Tony Redmond
Well-known author and Microsoft MVP for Exchange Server
Technical review by Paul Robichaux

Microsoft

Microsoft® Exchange Server 2010

INSIDE OUT

- **The ultimate, in-depth reference**
- **Hundreds of timesaving solutions**
- **Supremely organized, packed with expert advice**

Includes
YOUR BOOK—ONLINE!
See back

Microsoft®

Microsoft® Exchange Server 2010 Inside Out

Tony Redmond

PUBLISHED BY
Microsoft Press
A Division of Microsoft Corporation
One Microsoft Way
Redmond, Washington 98052-6399

Copyright © 2010 by Tony Redmond

All rights reserved. No part of the contents of this book may be reproduced or transmitted in any form or by any means without the written permission of the publisher.

Library of Congress Control Number: 2010935972
ISBN: 978-0-7356-4061-0

3 4 5 6 7 8 9 10 11 QGT 7 6 5 4 3 2

Printed and bound in the United States of America.

Microsoft Press books are available through booksellers and distributors worldwide. For further information about international editions, contact your local Microsoft Corporation office or contact Microsoft Press International directly at fax (425) 936-7329. Visit our Web site at www.microsoft.com/mspress. Send comments to mspinput @microsoft.com.

Microsoft and the trademarks listed at http://www.microsoft.com/about/legal/en/us/IntellectualProperty /Trademarks/EN-US.aspx are trademarks of the Microsoft group of companies. All other marks are property of their respective owners.

The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious. No association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred.

This book expresses the author's views and opinions. The information contained in this book is provided without any express, statutory, or implied warranties. Neither the authors, Microsoft Corporation, nor its resellers, or distributors will be held liable for any damages caused or alleged to be caused either directly or indirectly by this book.

Acquisitions Editor: Martin DelRe
Developmental Editor: Karen Szall
Project Editor: Karen Szall
Editorial Production: nSight, Inc.
Technical Reviewer: Paul Robichaux; Technical Review services provided by Content Master, a member of CM Group, Ltd.
Cover: Tom Draper Design

Body Part No. X17-21593

[2012-03-02]

Contents at a Glance

Chapter 1
Introducing Microsoft Exchange 2010 1

Chapter 2
Installing Microsoft Exchange 2010 39

Chapter 3
The Exchange Management Shell 75

Chapter 4
Role-Based Access Control 147

Chapter 5
Exchange Management Console and Control Panel 181

Chapter 6
Managing Mail-Enabled Recipients 255

Chapter 7
The Exchange 2010 Store 357

Chapter 8
Exchange's Search for High Availability 425

Chapter 9
Backups and Restores 527

Chapter 10
Clients 555

Chapter 11
Client Access Server 651

Chapter 12
Mailbox Support Services 707

Chapter 13
The Exchange Transport System 801

Chapter 14
Message Hygiene 907

Chapter 15
Compliance 973

Chapter 16
Rules and Journals 1083

Chapter 17
The Exchange Toolbox 1129

Table of Contents

Foreword xix
Introduction xxi
Service Pack 1 xxii
Writing style and general approach to content xxii
Examples used in the book xxiii
Thanks xxiv
In conclusion xxvi
Support for this book xxvi
We want to hear from you xxvii
Your Companion eBook xxvii

Chapter 1: Introducing Microsoft Exchange 2010 . . . . . **1**

The motivation to upgrade 3
Moving from Exchange 2003 or Exchange 2007 4
Testing and beta versions 6
Fundamental questions before you upgrade 7
No in-place upgrades 8
What version of Windows? 10
Preparing for Exchange 2010 11
The test plan 12
Testing for operational processes 14
Testing for programming and customizations 14
Bringing Exchange 2007 up to speed 16
Deploying earlier versions of Exchange servers alongside Exchange 2010 17
Web-based Deployment Assistant 18
Exchange 2010 editions 18
Active Directory 19
The strong link between Exchange and Active Directory 20
ADSIEdit 22

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

Microsoft is interested in hearing your feedback 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/

Types of Active Directory deployments that support Exchange . . . 23
The role of ADAccess . . . 25
Planning for global catalogs . . . 29
Preparing Active Directory for Exchange . . . 31
The joys of a customizable schema . . . 34
Ready-to-go custom attributes . . . 35
Let's install . . . 37

Chapter 2: Installing Microsoft Exchange 2010 . . . **39**

Approaching the installation . . . 39
Running /PrepareAD . . . 41
Installing prerequisite system components . . . 42
Installing the Microsoft Filter Pack . . . 46
Running Setup . . . 46
Setup logs . . . 49
Uninstalling Exchange . . . 51
Repairing Exchange . . . 53
Installing an edge server . . . 54
Language packs . . . 54
Recovering a failed server . . . 55
Customer Experience Improvement Program . . . 58
The services of Exchange . . . 60
Versions, roll-up updates, and service packs . . . 63
Exchange 2010 Service Pack 1 . . . 65
Version numbers . . . 66
Object versions . . . 68
Reporting licenses . . . 69
Security groups and accounts created by Exchange . . . 71
Contemplating management . . . 74

