

# Microsoft® Exchange Server 2007 Administrator's Pocket Consultant

*William R. Stanek*

To learn more about this book, visit Microsoft Learning at  
<http://www.microsoft.com/MSPress/books/9546.aspx>

9780735623484

**Microsoft®**  
*Press*

© 2007 William R. Stanek. All rights reserved.

# Table of Contents

<i>Acknowledgments</i> .....	<i>.xix</i>
<i>Introduction</i> .....	<i>.xxi</i>
<i>Who Is This Book For?</i> .....	<i>.xxi</i>
<i>How Is This Book Organized?</i> .....	<i>.xxii</i>
<i>Conventions Used in This Book</i> .....	<i>.xxiii</i>
<i>Support</i> .....	<i>.xxiv</i>

## Part I   **Exchange Server 2007 Administration Fundamentals**

<b>1   Microsoft Exchange Server 2007 Administration Overview</b> .....	<b>3</b>
Exchange Server 2007 and Your Hardware .....	4
Microsoft Exchange Server 2007 Editions .....	6
Exchange Server and Windows .....	10
Services for Exchange Server .....	10
Exchange Server Authentication and Security .....	12
Exchange Server Security Groups .....	13
Exchange Server and Active Directory .....	14
Understanding How Exchange Stores Information .....	14
Understanding How Exchange Routes Messages .....	15
Using the Graphical Administration Tools .....	16
Using the Command-line Administration Tools .....	18
<b>2   Deploying Microsoft Exchange Server 2007</b> .....	<b>21</b>
Exchange Server Messaging Roles .....	21
Understanding Exchange Server Messaging Roles .....	21
Deploying Mailbox Servers: The Essentials .....	22
Deploying Client Access Servers: The Essentials .....	24
Deploying Unified Messaging Servers: The Essentials .....	24
Deploying Transport Servers: The Essentials .....	25

**What do you think of this book?**  
We want to hear from you!

Microsoft is interested in hearing your feedback about this publication so we can continually improve our books and learning resources for you. To participate in a brief online survey, please visit: [www.microsoft.com/learning/booksurvey/](http://www.microsoft.com/learning/booksurvey/)

Integrating Exchange Server Roles with Active Directory .....	26
Using Hub Transport Servers with Active Directory .....	26
Using Client Access Servers with Active Directory .....	26
Using Unified Messaging Servers with Active Directory .....	27
Using Mailbox Servers with Active Directory .....	27
Using Edge Transport Servers with Active Directory .....	27
Integrating Exchange Server 2007 into Existing Exchange Organizations. ....	28
Preparing Active Directory for Exchange Server 2007 .....	29
Configuring Exchange Server 2007 for Use with Existing Exchange Organizations .....	29
Moving to Exchange Server 2007 .....	31
Running and Modifying Exchanger Server 2007 Setup .....	34
Installing New Exchange Servers .....	34
Installing Exchange Server .....	35
Adding, Modifying, or Uninstalling Server Roles. ....	39
<b>3 Managing Microsoft Exchange Server 2007 Clients .....</b>	<b>41</b>
Configuring Mail Support for Outlook 2007 and Windows Mail .....	42
Configuring Outlook 2007 for the First Time .....	42
Configuring Windows Mail for the First Time .....	46
Configuring Outlook 2007 for Exchange .....	48
Adding Internet Mail Accounts to Outlook 2007 and Windows Mail .....	48
Repairing and Changing Outlook 2007 Mail Accounts .....	49
Leaving Mail on the Server with POP3 .....	51
Leaving Mail on the Server: Outlook 2007 .....	51
Leaving Mail on the Server: Windows Mail .....	52
Checking Private and Public Folders with IMAP4 and UNIX Mail Servers .....	52
Checking Folders: Outlook 2007 .....	53
Checking Folders: Windows Mail .....	54
Managing the Exchange Server Service in Outlook 2007 .....	54
Managing Delivery and Processing E-mail Messages .....	54
Accessing Multiple Exchange Server Mailboxes .....	59
Granting Permission to Access Folders Without Delegating Access .....	62

Using Mail Profiles to Customize the Mail Environment . . . . .	64
Creating, Copying, and Removing Mail Profiles . . . . .	65
Selecting a Specific Profile to Use on Startup . . . . .	66
<b>4 Managing Mobile Messaging Users . . . . .</b>	<b>67</b>
Mastering Outlook Web Access Essentials . . . . .	67
Getting Started with Outlook Web Access . . . . .	68
Connecting to Mailboxes and Shared Non-Mail Data Over the Web . . . . .	70
Working with Outlook Web Access . . . . .	71
Enabling and Disabling Web Access for Users . . . . .	74
Mastering Mobile Device and Wireless Access Essentials . . . . .	75
Using Exchange ActiveSync . . . . .	76
Managing Wireless Synchronization and Exchange Server ActiveSync . . . . .	76
Managing Wireless Browsing on Mobile Devices . . . . .	77
Configuring Mobile Device Access and Wireless Browsing . . . . .	79
Mastering Remote Mail and Outlook Anywhere Essentials . . . . .	79
Using Remote Mail and Outlook Anywhere . . . . .	80
Creating Outlook Profiles for Dial-Up Connections to Corporate Networks . . . . .	81
Creating Outlook Profiles for Outlook Anywhere . . . . .	85

## **Part II Exchange Server 2007 Administration Essentials**

<b>5 Microsoft Exchange Server 2007 Administration Essentials . . . . .</b>	<b>91</b>
Understanding Exchange Server 2007 Organizations . . . . .	91
Using Site-Based Routing Instead of Routing Groups . . . . .	92
Using Configuration Containers Instead of Administrative Groups . . . . .	94
Understanding Data Storage in Exchange Server 2007 . . . . .	103
Working with the Active Directory Data Store . . . . .	103
Working with the Exchange Server Information Store . . . . .	104
Working with the Exchange Server Message Queues . . . . .	108
Using and Managing Exchange Server Services . . . . .	110
Starting, Stopping, and Pausing Exchange Server Services . . . . .	112
Configuring Service Startup . . . . .	112
Configuring Service Recovery . . . . .	113

<b>6</b>	<b>Configuring Microsoft Exchange Server with Exchange Management Shell</b>	<b>115</b>
	Using the Windows PowerShell	115
	Introducing the Windows PowerShell	115
	Running and Using Windows PowerShell	116
	Running and Using Cmdlets	117
	Running and Using Other Commands and Utilities	119
	Working with Cmdlets	119
	Using Windows PowerShell Cmdlets	120
	Using Cmdlet Parameters	121
	Understanding Cmdlet Errors	122
	Using Cmdlet Aliases	122
	Using the Exchange Management Shell	124
	Running and Using Exchange Management Shell	124
	Working with Exchange Cmdlets	125
	Working with Object Sets and Redirecting Output	125
	Working with Exchange Cmdlets	126
	Using General-Purpose Cmdlets	126
	Using Contact Management Cmdlets	127
	Using User Management Cmdlets	128
	Using Distribution Group Management Cmdlets	129
	Using Mailbox Management Cmdlets	131
	Using Database Management Cmdlets	133
	Using Storage Group Management Cmdlets	135
<b>7</b>	<b>User and Contact Administration</b>	<b>137</b>
	Understanding Users and Contacts	137
	Understanding the Basics of E-mail Routing	138
	Managing User Accounts and Mail Features	139
	Finding Existing Mailboxes, Contacts, and Groups	139
	Creating Mailbox-Enabled and Mail-Enabled User Accounts	140
	Adding Mailboxes to Existing Domain User Accounts	142
	Setting or Changing the Display Name and Logon Name for User Accounts	148
	Setting or Changing Contact Information for User Accounts	148

