analyzer
Check the configuration and health of your Exchange topology

Details Templates Editor
Allows management of Exchange Server 2007 Details Templates. © 2006 Microsof...

Public Folder Management Console
Allows management of Exchange Server 2007 Public Folders. © 2007 Microsoft Co...

Disaster recovery tools

Database Recovery Management
Manage disaster recovery scenarios

Database Troubleshooter
Troubleshoot store mounting and other database-related problems

Mail flow tools

Mail Flow Troubleshooter
Troubleshoot mail flow and transport-related problems

Message Tracking
Examine message tracking logs

Queue Viewer
Manage Exchange mail queues. © 2006 Microsoft Corporation. All rights reserved.

Routing Log Viewer
Examine routing table logs

Performance tools

Performance Monitor
Monitor server performance and overall health

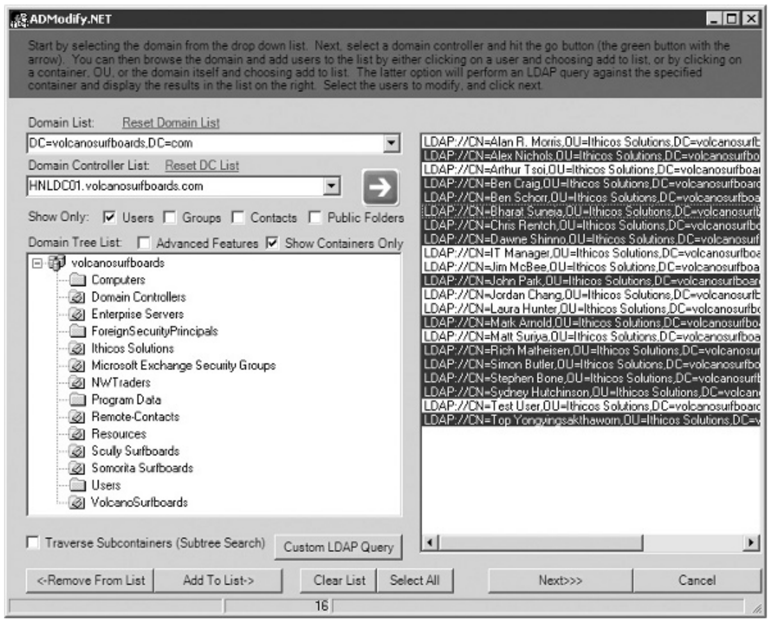
Performance Troubleshooter
Troubleshoot Exchange performance problems

PFDAVAdmin Public Folder Management Tool If your organization uses a lot of Exchange public folders, you may find that Exchange Public Folder Management Console is just not sufficient for managing permissions and propagating settings throughout the public folder tree. PFDAVAdmin is a powerful, free tool from Microsoft that is intended for advanced public folder property manipulation. You can download PFDAVAdmin from here:

`http://technet.microsoft.com/en-us/exchange/bb330849.aspx`

ADModify.NET If you need to make bulk changes to Active Directory Objects such as users, groups, contacts, or even public folders, then you need ADModify.NET (shown in Figure 1.11). This powerful and free tool allows you to find and select objects from the Active Directory and then use a simple interface to modify one or more attributes of the selected objects. You can even use other attributes of that object to build a new attribute.

There are a few important things to note about ADModify. You must run it from your local hard disk, and the Microsoft .NET Framework v2.0 is required. Another important item to note is that it has an ''undo'' feature; you can back out a bulk change that you made if it turns out to be wrong. You can download ADModify from `http://www.codeplex.com/admodify`.



Start by selecting the domain from the drop down list. Next, select a domain controller and hit the go button (the green button with the arrow). You can then browse the domain and add users to the list by either clicking on a user and choosing add to list, or by clicking on a container, OU, or the domain itself and choosing add to list. The latter option will perform an LDAP query against the specified container and display the results in the list on the right. Select the users to modify, and click next.

Domain List: DC=volcanosurfboards,DC=com

Domain Controller List: HNLDC01.volcanosurfboards.com

Show Only: [✓ Users] Groups Contacts Public Folders

Domain Tree List: Advanced Features [✓ Show Containers Only]

```
volcanosurfboards
    Computers
    Domain Controllers
    Enterprise Servers
    ForeignSecurityPrincipals
    Ithicos Solutions
    Microsoft Exchange Security Groups
    NWTraders
    Program Data
    Remote-Contacts
    Resources
    Scully Surfboards
    Somorita Surfboards
    Users
    VolcanoSurfboards
```

Traverse Subcontainers (Subtree Search)

```
LDAP://CN=Alan R. Morris,OU=Ithicos Solutions,DC=volcanosurfb
LDAP://CN=Alex Nichols,OU=Ithicos Solutions,DC=volcanosurfbo
LDAP://CN=Arthur Tsoi,OU=Ithicos Solutions,DC=volcanosurfboa
LDAP://CN=Ben Craig,OU=Ithicos Solutions,DC=volcanosurfboard
LDAP://CN=Ben Schorr,OU=Ithicos Solutions,DC=volcanosurfboa
LDAP://CN=Bharat Suneja,OU=Ithicos Solutions,DC=volcanosurfb
LDAP://CN=Chris Rentch,OU=Ithicos Solutions,DC=volcanosurfbo
LDAP://CN=Dawne Shinno,OU=Ithicos Solutions,DC=volcanosurf
LDAP://CN=IT Manager,OU=Ithicos Solutions,DC=volcanosurfboa
LDAP://CN=Jim McBee,OU=Ithicos Solutions,DC=volcanosurfboa
LDAP://CN=John Park,OU=Ithicos Solutions,DC=volcanosurfboard
LDAP://CN=Jordan Chang,OU=Ithicos Solutions,DC=volcanosurfb
LDAP://CN=Laura Hunter,OU=Ithicos Solutions,DC=volcanosurfbo
LDAP://CN=Mark Arnold,OU=Ithicos Solutions,DC=volcanosurfbo
LDAP://CN=Matt Suriya,OU=Ithicos Solutions,DC=volcanosurfboa
LDAP://CN=Rich Matheisen,OU=Ithicos Solutions,DC=volcanosur
LDAP://CN=Simon Butler,OU=Ithicos Solutions,DC=volcanosurfbo
LDAP://CN=Stephen Bone,OU=Ithicos Solutions,DC=volcanosurfb
LDAP://CN=Sydney Hutchinson,OU=Ithicos Solutions,DC=volcanc
LDAP://CN=Test User,OU=Ithicos Solutions,DC=volcanosurfboard
LDAP://CN=Top Yongyingsakthaworn,OU=Ithicos Solutions,DC=v
```

16

FIGURE 1.11 The main search screen of ADModify.NET

Quest ActiveRoles Management Shell for Active Directory Quest Software is giving away one of the most useful add-in tools for Microsoft PowerShell that I have ever used. Its Management Shell for Active Directory allows you to manage users and groups using PowerShell even if you don't yet have Exchange Server 2007. I use this tool almost daily in organizations that have not yet migrated to Exchange Server 2007. You can download this free tool from Quest at:

`http://www.quest.com/powershell/activeroles-server.aspx`

PowerGUI If you like the Quest ActiveRoles Management Shell, you will also like PowerGUI. PowerGUI is a graphical interface that ''wraps itself'' around PowerShell and allows you to see the results of PowerShell cmdlets. It will help you in writing scripts utilizing the PowerShell and other extensions to PowerShell. A sample of PowerGUI is shown in Figure 1.12.

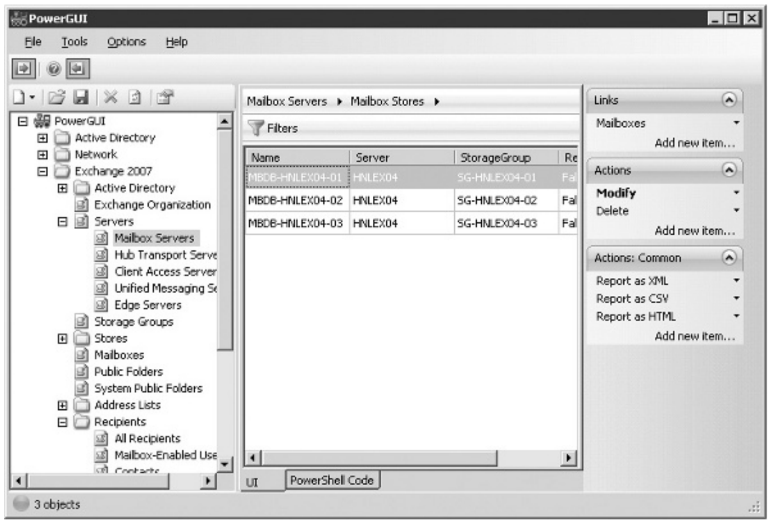
You can download PowerGUI from `http://powergui.org`.

Standards and Protocols

E-mail didn't just spring forth one day from the heads of the founders of the Internet fully formed and fully functional. Although few people fully appreciate it, the global e-mail system we take for granted every day is in a constant state of evolution. Countless numbers of systems participate in millions of daily connections, no two the same.

Every day, you and other people around the world effortlessly send and receive e-mail. By this point in time there have been hundreds, perhaps thousands, of e-mail clients and servers that have been, are being, and will be used, each with different design philosophies, features, and underlying programming languages.

FIGURE 1.12
Using the PowerGUI tool



Name	Server	StorageGroup	Re
MBDB-HNLEX04-01	HNLEX04	SG-HNLEX04-01	Fal
MBDB-HNLEX04-02	HNLEX04	SG-HNLEX04-02	Fal
MBDB-HNLEX04-03	HNLEX04	SG-HNLEX04-03	Fal

In order for these servers and clients to interoperate, there has to be at least some level of standardization. Although modern e-mail systems are overwhelmingly based on SMTP and a handful of related protocols and technologies, that wasn't always the case.

Components of an E-mail System

No matter the specifics of application and server, a handful of components and roles are involved in any e-mail transmission:

- The **mail user agent (MUA)** is the component that the user directly interacts with. If I were to use a postal metaphor, the MUA is roughly the equivalent of your local mailbox at the end of the driveway. Traditionally, the MUA has been a stand-alone client application such as Outlook; however, a web-based client such as Outlook Web Access also offers MUA functionality, even though it is a technically a server-side application.
- The **mail retrieval agent**, closely related to the MUA, is the component that handles retrieving messages from the main mail store. Depending on which protocols you are using, such as the Post Office Protocol (POP) or Internet Mailbox Access Protocol (IMAP), you can't just rely on new messages to be pushed to your MUA; something needs to pull them down for you. Typically, the MRA is not a separate component in modern systems, but a set of additional routines in the MUA that support message retrieval.
- If the MUA is the local mailbox, the **mail transport agent (MTA)** is the Post Office infrastructure connecting different towns and cities with each other. The MTA is responsible for accepting messages from other systems such as MUAs and MTAs, routing them, and ensuring their delivery on to their recipients. Messages typically travel though two

MTAs — the sender's and the recipient's (unless, of course, they share an MTA). In an Exchange 2007 system, the Hub Transport and Edge Transport roles fill the MTA role.

- Just as the MRA is a variant role often performed by the MUA, the **mail submission agent (MSA)** is a specialized form of the MTA. It's adapted to accept mail submissions from the MUA, introduce them into the mail flow, and handle any specialized processing that may be required. In Exchange 2007, this function is handled both in the Mailbox role as well as the Client Receive Connector on the Hub Transport role.
- What's missing from this picture? In this case, it's the equivalent of the local Post Office (or, if you prefer, the mail room in the big corporation) — the **mail delivery agent (MDA)** or **local delivery agent (LDA)**. Once the incoming message has been delivered to the proper collection of systems, the MDA/LDA is responsible for ensuring it's been put into the correct mailbox.

Each e-mail system can use a wide variety of solutions to implement these functions. Some applications, such as Exchange, incorporate all of these functions into a single end-to-end offering, whereas others provide just one piece of the puzzle, relying on other applications to provide the missing functionality. Even when using a complete solution, however, you can always mix and match pieces to provide functionality (such as using a third-party client for MUA functionality, or an edge mail appliance as an MTA to other mail systems). To ensure that these implementations work together, a series of standards have been developed over time.

Defining the Standards

When you have this many moving parts in a system — especially when they can be created, implemented, and configured by anyone — it helps if you define how the various parts work together. Although trying to create a true central authority for the Internet is a futile exercise, over the years the Internet community has evolved a method for describing, proposing, discussing, and documenting the various protocols that are in use across the world: a document known as the *Request For Comment*, or RFC.