Chapter 3: The Exchange Management Shell . . . **75**

How Exchange leverages Windows PowerShell . . . 76
Remote PowerShell . . . 79
Flowing remotely . . . 81
Connecting to remote PowerShell . . . 84
Be careful where you execute . . . 86
A more complex environment to manage . . . 86
Advantages of remote PowerShell . . . 91
EMS basics . . . 93
Command editing . . . 96
Handling information returned by EMS . . . 99
Selective output . . . 100
Using common and user-defined variables . . . 103
Identities . . . 106
Piping . . . 109
Adding recipient photos . . . 111
OPATH filters . . . 113

Server-side and client-side filters . . . 114
Transcripts . . . 117
Bulk updates . . . 118
Code changes required by remote PowerShell . . . 120
Command line versus Integrated Scripting Environment . . . 122
Calling scripts . . . 123
Profiles . . . 124
Script initialization . . . 125
Active Directory for PowerShell . . . 126
Setting the right scope for objects in a multidomain forest . . . 127
Some useful EMS snippets . . . 129
Looking for large folders . . . 129
Outputting a CSV file . . . 130
Creating a report in HTML . . . 131
Finding disconnected mailboxes . . . 132
Creating and sending messages from the shell . . . 132
Reporting database size and mailbox count via email . . . 134
Verbose PowerShell . . . 136
Setting language values . . . 136
Execution policies . . . 137
Testing cmdlets . . . 139
Test-SystemHealth . . . 139
Test-ServiceHealth . . . 140
Test-MAPIConnectivity . . . 141
Test-ReplicationHealth . . . 141
Test-ExchangeSearch . . . 142
Test-OWAConnectivity . . . 143
Test-ECPConnectivity . . . 143
Test-MRSHealth . . . 144
Testing POP3 and IMAP4 Connectivity . . . 144
Testing mail flow . . . 145
But we need some control . . . 146

Chapter 4: Role-Based Access Control . . . **147**

RBAC basics . . . 148
Roles . . . 151
Using role assignment policy to limit access . . . 152
Creating roles for specific tasks . . . 154
Scopes . . . 155
Role groups . . . 156
Creating a new role group . . . 159
Role assignment . . . 160
Specific scopes for role groups . . . 162
Special roles . . . 164
Unscoped roles . . . 165
What role groups do I belong to? . . . 166
Assignment policies . . . 168

RBAC enhancements in SP1 . . . 170
Managing role groups through ECP . . . 170
Database scoping . . . 174
Implementing a split permissions model . . . 175
RBAC reports in ExBPA . . . 178
RBAC validation rules . . . 179
Exchange Control Panel and roles . . . 179
Figuring out RBAC . . . 179
On to management . . . 180

Chapter 5: Exchange Management Console and Control Panel . . . **181**

Exchange Management Console . . . 182
Changes to EMC in Exchange 2010 . . . 182
A different console philosophy from Exchange 2003 . . . 185
Managing objects across Exchange 2010 and Exchange 2007 . . . 187
EMC startup . . . 188
How EMC accesses Exchange data . . . 190
Changing EMC columns . . . 194
Auto-generated PowerShell commands . . . 195
Using EMS command logs . . . 197
Naming conventions . . . 199
Organizational health data . . . 201
Managing multiple organizations . . . 204
Sharing policies . . . 205
Certificate management . . . 208
Exchange Control Panel . . . 213
SP1 updates for ECP . . . 215
An overview of the ECP application . . . 215
Basic ECP user options . . . 216
Inbox rules . . . 220
Delivery reports . . . 224
ECP administrator options . . . 227
Administrator searches for delivery reports . . . 228
Running ECP without an Exchange mailbox . . . 235
Managing groups with ECP . . . 237
Defining a default group location and group naming policy . . . 238
Creating new groups . . . 242
Creating security groups with ECP . . . 243
Users and groups . . . 244
Allowing users to create new groups through ECP . . . 247
Planning for user-created groups . . . 248
Maintain groups but don't create! . . . 249
Setting diagnostics for Exchange servers . . . 251
But what will we manage? . . . 253

