Part 1

Exchange Fundamentals

Chapter 1

Putting Exchange Server 2013 in Context

Email is one of the most visible services that IT professionals provide; most organizations have become dependent on "soft" information to run their business. As a result, users have developed an attachment to email that goes beyond the hard value of the information it contains. If there's a problem with email, it affects users' confidence in their ability to do their jobs—and their confidence in IT.

Microsoft's Exchange Server products play a key role in electronic messaging, including email. This chapter is a high-level primer on Exchange Server–based email administration and good administration practices, and it prepares you to put Exchange Server 2013 into the proper context. An experienced email administrator may want to proceed to more-technical chapters. However, if you are new to the job or need a refresher, or maybe you just want to put email services back into perspective, this chapter is for you!

In this chapter, you will learn to:

◆ Understand email fundamentals

◆ Identify email-administration duties

Email's Importance

If you're responsible for electronic messaging in your organization, no one has to tell you about its steadily expanding use—you see evidence 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 aspects of electronic mail and the ever-changing nature of email. Even experienced Exchange Server administrators may want to review this section to better understand how their users and requirements are evolving.

Billions of emails are sent every day (more than 500 billion worldwide, according to Research firm The Radicati Group). That's a lot of email messages, on a lot of servers—many of them Exchange servers.

Sure, sending simple text email and file attachments is the most basic function, but email 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.

- Provide notifications of workflow processes, such as finance/accounting activities, IT events (server status information), and more.

- Archive important attachments, text messages, and many other types of information.

- Allow users to access their "email data" through a variety of means, including clients running on Windows computers, Apple computers, Unix systems, web browsers, mobile phones, and even a regular telephone.

- Perform records management and enable long-term storage of important information or information that must be archived.

- Enable near-time communication of sales and support information with vendors and customers.

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

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 different ways depending on the vendor or even the version of the product. Exchange Server has evolved dramatically over the past 18 years, and its current architecture is almost nothing like Exchange Server 4.0 from 1996. Common components of most messaging systems include the following:

- A message transport system that moves messages from one place to another. Examples include the Simple Mail Transport Protocol (SMTP).

- 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 email 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, 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 App, and a mobile device such as a Windows Phone, an iPhone, or an Android device.

Working in tandem with real-time interactive technologies, electronic messaging systems have already produced a set of 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 email system usually deploy additional components from their email software vendor or third parties that extend the capabilities of the email system or provide required services. These include the following:

◆ Integration with existing phone systems or enterprise voice deployments to pull voice messages into the mailbox

◆ Message-hygiene systems that help reduce the likelihood of a malicious or inappropriate message being delivered to a user

◆ Backup and recovery, disaster recovery, and business continuity solutions

◆ Message archival software to allow for the long-term retention and indexing of email data

◆ Electronic forms routing software that may integrate with accounting, order entry, or other line-of-business applications

◆ Mail gateways to allow differing mobile devices, such as BlackBerry devices, to access the mail server, along with native access through Exchange ActiveSync

◆ Email security systems that improve the security of email data either while being transferred or while sitting in the user's mailbox

◆ A link load balancer to balance the load between multiple Internet-facing servers or internal servers

What Is Exchange Server?

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

However, Exchange Server, when used with other clients such as Outlook or Outlook Web App, turns the "mailbox" into a point of storage for personal information management such as your calendar, contacts, task lists, and any file type. 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 App 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.

The Unified Messaging features in Exchange Server 2013 further extend the functions of Exchange Server in your organization by allowing your Exchange Server infrastructure to also act as your voicemail system and direct voicemails and missed-call notifications automatically to the user's mailbox.

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

Exchange Server 2013 tightens the integration of collaborative tools in its integration with Lync Server 2013, the Lync client, and the Lync Mobile client. Lync provides a core set of SIP-based enterprise voice capabilities that allows it to act as a PBX in many cases. With Exchange Server, Lync, Outlook, and the Lync client, users enjoyed full Unified Messaging with software-based telephony from their computer, including the new voicemail and missed-call notification provided by Exchange Server and Outlook. Furthermore, Lync could log chat and instant-message conversation logs to a folder in the user's mailbox. Exchange Server 2013 further pushes this integration, embedding basic IM and presence capabilities into the Outlook Web App premium experience.

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

About Messaging Services

Electronic messaging is now far more than email. Together, Exchange Server 2013 and its clients perform a variety of messaging-based functions. These functions include email, unified messaging, message routing, scheduling, and support for several types of custom applications. Together these features are called messaging services.

Many Modes of Access

For years, the only way to access your email system was to use a Windows, Macintosh, or Unix-based client and access the email system directly. In the case of Outlook and Exchange Server, this access was originally in the form of a MAPI client directly against the Exchange server. As Exchange Server has evolved, it has included support for the POP3 and IMAP4 protocols, then web-based email access, and finally mobile device access. Exchange Server 2013 doesn't offer any radically new modes of mailbox access as Exchange Server 2007 did, but it does provide ongoing support and refinement of existing Exchange Server 2007 technologies, such as Exchange Web Services, that can provide additional mechanisms for accessing data in mailboxes and a move away from RPC in client connectivity in favor of RPC over HTTPs, also known as Outlook Anywhere.