RFCs began as informal memos among members of the academic teams that invented and programmed the hardware and software used to develop the Advanced Research Projects Agency Network (ARPANET), which was an early predecessor of the Internet. Whenever a team member wanted to suggest a new feature or protocol to the rest of the team, they'd write up an RFC describing their idea as a formal invitation for feedback. Over the years, as the ARPANET grew into the core of the Internet, RFCs expanded in scope to become more formal definitions and descriptions, sometimes retroactive.

Taken in total, the RFC archives provide a fascinating history of the Internet, although not all of the various technologies that have been used were documented through RFCs. Many of these alternatives, such as the X.400 standard for Message Handling Systems, were much more complicated and harder to implement than the systems described in the RFCs. In the end, the commercially developed X.400 standard and accompanying X.500 directory service protocols lost to SMTP, LDAP, and DNS. The latter protocols were all freely available in RFCs, allowing anyone with the time and interest to develop their own implementations.

RFCs are not static; they can contain errors or even just erroneous assumptions. As the world has changed, older RFCs are often updated, modified, or even superseded by newer RFCs. You can examine any of the RFCs online at `www.rfc-editor.org`. Table 1.1 summarizes some of the important mail transport RFCs.

TABLE 1.1: Some of the Common RFCs Relating to SMTP

RFC	TITLE	DESCRIPTION
821	Simple Mail Transfer Protocol	Defines how clients and servers transmit messages to each other using a simple, text-based conversational model. Released in August 1982; modified by 974 and 1869 among others; obsoleted by 2821.
822	Standard for the Format of ARPA Internet Text Messages	Defines the format and some of the standard headers used in messages passed by SMTP. Released in August 1982; modified by 1123 among others; obsoleted by 2822.
974	Mail Routing and the Domain System	Introduces the MX record into DNS and explains how it affects mail routing to remote systems. Released in January 1986; obsoleted by 2821.
1123	Requirements for Internet Hosts — Application and Support	An attempt to codify best practices and identify (and fix) errata for a variety of protocols; Chapter 5 focuses on SMTP. Released in October 1989; updated by 5321.
1869	SMTP Service Extensions	Defines the EHLO mechanism for Extended SMTP (ESMTP), allowing new features to be easily added to SMTP without requiring a complete updated RFC. Critical for certain features we now take for granted, such as SMTP authentication, TLS support, and streamlined data transfers of binary messages. Released in November 1995; obsoleted by 2821.
2554	SMTP Service Extension for Authentication	One of the main SMTP extensions, used to allow clients and servers to provide authentication before message submission. Provided an alternative to an open relay configuration. Released in March 1999; obsoleted by 4954.
2821	Simple Mail Transfer Protocol	This update ties SMTP, Extended SMTP, and many of the updates and fixes into a single document. It was the operational standard for systems such as Exchange 2007. Released in April 2001; obsoleted by 5321.
2822	Internet Message Format	An update to 822, published as a companion to 2821. Released in April 2001; obsoleted by 5322.
4954	SMTP Service Extension for Authentication	Provides several minor updates to the existing SMTP AUTH mechanism. Released in July 2007.
5321	Simple Mail Transfer Protocol	The third release of SMTP, intended to be the standard for future messaging systems. Addresses several lingering operational prohibitions and loopholes that make it hard for messaging administrators to fight spam within strict RFC compatibility. Released in October 2008.
5322	Internet Message Format	An incremental update to the message format used by SMTP. Released in October 2008.

Part of what makes SMTP so relatively simple is that it isn't based on any complicated routing schemes; at its heart, it is a *store-and-forward* system. Once an SMTP system accepts responsibility for a message, it then attempts to deliver it to the next best hop that it knows about. An SMTP connection, in its simplest form, is easy to understand. It involves just two systems: the client, which opens the connection so that it can submit e-mail, and the server, which accepts inbound connections and determines if it can accept the message that has been submitted. This is what a typical session, sending a message from `sender@client.tld` to `recipient@server.tld`, would look like:

```
{The client connects to the server}
01 S: 220 smtp.server.tld ESMTP mail system ready
   C: HELO desktop.client.tld
   S: 250 Hello desktop.client.tld, I am glad to meet you
02 C: MAIL FROM:<sender@client.tld>
   S: 250 Sender ok
03 C: RCPT TO:<recipient@server.tld>
   S: 250 Recipient ok
04 C: DATA
   S: 354 End data with <CR><LF>.<CR><LF>
05 C: From: "Client Sender" <sender@client.tld>
   C: To: Another User <another@otherdomain.tld>
   C: Date: Wed, 22 Oct 2008 01:13:22 -0800
   C: Subject: Test message
06 C:
   C: Isn't SMTP easy?
07 C: .
   S: 250 Ok: queued as 918273645
08 C: QUIT
   S: 221 Bye
{The server closes the connection}
```

There are eight key points to note in this conversation:

1. The initial greeting from the server, combined with the HELO or EHLO (''hello'') response from the client.
2. The client begins sending the *SMTP envelope* information that tells who the actual sender is. If the message is rejected or bounced, this envelope sender is the person who will receive the notification — not necessarily the person in the ''From:'' header.
3. The client continues the envelope by listing one or more recipients. Like the envelope sender, these recipients don't have to match the ones listed in the actual message. In fact, having an envelope recipient not listed in the message is precisely how Blind Carbon-Copy works.
4. The envelope is done; the client now begins submitting the actual message. Unless it uses modern SMTP extensions, this is just simple text.
5. First come the message headers. Note that these recipients don't match the ones in the envelope. Though these headers may be used for filtering, they won't be used for routing.
6. When the headers are finished, the client sends a blank line and continues with the message body starting on the next line.

7. When the message body is done, the client sends the *End of Data* sequence. The server now accepts or rejects the message; if the message is accepted, the server reports the associated queue information (much like a receipt).
8. The client indicates that it's finished with the connection. If it had another message to send to recipients on this system, it could reset this connection and reuse it for one or more following messages.

If the server were not the final destination for this message — perhaps it's just an edge mail system, handling all interaction with the Internet and performing message hygiene functions before routing accepted messages further into the organization — it would pass the system on to the next hop. Note that SMTP only handles message flow from the MUA through the MTA or MSA, and from one MTA to another MTA; it is not used to transfer messages to an MDA or LDA, or from the user's mailbox database back to the MUA or MRA.

DNS — Name Resolution

In the early days of the Internet, all of the hosts on the network were maintained in a single file called `HOSTS`. If you wanted to add a new computer to the Internet, you called up the guy who maintained the file and asked him to add your computer's name and IP address. Periodically, everyone on the Internet downloaded the latest version of the `HOSTS` file.

You can see how that would be a bit difficult today; the file would be huge, slow to search, and changing constantly. Plus it would require a massive team of people to maintain it. In 1984, a standard was proposed and adopted to allow name resolution by creating a hierarchical name space where different owners of the name space would be responsible for maintaining their own hosts. The pieces of the name space are called *domains*. Root servers would provide referrals to clients so that they could contact the correct servers that held the information. So, for example, I own a domain called `somorita.com` and I can control any hosts in that particular domain. The root servers on the Internet provide referrals to other servers on the Internet and point them to the servers that hold the `somorita.com` data.

As you may have guessed by now, I am talking about the Domain Name System (DNS); this is documented in RFC 1034. You may be wondering how DNS directly (or indirectly) affects mail servers, though. A full discussion of DNS can consume an entire book, but I wanted to discuss a few topics about which you should know some basic information.

Mail Exchanger (MX) records The Mail Exchanger or MX record is the DNS entry for your domain that allows mail servers on the Internet to look up the host names of systems that accept mail for your domain. These DNS records point to public address (A) records or aliases (CNAMES) of the SMTP servers that accept mail for your organization; this may be your Hub Transport or Edge Transport server, a third-party message hygiene system, a simple SMTP relay server, or even a third-party managed provider. You can use the NSLOOKUP command to look up and validate MX records for your domain. Here is an example of looking up the servers that accept mail for `apple.com`:

```
nslookup -q=mx apple.com
Server:  dnsserver1.somorita.local
Address:  192.168.254.71

Non-authoritative answer:
apple.com    MX preference = 100, mail exchanger = mail-in3.apple.com
apple.com    MX preference = 10, mail exchanger = mail-in11.apple.com
apple.com    MX preference = 10, mail exchanger = mail-in12.apple.com
apple.com    MX preference = 20, mail exchanger = mail-in1.apple.com
apple.com    MX preference = 20, mail exchanger = mail-in2.apple.com
```

Notice in the MX records that the preference values are different for different records. Mail servers are always supposed to choose the record with the lowest preference value first and then only use the records with the higher preference values if the lower ones do not respond. If two or more records are equal, the sending mail server is supposed to load balance. For more information see `http://en.wikipedia.org/wiki/MX_record`.

Address records and aliases Address records (also called A records) are DNS entries that point directly to an IP address. Aliases (also called CNAMES) point to other address records.

Pointer records A pointer record (PTR) is not added specifically to your domain, but rather to a domain name that represents an IP subnet. All of these ''domains'' are found in DNS in the `in-addr.arpa` domain; PTR records allow clients to take the IP address of an Internet host and look up the name associated with the IP address. PTR records are usually created and managed by the owner of the IP subnet block, not the manager for a domain like `somorita.com`.

Sender Policy Framework Sender Policy Framework (SPF) records are DNS records that help a receiving mail server determine whether or not the mail server that originally sent a message is authorized to send mail for that domain. SPF is often billed as an anti-spam system but it is better defined as an anti-spoofing system. For more information see `www.openspf.org`.

Service location records Service location records (SRV) records are DNS records that help a client locate a specific server type that is provided for a domain. Active Directory publishes SRV records for a number of different services including the global catalog server, so there might be multiple global catalog SRV records for a particular domain. Another good example of the use of an SRV record is Outlook 2007 Service Pack 1 looking for Autodiscover services. For more information see `http://en.wikipedia.org/wiki/SRV_record`.

Split brain DNS Split brain DNS systems are systems that (usually) have two different sets of DNS servers that host the same DNS server. Usually one set of DNS servers hosts the domain name for an internal network and is used to resolve the host names to internal IP addresses, while the other set is used to resolve IP addresses for hosts on the Internet. These are used when the organization's public and private domain name are the same and the organization does not want internal host information available to Internet users. For more information see `http://msdn.microsoft.com/en-us/library/ms954396.aspx`.

Accessing the Mailbox

As just mentioned, SMTP is designed for message transport — getting a message from one system to another. Once the message has been received and delivered to a mailbox store, there's one final hop that needs to happen — to the user's MUA. SMTP would be a poor protocol to handle this function, so the Internet community developed two common alternatives: POP3 and IMAP.

- **POP3**, the third version of the Post Office Protocol, is intended to be a simple, no-frills mailbox retrieval protocol. A POP3 client connects to the Mailbox server, provides the user's credentials, gets a list of messages that have arrived in the user's Inbox folder, and downloads one or more of them to the local client. POP3 is simple and lightweight to implement, ensuring that just about every e-mail client application supports it, but it has two main flaws in a traditional business setting:

1. It doesn't understand the concept of folders. POP3 will only pull messages from the Inbox; if you have server-side rules that move messages to alternate folders (or have moved them manually, perhaps through a web-based mail client), they will not be visible to POP3. Once messages have been downloaded to the MUA, they can of course be filed in any local folders.

2. By default, POP3 is designed to delete mail from the message store; it's really intended for situations where the mail store holds messages for a short time before the user picks them up. It wasn't designed to interact with a modern messaging system where the goal is to keep messages on the server, where they can be managed. Although you can configure most POP3-aware MUAs not to delete the server copy, this can cause other problems.

- **IMAP**, by contrast, was specifically designed to allow an MUA to interface with a message store that stays resident on the server. The current version of IMAP, 4.1, is a sophisticated protocol with many options. It easily handles folder hierarchies, local caching, and other high-end features required by heavy e-mail consumers. However, IMAP was designed with e-mail in mind; most IMAP clients do not synchronize calendar and contact information.

Both POP3 and IMAP are supported by Exchange 2007 out of the box, but they're not configured by default. Also, both require the users to have access to an SMTP server to submit any new messages or replies to; neither POP3 nor IMAP handles submitting new messages back from the MUA to the server. They're strictly *pull* protocols and require a *push* protocol like SMTP to ensure new messages get delivered.

POP3 Connectors Considered Harmful

Back when local messaging systems were becoming common and high-end dedicated Internet links were rare and costly, companies that wanted to host their own mail server faced a choice: spend money to get a dedicated line and IP address to run their own SMTP service on the Internet, or get an ISP to handle and queue their incoming SMTP traffic for them. The queuing strategy was definitely more economical, but in order to work well, you needed to find an ISP that was technically savvy and supported the SMTP enhanced turn (ETRN) extension. In what I hope isn't too much of a surprise twist, these providers usually charged more for e-mail service.