Chapter 6: Managing Mail-Enabled Recipients . . . **255**

Stop and think . . . 255
Mailbox naming conventions . . . 257

Creating new mailboxes . . . 259
Completing the new mailbox setup . . . 264
Creating new room and resource mailboxes . . . 265
Mailbox provisioning agent and database allocation . . . 265
Languages and folders . . . 269
Manipulating mailbox settings . . . 273
Bulk mailbox creation . . . 277
Setting quotas . . . 279
What's in a mailbox? . . . 284
Removing or disabling mailboxes . . . 285
Reconnecting mailboxes . . . 286
Email address policies . . . 290
Email policy priority . . . 292
Creating a new email address policy . . . 293
Creating email address policies with custom filters . . . 297
Setting priority for an email address policy . . . 297
Virtual list view (VLV) for Exchange address lists . . . 299
Discovery mailboxes . . . 299
Creating additional discovery mailboxes . . . 301
Setting mailbox permissions . . . 303
Mail flow settings . . . 303
The difference between Send on Behalf and Send As . . . 304
Managing full access permission . . . 306
Sending messages on behalf of other users . . . 309
Opening another user's mailbox . . . 310
Distribution groups . . . 312
Room lists . . . 314
Group owners . . . 316
Group expansion . . . 318
Protected groups . . . 319
Self-maintaining groups . . . 321
Viewing group members . . . 322
Tracking group usage . . . 324
Dynamic distribution groups . . . 324
OPATH queries . . . 325
Creating new dynamic distribution groups . . . 326
Creating dynamic groups using custom filters . . . 329
Moderated recipients . . . 334
Moderation requests . . . 337
Moderated mailboxes . . . 340
Mail-enabled contacts . . . 341
Mail users . . . 342
Resource mailboxes . . . 343
Defining custom properties for resource mailboxes . . . 345
Providing policy direction to the Resource Booking Attendant . . . 347
Processing meeting requests according to policy . . . 352
Equipment mailboxes . . . 355
Data, data, everywhere . . . 355

Chapter 7: The Exchange 2010 Store 357
Long live Jet! 358
Maximum database size 359
Database limits for the standard edition 361
Mailboxes per database (or per server) 362
Dealing with I/O 364
Maintaining contiguity 370
A new database schema 372
Database management 374
Creating new mailbox databases 377
Updating mailbox databases after installation 381
Background maintenance 383
Scheduling background maintenance 387
Content maintenance tasks 388
Tracking background maintenance 390
Corrupt item detection and isolation 391
Backups and permanent removal 394
Protection against high latency 395
Protection against excessive database or log growth 396
Store driver fault isolation 397
The death of ISINTEG 398
Controlling named properties 401
Database defragmentation 404
Using ESEUTIL 406
Database usage statistics 407
Transaction logs 409
Log sets 410
Transactions, buffers, and commitment 413
Transaction log checksum 417
Transaction log I/O 418
The question of circular logging 419
Noncircular logging 421
Reserved logs 422
And now for something completely different 423

Chapter 8: Exchange's Search for High Availability 425
Breaking the link between database and server 426
Introducing Database Availability Groups 428
The dependency on Windows clustering 431
Active Manager 433
Automatic database transitions 435
Best copy selection 437
ACLL: Attempt copy last logs 439
Transaction log replay: The foundation for DAG replication 440
Transaction log compression 445
Block replication 446
Transaction log truncation 448

Incremental resynchronization . . . 449
Seeding a database . . . 451
Unique database names . . . 451
Changes in message submission within a DAG . . . 455
Day-to-day DAG management and operations . . . 455
Building the DAG . . . 462
Investigating DAG problems . . . 468
Managing DAG properties . . . 469
DAG networks . . . 471
Using circular logging with database copies . . . 475
Adding new database copies to a DAG . . . 477
Handling initial seeding errors . . . 479
Monitoring database copies . . . 480
Reseeding a database copy . . . 481
Adding database copies with EMS . . . 482
Using a lagged database . . . 484
Activating a mailbox database copy . . . 488
Applying updates to DAG servers . . . 492
Dealing with a failed server . . . 493
AutoDatabaseMountDial and potential issues moving databases . . . 495
Activation blocks . . . 499
Moving database locations within a DAG . . . 500
Removing database copies . . . 502
Removing servers from a DAG . . . 506
Handling storage hangs . . . 507
Upgrading servers in a DAG . . . 508
Datacenter Activation Coordination . . . 510
Planning for datacenter resilience . . . 511
Managing cross-site connections . . . 513
Crimson events . . . 514
Approaching DAG designs . . . 515
Scripts to help with DAG management . . . 520
On to protecting data . . . 525

Chapter 9: Backups and Restores . . . **527**

An interesting philosophical question . . . 527
The Windows Server Backup plug-in for Exchange . . . 530
Exchange and Volume ShadowCopy Services . . . 531
Making an Exchange 2010 backup . . . 533
The backup complexities posed by passive database copies . . . 537
Restoring to a recovery database . . . 538
Performing a restore . . . 540
Validating the recovered database . . . 543
Mounting a recovery database . . . 544
Restoring mailbox data . . . 547
Complete server backups . . . 552
Clients . . . 553