Outlook Web App (OWA) has evolved quickly and, in Exchange Server 2013, bears almost no resemblance to the original version found in Exchange Server 5.0 in terms of features, functions, and the look of the interface. Exchange Server 2013 OWA is even a radical step beyond Exchange Server 2010. It also expands the previous option configuration experience of the Exchange Control Panel (ECP), which gives users a much greater degree of control over their mailbox, contacts, and group memberships. ECP is now built into the Outlook Web App interface. Using ECP, end users can create and join distribution groups (where permissions have been assigned), track their own messages throughout the organization, and perform other functions that previously required help-desk or IT professional intervention. Another significant feature of Outlook Web App is the ability to use the web-based interface when working offline and completely disconnected from the network.

With Exchange Server 2013, Exchange ActiveSync (EAS) continues to offer significant partnerships with and control over mobile devices. Many vendors have licensed EAS to provide their mobile devices with a high-performance, full-featured push mobile synchronization experience now extending beyond mobile phones and into tablet devices.

With all of these mechanisms for retrieving and sending email, it is not unusual for users to access their mailbox using more than one device. In some cases, we have seen a single user accessing her mailbox from her desktop computer, her tablet device using Outlook Anywhere, and her Windows Phone device.

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

WHAT'S GONE?

When Exchange Server 2007 was released, Microsoft introduced new core APIs (including Web Services, the new management API based on the .NET Framework) intended to replace existing Exchange Server APIs. Several of those legacy APIs were completely removed, whereas others were *deprecated*—while they still worked, developers were encouraged to port their applications over to the new APIs. The deprecated APIs were not guaranteed to be continued in future versions of Exchange Server.

With Exchange Server 2010, those deprecated APIs were eliminated. One of the biggest was WebDAV, which was the previous HTTP-based access protocol prior to Exchange Web Services. WebDAV calls are somewhat simpler to develop but are more fundamentally limited in what they can do.

HOW MESSAGING SERVICES ARE USED

Certainly, email 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 2013 together with Exchange Server 2013 introduces even more synergy. Figure 1.1 and Figure 1.2 show the Outlook 2013 client Calendar and Inbox in action.

FIGURE 1.1
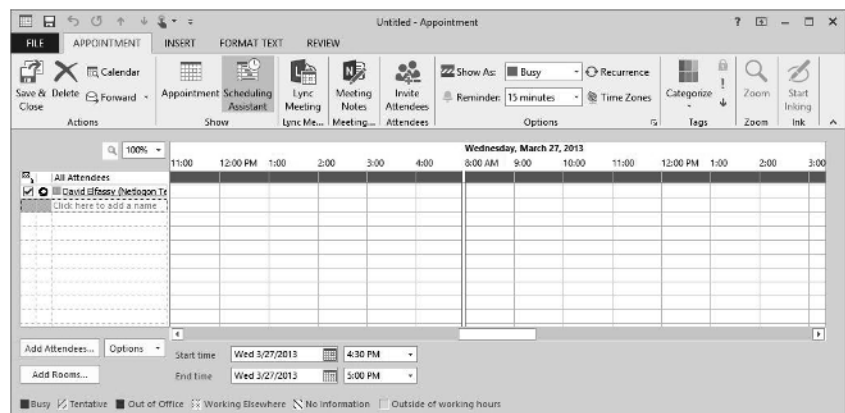Outlook 2013 Appointment scheduling on an Exchange Server 2013 mailbox

FIGURE 1.2
The Outlook 2013 client Inbox on an Exchange Server 2013 mailbox
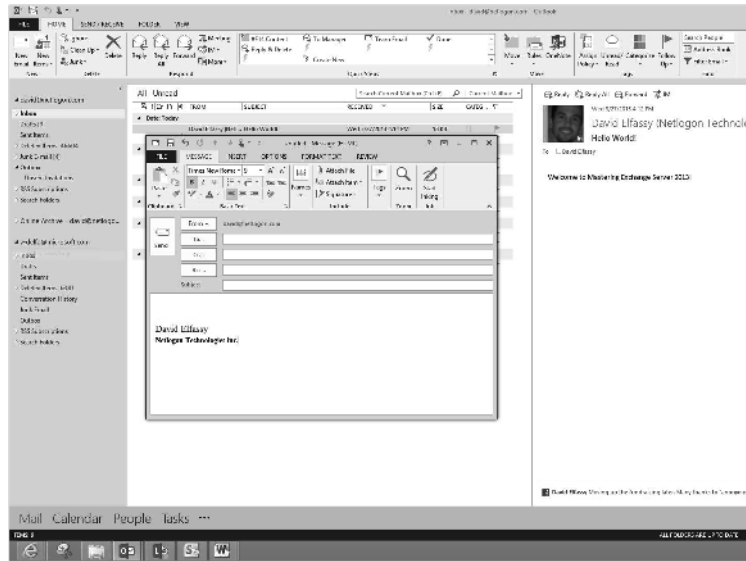
Figure 1.3 shows the new Outlook Web App 2013 web browser client. For the first time, Outlook Web App 2013 provides the full premium user experience for browsers other than Internet Explorer; it also supports Mac OS X Safari, Firefox, and Chrome. Those coming from previous versions will immediately notice a cleaner, less-cluttered interface in OWA 2013 and amazing new functionalities such as Offline Usage.