As a result, some bright person decided to write a POP3 connector. The idea was simple: get my ISP to deliver all e-mail for my domain to a single POP3 mailbox. I can use a cheap POP3 dial-up account and a POP3 connector to download this e-mail and deliver it into my on-premises SMTP server, which will then send it on to the mailboxes of my users. At first glance, it's an ingenuous solution to a common problem.

However, POP3 connectors violate the basic underlying assumptions of SMTP. Once a message has been delivered to a mailbox at the ISP, it often loses the SMTP envelope information that helps route the message, especially in the case of a Blind Carbon-Copy recipient. It may have been subject to extra processing — or, more commonly, hasn't been put under any filtering at all. Now the POP3 connector must download the message and inject it back into the local SMTP server for any further filtering to happen, leaving the local SMTP server stuck with handling the bounces for any messages that it wouldn't have accepted (such as those to non-existent recipients or those that could have been easily filtered as spam before the message was accepted).

Proponents of POP3 connectors focus almost entirely on the cost and point out that there are solutions to each objection that people can make about connectors. They're typically right, to a point; any specific problem can be handled through a clever kludge or gimmick. Though they're quick to point out their own anecdotal evidence, what they don't say is that each of these gimmicks relies on you being lucky enough to have an ISP that does things in a way that the connector recognizes. Many of these ''fixes'' require the ISP to inject extra headers into the message that the connector will then strip right back out; these headers are non-standard and often proprietary, and ISP administrators are understandably not eager to implement these systems for a simple no-frills bargain POP3 account.

Most importantly, POP3 connectors can place extra burden not just on your ISP's system, but on your users and correspondents as well. Although SMTP was never intended to be an instantaneous way to transmit e-mail, it turns out that it's pretty darned efficient at doing just that, enough so that people now have the expectation that e-mail is received instantly. Anything that impairs this process is seen as a problem, whether or not it really is. POP3, by its nature, is a polling mechanism, and most ISPs that offer it intend users to poll their mailboxes no more than every 15 minutes. If your users believe they need to receive e-mail more frequently, your POP3 connector will have to poll your ISP's server more frequently in turn.

There are a handful of situations for which POP3 connectors are the right solution. In most deployments, however, they are just another case of penny wise, pound foolish; spending the small additional sum for a dedicated line (such as ADSL or business-grade cable modem) would have more than returned its value in increased speed and reliability.

Finding People and Services: Directory Services

One additional aspect to messaging systems is how they look up message senders and recipients. Early messaging systems used simple flat-file lookups — typically out of the same files that listed user accounts, such as the `passwd` files on a Unix system. Flat files get quickly unwieldy for a large number of users, though, so various systems began using alternatives:

- Sun's Unix variants offered the Network Information Services (NIS), which provided a shared directory service, including messaging aliases and delivery information, across hundreds or even thousands of hosts within an organization.
- Early versions of Windows NT offered the local SAM database — their equivalent of the Unix `passwd` file — as well as Windows NT domains. NT domains allowed multiple Windows machines to share the same user information, including e-mail addresses.

Other messaging systems often provided tight integration with a corresponding directory service. The X.400 messaging system required the corresponding X.500 directory service. Though the concept of X.500 was interesting and useful, it was far too complicated to implement for SMTP. However, many of the same constructs and architectural features went into the design of the Lightweight Directory Access Protocol (LDAP) service, which was described in a series of RFCs.

LDAP proved to be a good complement for messaging systems; almost all modern MUAs and MTAs provide the ability to look up contacts and users from an LDAP-based directory service. Microsoft's Active Directory, Novell's Novell Directory Service, and eDirectory products are direct adaptations and extensions of the LDAP standard. By using LDAP, a distributed mail system can efficiently route and process millions of messages with minimal administrative overhead. Exchange Server 2007 relies heavily upon Active Directory to do its work.

Messaging Application Programming Interface

The Messaging Application Programming Interface (MAPI) is a programming interface that allows a developer to more easily write an application that accesses e-mail or directory functions and services. Though often considered an industry standard, MAPI was actually developed by Microsoft in the early 1990s and the API set is published so that anyone can use it.

Outlook, for example, uses MAPI to access data on the Exchange server as well as accessing directory information from Active Directory. An underlying directory service provider and e-mail service provider allows MAPI to access these systems. Outlook includes an underlying service provider for Exchange server as well as a service provider that allows access to PST files and IMAP4/POP3 servers. Third parties have developed service providers that will allow Outlook to access other messaging systems such as Lotus Notes, HP OpenMail, Hotmail, and even Zimbra.

A number of variations and versions of MAPI have been released over the years as new messaging functionality has been developed. To communicate with a client/server messaging system such as Exchange server, MAPI relies on Remote Procedure Calls (RPCs) to transport MAPI requests and data between the client and the server.

Message Security

The SMTP protocol that allows us to easily send an e-mail to any e-mail server in the world is very simple, but it is also very insecure. Essentially, in order for my mail server to be able to send e-mail to, say, `obama@whitehouse.gov`, the server at `whitehouse.gov` needs to be able to accept the mail from me anonymously. If SMTP required authentication, `whitehouse.gov` would need to have a user account and password for anyone who was going to send them e-mail.

Due to the open nature of Internet mail, it is very easy for unscrupulous people to send unsolicited commercial e-mail (UCE), hereafter known as *spam*. It is also easy for mail to be spoofed so that it appears to come from a credible source (such as your bank) and encourages you to take an action such as logging on to a fake URL and providing your banking credentials; this is known as *phishing*.

It is easy for these unscrupulous people to send e-mails with malicious attachments that might spread a virus, load a program onto your computer that will further spread itself (such programs are called *worms*), load a program that will then generate spam to send to others (this is a *bot*), or load a monitoring or remote control program on your computer that a malicious hacker can then use. These viruses, worms, and Trojan horse–type programs are collectively known as *malware*.

Finally, e-mail is such an easy way to send information back and forth that your users may misuse use it by sending inappropriate information to their friends and colleagues. Inappropriate use of e-mail can open an organization up to bad publicity and even potential lawsuits, not to mention getting the senders and recipients into big trouble. See `http://www.khon2.com/home/ticker/29852384.html` for a good example of this.

Message Hygiene

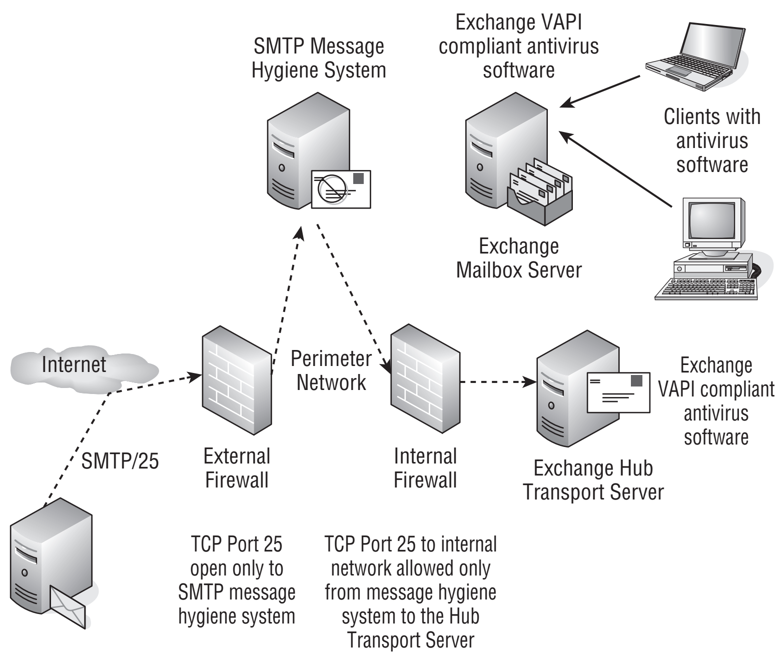
Collectively, the science of scanning messages for inappropriate content is known as *message hygiene*. All mail systems today should include some type of message hygiene system that, at a minimum, protects against viruses and reduces the amount of spam that makes its way into the user's mailbox.

Out of the box, Exchange Server 2007 provides fairly decent protection against spam through the anti-spam components of the Edge Transport and Hub Transport servers. But you will need additional software for protection against viruses. Customers who have Enterprise Client Access Licenses (CAL) for all users can use Microsoft's Forefront Security for Exchange Server product (`www.microsoft.com/forefront`).

I discuss this issue in a bit more depth in Chapter 25, ‘‘Securing Exchange Server,’’ but here are some basics of message hygiene security.

You may choose to implement your own message hygiene system, in which case you have your own servers performing the message hygiene functions. Figure 1.13 shows a multi-layer message hygiene system.

Figure 1.13
Implementing your own multi-layer message hygiene system


SMTP Message Hygiene System
Exchange VAPI compliant antivirus software
Clients with antivirus software
Exchange Mailbox Server
Internet
Perimeter Network
Exchange VAPI compliant antivirus software
SMTP/25
External Firewall
Internal Firewall
Exchange Hub Transport Server
TCP Port 25 open only to SMTP message hygiene system
TCP Port 25 to internal network allowed only from message hygiene system to the Hub Transport Server

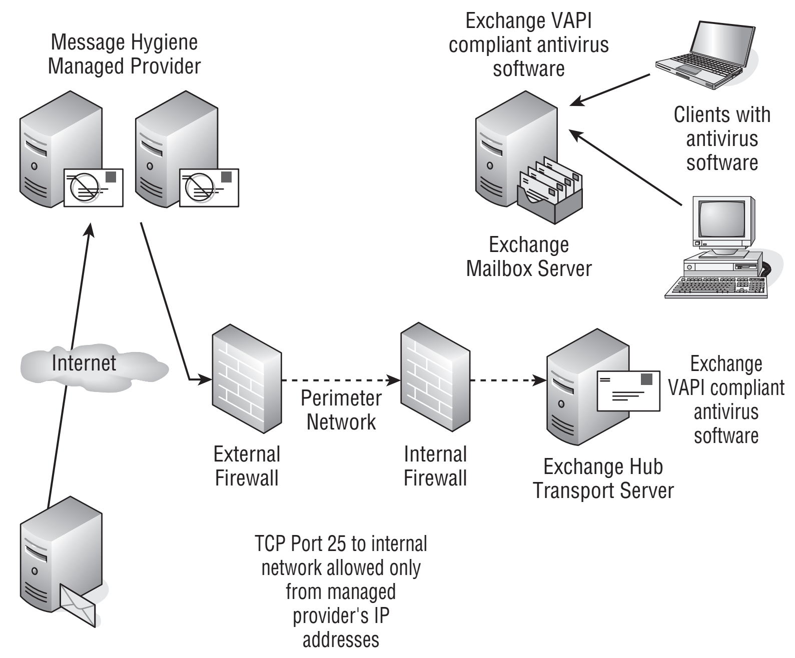
The message hygiene system in Figure 1.13 is a multi-layer system; this system has more than one place that may stop an e-mail-borne threat. Ideally, the majority of spam and malicious e-mail will be stopped by the message hygiene system in the perimeter network; this could be the Microsoft Exchange Edge Transport server or it could be one of the dozens of available SMTP-based message security systems available from third parties. The point of the hygiene system in the perimeter is to keep as much undesirable content as possible from reaching your production mail system and to protect the internal mail servers from possible attempts to compromise them.

Once a message is scanned in the perimeter network, it is then passed on to the Exchange servers on your internal network. There additional scanning takes place either when the message is moving through the message transport system or when the message is placed in the user’s Inbox. Ideally, the scanning system (or scanning engine) on the inside of the network should be a different scanning engine from the one that is used on the perimeter.

The final layer of protection is implemented at the client. The client has a file and memory virus/malware scanner that looks at any content as it is opened, whether that content is in the user’s Inbox or something downloaded from the Internet or something on a CD-ROM in the CD drive. Once again, ideally the software running on the client will be from a different vendor than the software running on the server. Running multiple types of scanning software improves the likelihood that newer threats will be stopped.

Some organizations decided that they don't want to have to maintain perimeter-based message hygiene systems so they use a third-party vendor that provides Internet-based scanning for them. These are usually known as managed providers and they have SMTP-based scanning systems that will scan messages coming to your mail system before they are delivered to your Exchange servers. Figure 1.14 shows an example of using a managed provider.