Chapter 10: Clients 555
The Outlook question 557
Missing functionality when using earlier versions of Outlook 559
Why new mail notifications seem slower on Outlook 561
Forcing faster Outlook Anywhere connections 562
Conversation views 563
Conflict resolution 567
Listing client connections 569
Blocking client connections to a mailbox 570
Blocking client access to a mailbox server 573
Outlook Web App 574
A refresh for OWA provided by Exchange 2010 SP1 575
OWA functionality deprecated in Exchange 2010 578
Different browsers, different experiences 579
OWA configuration file 583
Missing favorites 584
Forwarding meeting requests 585
OWA Web parts 586
Long signatures 587
Sharing calendars 588
Sharing calendars with Internet users 590
Mailbox quota exceeded 594
Handling attachments 595
OWA themes and customizations 597
OWA mailbox policies and feature segmentation 600
More than just segmentation 604
Attachment processing 608
Applying an OWA mailbox policy 609
POP3 and IMAP4 clients 610
Configuring the IMAP4 server 612
Configuring IMAP4 client access 615
Exchange ActiveSync 618
Setting ActiveSync policies 620
Generating ActiveSync reports 622
Reporting synchronized devices 623
Blocking types of mobile devices 626
Blocking devices on a per-user basis 631
Wiping lost devices 632
Debugging ActiveSync 635
Testing mobile connectivity 636
ActiveSync for BlackBerry 636
Client throttling 637
Unified Messaging 641
Voice mail preview 642
Fax integration 647
Exchange 2010 APIs 647
Exchange Web Services 648
A common connection point 650

Chapter 11: Client Access Server 651
The CAS role 652
Benefits of relocating the MAPI endpoint 653
CAS installation priority 655
The RPC Client Access layer 657
Linking CAS to mailbox databases 659
Supporting Outlook 2003 clients 661
CAS access to directory information 662
The Autodiscover service 663
Accessing a Service Connection Point 663
CAS settings 666
Site scope 668
AutoConfiguration 668
Logging Autodiscover actions 670
Static Autodiscover 673
SRV pointers to Autodiscover 675
Client Access Server arrays 676
Creating a CAS array 678
Managing cross-site connections with the RPC Client Access service 679
Load balancing and CAS arrays 681
Upgrading a Client Access Server in an array 682
CAS and perimeter networks 684
RPC Client Access logging 685
Certificates 688
Outlook Anywhere 691
An increased load for the CAS 692
Load balancing the CAS 693
The importance of affinity 696
Assigning static ports to the CAS 698
Web services URLs and load balancing 701
Changes to facilitate SSL offloading 702
Domain controllers 702
Preparing for transition and interoperability 703
A matter of manipulation 705

Chapter 12: Mailbox Support Services 707
The Mailbox Replication Service 707
MRS configuration file 708
Moving mailboxes 709
Asynchronous moving 711
Mailbox Replication Service processing 713
Preventing loss of data 716
Moving mailboxes 717
Clearing move requests 722
Managing mailbox moves with EMS 723
Preserving the mailbox signature 726
Moving mailboxes between versions of Exchange 727
Moving mailboxes with personal archives 729

Checking move request status . . . 731
Planning mailbox moves . . . 732
Ensuring high availability . . . 736
Reporting mailbox moves . . . 738
Accessing move report log data . . . 740
Moves and mailbox provisioning . . . 743
Handling move request errors . . . 744
Mailbox import and export . . . 747
Gaining permission through RBAC to execute mailbox import and export . . . 749
Planning the import of PST data . . . 750
Exporting mailbox data . . . 758
Limiting user access to PSTs . . . 760
MailTips and group metrics . . . 762
Configuring MailTips . . . 766
User experience . . . 768
Custom MailTips . . . 770
Multilingual custom MailTips . . . 771
The Offline Address Book . . . 772
OAB download . . . 773
OAB generation . . . 776
Updating OAB files . . . 781
Moving the OAB generation server . . . 782
Web-based distribution . . . 783
Creating and using customized OABs . . . 785
OAB support for MailTips . . . 790
OABInteg and Dave Goldman's Blog . . . 791
Hierarchical address book . . . 791
Mailbox assistants . . . 793
Calendar Repair Assistant (CRA) . . . 794
Work cycles . . . 797
Time to transport . . . 799

Chapter 13: The Exchange Transport System . . . **801**

Overview of the transport architecture . . . 802
Active Directory and routing . . . 806
Overriding Active Directory site link costs . . . 808
Delayed fan-out . . . 810
The critical role of hub transport servers . . . 811
Version-based routing . . . 813
Transport configuration settings . . . 816
Limits on user mailboxes . . . 822
Transport configuration file . . . 823
Caching the results of group expansion . . . 825
Routing tables . . . 826
TLS security . . . 830
Receive connectors . . . 831
Creating a receive connector . . . 835

Send connectors . . . 841
Creating a send connector . . . 845
Selecting a send connector . . . 851
Linked connectors . . . 853
Throttling . . . 854
Back pressure . . . 857
Transport queues . . . 859
How messages enter the submission queue . . . 861
Moving messages to delivery queues . . . 861
Viewing queues . . . 862
Problem queues . . . 865
Exchange Queue Viewer . . . 867
Submitting messages through the pickup directory . . . 869
Replay directory . . . 871
Customizable system messages . . . 871
Exchange DSNs . . . 871
Customizing NDRs . . . 875
Customizing quota messages . . . 878
Logging . . . 880
Controlling connectivity logging . . . 881
Interpreting a connectivity log . . . 883
Protocol logging . . . 884
Accepted domains . . . 886
Creating a new accepted domain . . . 888
Updating accepted domains . . . 889
Remote domains . . . 889
Transport pipeline . . . 891
Foreign and delivery connectors . . . 893
Shadow redundancy . . . 894
Linking Exchange 2003 to Exchange 2010 . . . 898
Decommissioning Exchange 2003 routing groups . . . 900
Handling Exchange 2003 link state updates . . . 900
Changes in Exchange 2010 SP1 . . . 901
Better SMTP load balancing . . . 902
Monitoring the submission queue . . . 903
Mailbox delivery prioritization . . . 904
Upgraded shadow redundancy . . . 906
Squeaky-clean email . . . 906