FIGURE 1.3
Outlook Web App on an Exchange Server 2013 mailbox
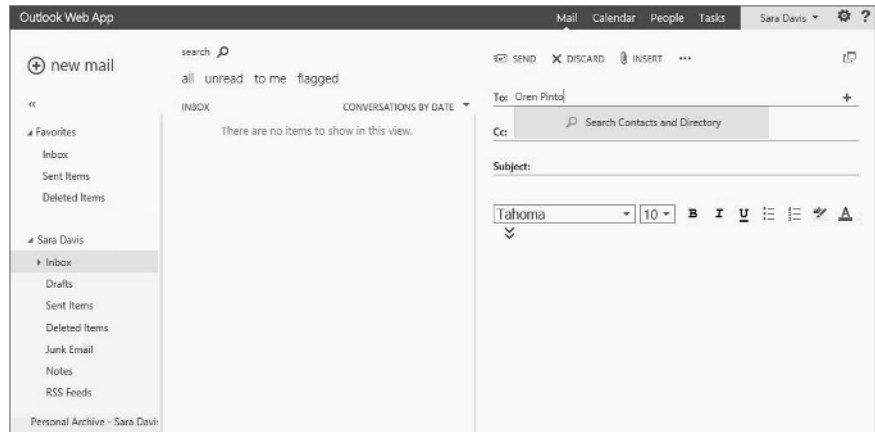
Email clients are exciting and sexy, but to get the most out of Exchange Server 2013 you need to throw away any preconceptions you have that messaging systems are only for email and scheduling. The really exciting applications are not those that use simple email or scheduling but those that are based on the routing capabilities of messaging systems. These applications bring people and computers together for improved collaboration.

The Universal Inbox

Email systems are converging with their voicemail and enterprise voice-solution cousins. The concept of unified messaging is nothing new to email users. For the past 20 years, third-party vendors have included email integration tools for voicemail, network faxing solutions, and third-party integration. However, for most organizations, integrated voicemail remains the exception rather than the rule. Exchange Server 2007 introduced integrated voice, which Exchange Server 2013 continues to improve on.

Organizations with IP-based telephone systems or telephone systems with an IP gateway can easily integrate a user's voicemail with the Exchange Server user's mailbox. The Exchange Server 2013 Unified Messaging features handle the interaction between an organization's telephone system and Exchange Server mailboxes. Inbound voicemail is transferred into the user's mailbox as a cross-platform-friendly MP3 file attachment; this message includes an Outlook or OWA form that allows the user to play the message. As well, the voicemail text can be transcribed into the body of the email message for quick reading by the user during meetings or rapid glancing at the Inbox. Because the default format is MP3 in Exchange Server 2013 (it was a Windows Media file in Exchange Server 2007, using a custom codec), this file can be easily played on mobile devices from any manufacturer, allowing easy on-the-go access to voicemail. A short voicemail message may be anywhere from 40 KB to 75 KB in size, whereas longer voicemail messages may be range from 200 KB to 500 KB in size. One estimate that is frequently used for the size of a voicemail message is around 5 KB per second of message.

Inbound voicemail increases the demands on your Exchange server from the perspective of required disk space and possible additional server hardware. As an administrator, you need to consider this.

Just the Fax, Ma'am

In Exchange Server 2007, the Unified Messaging features included the out-of-the-box capability to capture incoming facsimile (fax) messages. There were some limitations, but it provided good basic functionality. For outbound fax capability, organizations had to deploy some other solution, typically a third-party fax package.

For Exchange Server 2010 and Exchange Server 2013, Microsoft made the decision to cut this feature. When talking with the product group, it's not hard to figure out why; the inbound-only fax functionality wasn't enough for the customers who needed fax integration. Exchange Server 2013 needed to either add outgoing fax capability and beef up its feature set (and lose other desired functionality) or drop the existing functionality since the majority of Exchange Server 2007 customers needed a third-party product anyway. While it's always disappointing to lose a feature, most of the organizations we've talked to didn't use it to begin with. We think that Microsoft definitely made the right call, if you'll pardon the pun.

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.

THE EXTENSIBLE STORAGE ENGINE

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 1.4 shows 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 1.4
Exchange data and transaction logs
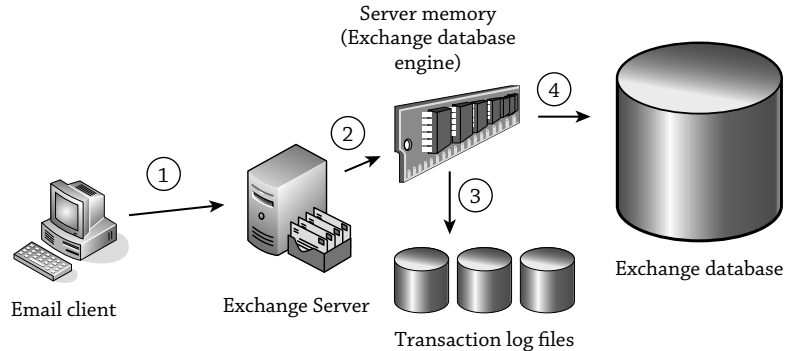
The transaction log that is always written to is the current transaction log for that particular database (e00.log, for example). Each transaction log file is exactly 1 MB in size, so when the transaction log is filled up, it is renamed to the next sequential number. For example, an old transaction log file might be named like this: e000004032.log. I often get questions about the logic of the transaction logs, and how they reserve space on the disk, whether they are empty or full. An easy way to look at it is to compare a log file to a carton of milk. When you have a carton of milk, it always takes up the same space in your fridge, empty or full. The same is true of the log files. Empty log files (current log file and *reserved* log files) are empty, or partially full; the renamed, *old*, log files are full. However, they take up the same amount of space on the disk.