FIGURE 1.14
Using a managed provider


Message Hygiene Managed Provider
Exchange VAPI compliant antivirus software
Clients with antivirus software
Exchange Mailbox Server
Internet
Perimeter Network
External Firewall
Internal Firewall
Exchange Hub Transport Server
Exchange VAPI compliant antivirus software
TCP Port 25 to internal network allowed only from managed provider's IP addresses

The additional cost of using a managed provider is offset by the fact that you don't have to maintain your own perimeter-based scanning system and that most malicious or unwanted e-mail content can be stopped prior to entering your network in the first place. Some third-party managed providers can also provide additional message security, disaster recovery, and message archival functions.

Approaches to Spam Protection

There are a lot of ways that your message hygiene system can determine if something is spam or is being sent by an unauthorized sender. Though each of these topics deserves in-depth treatment, my intent here is merely to familiarize you with the concepts.

Content inspection The most common way that a message is determined to be spam or a phishing message is through content inspection. The software opens the message and looks for characteristics of spam messages, such as a message with nothing but a URL or image, messages that mention certain words or phrases, and so on. Based on the content, the software ranks the message with a number (usually called the Spam Confidence Level or SCL) from 0 to 9 with 0 being likely the message is not spam and 9 being very spammy. Internal messages are set to an SCL level of negative one (–1). The message transport can then be configured with your tolerance level for spam and can reject, delete, or quarantine messages with higher SCL values. Arguably, content inspection is considered the most accurate method of detecting spam.

Block lists Block lists are lists that either you or a third party maintains. The lists contain IP addresses of known spammers, dial-up IP addresses, DHCP IP addresses, or IP addresses of systems that will allow spammers to send through. The third-party lists are often known as real-time block lists (RBLs) and are maintained (usually by volunteers).

Sender Protection Framework / Sender ID / Domain Keys The Sender Protection Framework (SPF), now known as Sender ID, is an initiative backed by Microsoft that requires that all known senders on the Internet register the addresses of mail servers that will send mail on their behalf. The Domain Keys initiative (DKIM) is backed by Yahoo and requires that a sending system include a calculation in the header of each outgoing message that the receiving system then verifies. Both of these initiatives are more directly ''anti-spoofing'' systems than they are ''anti-spam'' systems, but they are useful in helping to ensure that messages really are coming from the stated sender — which can help reduce spam.

DNS Name and IP verification Though Exchange Server cannot do some of these verifications, some SMTP systems will verify things such as whether or not your public IP address has a valid pointer (PTR) record and whether the DNS domain name you are using is valid. This can help reduce spam but also increases the probability of false positives. (A false positive occurs when the message hygiene mistakenly tags a legitimate message as spam and either quarantines, deletes, or rejects it.)

Sender inspection Sender inspection or sender filtering is the least useful method of blocking spam because it requires maintaining lists of senders' SMTP addresses or lists of domains from which you will not accept inbound mail. The problem with this approach is that spammers usually do not use the same sender address twice.

Why Is My Mail Being Rejected?

Naturally, if you put a system in place that scans and possibly rejects e-mail based on the characteristics or the sender of the message, you are occasionally going to end up with false positives. These false positives will always be in the form of an important e-mail that is being sent to your CEO or one that she is waiting to receive. Take a look at the non-delivery report (NDR) shown in Figure 1.15; this message was rejected by the receiving mail system.

Exchange Server 2007's message transport system does a pretty good job of examining rejection codes and letting you know why a message was rejected, but you may still have to do some detective work. A remote mail system might reject mail from your users for a number of reasons:

- The public IP address from which you send mail is on a real-time block list (RBL) provider.
- Your public IP address is registered as a DHCP IP address on some RBLs; your Internet service provider (ISP) must correct this.
- You do not have Sender ID records registered in DNS for your public IP addresses, or the records are incorrect.
- The message has content that makes it look like spam such as suspicious words (mortgage, Viagra, enlargement, free) or the messages is very short (one sentence, for example).
- Your public IP address does not have a DNS PTR record; your ISP or the owner of the IP address must fix this.

- Your mail server is sending out the wrong mail domain name when it connects to a remote mail system. For example, your mail domain is `somorita.com` but it is saying your domain name is `somorita.local`.

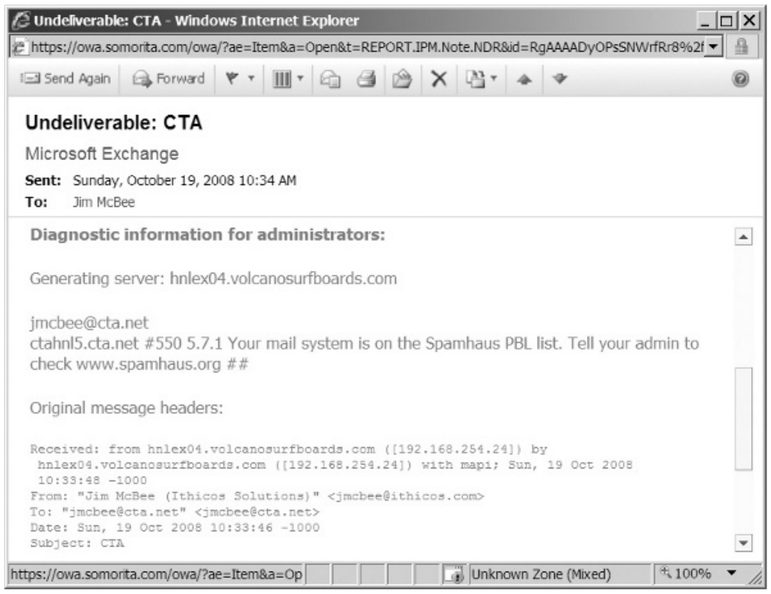


FIGURE 1.15
Examining the report of a rejected message

Undeliverable: CTA - Windows Internet Explorer
https://owa.somorita.com/owa/?ae=Item&a=Open&t=REPORT.IPM.Note.NDR&id=RgAAAADyOPsSNWrfRr8%2f
Send Again
Forward

Undeliverable: CTA
Microsoft Exchange
Sent: Sunday, October 19, 2008 10:34 AM
To: Jim McBee

Diagnostic information for administrators:

Generating server: hnlex04.volcanosurfboards.com

jmcbee@cta.net
ctahnl5.cta.net #550 5.7.1 Your mail system is on the Spamhaus PBL list. Tell your admin to check www.spamhaus.org ##

Original message headers:

```
Received: from hnlex04.volcanosurfboards.com ([192.168.254.24]) by
 hnlex04.volcanosurfboards.com ([192.168.254.24]) with mapi; Sun, 19 Oct 2008
 10:33:48 -1000
From: "Jim McBee (Ithicos Solutions)" <jmcbee@ithicos.com>
To: "jmcbee@cta.net" <jmcbee@cta.net>
Date: Sun, 19 Oct 2008 10:33:46 -1000
Subject: CTA
```

The Role of Storage

In medium-sized and large organizations, the Exchange administrator is usually not also responsible for storage. Many medium-sized and large organizations use specialized storage area networks (SANs) that require additional training to master. Storage is a massive topic and many books have been dedicated to discussing storage, but I feel it is important that you at least be able to speak the language of storage.

From the very beginning, messaging systems have had a give-and-take relationship with the underlying storage system. Even on systems that aren't designed to offer long-term storage for e-mail (such as ISP systems that offer only POP3 access), e-mail creates demands on storage:

- The transport (MTA) components must have space to queue messages that cannot be immediately transmitted to the remote system.
- The MDA component must be able to store incoming messages that have been delivered to a mailbox until users can retrieve them.
- The message store, in systems like Exchange, permits users to keep a copy of their mailbox data on central servers.
- As the server accepts, transmits, and processes e-mail, it keeps logs with varying levels of detail so administrators can troubleshoot and audit activities.

Though you'll have to wait for subsequent chapters to delve into the details of planning storage for Exchange, the following sections go over the broad categories of storage solutions that are used in modern systems.

Direct Attached Storage (DAS)

Direct attached storage (DAS) is the most common type of storage. DAS disks are usually internal disks or directly attached via cable. Just about every server, except for some high-end varieties, uses DAS at some level; typically, the operating system is on some DAS configuration. The main problem with DAS, historically, has been performance and capacity. To get around these problems, one can always just add more disks to the server. This gives you a configuration known as JBOD — Just a Box Of Disks.

Just a Box Of Disks (JBOD)

Although JBOD can usually give you the raw capacity you need, it has three flaws that render it unsuitable for most Exchange deployments:

- JBOD forces you to partition your data. Because each disk has a finite capacity, you can't store data on that disk if it is larger than the capacity. For example, if you have four 250GB drives, even though you have approximately one terabyte of storage in total, you have to break that up into separate 250GB partitions. Historically, this has caused some interesting design decisions in messaging systems that rely on file system–based storage.
- JBOD offers no performance benefits. Each disk is responsible for only one chunk of storage, so if that disk is already in use, subsequent I/O requests will have to wait for it to free up before they can go through. A single disk can thus become a bottleneck for the system, which can slow down mail for all of your users (not just those on the affected disk).
- JBOD offers no redundancy. If one of your disks dies, you're out of luck unless you can restore that data from backup. True, you haven't lost all of your data, but the one-quarter of your users who have just lost their e-mail are not likely to be comforted by that observation.

Luckily, some bright people came up with a great answer to JBOD: the Redundant Array of Inexpensive Disks, or RAID.

Redundant Array of Inexpensive Disks (RAID)

The basic premise behind RAID is to group the JBOD disks together in various configurations, allowing the computer (and applications) to see them as one very large disk device. These collections of disks are known as *arrays*; the arrays are presented to the operating system, partitioned, and formatted as if they were just regular disks. The common types of RAID configurations are shown in Table 1.2.

Note that several of these types of RAID arrays may be suitable for your Exchange server. Which one should you use? That question depends entirely on how many mailboxes your servers are holding, how they're used, and other types of business needs. Beware of anyone who tries to give hard and fast answers like, ''Always use RAID-5 for Exchange database volumes.'' To determine the true answer, you need to go through a proper storage sizing process, find out what your I/O and capacity requirements are really going to be, think about your data recovery needs and service level agreements (SLAs), and then decide what storage configuration will meet those needs for you in a fashion you can afford. There are no magic bullets.

In every case, the RAID controller you use — the piece of hardware, plus drivers, that aggregates the individual disk volumes for you into a single pseudo-device that is presented to Windows — plays a key role. You can't just take a collection of disks, toss them into slots in your server, and go to town with RAID. You need to install extra drivers and management software, you need to take extra steps configuring your arrays before you can even use them in Windows, and you may even need to update your disaster recovery procedures to ensure that you can always recover data from drives in a RAID array. Generally, you'll need to test whether you can move drives in one array between two controllers, even those from the same manufacturer; not all controllers support all options. After your server has melted down and your SLA is fast approaching is not a good time to find out that you needed to carry a spare controller on hand.

Table 1.2: RAID Configurations

RAID Level	Name	Description
None	Concatenated drives	Two or more disks are joined together in a contiguous data space. As one disk in the array is filled up, the data is carried over to the next disk. Though this solves the capacity problem and is easy to implement, it offers no performance or redundancy whatsoever, and makes it more likely that you're going to lose all of your data through a single disk failure, not less. These arrays are not suitable for use with Exchange.
RAID-0	Striped drives	Two or more disks have data split among them evenly. If you write a 1MB file to a two-disk RAID-0 array, half the data will be on one disk, half on the other. Each disk in the array can be written to (or read from) simultaneously, giving you a noticeable performance boost. However, if you lose one disk in the array, you lose all your data. These arrays are typically used for fast, large, temporary files, such as those in video editing. These arrays are not suitable for use with Exchange.
RAID-1	Mirrored drives	Typically done with two disks (although some vendors allow more), each disk receives a copy of all the data in the array. If you lose one disk, you've still got a copy of your data on the remaining disk; you can either move the data or plug in a replacement disk and rebuild the mirror. RAID-1 also gives a performance benefit; reads can be performed by either disk, because only writes need to be mirrored. However, RAID-1 can be one of the more costly configurations; to store 500GB of data, you'd need to buy two 500GB drives. These arrays are suitable for use with Exchange, depending on the type of data and the performance of the array.
RAID-5	Parity drive	Three or more disks have data split among them. However, one disk's worth of capacity is reserved for *parity checksum* data; this is a special calculated value that allows the RAID system to rebuild the missing data if one drive in the array fails. The parity data may be kept on one disk, or spread across all the disks in the array. If you had a four-disk 250GB RAID-5 array, you'd only have 750GB of usable space. RAID-5 arrays offer better performance than JBOD, but worse performance than other RAID configurations, especially on the write requests; the checksum must be calculated and the data + parity written to all the disks in the array. Also, if you lose one disk, the array goes into *degraded mode*, which means that even read operations will need to be recalculated and will be slower than normal. These arrays are suitable for use with Exchange, depending on the type of data and the performance of the array.