Chapter 14: Message Hygiene . . . **907**

To Edge or not to Edge, that's the question . . . 908
Edge servers . . . 909
Edge synchronization . . . 911
Validating Edge synchronization . . . 915
Ongoing synchronization . . . 919
Exchange anti-spam agents . . . 923
Installing the anti-spam agents on a hub transport server . . . 924

Order of anti-spam agent processing . . . 925
X-headers added by anti-spam agents . . . 926
Header firewalls . . . 929
Connection filtering . . . 931
Sender filtering . . . 934
Backscattering . . . 935
Sender reputation . . . 936
Recipient filtering . . . 939
Tarpits . . . 940
Sender ID . . . 940
Content filtering . . . 946
Attachment filtering . . . 953
Address rewriting . . . 955
Agent logs . . . 957
Safelist aggregation . . . 961
Choosing an antivirus product . . . 964
Client defense . . . 965
Outlook's junk mail filter . . . 966
Cleansed email, but compliant? . . . 972

Chapter 15: Compliance . . . **973**

The joy of legal discovery . . . 974
Personal archives . . . 976
Enabling a personal archive . . . 979
Default archive policy . . . 985
Disabling a personal archive . . . 987
Using a personal archive . . . 987
Messaging records management . . . 989
The new approach to messaging records management in Exchange 2010 . . . 990
System tags . . . 994
Designing a retention policy . . . 995
Naming retention tags . . . 997
Creating retention tags . . . 998
Creating a retention policy . . . 1004
Applying a retention policy to mailboxes . . . 1007
Modifying a retention policy . . . 1009
Customizing retention policies for specific mailboxes . . . 1010
User interaction with retention policies . . . 1012
Removing a retention policy . . . 1017
Upgrading from managed folders . . . 1018
How the Managed Folder Assistant implements retention policies . . . 1018
Putting a mailbox on retention hold . . . 1021
Putting a mailbox on litigation hold . . . 1022
The very valuable dumpster . . . 1025
Dumpster basics . . . 1025
Dumpster 2.0 arrives . . . 1027
Single item recovery . . . 1029

Knowing what's in the dumpster . . . 1031
Managing dumpster parameters . . . 1032
Discovery searches . . . 1033
Unsearchable items . . . 1035
Creating and executing a multimailbox search . . . 1037
Accessing search results . . . 1040
Deduplication of search results . . . 1043
Search logging . . . 1045
Search annotation . . . 1046
Executing searches with EMS . . . 1047
Auditing administrator actions . . . 1049
The audit mailbox . . . 1052
How administrator auditing happens . . . 1052
Auditing mailbox access . . . 1057
Enabling mailboxes for auditing . . . 1059
Accessing mailbox audit data . . . 1061
Message classifications . . . 1064
Creating a message classification . . . 1065
Localized message classifications . . . 1067
Client access to message classifications . . . 1067
Protecting content . . . 1070
Active Directory Rights Management Services . . . 1072
Installing Active Directory Rights Management . . . 1073
Using AD RMS to protect content . . . 1076
Rights management enhancements in Exchange 2010 SP1 . . . 1080
Outlook Protection Rules . . . 1080
Rules help compliance, too . . . 1082

Chapter 16: Rules and Journals . . . **1083**

Transport rules . . . 1083
Examples of transport rules . . . 1085
Rules and ECP . . . 1087
Basic structure of transport rules . . . 1088
Edge versus hub rules . . . 1088
Setting transport rule priority . . . 1089
Creating a corporate disclaimer . . . 1091
Basic moderated workflow . . . 1097
Evaluating Active Directory attributes in transport rules . . . 1099
Ethical firewalls . . . 1101
Blocking certain users from sending external email . . . 1102
Scanning attachments with transport rules . . . 1105
Using message classifications and rights management templates in transport rules . . . 1108
Caching transport rules . . . 1110
Transferring rules between Exchange versions . . . 1111
Transport rule actions . . . 1112
Developing custom transport agents . . . 1113

Transport rule priority 1114
Journaling 1114
When journaling happens 1115
Journaling options 1116
Journal reports 1116
Alternate journal recipient 1120
Standard journaling 1121
Journal rules 1122
Creating a journal rule 1123
Assessing journal load 1125
Securing a mailbox used as a journal recipient 1126
Interversion and interorganization journaling 1127
To the toolbox 1127