The data, such as new email messages that enter my organization, is retained in RAM for some period of time (maybe as little as 5 seconds or maybe even 60 seconds or more) before it is flushed to the database file. The actual period that data is retained in memory will depend on how much cache memory is available, what types of operations are happening in the data, 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 store process) will automatically read the transaction log files once the server is brought back up and compare them with the data that's stored in the corresponding mailbox databases. Any inconsistency is resolved by replaying the missing data operations from the transaction logs back into the database, assuming that the entire transaction is present; if it's not, the operations are not written (and you can be confident that the operation wasn't completed at the time the crash happened). This helps ensure that the integrity of the mailbox database is preserved and that half-completed data operations aren't written back into the database and allowed to corrupt good data.

The transaction log files are important for a number of reasons. They are used by Microsoft replication technologies (as you'll learn in Chapter 19, "Creating and Managing Mailbox Databases"), but they can also be used in disaster recovery. The transaction logs are not 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 the missing transactions back into the database—provided you have all the transactions since the last full backup.

In previous versions of Exchange Server, you had two separate mail store objects: the *storage group*, which was a logical container that held an associated set of transaction logs, and the *mailbox database*, a set of files that held the actual permanent copies of user mailboxes. You often had multiple mailbox databases per storage group, meaning that one set of transaction logs contained interwoven transaction data for multiple databases (which could have detrimental effects on performance, space, and backups). In Exchange Server 2007, the recommendation changed; while you could still assign mailbox databases to a storage group in a many-to-one ratio, Microsoft encouraged you to assign them 1:1. In fact, to use the continuous-replication features in Exchange Server 2007, you had to do so.

In Exchange Server 2013, you still have mailbox databases. However, storage groups were removed in Exchange Server 2010; each mailbox database now has its own integral set of transaction log files. In fact, mailbox databases—which were once tightly coupled with specific servers—can have copies on multiple servers in the organization, even spread across multiple sites. This functionality was introduced by moving the mailbox databases from the Server hierarchy to the Organization hierarchy, essentially rendering them a *shared* object that can become active on any server in the organization. The *database availability group* container is now available to contain servers that participate in the replication of mailbox databases with each other.

EXCHANGE AND ACTIVE DIRECTORY

We 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 email-specific properties for mail-enabled objects such as users, contacts, groups, and public folders. Look at Figure 1.5 to see some of the different types of interactions that occur between Exchange Server and the Active Directory.
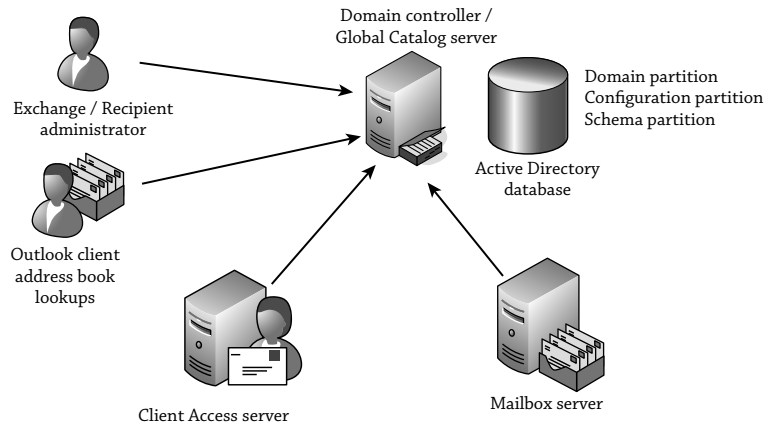
Because most of the Exchange Server configuration data for an Exchange server is stored in the Active Directory, all Exchange Server roles must contact a domain controller to request its configuration data; this information is stored in a special partition of the Active Directory database called the configuration partition. The configuration partition is replicated to all domain controllers in the entire Active Directory forest.

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

Mailbox Servers   Exchange Server 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. They also require access to global catalog servers to look up email addressing information, distribution list membership information, and other data related to message routing.

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

Controlling Mailbox Growth

As users have become more savvy and competent at using Outlook and the features of Exchange Server, and email messages themselves have become more complex, the need for email storage has grown. Back in the days of Exchange Server 4.0, an organization that gave its users a 25 MB mailbox was considered generous. With Exchange Server 2003, a typical user's mailbox may have a storage limit of 300 to 500 MB, with power users and VIPs requiring even more. At TechEd 2006, Exchange Server gurus were tossing about the idea that in the future a default mailbox limit would be closer to 2 GB as users start incorporating Unified Messaging features. Current discussions now look forward to and assume 20 GB mailboxes within the next few years.

We all see users with mailbox sizes in the gigabyte range, but is your organization prepared for a typical user with a 20 GB mailbox size limit? What sort of concerns will you face when your average user has 5 GB, 10 GB, 15 GB, or even 20 GB of content (not just email!) in their 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. While early versions of Exchange Server were primarily *performance-bound* /meaning that they would require more drive performance before they required more disk capacity/versions since Exchange Server 2007 have solidly pushed that to being *capacity-bound*. With the performance characteristics and capacities of modern drives, it becomes feasible to economically provision Exchange Server storage in support of large mailboxes.

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 300 GB mailbox database will take some time to restore from backup media. However, these issues have largely been mitigated by the use of database availability groups (DAGs), which ensure constant *copies* of mailbox databases that reside on other servers, essentially providing a constant *live* backup of mailbox databases on other servers, and in other datacenters.

Microsoft recommends that you do not allow an Exchange Server mailbox database to grow larger than 200 GB unless you are implementing continuous-replication technologies in Exchange Server 2013. If you use database availability groups to replicate databases to multiple servers, the maximum database size recommendation goes up (way up) to 2 TB. However, the maximum supported database size is actually 64 TB. If you require more than the maximum recommend database storage, Exchange Server 2013 Standard Edition allows you to have up to 5 mailbox databases and Exchange Server 2013 (CU2 and later) Enterprise Edition allows you to have up to 100.