Table 1.2: RAID Configurations *(CONTINUED)*

RAID Level	Name	Description
RAID-6	Double parity drive	This RAID variant has become common only recently, and is designed to provide RAID-5 arrays the ability to survive the loss of two disks. Other than offering two-disk resiliency, base RAID-6 implementations offer mostly the same benefits and drawbacks as RAID-5. Some vendors have built custom implementations that attempt to solve the performance issues. These arrays are suitable for use with Exchange, depending on the type of data and the performance of the array.
RAID-10	Mirroring plus striping	A RAID-10 array is the most costly variant to implement because it uses mirroring. However, it also uses striping to deliver blistering performance, which makes it a great choice for high-end arrays that have to sustain a high-level of I/O. As a side bonus, it also increases your chances of surviving the loss of multiple disks in the array. There are two basic variants. RAID 0+1 takes two big stripe arrays and mirrors them together; RAID 1+0 takes a number of mirror pairs and stripes them together. Both variants have essentially the same performance, but 1+0 is preferred because it can be rebuilt more quickly and has far higher chances of surviving the loss of multiple disks. These arrays are suitable for use with Exchange, usually for highly loaded mailbox database volumes.

If you choose the DAS route (whether JBOD or RAID), you'll need to think about how you're going to house the physical disks. Modern server cases don't leave a lot of extra room for disks; this is especially true of rack mounted systems. Usually, this means you'll need some sort of external enclosure that hooks back into a physical bus on your server, such as SCSI or eSATA. Make sure to give these enclosures suitable power and cooling; hard drives pull a lot of power and return it all as heat. Also make sure that your drive backplanes (the physical connection point) and enclosures support *hot-swap* capability, where you can pull the drive and replace it without powering the system down. Keep a couple of spare drives and drive sleds on hand, too. You really don't want to have to schedule an outage of your Exchange server in order to replace a failed drive in a RAID-5 array, letting all of your users enjoy thrashed performance because the array is in degraded mode until the replacement drives arrive.

Choosing RAID Controllers

Beware! Not all kinds of RAID are created equal. Before you spend a lot of time trying to figure out which configuration to choose, you need to first think about your RAID controller. There are three kinds of them, and unlike RAID configurations, it's pretty easy to determine which kind you need for Exchange.

- **Software RAID** avoids the whole problem of having a RAID controller by performing all of the magic in the operating system software. If you convert your disk to dynamic volumes, you can do RAID-0, RAID-1, and RAID-5 natively in Windows 2003 and Windows 2008 without any extra hardware. However, Microsoft strongly recommends that you not do this with Exchange, and the Exchange community echoes that recommendation. It takes extra memory and processing power, and inevitably slows your disks down from what you could get with a simple investment in good hardware. You will also not be able to support higher levels of I/O load with this configuration, in my experience.

- **BIOS RAID** attempts to provide ''cheap'' RAID by putting some code for RAID in the RAID chipset, which is then placed either directly on the motherboard (common in workstation-grade and low-end server configurations) or on an inexpensive add-in card. The catch is that the card isn't really doing the RAID; it's again all happening in memory, this time in the associated Windows driver. If you're about to purchase a RAID controller card for a price that seems too good to be true, it's probably one of these cards. Although you can get Exchange to work with these, you can do so only with very low numbers of users. Otherwise, you'll quickly hit the limits these cards have and stress your storage system. Just avoid them; the time you save will more than make up for the up-front price savings.
- **Hardware RAID** is the only kind of RAID you should even be thinking about for your Exchange servers. This means good quality, high-end cards that come from reputable manufacturers that have taken the time to get the product on the Windows Hardware Compatibility List (HCL). These cards do a lot of the work for your system, removing the CPU overhead of parity calculations from the main processors, and they are worth every penny you pay for them. Better yet, they'll be able to handle the load your Exchange servers and users throw at them.

If you can't tell whether a given controller you're eyeing is BIOS or true hardware RAID, get help. Lots of forums and websites on the Internet will help you sort out which hardware to get and which to avoid. And while you're at it, spring a few extra bucks for good, reliable disks. Again, the time you save will be your own.

Network Attached Storage (NAS)

When early versions of Exchange Server came on the market, DAS was just the way you did things. As mailbox databases got larger and traffic levels rose, pretty soon people wanted to look for alternatives; DAS storage under Exchange 2000 and Exchange 2003 required a lot of disks, because they were optimized for the smaller disks that were on the market at the time.

One of the potential solutions people wanted to use was NAS devices. These machines — giant file servers — sit on the network and share out their disk storage. They range in price and configuration from small plug-in devices with fixed capacity to large installations with more configuration options than most luxury cars (and a price tag to match). Companies who bought these were using them to replace file servers, web server storage, SQL Server storage — why not Exchange?

For many years, Exchange Server wasn't compatible with NAS devices; Microsoft didn't support moving Exchange storage to NAS, and vociferously argued against the idea. In order to understand why, you have to understand the difference between file-level and block-level access.

When one network device shares storage with another device, it has two basic ways to do it.

- *File-level sharing* is the type of sharing we're used to any time somebody uses SMB/CIFS to create or mount a Windows file share; the client (the machine opening the file share) doesn't worry about how the files and folders are laid out on the physical disk, it just asks the server to perform certain operations on its behalf.
- With *block-level sharing*, the client computer is essentially presented with a raw device (the shared volume) that appears, thanks to the appropriate drivers, to be a locally attached

drive. The client manipulates the raw disk blocks and file system just as it would on a physical disk; the server simply translates the client's view of the storage into the actual view.

Most NAS devices at the time, especially the inexpensive ones, only offered file-level sharing. Though you could (in theory) move the Exchange databases to NAS volumes, doing so created huge performance bottlenecks that would quickly overwhelm your system. This happened for two main reasons: network bandwidth and server latency. As it turns out, a Fast Ethernet 100Mbps network isn't nearly as fast as the physical bus connecting your drive controller and hard drives. On top of that, Exchange expects to have a fine level of control over when specific I/O operations go to disk; when you filter that all through file-level sharing, Exchange loses that control.

With all that said, Microsoft did finally provide a way for Exchange 2003 administrators to use NAS configurations. It required the following:

- Use of a HCL-certified Windows Storage Server 2003 storage device. You couldn't do it with just any random NAS server.
- Addition of the Windows Storage Server Feature Pack, installed on both the storage server and the Exchange server.
- Adequate network connectivity. The recommendation was for a dedicated Gigabit Ethernet segment between the storage server and the Exchange servers.

Apparently, despite all of the people asking for such a configuration, it didn't turn out to be a popular option, because NAS devices are no longer supported for Exchange Server 2007. Instead, the push is on reducing the overall I/O requirements so that DAS configurations become practical for small to midsize organizations.

Storage Area Networks (SAN)

The final type of network storage used with Exchange is SAN. The premise here is to move disks to dedicated storage units that can handle all the advanced features you need — high-end RAID configurations, hot-swap replacement, on-the-fly reconfiguration, rapid disk snapshots, tight integration with backup and restore solutions, and more. This helps consolidate the overhead of managing storage, often spread out on dozens of servers and applications (and their associated staff), into a single set of personnel. Then, dedicated network links connect these storage silos with the appropriate application servers.

Initially SAN solutions used fiber optic solutions to provide the necessary bandwidth for storage operations. As a result, these systems were incredibly expensive and were used only by organizations with deep pockets. Over time, however, many vendors have appeared offering SAN solutions that were increasingly affordable even for small companies. The main reason they've been able to do so is the iSCSI protocol; block-based file access routed over TCP/IP connections. Add iSCSI with ubiquitous Gigabit Ethernet hardware, and SAN deployments have become a lot more common.

Clustering is the other factor in the growth of Exchange/SAN deployments. Exchange 2003 supported clustered configurations, but they required the cluster nodes to have a shared storage solution. As a result, any organization that wanted to deploy an Exchange cluster needed some sort of SAN solution (apart from the handful of people who stuck with shared SCSI configurations). A SAN has a certain elegance to it; you simply create a virtual slice of drive space for Exchange (called a LUN, or *logical unit number*), use Fibre Channel or iSCSI (and corresponding drivers) to present it to the Exchange server, and away you go.

However, SAN solutions don't fix all problems, even with (usually because of) their price tag. Because SANs cost so much, there is often a strong drive to use the SAN for all storage and make full use of every last free block of space. Unfortunately, Exchange's I/O characteristics are very different than those of just about any other application, and few dedicated SAN administrators really know how to properly allocate disk space for Exchange:

- SAN administrators do not usually understand that total disk space is only one component of Exchange performance. For day-to-day operations, it is far more important to ensure enough I/O capacity. Traditionally, this is delivered by using lots of physical disks (commonly referred to as ''spindles'') to increase the amount of simultaneous read/write operations supported. It is important to make sure the SAN solution provides enough I/O capacity, not just free disk space, or Exchange will crawl.
- Even if you can convince them to configure LUNs spread across enough disks, SAN administrators immediately want to reclaim that wasted space. As a result, you end up sharing the same spindles between Exchange and some other application with its own performance curve, and then suddenly you have extremely noticeable but hard-to-diagnose performance issues with your Exchange servers. Shared spindles will kill Exchange.
- Although some SAN vendors have put a lot of time and effort into understanding Exchange and its I/O needs so that their salespeople and certified consultants can help you deploy Exchange on their products properly, not everyone does the same. Many vendors will shrug off performance concerns by telling you about their extensive write caching and how good write caching will smooth out any performance issues. They're true. . .up to a point. A cache can help isolate Exchange from effects of transient I/O events, but it won't help you come Monday morning when all your users are logging in and the SQL Server databases that share your spindles are churning through extra operations.

The moral of the story is simple: don't believe that you need to have a SAN. But if you do, get the best one you can afford. Make sure that your vendors know Exchange storage inside and out; if possible, get them to put you in contact with their on-staff Exchange specialists. Have them work with your SAN administrators to come up with a storage configuration that meets your real Exchange needs.

For more information on using SANs and sizing disks for Exchange Server, see the free eBook *The Shortcut Guide to Exchange Server 2007 Storage Systems* at `http://nexus.realtimepublishers.com/SGES2k7SS.htm`.

Compliance and Governance

Quite simply, today's legal system considers e-mail to be an official form of business communication just like written memos. This means that any type of legal requirement or legal action against your organization (regarding business records) will undoubtedly include e-mail. Unless you work in a specific vertical market such as healthcare or financials, the emergence of compliance and governance as topics of import to the messaging administrator is a relatively recent event. The difference between compliance and governance can be summarized simply:

> *Governance is the process of defining and enforcing policies, while compliance is the process of ensuring that you meet external requirements.*

However, both of these goals share a lot of common ground:

- They require thorough planning to implement, based on a detailed understanding of what behaviors are allowed, required, or forbidden.
- Though they require technical controls to ensure implementation, they are at heart about people and processes.

- They require effective monitoring in order to audit the effectiveness of the compliance and governance measures.

In short, they require all the same things you need in order to effectively manage your messaging data. As a result, there's a useful framework you can use to evaluate your compliance and governance needs: Discovery, Compliance, Archival, and Retention, also known as the DCAR framework.

DCAR recognizes four key pillars of activity, each historically viewed as a separate task for messaging administrators. However, all four pillars involve the same mechanisms, people, and policies; all four in fact are overlapping facets of messaging data management. These four pillars are described in the following list:

- **Discovery** is finding messages in the system quickly and accurately, whether for litigation, auditing, or other needs. It requires:
  - Good storage design to handle the additional overhead of discovery actions.
  - The accurate and thorough indexing of all messaging data that enters the Exchange organization through any means.
  - Control over the ability of users to move data out of the messaging system through mechanisms such as personal folders (PSTs).
- **Compliance** is meeting all legal, regulatory, and governance requirements, whether derived from external or internal drivers. It requires:
  - Clear guidance on which behaviors are allowed, required, or prohibited, as well as a clear description of which will be enforced through technical means.
  - The means to enforce required behavior, prevent disallowed behavior, and audit for the success or failure of these means.
  - The ability to control and view all messaging data that enters the Exchange organization through any means.
- **Archival** is the ability to preserve the messaging data that will be required for future operations, including governance tasks. It requires:
  - Clear guidance on which data must be preserved and a clear description of procedural and technical measures will be used to enforce archival.
  - The accurate and thorough indexing of all messaging data that enters the Exchange organization through any means.
  - Control over the ability of users to move data out of the messaging system through mechanisms such as personal folders (PSTs).
- **Retention** is the ability to identify data that can be safely removed without adverse impact (whether immediate or delayed) to the business. It requires:
  - Clear guidance on which data is safe to remove and a clear description of the time frames and technical measures that will be used to enforce removal.
  - The accurate identification of all messaging data that enters the Exchange organization through any means.