Chapter 17: The Exchange Toolbox **1129**
Display or Details Templates Editor 1130
Message tracking 1135
Message tracking log files generated on servers 1139
Interpreting entries in message tracking logs 1142
Measuring message latency 1151
Using the Tracking Log Explorer 1153
Other options for analyzing messaging tracking logs 1158
Performance Monitor 1159
Exchange Performance Troubleshooter 1162
ExPerfWiz 1162
ExPerfWiz limitations 1164
Exchange Load Generator 2010 1165
Remote Connectivity Analyzer 1167
Searching for more information 1170

Index to Troubleshooting Topics 1171

Index 1173

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

Microsoft is interested in hearing your feedback 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/

Foreword

I took over the Exchange team in 2008 after 18 years in various roles at Microsoft, and was welcomed to the team appropriately via a post on the "You Had Me at EHLO" Exchange team blog. In November 2009, not too long after taking on this new mantle, I was in Las Vegas preparing to keynote the Exchange Connections conference to celebrate the launch of Exchange Server 2010. Knowing that I'd have some time to get to know members of the much-lauded Exchange community at the event, I reached out to my team for suggestions on which customers to seek out, which partner booths to visit, and any other advice they had. If one thing was universally clear it was that I had to—simply must—meet Tony Redmond.

Tony has been a fixture in the Exchange community for over a decade. Whether it is his advocacy for the Exchange customers or his critical feedback toward improving the product, Tony has played a significant role in the Exchange ecosystem since before the first Exchange Server ever shipped out of Redmond. He is one of the most popular speakers and authors on Exchange and an important voice for every one of the executives who preceded me as leader of Exchange at Microsoft.

It is appropriate that, after 14 years, Tony is publishing a book on Exchange Server 2010 SP1, a release that has so much to do with responding to customer and community feedback through early adopter and beta programs in which he has been so active over the years. Although an important milestone for the server, Exchange 2010 SP1 is also a significant milestone in our cloud strategy. This is the version of Exchange that we carry to our own datacenter as we bring the capabilities of Exchange 2010 to the cloud. It is unprecedented in the industry to provide a product that so comprehensively increases the operational efficiency of IT, makes users more productive in their daily workflow, reduces the risk profile of an organization, and brings this value to customers whether they choose to deploy servers on premises or migrate mailboxes into the cloud.

It is this unprecedented choice and flexibility that makes Exchange so unique and SP1 so important. It is with this focus that my team marches forward as we plan for the next updates to Exchange Online and the next versions of Exchange Server. Exchange 2010 SP1 makes me excited for the future of Exchange and I hope it does the same for you.

On behalf of my entire team, thank you for being part of our Exchange family and enjoy the book!

Rajesh Jha
Corporate Vice-President, Exchange
Microsoft Corporation

Introduction

Welcome to my tenth book covering the technology in Microsoft Exchange Server and its surrounding ecosystem. I seem to have been constantly writing about Exchange since before its introduction as version 4.0 in March 1996 in books and many articles printed in the redoubtable *Windows IT Pro* magazine (*http://www.windowsitpro.com*). All my previous books were published under the Digital Press imprint, which has now disappeared as a result of corporate upheavals. This is my first book working with Microsoft Press and it's been an interesting and productive experience for me to work with the publishing arm of the company that engineers Exchange. I look forward to future collaboration.

No book can cover every aspect of a huge product such as Exchange. To attempt to do so would require a multivolume set spanning many thousands of pages and create something that would probably be too expensive for most administrators to buy. This book covers the topics in Exchange that are most interesting to me and those that I think are most useful to the majority of administrators who need to understand how to manage an Exchange organization. There are some notable omissions, such as Unified Messaging and Exchange's connection to other Microsoft products such as Office Communications Server, which you might find surprising. However, the truth is that there are other books available that do a good job of covering these topics, so I feel able to concentrate on the areas that I think deserve the most investigation (or are most interesting to me). There's also an incredible amount of information posted in blogs and other commentaries available on the Web, so if your interest is piqued by a topic and you want to find more information, plug the topic into your search engine of choice and you're likely to find additional insights and observations. Apart from anything else, you'll discover information that is up to date and reflects advances due to software updates (I predict that Microsoft will continue to upgrade Exchange 2010 after Service Pack 1!) and the knowledge that accumulates over time about any product that's used in production environments.

Exchange 2010 has been an interesting journey because it provoked more new thoughts and ideas for me than any other version released by Microsoft. Although Exchange 2007 laid down much of the architecture that Exchange 2010 exploits, there is a mass of detail in the changes between the two versions. Two of the three big changes effected in Exchange 2007—Windows PowerShell, the transition to a pure SMTP-based transport system, and the introduction of transaction log shipping as the basis for database replication—have been expanded and enhanced in a very effective manner. Remote Windows PowerShell and the Database Availability Group might be what people remember as the big new things that appeared in Exchange 2010, but their foundation was laid many years previously and what we see today is simply the result of a lot of development and hard work since Microsoft finished the development of Exchange 2003. Maybe that's why there is so much to discuss and comment on.