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 arrives in a user's mailbox and users have additional mechanisms for accessing the data stored in their mailbox, keeping mail data around longer is a demand and a requirement for your user community. The Exchange Server 2013 archive mailbox feature also drives the need for more storage, as message archival moves away from the PST files and back into Exchange Server in the form of archive mailboxes. Those archive mailboxes can be *segregated* to a dedicated mailbox database and be set to a different backup schedule and their own set of management practices.

Personal Folders or PST Files

And while we're on the subject of PST files, let's discuss this pesky feature of client management. The Outlook Personal Folder, or PST files, can be the very bane of your existence. Outlook allows users to create a local database, named Personal Folder, in which users can

create folders and archive email. Although this seems like a good feature on the surface, there are a few downsides:

◆ 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. PSTs can become a management and security nightmare as data is suddenly distributed all over your network.

◆ 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, become misplaced, or even be lost entirely. PSTs are not designed for access over a network connection; they're meant to be on the local hard drive, which wastes space, as well as complicates the backup and management scenarios.

◆ Starting with Exchange Server 2010, Personal Archives stored on the server can be populated from PST files, therefore offering a true alternative to those pesky local files.

Email Archiving

Sometimes, managing a mail server seems like a constant race between 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 a business reason for them to do so, 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 email databases. Worse still, backups and data recovery take longer.

As discussed in the previous section, Exchange Server 2013 does provide some archiving features, such as the Personal Archive. Also, large mailboxes could be moved to an Office 365 subscription, in a hybrid coexistence model.

For those organizations that are *not* opting to head out to the cloud or do *not* choose Office 365 as their email solution, this is where email archiving becomes useful. The last time we counted, there were several dozen companies in the business of supplying email archiving tools and services. Archiving products all have a lot of functions in common, including the ability to keep data long term in email 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 email is archived, archive systems generally come in one of three flavors:

◆ Systems that depend on journaling to automatically forward every email sent or received by specified users on to the archive system.

◆ Systems that perform a scheduled "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 using storage, providing a complete archive, and dealing with performance overhead.

In the previous section, I discussed briefly the archive mailbox as an alternative to the management of PST files. However, its ability goes beyond the manual move of email messages

to a dedicated location on the server. For any user who requires email archival, a Personal Archive, can be created for that user. As email ages past a certain point, the mail is moved from the active mailbox to the archive mailbox by using Retention Policies. The user can still access and search the archive mailbox from Outlook Web App or Outlook, though. The email data remains on the Exchange server and thus does not require an additional email archival infrastructure.

We often get asked if this information can be made available offline; keep in mind that it cannot. Personal Archives cannot be included in Offline Stores (OST) files. This is by design, and we're kind of glad that it works this way, since we are continuously trying to reduce the email footprint on the client computers. OST files get very large, very fast, and can cause plenty of headaches as well.

If I Use a Third-Party Solution, Does It Matter How I Archive?

Every third-party archival vendor is going to tell you how their product is best and give you long technical reasons why their approach is so much better than the competition's. The dirty little secret is that all three approaches have their pros and cons:

Journaling   is based on SMTP. If content doesn't run across SMTP, it won't get journaled and thus won't get archived. Journaling is great for capturing messaging and calendaring traffic that involve multiple parties or external entities, but it won't capture what happens to messages and other mailbox data once they're in the mailbox. Journaling can also place an additional load on the Hub Transport servers, depending on the amount and type of messaging traffic your users generate.

Crawling   can capture changes only at certain intervals; it can't capture every single change, even though it overcomes many of the limitations of journaling. For example, if one user sends a message to another in violation of policy and both hard-delete their copy of the message before the next crawl interval, that message won't be detected and archived. The more often you schedule the crawl, the more of a performance impact your Mailbox servers will suffer.

Log shipping   is the best of all options; it captures every transaction and change, allowing you to capture the entire history of each object while offloading the performance hit from your Exchange servers. However, the Exchange Server product team does not like the concept of log shipping and tries to discourage its use—mainly because there are vendors who try to inject data back into Exchange Server by modifying logs. This, needless to say, results in mailbox data that won't be supported by Microsoft.

Public Folders

The end-user experience for public folders has not changed in Exchange Server 2013, though the architecture has changed a lot—mainly the storage of the public folders which is now in a mailbox database, instead of the public folder database. 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 perceived as being deemphasized since Exchange Server 2007, Microsoft continues to support public folders, and many organizations will continue to find useful applications for public folders for the foreseeable future.

A key change in public-folder storage occurs in Exchange Server 2013, one that finally breaks the paradigm of dedicated public folder databases and public folder replication. Although we discuss this change in Chapter 2, "Introducing the Changes in Exchange Server 2013," we just briefly note here that public folders are now stored in mailbox databases and can be replicated as mailbox database copies in a database availability group.

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 Server or Outlook Forms Designer, client or server scripting, or the Exchange Server API set. You can use public folders to replace many of the maddening paper-based processes that abound in every organization.

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.

Things Every Email Administrator Should Know

The information in this section is something that we often find even our own email 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.

A Day in the Life of the Email Administrator

We know and work with a lot of email administrators, and we 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 email administrator? These tasks will depend on the size of your organization, the number of administrators you have running your Exchange Server organization, and how administrative tasks are divided up.