Changing a User's Exchange Server Alias and Display Name. . . . .	149
Adding, Changing, and Removing E-mail Addresses . . . . .	150
Setting a Default Reply-To Address for a User Account . . . . .	151
Changing a User's Web, Wireless Service, and Protocol Options. . . . .	151
Requiring User Accounts to Change Passwords. . . . .	152
Deleting Mailboxes from User Accounts. . . . .	153
Deleting User Accounts and Their Mailboxes. . . . .	154
Managing Contacts . . . . .	155
Creating Mail-Enabled Contacts. . . . .	155
Mail-Enabling Existing Contacts. . . . .	157
Setting or Changing a Contact's Name and Alias . . . . .	158
Setting Additional Directory Information for Contacts. . . . .	159
Changing E-mail Addresses Associated with Contacts . . . . .	159
Disabling Contacts and Removing Exchange Attributes. . . . .	160
Deleting Contacts . . . . .	160
<b>8 Mailbox Administration . . . . .</b>	<b>163</b>
Creating Special-Purpose Mailboxes . . . . .	163
Using Room and Equipment Mailboxes . . . . .	163
Creating Room and Equipment Mailboxes . . . . .	166
Creating Linked Mailboxes . . . . .	168
Creating Forwarding Mailboxes . . . . .	170
Managing Mailboxes: The Essentials . . . . .	171
Viewing Current Mailbox Size, Message Count, and Last Logon . . . . .	171
Setting Alternate Mailbox Display Names for Multilanguage Environments . . . . .	173
Hiding Mailboxes from Address Lists. . . . .	173
Defining Custom Mailbox Attributes for Address Lists . . . . .	173
Moving Mailboxes. . . . .	174
Moving Mailboxes: The Essentials . . . . .	174
Moving Mailboxes Using Exchange Management Console . . . . .	175
Moving Mailboxes Using Exchange Management Shell . . . . .	177

Configuring Mailbox Delivery Restrictions, Permissions, and Storage Limits. . . . .	179
Setting Message Size Restrictions for Contacts . . . . .	179
Setting Message Size Restrictions on Delivery to and From Individual Mailboxes . . . . .	179
Setting Send and Receive Restrictions for Contacts . . . . .	180
Setting Message Send and Receive Restrictions on Individual Mailboxes . . . . .	180
Permitting Others to Access a Mailbox. . . . .	181
Forwarding E-mail to a New Address . . . . .	182
Setting Storage Restrictions on an Individual Mailbox . . . . .	183
Setting Deleted Item Retention Time on Individual Mailboxes . . . . .	185
<b>9 Working with Distribution Groups and Address Lists. . . . .</b>	<b>187</b>
Using Security and Distribution Groups . . . . .	187
Group Types, Scope, and Identifiers . . . . .	187
When to Use Security and Standard Distribution Groups . . . . .	188
When to Use Dynamic Distribution Groups. . . . .	189
Working with Security and Standard Distribution Groups . . . . .	190
Creating Security and Standard Distribution Groups . . . . .	190
Assigning and Removing Membership for Individual Users, Groups, and Contacts. . . . .	195
Working with Dynamic Distribution Groups. . . . .	196
Creating Dynamic Distribution Groups. . . . .	196
Changing Apply-To Filters. . . . .	199
Changing Filter Conditions. . . . .	200
Designating an Expansion Server. . . . .	200
Modifying Dynamic Distribution Groups Using Cmdlets. . . . .	201
Previewing Dynamic Distribution Group Membership . . . . .	202
Other Essential Tasks for Managing Groups . . . . .	202
Changing a Group's Name Information . . . . .	202

Changing, Adding, or Deleting a Group's E-mail Addresses . . . . .	203
Hiding Groups from Exchange Address Lists . . . . .	204
Setting Usage Restrictions on Groups . . . . .	204
Setting Message Size Restrictions for Delivery to Groups . . . . .	206
Setting Out-of-Office and Delivery Report Options for Groups . . . . .	206
Deleting Groups . . . . .	206
Managing Online Address Lists . . . . .	207
Using Default Address Lists . . . . .	207
Creating and Applying New Address Lists . . . . .	208
Configuring Clients to Use Address Lists . . . . .	211
Updating Address List Configuration and Membership Throughout the Domain . . . . .	212
Editing Address Lists . . . . .	212
Renaming and Deleting Address Lists . . . . .	214
Managing Offline Address Books . . . . .	214
Creating Offline Address Books . . . . .	214
Configuring Clients to Use an Offline Address Book . . . . .	217
Assigning a Time to Rebuild an Offline Address Book . . . . .	217
Rebuilding Offline Address Books Manually . . . . .	218
Setting the Default Offline Address Book . . . . .	218
Changing Offline Address Book Properties . . . . .	218
Changing the Offline Address Book Server . . . . .	219
Deleting Offline Address Books . . . . .	220
<b>10 Implementing Exchange Server 2007 Security . . . . .</b>	<b>222</b>
Controlling Exchange Server Administration and Permissions . . . . .	223
Assigning Exchange Server Permissions to Users, Contacts, and Groups . . . . .	223
Understanding the Exchange Administration Groups . . . . .	224
Assigning Exchange Administrator Permissions in Active Directory Users and Computers . . . . .	226



Understanding Advanced Exchange Server Permissions . . . .	228
Assigning Advanced Exchange Server Permissions . . . . .	229
Adding Exchange Administrator Permissions in Exchange Management Console . . . . .	231
Understanding Administration Roles . . . . .	231
Adding Exchange Permissions to a User or Group . . . . .	232
Removing Delegated Exchange Permissions . . . . .	233
Auditing Exchange Server Usage . . . . .	234
Using Auditing . . . . .	234
Configuring Auditing . . . . .	235

## **Part III Server and Group Administration**

<b>11 Managing Microsoft Exchange Server 2007 Data and Storage Groups . . . . .</b>	<b>239</b>
Navigating the Information Store . . . . .	239
Using Storage Groups and Databases . . . . .	239
Configuring Storage Groups and Databases for Availability . . . . .	240
Improving Availability . . . . .	243
Controlling the Information Store . . . . .	244
Creating Storage Groups . . . . .	244
Enabling Local Continuous Replication for an Existing Storage Group . . . . .	246
Initiating or Resynchronizing Local Continuous Replication . . . . .	248
Disabling, Suspending, or Resuming Local Continuous Replication . . . . .	249
Moving Storage Groups . . . . .	250
Tracking Local Continuous Replication Status . . . . .	252
Verifying Your Local Continuous Replication Copies . . . . .	256
Enabling and Disabling Circular Logging . . . . .	256
Renaming Storage Groups . . . . .	257
Deleting Storage Groups . . . . .	258
Content Indexing . . . . .	258
Understanding Indexing . . . . .	258
Managing Full-Text Indexing . . . . .	259

<b>12</b>	<b>Mailbox and Public Folder Database Administration</b>	<b>261</b>
	Using Mailbox Databases	261
	Understanding Mailbox Databases	261
	Creating Mailbox Databases	262
	Setting the Default Public Folder Database and Default Offline Address Book	264
	Setting Mailbox Database Limits and Deletion Retention	265
	Recovering Deleted Mailboxes	268
	Recovering Deleted Items from Mailbox Databases	269
	Using Public Folder Databases	270
	Understanding Public Folder Databases	270
	Creating Public Folder Databases	270
	Setting Public Folder Database Limits	272
	Configuring Public Folder Replication	275
	Recovering Deleted Items from Public Folder Databases	276
	Managing Databases	276
	Mounting and Dismounting Databases	276
	Setting the Maintenance Interval	281
	Moving Databases	282
	Renaming Databases	283
	Deleting Databases	284
<b>13</b>	<b>Implementing Managed Folders and Managed Records</b>	<b>287</b>
	Introducing Messaging Records Management	287
	Implementing Records Management	288
	Managing Mailbox Folders	289
	Managing Content Settings	294
	Deploying Managed Custom Mailbox Folders	300
	Applying Records Management to User Mailboxes	303
<b>14</b>	<b>Accessing and Managing Public Folders</b>	<b>307</b>
	Accessing Public Folders	307
	Accessing Public Folders in Mail Clients	308
	Accessing Public Folders Through the Information Store	308