Service Pack 1

I decided to base the book on Exchange 2010 Service Pack 1 (SP1), which Microsoft released to customers in August 2010, because I felt that there were a number of important areas that were incomplete in the original (RTM) version of Exchange 2010 released in October 2009. The fact that Microsoft needed additional time and effort to fully build out the features in Exchange 2010 should come as no surprise to anyone who has any experience with Exchange.

Don't get me wrong: The software that Microsoft shipped to customers in late 2009 was highly functional and had many strong points. However, the nature of software development is that a little extra time always helps to smooth rough edges and deliver the complete plan that the engineers wanted to build when they started to work on Exchange 2010. SP1 had the benefit of nine months' more development, testing, and documentation, plus the invaluable exposure that software receives when customers deploy it into production environments. The experience gained from this time, the feedback from customers and the Microsoft Most Valuable Professional (MVP) community, the insight shared in blogs and forums, and the bugs that were uncovered and fixed made SP1 a better target about which to write. Besides, I hate books that are rushed to market to meet an arbitrary date so that they can be first to market to cover new technology, because I know the dangers involved in writing about beta versions of technology. You can hope that the developers won't change the software between the beta and final version, but all too often a diversion appears between the description on the printed page and what the software actually does. It's safer to wait and see what the experience is with the software after it is released rather than rush to market to sell a few more books.

Many major and minor differences exist between the original version of Exchange 2010 and SP1 and I have attempted to indicate clearly where these differences exist.

Writing style and general approach to content

My writing approach to books is best described as chaotic and not very organized. I don't start with a list of topics and work through from A to Z until the book is done. I think I would find such an approach tiresome and would lose interest halfway through. Instead, I build the book from page to page and split content into chapters as the form of the book develops. Text is added as I discover new and interesting facts about the technology. I make no apologies for this approach, because it has served me well in my previous 12 books. However, I acknowledge that topics that don't interest me are omitted or receive short shrift. Ah well, you can't be brilliant at everything and you have to leave room for others to prosper.

Along these lines, I made some deliberate decisions about topics that I would not cover in this book to concentrate on what I believe are the most important technical and functional

changes in Exchange 2010. In effect, I used the 80–20 rule to select topics that I believe are of interest to the vast majority of the Exchange community and omitted others. So, to take two examples, if you are interested in the finer technical points of Unified Messaging or Active Directory Rights Management Services, you'll probably want to buy a different book. I think that these topics deserve specialized in-depth attention that cannot be justified in a book that attempts to cover the fundamental principles of Exchange. The same is true about connecting Exchange organizations with federation or integrating Exchange with various other products such as Microsoft Office Communications Server. These tasks can be done and are often done, and the subjects are explained better elsewhere. Be warned, therefore, that you might be disappointed if your favorite topic is not covered. On the other hand, you might be delighted that a topic of special importance to you is covered across many pages because we share a joint interest in it. In passing, I note that three chapters were removed from the original draft of the book to get the book down to a reasonable size. These chapters covered public folders, hardware planning, and cloud-based Exchange.

I doubt that many will read the book from beginning to end. At least, I have not written the text to flow from one chapter to another in the way that a novel or a history does. I expect most people to plunge into a part that interests them and then discover new topics as the need arises or curiosity takes over. I like technical books where chapters are self-sufficient and I hope that this book meets that goal.

In terms of other sources of technical information about Exchange 2010, I recommend that you download the latest version of the compiled help file (CHM) and keep it close at hand, because it will provide you with an invaluable guide to Exchange 2010 that you can use offline. TechNet provides an online copy, but Microsoft is quite good at updating the downloadable CHM regularly and did an excellent job for SP1. They've also gotten a lot better in terms of the breadth and depth of the content, even if it sometimes leaves gaps in the explanation. Of course, those gaps are exploited by the mass of Exchange 2010 books, magazine articles, and blogs. I particularly like the blogs of my fellow Exchange MVPs, even if it took me a long time before I got around to writing a blog myself. I now blog regularly at *http://thoughtsofanidlemind.wordpress.com/* and hope that I add some value to the Exchange community there.

Examples used in the book

I've tried to include as many examples as I can to illustrate points and show you exactly what you can expect to see when you execute a command. There are three kinds of examples:

- A simple screen shot. Hopefully these are self-evident. I've tried hard to avoid liberally scattering screen shots across the pages of the book because I hate big, thick, technical books that are half composed of screen shots. However, given the graphic nature of Windows and the Exchange management GUI, some screen shots are inevitable! In

most cases I have opted to use Outlook Web App to illustrate client functionality simply because you can be guaranteed that Outlook Web App is available within every Exchange 2010 deployment. In addition, huge variation exists in the feature set available in the Outlook versions supported by Exchange 2010; Outlook 2010 is therefore only used to illustrate unique features that are only exposed in this version.