Recipient Management Tasks   These are certainly the biggest day-to-day tasks that most Exchange Server administrators in medium and large organizations will experience. 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 email addresses to a user, or adding/removing group members

Basic Monitoring Tasks   These 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 event logs for unusual activity, errors, or warnings

Daily Troubleshooting Tasks   These 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 event logs to determine if they are serious and warrant corrective action

◆ Looking at mail flow in the organization to identify why delivery is taking a long time to some recipients

Security-Related Tasks   Some of these are performed daily, while others are performed only weekly or monthly:

◆ Looking at server and service uptimes 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 IIS and SMTP and connectivity logs or even reviewing their content

Email Client Administration Tasks   These include the following:

◆ Troubleshooting Autodiscover connectivity and client connectivity issues

◆ Diagnosing problems with mobile or tablet devices that use Exchange ActiveSync connectivity

Application Integration Tasks These are performed on an as-needed basis and may include the following:

◆ Establishing and diagnosing SMTP connectivity with email-enabled third-party applications such as web servers

◆ Configuring, testing, and troubleshooting Unified Messaging interoperability with voice and SIP systems

◆ Configuring, testing, and troubleshooting connectivity with SharePoint Server 2013 site mailboxes.

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 are almost as important as delivering the IT service itself. Keeping users informed of full or partial service outages such as mobile or iPhone 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?

Also remember to have multiple avenues of communication available to your users. For example, you may need to get out to your users the message that you will be having downtime on the weekend. Postings on your company intranet or even the bulletin board in the cafeteria or on the wall of the elevator are good ways to keep your users informed.

PREPARING REPORTS

Maybe we have just worked in a large IT environment for too long now, but it seems to us that information technology is more and more about reports and metrics. We are 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 the following:

◆ Total number of mailboxes and mailbox sizes

◆ Top system users and top source/destination domains

◆ Antispam 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 an additional reporting product, such as System Center 2012 R2.

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

As the discussion over moving to "the cloud" becomes more prevalent in most industries, the common argument that keeps on coming back in favor for moving Exchange Server services to some version of Exchange Online or Office 365 is server availability. 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 uptime per month or to provide email services during certain hours. Unscheduled downtime is anything that happens during your stated hours of operation that keeps users from accessing their email.

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 and listed in the Microsoft Server Catalog.

◆ Server hardware should be installed using the vendor recommended procedures and updated regularly. Problems with servers are frequently caused by outdated firmware and device drivers.

◆ Once the server is in production, it should not be used as a test bed for other software. Keep an identically configured server that uses the same hardware for testing updates.

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 failover 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, and for less money.

⊕ Real World Scenario

Internal Staff Training Is Just as Important as Your Infrastructure

Company LMNOP invested hundreds of thousands of dollars in their infrastructure to improve server uptime. Three months into the operation of the new system, an untrained operator accidentally brought down a 15,000-mailbox database availability group (DAG) simply because he had been asked to do a task he had never done before and the organization did not have documentation on how to proceed. So keep in mind that documentation, training, and procedures are very important in improving uptime.


Even the biggest mailbox servers in large database availability groups 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. Thankfully, the DAG solution for high availability ensures that users may never notice the scheduled server downtime, since mailbox services can be switched over to another member server in the DAG. That being said, when you are driving your car with no spare tire in the trunk, you are more vulnerable to a flat tire. The same is true of the DAG, because when a member server is offline for maintenance, the DAG loses a potential mailbox server that is capable of taking over in the event of server failure.

When your scheduled downtime will affect components that can impact server availability for your users, that downtime should be well communicated. Also, You should document your scheduled downtime as part of your operational plans and let your user community know about these plans. The specific time window for maintenance should always be the same; for some organizations, this might be 6:30 p.m.–10:30 p.m. on Thursday once per month, whereas other organizations might schedule downtime from 11:00 p.m. Saturday until 4:00 a.m. 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 security updates for the operating system and applications if vulnerabilities are discovered. Every few months, Microsoft releases updates for Exchange Server 2013 that fix bugs or that may even add slight 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 Windows Software Update Service (WSUS), Microsoft Systems Center 2012 R2, or another third-party server inside your network. Whichever you choose, it is important that you make sure that the machine is a server and not a workstation. For example, make sure the automatic updates component of Windows Server is configured correctly. Figure 1.6 shows the Change Settings options for Windows Update.

FIGURE 1.6
Configuring automatic updates
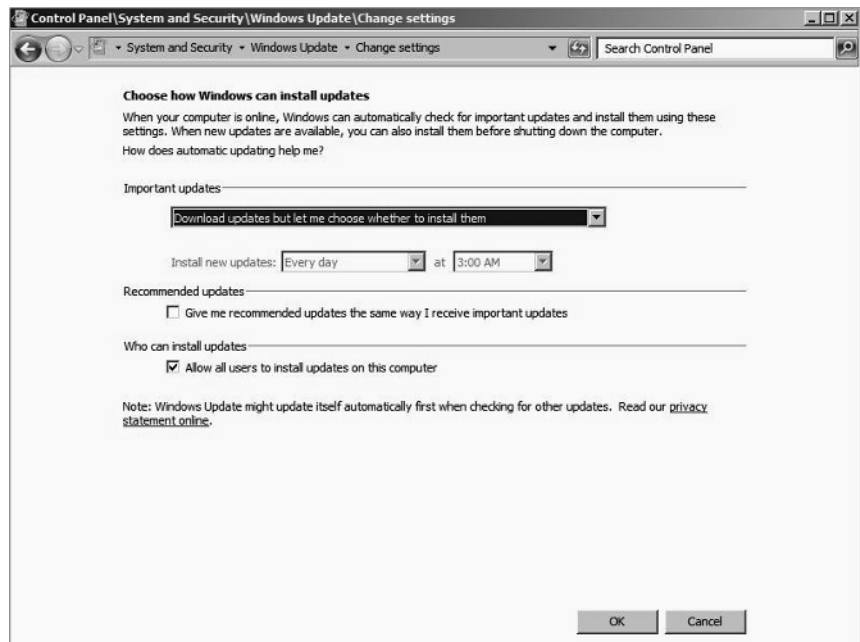
For production Exchange servers, you should configure the server with the option Download Updates But Let Me Choose Whether To Install Them. This is an important setting because if you choose the Install Updates Automatically (Recommended) option, the server will