Creating and Working with Public Folders .....	310
Creating Public Folders in Microsoft Outlook .....	311
Creating Public Folders Using Exchange Management Shell .....	312
Determining Public Folder Size, Item Count, and Last Access Time .....	312
Adding Items to Public Folders Using Outlook .....	313
Managing Public Folder Settings .....	316
Controlling Folder Replication, Messaging Limits, Quotas, and Deleted Item Retention .....	316
Setting Client Permissions .....	317
Propagating Public Folder Settings and Data .....	320
Manipulating, Renaming, and Recovering Public Folders .....	320
<b>15 Managing Hub Transport and Edge Transport Servers .....</b>	<b>323</b>
Working with SMTP Connectors, Sites, and Links .....	323
Connecting Source and Destination Servers .....	324
Viewing and Managing Active Directory Site Details .....	324
Viewing and Managing Active Directory Site Link Details .....	326
Creating Send Connectors .....	328
Viewing and Managing Send Connectors .....	334
Configuring Send Connector DNS Lookups .....	334
Setting Send Connector Limits .....	335
Creating Receive Connectors .....	337
Viewing and Managing Receive Connectors .....	340
Connecting to Exchange 2003 or Exchange 2000 Routing Groups .....	343
Completing Transport Server Setup .....	345
Configuring the Postmaster Address and Mailbox .....	345
Enabling Antispam Features .....	346
Subscribing Edge Transport Servers .....	347
Configuring Journal Rules .....	351
Configuring Transport Rules .....	354
Managing Message Pickup and Replay .....	355
Understanding Message Pickup and Replay .....	355

Configuring and Moving the Pickup and Replay Directories .....	357
Changing the Message Processing Speed .....	358
Configuring Messaging Limits for the Pickup Directory .....	359
Creating and Managing Accepted Domains .....	359
Understanding Accepted Domains, Authoritative Domains, and Relay Domains .....	359
Viewing Accepted Domains .....	360
Creating Accepted Domains .....	361
Changing the Accepted Domain Type and Identifier .....	363
Removing Accepted Domains .....	364
Creating and Managing E-mail Address Policies .....	364
Viewing E-mail Address Policies .....	365
Creating E-mail Address Policies .....	365
Editing and Applying E-mail Address Policies .....	368
Removing E-mail Address Policies .....	370
Creating and Managing Remote Domains .....	370
Viewing Remote Domains .....	371
Creating Remote Domains .....	371
Configuring Messaging Options for Remote Domains .....	373
Removing Remote Domains .....	375
Configuring Antispam and Message Filtering Options .....	375
Filtering Spam and Other Unwanted E-mail by Sender .....	375
Filtering Spam and Other Unwanted E-mail by Recipient .....	377
Filtering Connections with Real-Time Block Lists .....	378
Defining Block List Exceptions and Global Allow /Block Lists .....	382
<b>16 Managing Client Access Servers .....</b>	<b>385</b>
Managing Web and Mobile Access .....	385
Using Outlook Web Access and Exchange ActiveSync with HTTP Virtual Servers .....	385
Working with HTTP Virtual Servers .....	386
Enabling and Disabling Outlook Web Access Features .....	387

Configuring Ports, IP Addresses, and Host Names Used by HTTP Virtual Servers .....	389
Enabling SSL on HTTP Virtual Servers .....	391
Restricting Incoming Connections and Setting Time-Out Values .....	392
Redirecting Users to Alternate URLs .....	393
Controlling Access to the HTTP Server .....	394
Starting, Stopping, and Pausing HTTP Virtual Servers and Web Sites .....	397
Configuring URLs and Authentication for OAB .....	398
Configuring URLs and Authentication for OWA .....	399
Configuring URLs and Authentication for Exchange ActiveSync .....	399
Enabling POP3 and IMAP4 .....	400
Deploying Outlook Anywhere .....	401
Managing Exchange Server Features for Mobile Devices .....	405
Understanding and Using Autodiscovery .....	406
Understanding and Using Direct Push .....	407
Understanding and Using Exchange ActiveSync Mailbox Policy .....	408
Understanding and Using Remote Device Wipe .....	415
Understanding and Using Password Recovery .....	417
Understanding and Configuring Direct File Access .....	418
Understanding and Configuring Remote File Access .....	422
Understanding and Using WebReady Document Viewing .....	425

## **Part IV Exchange Server 2007 Optimization and Maintenance**

<b>17 Microsoft Exchange Server 2007 Maintenance, Monitoring, and Queuing .....</b>	<b>430</b>
Understanding Troubleshooting Basics .....	430
Performing Tracking and Logging Activities in the Organization .....	432
Using Message Tracking .....	432
Using Protocol Logging .....	438
Using Connectivity Logging .....	443

Monitoring Events, Services, Servers, and Resource Usage . . . . .	445
Managing Essential Services . . . . .	447
Monitoring Exchange Messaging Components . . . . .	448
Using Performance Alerting . . . . .	450
Working with Queues . . . . .	455
Understanding Exchange Queues . . . . .	455
Accessing the Queue Viewer . . . . .	457
Managing Queues . . . . .	458
Understanding Queue Summaries and Queue States . . . . .	458
Refreshing the Queue View . . . . .	459
Working with Messages in Queues . . . . .	459
Forcing Connections to Queues . . . . .	461
Suspending and Resuming Queues . . . . .	461
Deleting Messages from Queues . . . . .	461
<b>18 Backing Up and Restoring Microsoft Exchange Server 2007 . . . .</b>	<b>463</b>
Understanding the Essentials of Exchange Server	
Availability, Backup, and Recovery . . . . .	463
Ensuring Data Availability . . . . .	463
Backing Up Exchange Server: The Basics . . . . .	464
Creating a Disaster Recovery Plan Based on	
Exchange Roles . . . . .	466
Finalizing Your Exchange Server Disaster	
Recovery Plan . . . . .	468
Choosing Backup Options . . . . .	469
Backing Up Exchange Server . . . . .	470
Starting the Backup Utility . . . . .	471
Backing Up Exchange Server with the Backup Wizard . . . . .	472
Backing Up Exchange Server Manually . . . . .	475
Recovering Exchange Server . . . . .	478
Recovering Exchange Server with the Restore Wizard . . . . .	479
Recovering Exchange Server Manually . . . . .	481
Restoring Mailboxes Selectively from Backup . . . . .	484
Step 1: Creating and Using Recovery Storage Groups . . . . .	485
Step 2: Restoring and Mounting the	
Recovery Databases . . . . .	486
Step 3: Selecting and Restoring Mailboxes . . . . .	487

Step 4: Dismounting Recovery Databases and  
Removing the Recovery Storage Group ..... 488

Performing Additional Backup and Recovery Tasks ..... 489

    Using the Recover Server Mode. .... 489

    Cloning Edge Transport Server Configurations..... 490

    Troubleshooting Database Mount Problems..... 490

    Mounting Mailbox Databases on Alternate Servers ..... 491

**Index..... 493**

**What do you think of this book?**  
We want to hear from you!

Microsoft is interested in hearing your feedback about this publication so we can continually improve our books and learning resources for you. To participate in a brief online survey, please visit: [www.microsoft.com/learning/booksurvey/](http://www.microsoft.com/learning/booksurvey/)

## Chapter 6

# Configuring Microsoft Exchange Server with Exchange Management Shell

Microsoft Exchange Server 2007 introduces Exchange Management Shell to complement the expanding role of Exchange Server administrators and developers. Exchange Management Shell is an extensible command-line environment for Exchange Server 2007 that builds on the existing framework provided by Windows PowerShell. When you install Exchange Server 2007 on a server or the Exchange Server management tools on a workstation, you install Windows PowerShell and Exchange Management Shell as part of the process. This chapter introduces Windows PowerShell and its features and then details the available commands and options of the Exchange Management Shell.