- Control over the ability of users to move data out of the messaging system through mechanisms such as personal folders (PSTs).

If many of these requirements look the same, good; that emphasizes that these activities are all merely different parts of the same overall goal. You should be realizing that these activities are not things you do with your messaging system so much as they are activities that you perform *while managing* your messaging system. The distinction is subtle, but important; knowing your requirements helps make the difference between designing and deploying a system that can be easily adapted to meet your needs and one that you will constantly have to fight.

Many of these activities will require the addition of third-party solutions, even for Exchange 2007. What makes this space interesting is that a lot of these functions are now starting to be filled by hosted solutions, often at a competitive price.

Where Journaling Fits into DCAR

Journaling is a common technology that gets mentioned whenever compliance, archival, and discovery are discussed. However, I find that it often gets over-discussed. Journaling is not the end goal; it's simply a mechanism for getting data out of Exchange into some other system, which does what you're really wanting to do.

If journaling is a concern for you, you should stop and ask yourself why. What information are you trying to journal, and what do you want it for? I don't know a single Exchange administrator who has ever come up to me and said, ''I want to journal my data.'' Instead, they say, ''I need to archive my data and I have to use journaling to get it to my archival solution.'' Understanding why you need journaling will give you the ammunition you need to effectively design your Exchange organization. It will also help you identify when journaling may not be all that you need.

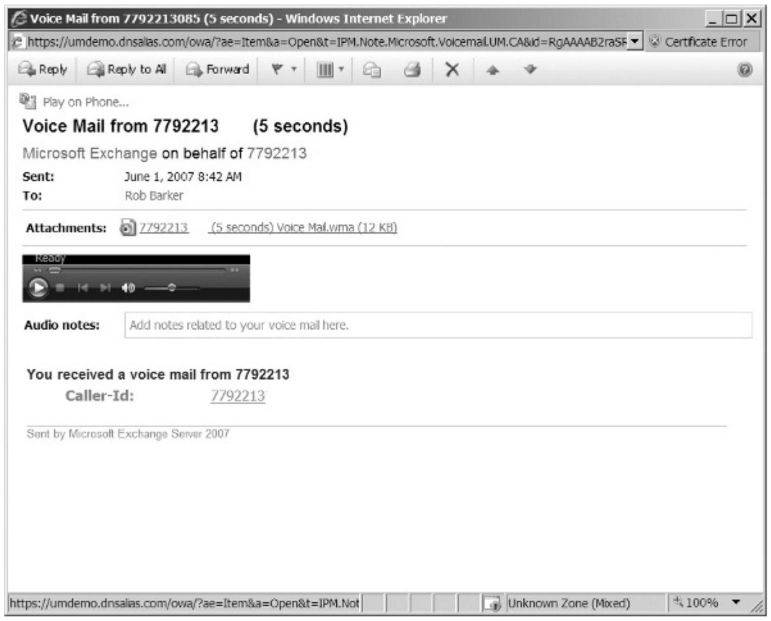
What Is Exchange Server?

In its very simplest form, Exchange Server provides the underlying infrastructure necessary to run a messaging system. Exchange Server provides the database to store e-mail data, the transport infrastructure to move the data from one place to another, and access points to access e-mail data via a number of different clients.

However, Exchange Server, when used with other clients such as Outlook or Outlook Web Access turn the ''mailbox'' into a point of storage for personal information management such as your calendar, contacts, task lists, and personal journal. Users can share some or all of this information in their own mailbox with other users on the message system and start to collaborate.

The Outlook and Outlook Web Access clients also provide access to public folders. Public folders look like regular mail folders in your mailbox, except that they are in an area where they can be shared by all users within the organization. A folder can have specialized forms associated with it to allow the sharing of contacts, calendar entries or even other specialized forms. Further, each public folder can be secured so that only certain users can view or modify data in that folder.

Exchange Server 2007's Unified Messaging server role further extends the functions of the Exchange Server in your organization by allowing your Exchange Server 2007 infrastructure to also act as your voicemail system and direct inbound faxes directly (and automatically) to the user's mailbox. Figure 1.16 shows a voice-mail message that arrived in a mailbox and can be checked via Outlook Web Access.


Voice Mail from 7792213085 (5 seconds) - Windows Internet Explorer
https://umdemo.dnsalias.com/owa/?ae=Item&a=Open&t=IPM.Note.Microsoft.Voicemail.UM.CA&id=RgAAAAB2raSF
Certificate Error
Reply
Reply to All
Forward
Play on Phone...
Ready
https://umdemo.dnsalias.com/owa/?ae=Item&a=Open&t=IPM.Not
Unknown Zone (Mixed)
100%

Voice Mail from 7792213 (5 seconds)

Microsoft Exchange **on behalf of** 7792213

Sent: June 1, 2007 8:42 AM
To: Rob Barker

Attachments: 7792213 (5 seconds) Voice Mail.wma (12 KB)

Audio notes: Add notes related to your voice mail here.

You received a voice mail from 7792213
Caller-Id: 7792213

Sent by Microsoft Exchange Server 2007

FIGURE 1.16
Checking your voice mail from Outlook Web Access

While integrated voicemail and faxing solutions are nothing new for Exchange customers, Microsoft is now providing these capabilities out of the box rather than having to rely on third-party products.

Microsoft has once again tightened the integration of collaborative tools with Office Communication Server 2007 and the Communicator client. OCS and the Communicator client can be used to integrate with Unified Messaging to provide notification of new voicemail and missed calls. Furthermore, Communicator can log chat and instant message conversation logs to a folder in the user's mailbox.

The capabilities of the client can be extended with third-party tools and forms routing software so that electronic forms can be routed through e-mail to users' desktops.

History of Exchange

The original version of Exchange Server that debuted in 1996 was called Exchange Server 4.0 because the previous version of Microsoft's e-mail offering was Microsoft Mail v3.2. However, Exchange Server 4.0 was nothing like MS Mail v3.2 in any way, shape, or form. Exchange Server v4.0 included client/server database technology, a much more comprehensive directory service, and built-in connectors for connectivity with the Internet (SMTP), as well as cc:Mail and Microsoft Mail.

Exchange continued to evolve with Exchange 5.0 and 5.5 as new capabilities such as improved Internet access, web-based e-mail, database engine optimizations, additional messaging connectors, and better scalability.

Exchange Server 2000 represented a big jump forward with even more scalability improvements, such as more databases on a single server. Exchange Server 2000 also moved from a dedicated directory service over to using Active Directory to store the Exchange configuration as well as recipient information. Internet connectivity and client interface improvements were also introduced in both Exchange 2000 as well as Exchange Server 2003.

Microsoft's Exchange Server products have played and will continue to play a key role in electronic messaging. Exchange Server 2007 is one of the most powerful, extensible, scalable, easy-to-use, and manageable electronic messaging back ends currently on the market. Combined with Microsoft's excellent Outlook clients, Internet-based clients from Microsoft and other vendors, mobile devices that use ActiveSync, and third-party or home-grown applications, Exchange Server 2007 can help your organization move smoothly and productively into the electronic messaging future.

The Universal Inbox

E-mail systems are converging with their voicemail and faxing cousins. The concept of unified messaging is nothing new to e-mail users. For at least the past 10 years, third-party vendors have included e-mail integration tools for voicemail and network faxing solutions. However, for most organizations, integrated voicemail and faxing solutions remain the exception rather than the rule. Exchange 2007 introduces integrated voice and fax solutions as part of the base product.

Organizations with IP-based telephone systems or telephone systems with an IP gateway can now easily integrate a user's voicemail and inbound faxing with the Exchange user's mailbox. The Exchange 2007 Unified Messaging server handles the interaction between an organization's telephone system and Exchange mailboxes. Inbound voicemail is transferred into the user's mailbox as a WMA file attachment; this message includes an Outlook form that allows the user to play the message. A short voicemail message may be anywhere from 40KB to 75KB in size, whereas longer voicemail messages may be 200KB to 500KB in size. One estimate that is frequently used for the size of a voicemail message is around 5KB per second of message.

Inbound faxes are transferred to the user's mailbox as messages containing an Outlook form with a Group IV TIFF attachment; a single-page fax can be as small as 25KB, whereas multipage faxes can easily be 200KB or larger. Incorporating third-party scanning and outbound faxing products (outbound faxing is not supported out of the box with Exchange 2007) can further increase the size of a mailbox.

With Outlook Voice Access, users can now dial in to the Exchange 2007 Unified Messaging server and access their mailbox, have e-mail read to them, have appointments read to them, and move or cancel appointments. If an appointment is changed, Outlook Voice Access will automatically notify attendees of scheduling changes; this is very useful if you are sitting in traffic on the freeway with nothing but your cell phone (using your headset of course)!

Inbound voicemail and inbound faxes will increase the demands on your Exchange server from the perspective of required disk space and possibly additional server hardware, though. This needs to be considered. Outlook Voice Access will increase the potential number of connections and usage of your Exchange mailbox servers and Unified Messaging servers.

Many Modes of Access

For years, the only point of access for one's e-mail system was to use a Windows, Macintosh, or Unix-based client and access the e-mail system directly. In the case of Outlook and Exchange, this access was originally in the form of a MAPI client directly against the Exchange server. As Exchange has evolved, POP3 and IMAP4 access has been included in the product, then web-based e-mail access, and finally mobile device access. Exchange Server 2007 supports additional technologies such as web services that can provide additional mechanisms for accessing data in mailboxes.

Outlook Web Access has evolved quickly and in Exchange 2007 bears almost no resemblance to the original version found in Exchange 5.0 in terms of features, functions, and the look of the interface.

Mobile device access was first provided to Exchange 2000 using Microsoft Mobile Information Server and then later included as part of Exchange 2003. Mobile device functionality has been further improved in Exchange 2007. Users are more frequently asking for integration of mobile devices with e-mail. The Radicati Group estimated that in 2006 there were 14 million wireless e-mail users but by 2010 that number will grow to 228 million. You can bet that your users will want to be included!

Unified Messaging and Outlook Voice Access now allow a user with nothing but a telephone to access his or her e-mail and calendar and even make changes via the telephone.

With all of these mechanisms for retrieving and sending e-mail, it is not unusual for users to be accessing their mailbox using more than one. In some cases, I have seen a single user accessing her mailbox from her desktop computer, her notebook computer (using RPC over HTTP), and her Windows Mobile device.

In medium and large organizations, the fact that users are now accessing their mailbox from more than one device and/or mechanism will affect not only hardware sizing but also, potentially, your licensing costs.

Architecture Overview

Understanding a bit about how Exchange Server works from an architectural perspective will help make you a better administrator. You don't have to be able to reproduce or write your own client/server messaging system, but it helps to know the basics.

Exchange as a Client/Server Mail System

Since Exchange Server 4.0, Exchange has been a client/server messaging system. Remember back in Figure 1.7, how the client sends a request to the Exchange server, the Exchange server does the work, and then sends back only the response to the client? Well, that is how Exchange Server works; the mechanism that the Outlook client uses is the Messaging Application Programming Interface (MAPI); the data is sent using Remote Procedure Calls (RPCs). The underlying network infrastructure uses whatever the network transport is to move the RPC request from the client to the server. In years past, this might have been IPX/SPX or even NetBEUI, but it is very rare that you will find any network transport today other than TCP/IP. MAPI clients such as Outlook 2003 or 2007 directly access an Exchange Mailbox server using MAPI over RPCs, such as is shown in Figure 1.17.

The client/server architecture does not just stop at Outlook clients, though; Internet clients such as Outlook Web Access, Outlook Anywhere, POP3, IMAP4, Windows Mobile, and the iPhone all go through a ''middle-ware'' layer to get to their data. In the case of Exchange Server 2007, ''web'' access has been abstracted from the mail storage system and is run by a server role called the Client Access server. Figure 1.17 shows a simple diagram of how this works.

The Extensible Storage Engine (ESE)

The Exchange Server database uses a highly specialized database engine called the Extensible Storage Engine (ESE). Generically, you could say it is almost like SQL Server, but this is technically not true. It is a client/server database and is somewhat relational in nature, but it is designed to be a single-user database (the Exchange server itself is the only component that directly accesses the data). Further, the database has been highly tuned to store hierarchical data such as mailboxes, folders, messages, and attachments.

Without going into a lot of techno-babble on the database architecture, it is important that you understand the basics of what the database is doing. Figure 1.18 show conceptually what is happening with the ESE database as data is sent to the database. In step 1, an Outlook client sends data to the Exchange Server (the information store service), the information store service places this data in memory and then immediately writes the data out to the transaction log files associated with that database.