automatically apply any update within a day or so of downloading it. This is not a desirable action for a production mail server. Instead, you want the server to download the updates and notify you via the updates icon in the system tray. You can then investigate the updates and schedule appropriate downtime to apply them manually.

Finding Answers

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

Too often techies 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. Key players in your organization will respect you much more when they know that you are willing to accept the limitations of your knowledge and have the appropriate resources to find the resolution to a problem or the answer to a question.

Helpful Resources

Exchange Server 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 we have about Exchange Server we can usually answer via the right search or by 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 many other resources are available:

Exchange Server Documentation   There is a world of free information on the Internet, but let's start right on the local hard disk of your Exchange Server or any place 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 2013 documentation is comprehensive and so readable you will wonder if it is really from Microsoft. Figure 1.7 shows an example of the Exchange Server 2013 documentation. Look for the following file,

```
C:\Program Files\Microsoft\Exchange Server\v15\Bin\ExchHelp.chm
```

or run it from the Microsoft Exchange Server 2013 folder on the Start menu.

FIGURE 1.7
Viewing the Exchange 2013 documentation
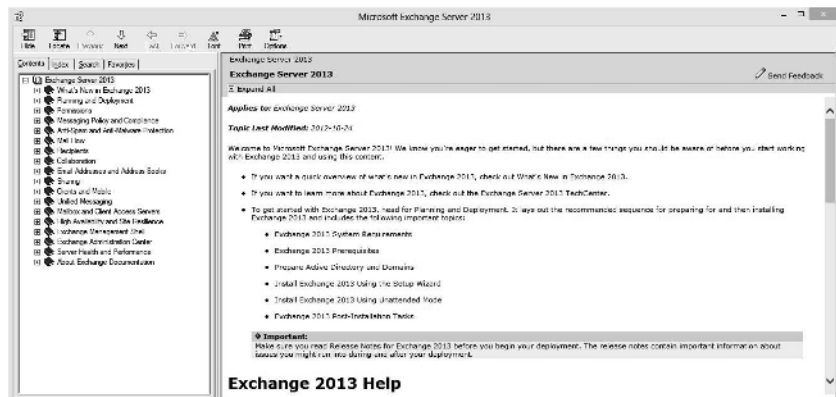
You can also download updated versions of `ExchHelp.chm` from the following URL:

`http://technet.microsoft.com/en-us/exchange/fp179701`

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\v15\`

Exchange Server 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 forums, also known as social.technet.microsoft .com. You can find the Exchange Server section here:

`http://social.technet.microsoft.com/forums/en-US/category/exchangeserver/`

When you post your question, please take a moment to think about what information the other readers are going to need to answer your question. While you can post a question like "Exchange is giving me an error," doing so is only going to result in (at best) delays while other forum participants have to request specific information from you. Instead, 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 you are using.

You Had Me at EHLO   This is the Microsoft Exchange Team's blog. This is the best site on the Internet for getting the inside scoop on how Exchange Server works, best practices, and the future of Exchange Server. You can read articles written by Exchange Server developers and Customer Support Services engineers. When changes to the product are announced, or customers are requesting changes in the product, you will hear *first* from the product group engineers about the way they have chosen to deal with the issue.

`http://blogs.technet.com/b/exchange/`

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 Server 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 to 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, where a web browser–based wizard guides you through your support options:

`https://gettechsupport.microsoft.com/default.aspx?locale=en-us&supportregion=en-us&pesid=14886`

If you do not have a Microsoft Premier agreement, Microsoft telephone support may seem to be a bit expensive, but believe me, when an Exchange server is down and the users are burning you in effigy in the company parking lot, a few hundred dollars for business hours support is cheap.

When you call and get a support technician on the phone, don't be surprised or offended if they start at the beginning and ask you a lot of elementary questions. They have to double-check everything you have done before they can look into more advanced problems. Frequently, one of these basic questions will help you locate a problem that you were convinced was more complicated than it really was. Though the beginning of the call may be underwhelming, the technician will stay with you on the phone until the problem is resolved or some kind of an acceptable resolution is put in place.

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 Server 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 backup if possible.

◆ 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 Lync 2013 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 with the right level of administrative access. If you don't have those, have someone nearby who can log you in.

◆ Save copies of the event logs. Be prepared to send these 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. We usually create a document with screen captures along with notes of what we were doing when we 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.

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 you may need to download other tools.

PowerShell and the Exchange Management Shell   Even here in the very first chapters, we are extolling the virtues of the new Windows management shell (or command-line and scripting interface) 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 this author. The Exchange Server team pioneered the adoption of PowerShell when they built the entire Exchange Server 2007 management interface, known as the Exchange Management Shell (EMS), as an extension to PowerShell. Exchange Server 2010 and Exchange Server 2013 continues to follow this pattern.