## Using the Windows PowerShell

Anyone with a UNIX background is probably familiar with the concept of a command shell. Most UNIX-based operating systems have several full-featured command shells available, including Korn Shell (KSH), C Shell (CSH), and Bourne Shell (SH). Although Microsoft Windows operating systems have always had a command-line environment, they've lacked a full-featured command shell, and this is where Windows PowerShell comes into the picture.

## Introducing the Windows PowerShell

Not unlike the less sophisticated Windows command prompt, the UNIX command shells operate by executing built-in commands, external commands, and command-line utilities and then returning the results in an output stream as text. The output stream can be manipulated in various ways, including redirecting the output stream so that it can be used as input for another command. This process of redirecting one command's output to another command's input is called piping, and it is a widely used shell-scripting technique.

The C Shell is one of the more sophisticated UNIX shells. In many respects, C Shell is a marriage of some of the best features of the C programming language and a full-featured UNIX shell environment. The Windows PowerShell takes the idea of a full-featured command shell built on a programming language a step further. It does this by implementing a scripting language based on C# and an object model based on the .NET framework.



Basing the scripting language for Windows PowerShell on C# ensures that the scripting language can be easily understood by current C# developers and also allows new developers to advance to C#. Using an object model based on the .NET framework allows the Windows PowerShell to pass complete objects and all their properties as output from one command to another. The ability to redirect objects is extremely powerful and allows for a much more dynamic manipulation of a result set. For example, not only can you get the name of a particular user, but you can also get the entire related user object. You can then manipulate the properties of this user object as necessary by referring to the properties you want to work with by name.

## Running and Using Windows PowerShell

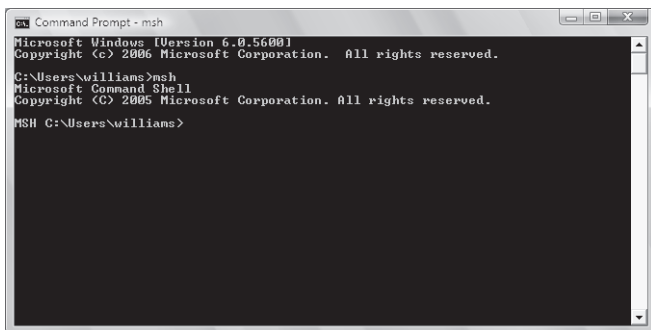
To invoke the Windows PowerShell, you must first open a command prompt window and then start the Windows PowerShell by typing **msh** at the command line. To exit the Windows PowerShell and return to the command prompt, type **exit**.

Usually, when the shell starts, you will see a message similar to the following:

```
Microsoft Command Shell
Copyright (C) 2005 Microsoft Corporation. All rights reserved.
```

You can disable this message by starting the shell with the `-nologo` parameter, such as **msh -nologo**

Regardless of how you start the shell, you know you are using the Windows PowerShell because the command prompt title bar changes to Command Prompt - msh and the current path is preceded by MSH, as shown in Figure 6-1.



**Figure 6-1** Start the Windows PowerShell.

When the shell starts, user and system profiles are run to set up the environment. The following is a listing and description of the profile files run, in the order of their execution:

1. %AllUsersProfile%\Documents\PSConfiguration\profile.ps1

A system-wide profile executed for all users. This profile is used by the system administrator to configure common settings for the Windows PowerShell.

2. %AllUsersProfile%\Documents\PSH\Microsoft.PowerShell\_profile.ps1

A system-wide profile executed for all users. This profile is used by the system administrator to configure common settings for the Windows PowerShell.

3. %UserProfile%\My Documents\PSConfiguration\profile.ps1

A user-specific profile executed only for the current user. This profile is used by individual users to configure common settings for the Windows PowerShell.

4. %UserProfile%\My Documents\PSH\Microsoft.PowerShell\_profile.ps1

A user-specific profile executed only for the current user. This profile is used by individual users to configure common settings for the Windows PowerShell.

You can start Windows PowerShell without loading profiles using the `-noprofile` parameter, such as:

**msh-noprofile**

The first time you start Windows PowerShell, you typically will see a message indicating that scripts are disabled and that none of the listed profiles are executed. This is the default secure configuration for the Windows PowerShell. To enable scripts for execution, enter the following command at the shell prompt:

**set-executionpolicy allsigned**

This command sets the execution policy to require that all scripts have a trusted signature to execute. For a less restrictive environment, you can enter the following command:

**set-executionpolicy remotesigned**

This command sets the execution policy so that scripts downloaded from the Web execute only if they are signed by a trusted source. To work in an unrestricted environment, you can enter the following command:

**set-executionpolicy unrestricted**

This command sets the execution policy to run scripts regardless of whether they have a digital signature.

## Running and Using Cmdlets

Windows PowerShell introduces the concept of a cmdlet (pronounced “command-let”). A cmdlet is the smallest unit of functionality in the Windows PowerShell. You can think of a cmdlet as a built-in command. Rather than being highly complex, most cmdlets are quite simple and have a small set of associated properties.

You use cmdlets the same way you use any other commands and utilities. Cmdlet names are not case-sensitive. This means you can use a combination of both uppercase and lowercase characters. After starting the Windows PowerShell, you can enter the name of the cmdlet at the prompt, and it will run in much the same way as a command-line command.

For ease of reference, cmdlets are named using verb-noun pairs. As Table 6-1 shows, the verb tells you what the cmdlet does in general. The noun tells you what specifically the cmdlet works with. For example, the `get-variable` cmdlet gets a named Windows PowerShell environment variable and returns its value. If you don't specify which variable to get as a parameter, `get-variable` returns a list of all Windows PowerShell environment variables and their values.

**Table 6-1   Common Verbs Associated with Cmdlets and Their Meanings**

Cmdlet Verb	Usage
New-	Creates a new instance of an item, such as a new mailbox.
Remove-	Removes an instance of an item, such as a mailbox.
Enable-	Enables a setting or mail-enables a recipient.
Disable-	Disables an enabled setting or mail-disables a recipient.
Set-	Modifies specific settings of an object.
Get-	Queries a specific object or a subset of a type of object, such as a specified mailbox or all mailbox users.

You can work with cmdlets in two ways:

- Executing commands directly at the shell prompt
- Running commands from within scripts

You can enter any command or cmdlet that you can run at the Windows PowerShell command prompt into a script by copying the related command text to a file and saving the file with the `.PS1` extension. You can then run the script in the same way you would any other command or cmdlet.

**Note**   Windows PowerShell also includes a rich scripting language and allows the use of standard language constructs for looping, conditional execution, flow control, and variable assignment. Discussion of these features is beyond the scope of this book.