FIGURE 1.17
Client Access server architecture

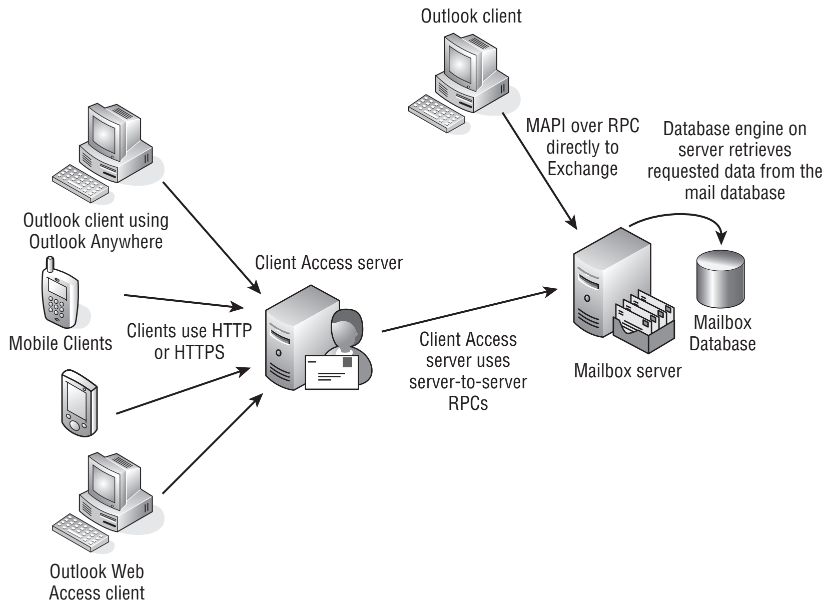
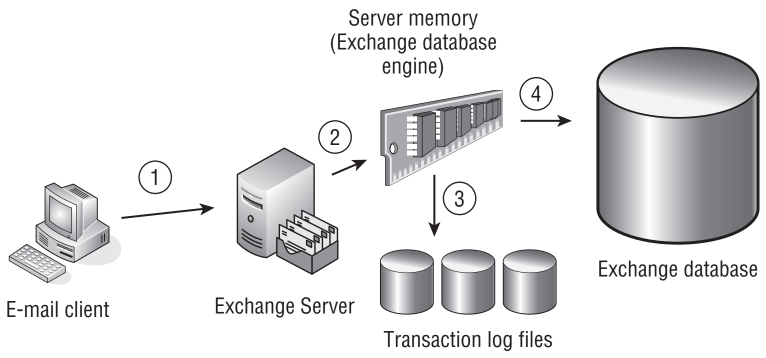

Outlook client
MAPI over RPC directly to Exchange
Database engine on server retrieves requested data from the mail database
Outlook client using Outlook Anywhere
Client Access server
Mobile Clients
Clients use HTTP or HTTPS
Client Access server uses server-to-server RPCs
Mailbox server
Mailbox Database
Outlook Web Access client

FIGURE 1.18
Exchange data and transaction logs



The transaction log that is always written to is the current transaction log for that particular database (`e00.log` for example). Each transaction log file is exactly 1MB in size, so when the transaction log is filled up, it is renamed to the next sequential number. For example, an old transaction log file might be named like this: `e000004032.log`.

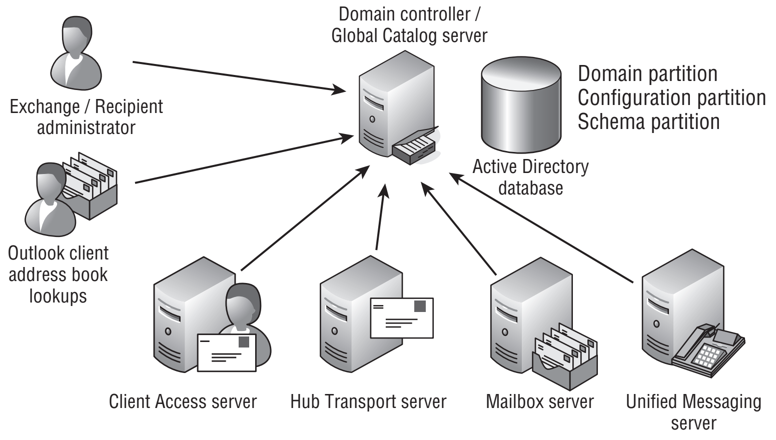
The data is retained in RAM for some period of time (maybe 5 seconds or maybe 60 seconds) before it is then flushed to the database file. The actual period that data is retained in memory will depend on how much cache memory is available and how busy the server is. The important operation, though, is to make sure that as soon as the data is sent to the Exchange server it is immediately flushed to the transaction log files. If the server crashes before the data is written to the database file, the database engine (the information store service) will automatically ‘‘replay’’ the missing data by reading the transaction log files once the server is brought back up.

The transaction log files are important for a number of reasons. They are used by Microsoft replication technologies (as you learn in Chapter 18, ''Implementing Replication Technologies'') but they can also be used in disaster recovery. The transaction logs are not actually purged off the log disk until a full backup is run; therefore every transaction that occurred to a database (new data, modifications, moves, deletes) is stored in the logs. If you restore the last good backup to the server, Exchange server can replay and rebuild all of the missing transactions back in to the database — provided you have all the transactions since the last full backup.

Exchange and Active Directory

I could easily write two or three chapters on how Exchange Server interacts with the Active Directory, but the basics will have to do for now. Exchange Server relies on the Active Directory for information about its own configuration, user authentication, and e-mail specific properties for mail-enabled objects such as users, contacts, groups, and public folders. Look at Figure 1.19 to see some of the different types of interactions that occur between Exchange and the Active Directory. Because most of the Exchange configuration data for an Exchange server is stored in the Active Directory, all Exchange server roles must contact a domain controller to request its configuration data; this information is stored in a special partition of the Active Directory database called the configuration partition. The configuration partition is replicated to all domain controllers in the entire Active Directory forest.

Figure 1.19
Active Directory and Exchange


Domain controller / Global Catalog server
Domain partition
Configuration partition
Schema partition
Active Directory database
Exchange / Recipient administrator
Outlook client address book lookups
Client Access server
Hub Transport server
Mailbox server
Unified Messaging server

Each of the individual Exchange Server roles uses the Active Directory for different things. Following is a list of some of those functions:

Mailbox servers Exchange Mailbox servers must query the Active Directory to authenticate users, enumerate permissions on mailboxes, look up individual mailbox limits, and determine which mailboxes are on a particular server.

Hub Transport servers Exchange Hub Transport servers require access to global catalog servers in order to look up e-mail addressing information, home server information, distribution list membership information, and other data related to message routing.

Client Access servers Exchange Client Access servers require access to the Active Directory to look up information about home servers for users, ActiveSync, and Outlook Web Access user restrictions.

Unified Messaging servers Unified Messaging servers require access to the Active Directory in order to retrieve and play the user's personalized outgoing message as well as to retrieve e-mail address information so that voicemail and faxes can be delivered to the user.

Exchange management tools The Exchange Server management tools must connect to the Active Directory in order to make configuration changes to Exchange server objects and to create, update, manage, or delete mail-enabled objects such as mailbox-enabled users or mail-enabled groups.

Outlook clients Outlook clients require access to Active Directory global catalog servers in order to retrieve information about the global address lists as well as individual recipient information.

Controlling Mailbox Growth

As users have become more savvy and competent at using Outlook and the features of Exchange, and e-mail messages themselves have become more complex, the need for e-mail storage has grown. Back in the days of Exchange 4.0, an organization that gave its users a 25MB mailbox was considered generous. With Exchange 2003, a typical user's mailbox may have a storage limit of 300 to 500MB, with power users and VIPs requiring even more.

At TechEd 2006, Exchange gurus were tossing about the idea that in the future a default mailbox limit would be closer to 2GB as users start incorporating Unified Messaging features. We all see users with mailbox sizes in the gigabyte range, but is your organization prepared for a typical user with a 2GB message size limit? What sort of concerns will you face when your average user has 1 to 2GB of content (not just e-mail!) in his mailbox?

Certainly the need for more disk storage will be the first factor that organizations need to consider. However, disk storage is reasonably cheap, and many larger organizations that are supporting thousands of mailbox users on a single mailbox server already have more disk space than they can practically use. This is due to the fact that they require more disk spindles to accommodate the number of simultaneous I/Os per second (IOPS) that are required by a large number of users.

For more administrators with large amounts of mail storage, the primary concern they face is the ability to quickly and efficiently restore data in the event of a failure. These administrators are often faced with service level agreements that bind them to maximum restoration times. In even the most optimal circumstances, a 300GB mailbox database will take some time to restore from backup media!

Microsoft recommends that you do not allow an Exchange mailbox database to grow larger than 100GB unless you are implementing any of the new continuous replication technologies in Exchange 2007. If you use local continuous replication or clustered continuous replication to keep a copy of the database ready to use in case of database corruption, do not let the mailbox database grow to larger than 200GB. If you require more than 100–200GB of mailbox database storage, Exchange 2007 Standard Edition allows you to have up to 5 mailbox databases and Exchange 2007 Enterprise Edition allows you to have up to 50.

The solution in the past was to restrain the user community by preventing them from keeping all of the mail data that they might require on the mail server. This was done by imposing low mailbox limits, implementing message archival requirements, keeping deleted items for only a few days, and keeping deleted mailboxes for only a few days.

However, as Unified Messaging data now starts to arrive in a user's mailbox and users have additional mechanisms for accessing the data stored in their mailbox, keeping mail data around longer is going to be a demand and a requirement for your user community.

Personal Folders or PST Files

The Outlook personal folder or PST file can be the very bane of your existence. Outlook allows users to create a local database in which they can create folders and archive e-mail. Though this seems like a good feature on the surface, there are a few downsides:

- The data in PST files take up more space than the corresponding data on the server.
- The default location for a PST is the local portion of the user's profile; this means it is stored on the local hard disk of their computer and is not backed up.
- PST files can get corrupted much more easily than the data on the Exchange server.
- Performance when accessing PST files is not very good once a PST file is around 1GB or larger. And you should never allow a PST file to grow larger than 2GB even though Outlook 2003 and Outlook 2007 will allow you to do so.
- Once data is in a user's PST file, you, as the server administrator, have lost control of it. If you ever had to find all copies of a certain message, perhaps for a lawsuit, you would be out of luck.

E-Mail Archiving

Sometimes, managing a mail server seems like a constant race between Information Technology (IT) and users to keep users from letting their mailbox run out of space. Users are pack rats and generally want to keep everything. If there is an actual business reason for them to do so, then you should look at ways to expand your available storage to accommodate them.

However, as databases become larger and larger, the Exchange server will be more difficult to manage. You might start requiring hundreds and hundreds of gigabytes (or even terabytes) of storage for e-mail databases. Worse still, backups and data recovery take longer.

This is where e-mail archiving becomes useful. The last time I counted, there were more than 45 companies in the business of supplying e-mail archiving tools and services. Archiving products all have a lot of functions in common, including the ability to keep data long term in the e-mail archival, to allow the users to search for their own data, and to allow authorized users to search the entire archive.

If you look at how e-mail is archived, archive systems generally come in one of three flavors:

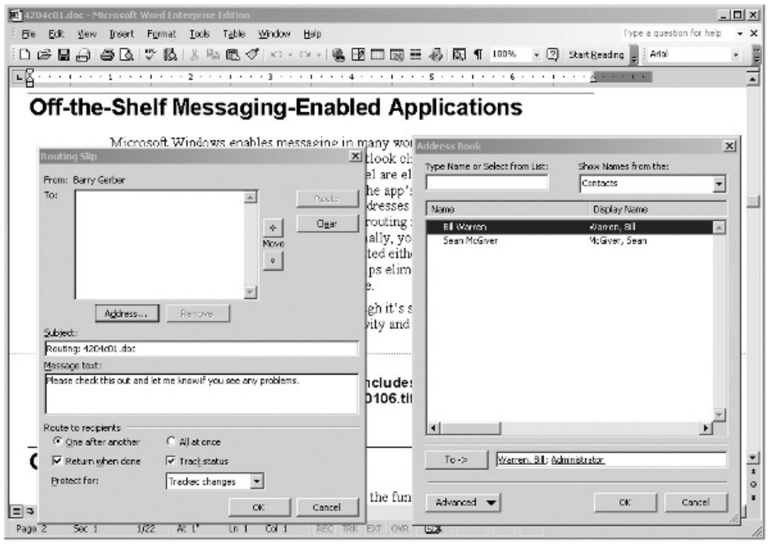
- Systems that depend on journaling to automatically forward every e-mail sent or received by specified users on to the archive system.
- Systems that perform a scheduled MAPI ''crawl'' of specified mailboxes, looking for messages that are eligible to be moved or copied to the archive.
- Systems that move data to the archive by copying the log files from the production mailbox servers and then replaying the logs in to the archive. This is called log shipping.