Although almost every chapter in this book will include at least some information about using EMS to perform Exchange Server management tasks, we have dedicated all of Chapter 5, "Introduction to PowerShell and the Exchange Management Shell," to helping you learn your way around EMS.

Exchange Management Shell Test Cmdlets   The Exchange Management Shell has a series of command-line tools that are very good for testing and diagnosing problems. These include tools for testing Outlook Web App connectivity, Unified Messaging connectivity, Outlook connectivity, and even mail flow. They are installed when you install the Exchange Server 2013 Management Tools. For more information, at the EMS prompt type **`Get-Excommand test*`**.

Microsoft Remote Connectivity Analyzer (Previously Exchange Remote Connectivity Analyzer)   Available at `www.testexchangeconnectivity.com`, the Remote Connectivity Analyzer is likely going to be the most useful tool in your troubleshooting arsenal. Initially started as a side project by two Microsoft employees, this website acts as the ultimate connectivity troubleshooting catch-all. The basic troubleshooting scenarios for Exchange Server 2013 (on-premises) are shown in Figure 1.8.

FIGURE 1.8
Viewing the Microsoft Remote Connectivity Analyzer
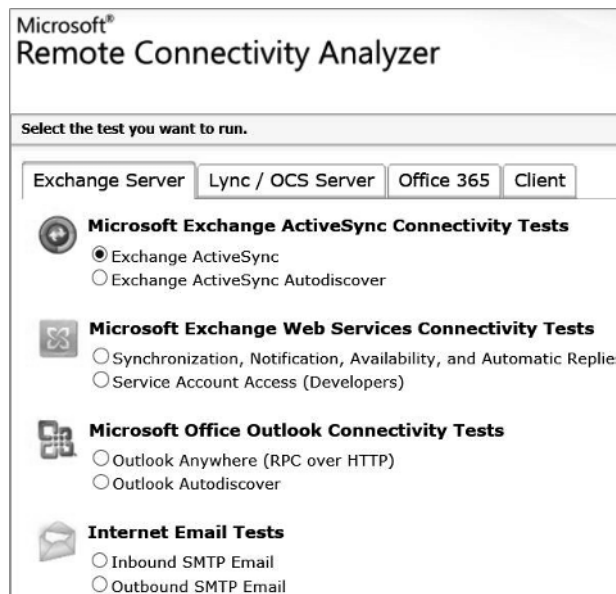
Those of you who have used "analyzers" from Microsoft in the past may remember the Exchange Best Practices Analyzer (ExBPA), the Remote Connectivity Analyzer should not be confused with the ExBPA. In fact, a new version of the ExBPA has not been released for Exchange Server 2013, as of the writing of this book, and will likely not be released.

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 1.9). 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.

FIGURE 1.9
The main search screen of ADModify.NET
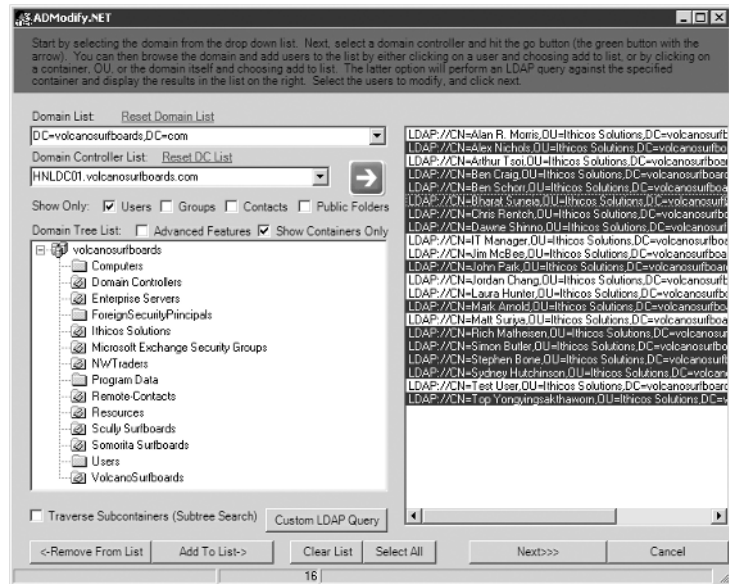
There are a few important things to note about ADModify. You must run it from your local hard disk, and the Microsoft .NET Framework v2.0 is required. Another important item to note is that it has an "undo" feature; you can back out a bulk change that you made if it turns out to be wrong. You can download ADModify from `www.codeplex.com/admodify`.

Quest ActiveRoles Management Shell for Active Directory   Quest Software is giving away one of the most useful add-on tools for Microsoft PowerShell that we 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 2013. We use this tool almost daily in organizations that have not yet migrated to Exchange Server 2013. You can download this free tool from Quest at: `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 PowerShell and extensions to PowerShell.

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

The Bottom Line

Understand email fundamentals.   To gain the best advantage from Exchange Server 2013, you should have a good grounding in general email applications and principles.

Master It   What two application models have email programs traditionally used? Which one does Exchange Server use? Can you name an example of the other model?

Identify email-administration duties.   Installing an Exchange Server system is just the first part of the job. Once it's in place, it needs to be maintained. Be familiar with the various duties and concerns that will be involved with the care and feeding of Exchange Server.

Master It   What are the various types of duties that a typical Exchange Server administrator will expect to perform?