From the Windows command-line environment or a batch script, you can execute Windows PowerShell cmdlets with the `-command` parameter. Typically when you do this, you will also want to suppress the Windows PowerShell logo and stop execution of profiles. After doing this, you can type the following command at a command prompt or insert it into a `.BAT` script:

```
msh Dnologo Dnoprofile Dcommand get-service
```

Finally, when you are working with Windows PowerShell, it is important to remember that the current directory may not be part of the environment path. Because of this, you may need to use “./” when you run a script in the current directory, such as:

```
./runtasks
```

## Running and Using Other Commands and Utilities

Because Windows PowerShell runs within the context of the Windows command prompt, you can run all Windows command-line commands, utilities, and graphical applications from within the Windows PowerShell. However, it is important to remember that the Windows PowerShell interpreter parses all commands before passing off the command to the command prompt environment. If the Windows PowerShell has a like-named command or a like-named alias for a command, this command, and not the expected Windows command, is executed. (See the “Using Cmdlet Aliases” section later in this chapter for more information on aliases.)

Non-Windows PowerShell commands and programs must reside in a directory that is part of the PATH environment variable. If the item is found in the path, it is run. The PATH variable also controls where the Windows PowerShell looks for applications, utilities, and scripts. In Windows PowerShell, you can work with Windows environment variables using \$env. If you want to view the current settings for the PATH environment variable, you type **\$env:path**. If you want to add a directory to this variable, you can use the following syntax:

```
$env:path += ";DirectoryPathToAdd"
```

Where DirectoryPathToAdd is the directory path you want to add to the path, such as:

```
$env:path += ";C:\Scripts"
```

To have this directory added to the path every time you start the Windows PowerShell, you can add the command line as an entry in your profile. Keep in mind that cmdlets are like built-in commands rather than stand-alone executables. Because of this, they are not affected by the PATH environment variable.

## Working with Cmdlets

Cmdlets provide the basic foundation for working with a computer from within the Windows PowerShell. Although there are many different cmdlets with many different available uses, cmdlets all have common features. I’ll examine these common features in this section.

# Using Windows PowerShell Cmdlets

At the Windows PowerShell prompt, you can get a complete list of cmdlets available by typing **help \***. To get help documentation on a specific cmdlet, type **help** followed by the cmdlet name, such as:

```
help get-variable
```

Table 6-2 provides a list of cmdlets you'll commonly use for administration. Although there are many other cmdlets available, these are the ones you're likely to use the most.

**Table 6-2 Cmdlets Commonly Used for Administration**

Cmdlet Name	Description
ConvertFrom-SecureString	Export a secure string to a safe format.
ConvertTo-SecureString	Create a securestring from a normal string.
Get-Alias	Returns alias names for cmdlets.
Get-AuthenticodeSignature	Gets the signature object associated with a file.
Get-Credential	Gets a credential object based on a password.
Get-Date	Gets the current date and time.
Get-EventLog	Gets the log data from the Windows log files.
Get-ExecutionPolicy	Gets the effective execution policy for the current shell.
Get-Host	Gets host information.
Get-Location	Displays the current location.
Get-MshDrive	Gets the drive information for the specified Msh drive.
Get-Service	Gets a list of services.
Import-Alias	Imports an alias list from a file.
New-Alias	Creates a new cmdlet-alias pairing.
New-Service	Creates a new service.
Push-Location	Pushes a location to the stack.
Read-Host	Reads a line of input from the host console.
Restart-Service	Restarts a stopped service.
Resume-Service	Resumes a suspended service.
Set-Alias	Maps an alias to a cmdlet.
Set-AuthenticodeSignature	Places an Authenticode signature in a script or other file.
Set-Date	Sets the system date and time on the host system.
Set-ExecutionPolicy	Sets the execution policy for the current shell.
Set-Location	Sets the current working location to a specified location.

Table 6-2 Cmdlets Commonly Used for Administration

Cmdlet Name	Description
Set-Service	Makes and sets changes to the properties of a service.
Start-Service	Starts a stopped service.
Start-Sleep	Suspends shell or script activity for the specified period.
Stop-Service	Stops a running service.
Suspend-Service	Suspends a running service.
Write-Output	Writes an object to the pipeline.

## Using Cmdlet Parameters

All cmdlet parameters are designated with an initial dash (-). To reduce the amount of typing required, some parameters are position-sensitive such that you can sometimes pass parameters in a specific order without having to specify the parameter name. For example, with `get-service`, you don't have to specify the `-Name` parameter, you can simply type:

```
Get-service ServiceName
```

where *ServiceName* is the name of the service you want to examine, such as:

```
Get-service MExchangeIS
```

This command line returns the status of the Microsoft Exchange Information Store service. Because you can use wildcards, such as `*`, with name values, you can also type `get-service mse*` to return the status of all Microsoft Exchange-related services.

All cmdlets support the common set of parameters listed in Table 6-3. However, for you to use these parameters, you must run the cmdlet in such a way that these parameters are returned as part of the result set.

Table 6-3 Common Cmdlet Parameters

Parameter Name	Description
-Confirm	Pauses processes and requires the user to acknowledge the action before continuing. Remove- and Disable- cmdlets have this parameter.
-Debug	Provides programming-level debugging information about the operation.
-ErrorAction	Controls the command behavior when an error occurs.
-ErrorVariable	Sets the name of the variable (in addition to the standard error) in which to place objects for which an error has occurred.
-OutBuffer	Sets the output buffer for the cmdlet.
-OutVariable	Sets the name of the variable in which to place output objects.

Table 6-3   Common Cmdlet Parameters

Parameter Name	Description
-Verbose	Provides detailed information about the operation.
-WhatIf	Allows the user to view what would happen if a cmdlet were run with a specific set of parameters. Remove- and Disable- cmdlets have this parameter.

## Understanding Cmdlet Errors

When you work with cmdlets, you'll encounter two standard types of errors:

- **Terminating errors**   Errors that halt execution
- **Nonterminating errors**   Errors that cause error output to be returned but do not halt execution

With both types of errors, you'll typically see error text that can help you resolve the problem. For example, an expected file might be missing, or you may not have sufficient permissions to perform a specified task.

## Using Cmdlet Aliases

For ease of use, Windows PowerShell lets you create aliases for cmdlets. An alias is an abbreviation for a cmdlet that acts as a shortcut for executing the cmdlet. For example, you can use the alias gsv instead of the cmdlet name get-service.

Table 6-4 provides a list of commonly used default aliases. Although there are many other aliases, these are the ones you'll use most frequently.

Table 6-4   Commonly Used Cmdlet Aliases

Alias	Cmdlet
clear, cls	Clear-Host
Diff	Compare-Object
cp, copy	Copy-Item
Epal	Export-Alias
Epcsv	Export-Csv
Foreach	ForEach-Object
Fl	Format-List
Ft	Format-Table
Fw	Format-Wide
Gal	Get-Alias
ls, dir	Get-ChildItem

Table 6-4 Commonly Used Cmdlet Aliases

Alias	Cmdlet
Gcm	Get-Command
cat, type	Get-Content
h, history	Get-History
gl, pwd	Get-Location
gps, ps	Get-Process
Gsv	Get-Service
Gv	Get-Variable
Group	Group-Object
lpal	Import-Alias
lpcsv	Import-Csv
R	Invoke-History
Ni	New-Item
Mount	New-MshDrive
Nv	New-Variable
rd, rm, rmdir, del, erase	Remove-Item
Rv	Remove-Variable
Sal	Set-Alias
sl, cd, chdir	Set-Location
sv, set	Set-Variable
Sort	Sort-Object
Sasv	Start-Service
Sleep	Start-Sleep
spps, kill	Stop-Process
Spsv	Stop-Service
write, echo	Write-Output

You can define additional aliases using the Set-Alias cmdlet. The syntax is:

```
Set-alias aliasName cmdletName
```

Where *aliasName* is the alias you want to use and *cmdletName* is the cmdlet for which you are creating an alias. The following example creates a “go” alias for the get-process cmdlet:

```
Set-alias go get-process
```



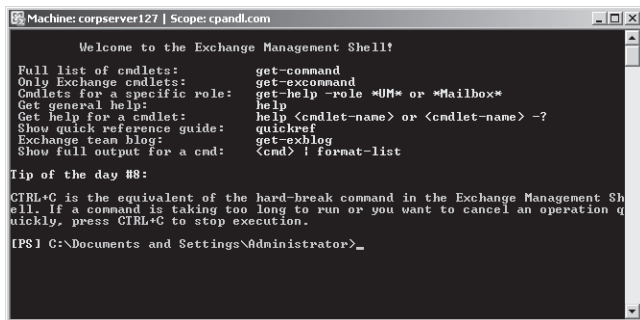
To use your custom aliases whenever you work with Windows PowerShell, enter the related command line in your profile.