Each of these methods has its advantages and disadvantages with respect to storage, providing a complete archive, and performance overhead.

Messaging-Enabled Applications

Microsoft Office enables messaging in many word-processing and spreadsheet applications. For example, when you install the Outlook client on your computer, Microsoft's Office products such as Word and Excel are enabled for electronic messaging. Let's say you want to send a Word document. You can select the Routing Recipient option from the application's File ➢ Send To menu. An electronic routing slip pops up. You then add addresses to the slip from your Exchange address books or from your Outlook contacts, select the routing method you want to use, and set other attributes for the route. Finally, you add the routing slip to the document with a click of the Add Slip button and ship it off to others using options on the File ➢ Send To menu.

As you can see in Figure 1.20, a file can be routed either sequentially or all at once to each address you selected. Routing sequentially helps eliminate problems associated with multiple users editing the same file at the same time. With applications such as Microsoft Word that keep track of each person's comments and changes, once the document has been routed, the original author can read the comments and incorporate or not incorporate them as he sees fit.


4204c01.doc - Microsoft Word Enterprise Edition
File Edit View Insert Format Tools Table Window Help
Start Reading
Off-the-Shelf Messaging-Enabled Applications
Routing Slip
Route
Clear
Move
Address...
Remove
OK
Cancel
Address Book
Type Name or Select from List:
Show Names from the:
Name
Display Name
To ->
Advanced
OK
Cancel

Routing Slip

From: Barry Gerber

Subject: Routing: 4204c01.doc

Message text: Please check this out and let me knowif you see any problems.

Route to recipients: (•) One after another ( ) All at once

☑ Return when done ☑ Track status

Protect for: Tracked changes

Address Book

Show Names from the: Contacts

Name	Display Name
Bill Warren	Warren, Bill
Sean McGiver	McGiver, Sean

To -> Warren, Bill; Administrator

FIGURE 1.20 Microsoft Word 2003 includes messaging-enabled functions for sending and routing.

Although it's simple, application-based messaging can significantly improve user productivity and speed up a range of business processes.

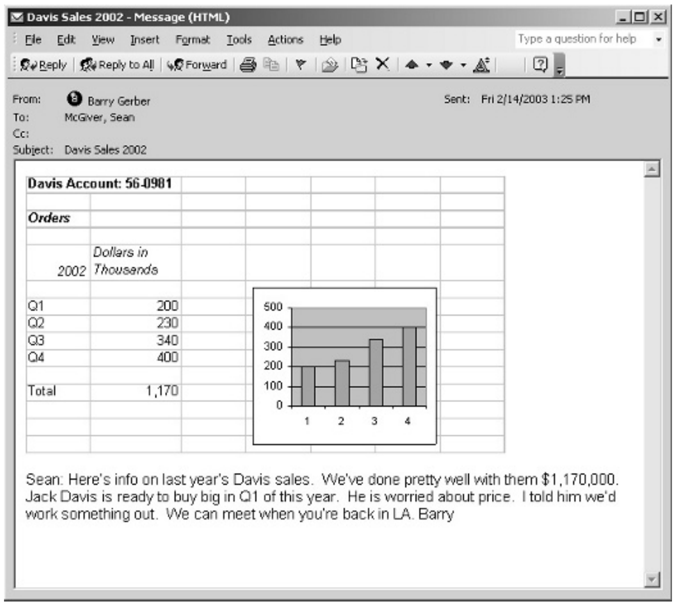
Objects

Object insertion and linking further enhance the functionality of the Exchange messaging system. Take a close look at Figure 1.21. Yes, the message includes an Excel spreadsheet and chart. The person who sent the message simply selected Object from the Insert menu that appears on every Exchange message. Then he specified a file with an existing spreadsheet as the source of the object to be inserted into the message. The Outlook client then inserted the file into the message as an object.

The recipient can see the spreadsheet as a graphic image in the message, as shown in the figure. When the recipient double-clicks the graphic image, Excel is launched inside the message, and Excel's menus and toolbars replace those of the message (see Figure 1.22). In essence, the message becomes an Excel document.

The Excel spreadsheet is fully editable if Excel is available to the recipient. Without Excel, the recipient can only look at the spreadsheet in graphic image form. The graphic image changes when someone else edits the spreadsheet in Excel.

FIGURE 1.21
Object insertion makes it easy to create sophisticated messaging-enabled applications.


Davis Sales 2002 - Message (HTML)
File Edit View Insert Format Tools Actions Help
Type a question for help
Reply Reply to All Forward

From: Barry Gerber Sent: Fri 2/14/2003 1:25 PM
To: McGiver, Sean
Cc:
Subject: Davis Sales 2002

Davis Account: 56-0981	
Orders	
2002	*Dollars in Thousands*
Q1	200
Q2	230
Q3	340
Q4	400
Total	1,170

Sean: Here's info on last year's Davis sales. We've done pretty well with them $1,170,000. Jack Davis is ready to buy big in Q1 of this year. He is worried about price. I told him we'd work something out. We can meet when you're back in LA. Barry

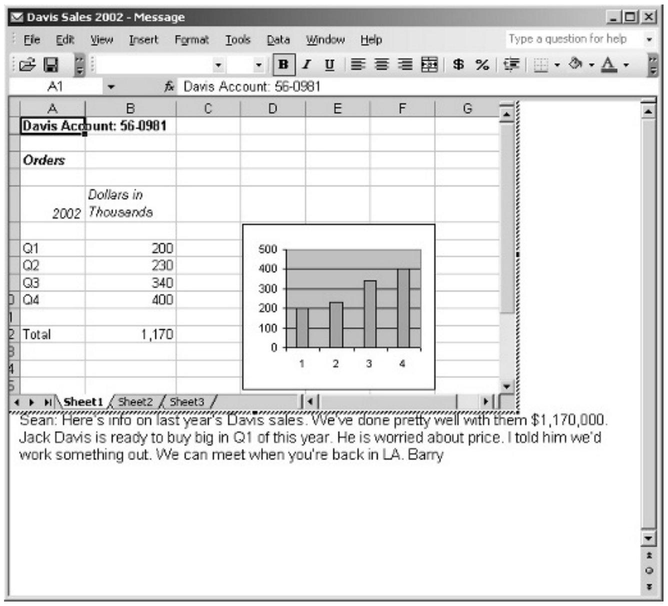
You can also insert in a message an object that is a link to a file that was created by an application such as Word or Excel. As with other kinds of object insertion, your recipient sees a graphic picture of the contents of the file and can edit the file by double-clicking the graphic picture. Links are a bit more flexible, because they allow users to work with files stored on a shared disk. With inserted objects, users work with a file embedded in the message itself.

Technically, this capability is provided not by the Exchange server, but by the Outlook client. However, OWA includes some of this same rich functionality, and the Exchange database and protocols are certainly built to support it, so many users consider it a feature of the Exchange system.

Public Folders

Public folders are for common access to messages and files. Files can be dragged from file-access interfaces, such as Windows Explorer, and dropped into public folders. The whole concept of public folders has many organizations in a quandary as they try to figure out the best place for these collaborative applications. Increasingly, applications that were once ''best suited'' for a public folder are now better suited for web pages or portals such as SharePoint workspaces. Although the whole concept of public folders is being deemphasized in Exchange 2007, this release continues to support public folders and many organizations will continue to find useful applications for public folders for the foreseeable future.

You can set up sorting rules for a public folder so that items in the folder are organized by a range of attributes, such as the name of the sender or creator of the item or the date that the item was placed in the folder. Items in a public folder can be sorted by conversation threads. Public folders can also contain applications built on existing products such as Word or Excel or built with Exchange or Outlook Forms Designer, client or server scripting, or the Exchange API set. You can use public folders to replace many of the maddening paper-based processes that abound in every organization.



	A	B
1	Davis Account: 56-0981	
2		
3	Orders	
4		
5	2002	Dollars in Thousands
6		
7	Q1	200
8	Q2	230
9	Q3	340
10	Q4	400
11		
12	Total	1,170

Sean: Here's info on last year's Davis sales. We've done pretty well with them $1,170,000. Jack Davis is ready to buy big in Q1 of this year. He is worried about price. I told him we'd work something out. We can meet when you're back in LA. Barry

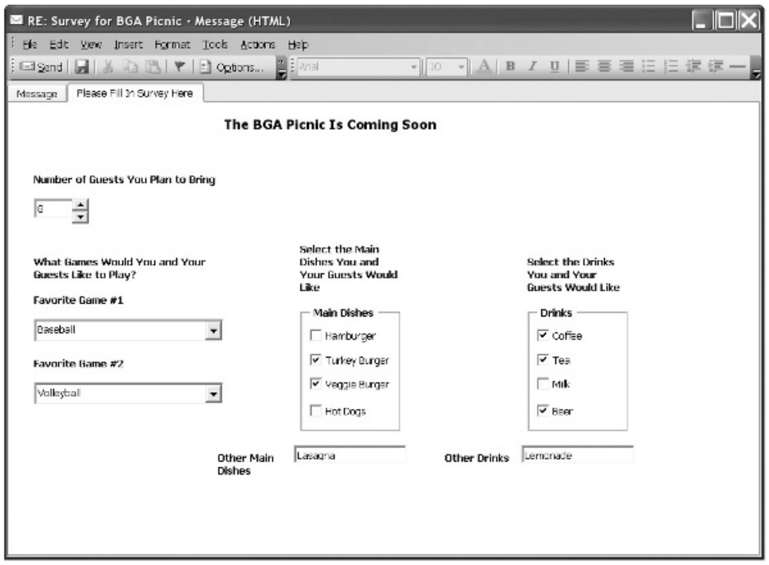
FIGURE 1.22
Double-clicking an Excel spreadsheet object in a message enables Excel menus and toolbars.

For easy access to items in a public folder, you can use a *folder link*. You can send a link to a folder in a message. When someone navigates to the folder and double-clicks a file, the file opens. Everyone who receives the message works with the same linked attachment, so everyone reads and can modify the same file. As with document routing, applications such as Microsoft Word can keep track of each person's changes to and comments on file contents. Of course, your users will have to learn to live with the fact that only one person can edit an application file at a time. Most modern end-user applications warn the user when someone else is using the file and if so allow the user to open a read-only copy of the file, which of course can't be edited.

Electronic Forms

Exchange Server 2007 continues to supports forms created with the Outlook Forms Designer (OFD). You can use OFD to build information-gathering forms containing a number of the bells and whistles that you're accustomed to in Windows applications. These include drop-down list boxes, check boxes, fill-in text forms, tab dialog controls, and radio buttons (see Figure 1.23).

OFD, which is easy enough for nontechnical types to use, includes a variety of messaging-oriented fields and actions. For example, you can choose to include a preaddressed To field in a form so that users of the form can easily mail it off to the appropriate recipient. (The preaddressed To field for the form shown in Figure 1.23 is on the page with the tab marked Message, which is not visible in this figure.) When you've designed a form, you can make it available to all users or only to select users; users can access the completed form simply by selecting it while in an Outlook client.


RE: Survey for BGA Picnic - Message (HTML)
File Edit View Insert Format Tools Actions Help
Send
Options...
Message
Please Fill In Survey Here

The BGA Picnic Is Coming Soon

Number of Guests You Plan to Bring: 0

What Games Would You and Your Guests Like to Play?

Favorite Game #1: Baseball

Favorite Game #2: Volleyball

Select the Main Dishes You and Your Guests Would Like

Main Dishes
- [ ] Hamburger
- [x] Turkey Burger
- [x] Veggie Burger
- [ ] Hot Dogs

Other Main Dishes: Lasagna

Select the Drinks You and Your Guests Would Like

Drinks
- [x] Coffee
- [x] Tea
- [ ] Milk
- [x] Beer

Other Drinks: Lemonade

FIGURE 1.23
Electronic forms turn messages into structured information-gathering tools.

Summary

I hope that this chapter has given you some idea of what it is like to be an e-mail administrator and some of the things that you need to know. I have been using e-mail for almost 30 years and managing e-mail systems in one form or another for 20 years. I have watched how important e-mail has become to an organization over these past 20 years and I see how much organizations and users depend on their e-mail.

This gives me a source of pride to be able to provide such services to the users and to work in an area that is almost always changing with new technologies, features, and enhancements. As you now start your journey toward learning more about Microsoft Exchange Server 2007, I hope you too will find this as interesting and challenging as I do.