## Using the Exchange Management Shell

Exchange Management Shell is a command-line management interface built on the Windows PowerShell. You use Exchange Management Shell to manage any aspect of Exchange Server 2007 configuration that you can manage in the Exchange Management Console. This means that you can typically use either tool to configure Exchange Server 2007. However, there are also some configuration settings that you can manage only by using Exchange Management Shell.

## Running and Using Exchange Management Shell

After you've installed the Exchange management tools on a computer, you can start the Exchange Management Shell by clicking Start, pointing to All Programs, selecting Microsoft Exchange Server 2007, and then selecting Exchange Management Shell. You know you are using the Exchange Management Shell because the command prompt title bar shows "Machine:" followed by the server name and the current working directory. The current path is preceded by [MSH], as shown in Figure 6-2.



**Figure 6-2** Use the Exchange Management Shell to manage Exchange Server from the command line.

When you start the Exchange Management Shell, it is initialized by a Microsoft Configuration (MCF1) file that instructs the shell to use administrator mode and obtain specific session settings about the Active Directory environment in which Exchange Server is being used. Because Exchange Management Shell is an extension of the Windows PowerShell, user and system profiles are also run to set up the environment. If you want to use specific environment settings every time you use Exchange Management Shell, you should put the settings in your user profile—either %UserProfile%\My Documents\PSConfiguration\profile.ps1 or %UserProfile%\My Documents\PSH\Microsoft.PowerShell\_profile.ps1.

When you work with Exchange Management Shell, all the Windows PowerShell cmdlets and aliases are available for your use. Although additional cmdlets are added, no additional aliases are added to the working environment. To end your session, you can exit the Exchange Management Shell and go to the command prompt by typing **exit**. Or you can close the shell window by clicking Close.

## Working with Exchange Cmdlets

When you are working with the Exchange Management Shell, additional Exchange-specific cmdlets are available. As with Windows PowerShell cmdlets, you can get help information on Exchange cmdlets:

- To view a list of all Exchange cmdlets, type **get-excommand** at the shell prompt.
- To view Exchange cmdlets related to a specific server role, type **get-help -role RoleName** where *RoleName* is the name of the server role you want to examine. Role names you can use are:
  - ❑ **\*UM\*** for cmdlets related to the Unified Messaging Server role
  - ❑ **\*Mailbox\*** for cmdlets related to the Mailbox Server role
  - ❑ **\*ClientAccess\*** for cmdlets related to the Client Access Server role

When you work with Exchange Management Shell, you'll often work with Get-, Set-, Enable-, Disable-, New-, and Remove- cmdlets. These cmdlets all accept the **-Identity** parameter, which identifies the unique object with which you are working.

Typically, a cmdlet that accepts the **-Identity** parameter has it as its first parameter, allowing you to specify the identity, with or without the parameter name. When identities have names as well as aliases, you can specify either value as the identity. For example, you can use any of the following techniques to retrieve the mailbox object for the user William Stanek with the mail alias WilliamS:

```
get-mailbox $identity Williams
get-mailbox $identity 'William Stanek'
get-mailbox Williams
get-mailbox "William Stanek"
```

With Get- cmdlets, you typically can return an object set containing all related items simply by omitting the identity. For example, if you type **get-mailbox** at the shell prompt without specifying an identity, you get a list of all mailboxes in the enterprise (up to the maximum permitted to return in a single object set).

## Working with Object Sets and Redirecting Output

When you are working with Exchange Management Shell, you'll often need to redirect the output of one cmdlet and pass it as input to another cmdlet. You can do this using the pipe (|) symbol. For example, if you want to view mailboxes for a specific mailbox

database rather than all mailboxes in the enterprise, you can pipe the output of `get-mailboxdatabase` to `get-mailbox`, as shown in this example:

```
get-mailboxdatabase ØIdentity ØEngineeringØ | get-mailbox
```

Here, you use `get-mailboxdatabase` to get the mailbox database object for the Engineering database. You then send this object to the `get-mailbox` cmdlet as input, and `get-mailbox` iterates through all the mailboxes in this database. If you don't perform any other manipulation, the mailboxes for this database are listed as output, as shown here:

Name	Alias	Server	ProhibitSendQuota
Administrator	Administrator	corpsvr127	unlimited
William S	williams	corpsvr127	unlimited
Tom G	tomg	corpsvr127	unlimited
David W	davidw	corpsvr127	unlimited
Kari F	karif	corpsvr127	unlimited
Connie V	conniev	corpsvr127	unlimited
Mike D	miked	corpsvr127	unlimited

You could also pipe this output to another cmdlet to perform an action on each individual mailbox in this database.

## Working with Exchange Cmdlets

You use Exchange cmdlets to manage the configuration of your Exchange organization. These cmdlets work with objects matching a specific set of criteria. The sections that follow provide an overview of the most commonly used cmdlets with their most commonly used syntaxes.

### Using General-Purpose Cmdlets

Several general-purpose cmdlets are provided. These cmdlets, along with their syntaxes, follow:

- **Get-ExchangeServer**    Retrieves a list of all or specified Exchange servers.

```
Get-ExchangeServer ØDomain 'DomainName'
[-DomainController 'DCName']
```

- **Get-Recipient**    Retrieves a list of all or specific recipients.

```
Get-Recipient [-RecipientType 'RecipientIdentifier']
[-Identity 'Identifier'] [Format-List'] [-DomainController
'DCName']
[ØOrganizationalUnit 'OUName'] [ØAnr 'String']
```

**Note** The `-Anr` parameter is used to specify a string on which to perform ambiguous name resolution. Any value entered is searched for within the specified objects.

## Using Contact Management Cmdlets

You can work with contacts using the following cmdlets and command-line syntaxes:

- **Enable-Mailcontact** Mail-enables a contact.

```
Enable-Mailcontact -Identity 'Identifier'  
-externalEmailAddress 'EmailAddress'  
[-alias 'Alias']  
[-DomainController 'DCName']
```

- **Disable-MailContact** Mail-disables a contact.

```
Disable-MailContact -Identity 'Alias'  
[-DomainController 'DCName']
```

- **Get-MailContact** Retrieves a list of all or specific mail-enabled contacts.

```
Get-MailContact [-Identity 'Identifier'] [Format-List]  
[-DomainController 'DCName'] [DOrganizationalUnit 'OUName']  
[DAnr 'String']
```

- **Set-MailContact** Changes the specified properties of the specified mail-enabled contact.

```
Set-MailContact -Identity 'Identifier' [-Alias 'NewAlias']  
[-AcceptMessagesOnlyFrom 'Recipient']  
[-DeliverToForwardingAddress <$false|$true>]  
[-DisplayName 'Name'] [-DomainController 'DCName']  
[-EmailAddresses 'ProxyAddress']  
[-EmailAddressPolicyEnabled: <$false|$true>]  
[-ExternalEmailAddress 'ProxyAddress']  
[-ForwardingAddress 'Recipient']  
[-GrantSendOnBehalfTo 'Mailbox']  
[-HiddenFromAddressListsEnabled <$false|$true>]  
[-MaxReceiveSize 'Size'] [-MaxRecipientPerMessage 'Size']  
[-MaxSendSize 'Size'] [-MessageBodyFormat 'Format']  
[-MessageFormat 'Format'] [-Name 'Name']
```

```
[-PrimarySmtpAddress 'SmtpAddress']
[-RejectMessagesFrom 'Recipient']
[-RejectMessagesFromDLMembers 'Recipient']
[-RequireSenderAuthenticationEnabled: <$false|$true>]
[-SimpleDisplayName 'Name']
```

- **Get-Contact** Retrieves a list of all or specific contacts, whether mail-enabled or not.

```
Get-Contact [-DomainController 'DCName'] [DOrganizationalUnit
'OUName'] [DResultSize 'Size']
```

- **Set-Contact** Changes or sets the specified properties of the specified contact.

```
Set-Contact -Identity 'Identifier' [-AssistantName 'Name']
[-City 'Name']
[-Company 'Name'] [-CountryOrRegion 'Name']
[-Department 'Name']
[-DisplayName 'Name'] [-DomainController 'DCName']
[-Fax 'FAXNUMBER']
[-FirstName 'Name'] [-HomePhone 'PhoneNumber']
[-Initials 'Value']
[-LastName 'Name'] [-Manager 'RecipientId']
[-MobilePhone 'PhoneNumber']
[-Name 'Name'] [-Notes 'Value'] [-Office 'Value']
[-Phone 'PhoneNumber']
[-PostalCode 'Code'] [-SimpleDisplayName 'Name']
[-StateOrProvince 'Value']
[-StreetAddress 'Value'] [-TelephoneAssistant 'Value']
[-Title 'Value']
[-WebPage 'Value']
```

## Using User Management Cmdlets

You can manage users using the following cmdlets and command-line syntaxes:

- **Get-User** Retrieves a list of all or specific Active Directory users.

```
Get-User [-Identity 'Identifier'] [-DomainController 'DCName']
[DOrganizationalUnit 'OUName'] [DResultSize 'Size'] [-SortBy
'Value']
```

- **Disable-MailUser** Mail-disables the specified Active Directory user.

```
Disable-MailUser -Identity 'Identifier'  
[-DomainController 'DCName']
```

- **Get-MailUser** Retrieves a list of all or specified mail-enabled users.

```
Get-MailUser [-Identity 'Identifier'] [-DomainController  
'DCName']  
[OrganizationalUnit 'OUName'] [ResultSize 'Size'] [-SortBy  
'Value']
```

- **Set-MailUser** Sets the specified properties for the specified user.

```
Set-MailUser [-Identity 'Identifier'] [-AllowMerge  
<$false|$true>]  
[-AttachmentFileNames 'Names'] [-BadItemLimit 'Num']  
[-ContentKeywords 'Strings'] [-DomainController 'DCName']  
[-EndDate 'DateTime'] [-ExcludeFolders 'MapiFolderPath']  
[-GlobalCatalog 'GCName'] [-IgnorePolicyMatch <$false|$true>]  
[-IncludeFolders 'MapiFolderPath'] [-Locale 'Value']  
[-MaxThreads 'Num']  
[-NTAccountOU 'OUId'] [-PreserveMailboxSizeLimit <$false|$true>]  
[-ReportFile 'LocalPath'] [-RetryInterval 'TimeSpan']  
[-RetryTimeout 'TimeSpan'] [-StartDate 'DateTime']  
[-SubjectKeywords 'Values'] [-ValidateOnly <$false|$true>]
```

## Using Distribution Group Management Cmdlets

You can work with distribution groups using the following cmdlets and command-line syntaxes:

- **Enable-DistributionGroup** Mail-enables an existing universal distribution group.

```
Enable-DistributionGroup -Identity 'Identifier' [-Alias 'Alias']  
[-DisplayName 'Name'] [-DomainController 'DCName']
```

- **Disable-DistributionGroup** Mail-disables a specified universal distribution group.

```
Disable-DistributionGroup -Identity 'Identifier'
[-DomainController 'DCName']
```

- **Get-DistributionGroup** Retrieves a list of all or specific mail-enabled universal distribution groups.

```
Get-DistributionGroup [-Identity 'Identifier']
[-DomainController 'DCName']
[-ManagedBy 'RecipientId'] [DOrganizationalUnit 'OUnName']
[DResultSize 'Size'] [-SortBy 'Value']
```

- **Set-DistributionGroup** Changes the specified properties of the specified distribution group.

```
Set-DistributionGroup -Identity 'Identifier' [-Alias 'NewAlias']
[-AcceptMessagesOnlyFrom 'Recipient'] [-DisplayName 'Name']
[-DomainController 'DCName'] [-EmailAddresses 'ProxyAddress']
[-EmailAddressPolicyEnabled <$false|$true>] [-ExpansionServer
'Server']
[-GrantSendOnBehalfTo 'Mailbox']
[-HiddenFromAddressListsEnabled <$false|$true>]
[-MaxReceiveSize 'Size'] [-MaxSendSize 'Size']
[-Name 'Name'] [-PrimarySmtpAddress 'SmtpAddress']
[-RejectMessagesFrom 'Recipient']
[-RejectMessagesFromDLMembers 'Recipient']
[-SimpleDisplayName 'Name']
```

- **Add-DistributionGroupMember** Adds the specified recipient to the universal distribution group.

```
Add-DistributionGroupMember -Identity 'Identifier' -Member
'RecipientID' [-DomainController 'DCName']
```

- **Remove-DistributionGroupMember** Removes the specified recipient from the universal distribution group.

```
Remove-DistributionGroupMember -Identity 'Identifier' -Member  
'RecipientID' [-DomainController 'DCName']
```

- **Get-DistributionGroupMember** Retrieves a list of all of the members of the specified distribution group.

```
Get-DistributionGroupMember -Identity 'Identifier'  
[-DomainController 'DCName'] [DResultSize 'Size']
```

- **Get-Group** Retrieves a list of all security and distribution groups.

```
Get-Group [-Identity 'Identifier'] [-DomainController 'DCName']  
[DOrganizationalUnit 'OUName'] [DResultSize 'Size'] [-SortBy  
'Value']
```

- **Set-Group** Sets the specified properties of the specified Windows group.

```
Set-Group -Identity 'Identifier' [-DisplayName 'NewDisplayName']  
[-DomainController 'DCName'] [-ManagedBy 'RecipientId']  
[DName 'Name'] [DNotes 'Value'] [-SimpleDisplayName 'Name']
```

## Using Mailbox Management Cmdlets

You can work with mailboxes using the following cmdlets and command-line syntaxes:

- **Enable-Mailbox** Mailbox-enables an existing Active Directory user account.

```
Enable-Mailbox -Identity 'Domain\UserName' -Database  
'MailboxDatabase' [-Alias 'Alias'] [-DomainController 'DCName']  
[-ManagedFolderMailboxPolicy 'PolicyId'] [DMobileMailboxPolicy  
'PolicyId']
```

- **Disable-Mailbox** Mailbox-disables the specified user account.

```
Disable-Mailbox -Identity 'Identifier' [-DomainController  
'DCName']
```



- **Set-Mailbox**   Changes the specified properties of the specified mailbox.

```
Set-Mailbox -Identity 'Identifier'
[-AcceptMessagesOnlyFrom 'RecipientId']
[-AcceptMessagesOnlyFromDLMembers 'RecipientId']
[-Alias 'Alias']
[-AntispamBypassEnabled: <$false|$true>]
[-DeliverToMailboxAndForward: <$false|$true>]
[-DisplayName 'Name'] [-DomainController 'DCName']
[-EmailAddresses 'ProxyAddresses'] [-EmailAddressPolicyEnabled
<$false|$true>] [-ForwardingAddress 'RecipientId']
[-GrantSendOnBehalfTo 'MailboxId']
[-HiddenFromAddressListsEnabled 'State']
[-IssueWarningQuota 'Size'] [-ManagedFolderMailboxPolicy
'MailboxPolicyId']
[-MaxReceiveSize 'Size'] [-MaxSendSize 'Size'] [-Name 'Name']
[-Office 'Value'] [-OfflineAddressBook 'OfflineAddressBookId']
[-PrimarySmtpAddress 'SmtpAddress'] [-ProhibitSendQuota 'Size']
[-ProhibitSendReceiveQuota 'Size'] [-RecipientLimits 'Size']
[-RejectMessagesFrom 'RecipientId']
[-RejectMessagesFromDLMembers 'RecipientId']
[-RequireSenderAuthenticationEnabled <$false|$true>]
[-RetainDeletedItemsFor 'Time'] [-RetainDeletedItemsUntilBackup:
<$false|$true>]
[-RetentionHoldEnabled <$false|$true>]
[-SamAccountName 'Name'] [-UserPrincipalName 'Name']
[-WindowsEmailAddress 'SmtpAddress']
```

- **Get-Mailbox**   Retrieves a list of all or specific mailboxes.

```
Get-Mailbox [-Identity 'Identifier'] | [-Database
'DatabaseName'] [-DomainController 'DCName']
[DOrganizationalUnit 'OUName'] [DResultSize 'Size'] [-SortBy
'Value']
```

- **Get-MailboxStatistics**   Retrieves summary statistics for all or specific mailboxes, as long as the mailboxes have been logged on to at least once.

```
Get-MailboxStatistics [-Identity 'Identifier'] [-Database
'MailboxDatabase'] [-Server 'Server']
```

**Move-Mailbox** Moves the mailbox of the specified user to the specified server.

```
Move-Mailbox -Identity 'Identifier' -TargetDatabase
'Server\MailboxDatabase'
[-AllowMerge <$false|$true>] [-AttachmentFileNames 'Values']
[-ContentKeywords 'Values'] [-DomainController 'DCName']
[-EndDate 'DateTime'] [-ExcludeFolders 'MapiFoldePath']
[-GlobalCatalog 'GCName'] [-IgnorePolicyMatch <$false|$true>]
[-IncludeFolders 'MapiFolderPath'] [-Locale 'Value']
[-MaxThreads 'Num']
[-NTAccountOU 'OUIId'] [-PreserveMailboxSizeLimit 'Switch']
[-ReportFile 'LocalPath'] [-RetryInterval 'TimeSpan']
[-RetryTimeout 'TimeSpan'] [-StartDate 'DateTime']
[-SubjectKeywords 'Values'] [-ValidateOnly: <$false|$true>]
```

## Using Database Management Cmdlets

You can manage Exchange databases using the following cmdlets and command-line syntaxes:

- **New-MailboxDatabase** Creates a new mailbox database in the specified storage group.

```
New-MailboxDatabase -Name 'MailboxDatabase' -StorageGroup
'StorageGroup'
[-CopyEdbFilePath 'EdbFilePath'] [-DomainController 'DCName']
[-EdbFilePath 'EdbFilePath'] [-HasLocalCopy <$false|$true>]
[-OfflineAddressBook 'OABId'] [-PublicFolderDatabase
'DatabaseId']
```

- **Remove-MailboxDatabase** Removes the specified mailbox database.

```
Remove-MailboxDatabase -Identity 'MailboxDatabase'
[-DomainController 'DCName']
```

- **Set-MailboxDatabase** Sets the specified properties of the specified mailbox database.

```
Set-MailboxDatabase [-Identity 'MailboxDatabase']
[-AllowFileRestore <$false|$true>] [-DomainController 'DCName']
[-EventHistoryRetentionPeriod 'TimeSpan'] [-FixedFont
<$false|$true>]
[-IndexEnabled <$false|$true>] [-IssueWarningQuota 'Size']
[-ItemRetention 'TimeSpan'] [-JournalRecipient 'RecipientId']
[-MailboxRetention 'TimeSpan'] [-MaintenanceSchedule 'Schedule']
[-MountAtStartup <$false|$true>] [-Name 'Name']
[-OfflineAddressBook 'OABId'] [-ProhibitSendQuota 'Size']
[-ProhibitSendReceiveQuota 'Size'] [-PublicFolderDatabase
'DatabaseId']
[-QuotaNotificationSchedule 'Schedule'] [-RestoreInProgress
<$false|$true>]
[-RetainDeletedItemsUntilBackup <$false|$true>]
[-SMimeSignatureEnabled <$false|$true>]
```

- **Get-MailboxDatabase** Retrieves a list of all or specified mailbox databases.

```
Get-MailboxDatabase [-Identity 'MailboxDatabase']
[-DomainController 'DCName']

Get-MailboxDatabase [-StorageGroup 'StorageGroup']
[-DomainController 'DCName']

Get-MailboxDatabase [-Server 'Server'] [-DomainController
'DCName']
```

- **Mount-Database** Mounts the specified mailbox database.

```
Mount-Database -Identity 'MailboxDatabase' [-DomainController
'DCName'] [-Force <$false|$true>]
```

- **Dismount-Database** Dismounts the specified mailbox database.

```
Dismount-Database -Identity 'MailboxDatabase'
[-DomainController 'DCName']
```

- **Enable-DatabaseCopy** Enables local continuous backup for the specified mailbox database.

```
Enable-DatabaseCopy -Identity 'MailboxDatabase'  
[-CopyEdbFilePath 'EdbFilePath'] [-DomainController 'DCName']
```

## Using Storage Group Management Cmdlets

You can manage Exchange storage groups using the following cmdlets and command-line syntaxes:

- **New-StorageGroup** Creates the named storage group on the specified server.

```
New-StorageGroup -Name 'StorageGroupName' -Server 'Server'  
[-CircularLoggingEnabled <$false|$true>] [-CopyLogFolderPath  
'LocalPath']  
[-CopySystemFolderPath 'LocalPath'] [-DomainController 'DCName']  
[-HasLocalCopy <$false|$true>] [-LogFolderPath 'LocalPath']  
[-SystemFolderPath 'LocalPath'] [-ZeroDatabasePages  
<$false|$true>]
```

- **Get-StorageGroup** Retrieves a list of all or specified storage groups.

```
Get-StorageGroup [-Identity 'StorageGroup'] [-DomainController  
'DCName']
```

- **Set-StorageGroup** Changes the name of the specified storage group to the name value provided.

```
Set-StorageGroup -Identity 'StorageGroup'  
[-CircularLoggingEnabled <$false|$true>] [-DomainController  
'DCName'] [-Name 'Name']  
[-ZeroDatabasePages <$false|$true>]
```

- **Remove-StorageGroup** Deletes the specified storage group.

```
Remove-StorageGroup -Identity 'StorageGroup'  
[-DomainController 'DCName']
```

- **Enable-StorageGroupCopy**   Enables local continuous backup for the specified storage group, provided all databases within the storage group already have this feature enabled.

```
Enable-StorageGroupCopy -Identity 'StorageGroup'  
[-CopyLogFolderPath 'LocalPath'] [-CopySystemFolderPath  
'LocalPath'] [-DomainController 'DCName']  
[-SeedingPostponed: <$false|$true>]
```

- **Disable-StorageGroupCopy**   Disables the continuous backup of the specified storage group.

```
Disable-StorageGroupCopy -Identity 'StorageGroup'  
[-DomainController 'DCName']
```