- Illustrative Windows PowerShell (Exchange Management Shell) commands. If only because it is the foundation of Exchange 2010, there are many EMS examples throughout the book. If you don't know how to use EMS to manage Exchange, you miss out on so much of the potential that the product offers. All references to cmdlets in the body of the text plus example commands are shown like this:

  ```
  Get-User –Identity Redmond
  ```

- Windows PowerShell commands and example output. In some cases I want to show you exactly what you will see when a command is executed. Windows PowerShell output can be pretty ugly and reading it from a screen shot is not always easy, so I show the Windows PowerShell command before and then the output in a separate block immediately following the command:

  ```
  Get-StoreUsageStatistics –Database 'VIP Data'
  ```

```
DigestCategory       SampleId         DisplayName               TimeInServer
----------------     ------------     -----------------         ---------------
TimeInServer         0                Mailbox –Redmond, Tony    485
```

I'm certainly not an elegant or skilled Windows PowerShell coder. Rather, I like to think of myself as a contented hacker who fully buys into the concept that the charm of Windows PowerShell is that you can quickly stitch together snippets of code to do remarkable things. I apologize if I offend any purists with the examples presented in the book. I just do my best to make things happen with the best code I can.

The examples are based on an Exchange organization called contoso.com that runs on my notebook. It's really amazing how notebook technology has evolved to a point where a portable computer can happily support several servers while still allowing you to run client applications such as Word and Outlook that are available through a simple Alt+Tab keystroke.

Thanks

David Espinoza, Senior Product Manager in the Exchange "ship team," has been a delight to work with for many years. The ship team does what its title indicates: It is responsible for the complex choreography involved in shepherding a product from development to a point where it can be used by customers. David's team runs the Technology Adaption Program

(TAP), which puts beta versions of Exchange into customer hands early so that code can be exercised in real-life environments. The ship team organizes regular calls to inform people about new features and development progress, makes new builds available on a regular basis, and acts as the interface for bugs and feature requests that flow back from customers into Microsoft. All in all, the team does a standout job and David and his faithful assistants Robbie Roberts and Kern Hardman deserve my full thanks.

I've also received great advice and information from many individual contributors in the Exchange development group, including Dmitri Gavrilov, Jim Edelen, Kumar Venkateswar, Crystal Flores, William Rall, Julian Zbogar-Smith, Derek Tan, Kamal Janardhan, David Los, Sanjay Ramaswamy, Frank Byrum, Robin Thomas, Yesim Koman, Todd Luttinen, Linden Goffar, Ed Banti, Jim Knibb, Mayerber Carvalho Neto, Greg Taylor, Paul Bowden, and Siddhartha Mathur. I also acknowledge the help that I received from Bill Long to make the ExFolders utility work with beta builds of SP1.

Martin DelRe first contacted me in August 2008 to explore the possibilities of writing a book for Microsoft Press. Exchange 14, which is what Exchange 2010 was known as then, seemed like a good target and there was certainly plenty of new stuff to write about, but I was busy and didn't become enthused about the project until June 2009. Martin guided me through the process of writing for a new publisher (they are all different) and made sure that I didn't have to cope with too much bureaucracy, for which I am very thankful. Karen Szall directed the production of the book from submission through copyediting, technical review, and eventual publication, and did a wonderful job of making everything come together on time, including the ability to manage a constant flow of changes from me as I attempted to update the technical content of the book to match new experiences, insights, and reports of problems and workarounds discovered in the field.

A group of willing volunteers within HP who have enormous experience with enterprise messaging systems reviewed the material as it was under development. I'd like to thank Marc Van Hooste, Kevin Laahs, Andreas Zit, and Mike Ireland for their advice.

HP has one of the largest deployments of Exchange, with more than 350,000 mailboxes supported by a highly centralized datacenter structure centered in the United States. Kathy Pollert, Mike Ireland, and Stan Foster have contributed greatly to my understanding of how Exchange functions in very large environments and I truly appreciate the insight that they have shared with me over the last decade. I'd also like to thank Aric Bernard and Guido Grillenmeier for allowing me to deploy new software into the sandbox environment used by HP consultants on a regular basis. Aric and Guido are Active Directory gurus and would probably prefer that Exchange was kept well away from their nice, clean domain controllers. Into every life a little rain must fall and every Active Directory administrator has to learn that the directory is there to serve applications rather than to explore the wonders of replication. We had fun getting Exchange 2010 to even install into HPQBOX, mostly because a previous deployment of Exchange 2007 had been removed through brutal surgery applied

to the directory, leaving many lingering objects that just got in the way. Guido cleaned everything up and Aric made the servers run nicely, and I appreciate their efforts.

Finally, the dedicated effort of Paul Robichaux as technical editor must be acknowledged. Paul read every page, challenged places where I didn't seem to make sense or had misstated something, and ferreted out errors, all to improve the overall quality and content of the book. I owe him a lot.

In conclusion

I hope that you enjoy the book and its contents prove valuable in your understanding of Exchange 2010 as you approach the assessment, design, deployment, operations, and management of the software in production. At the end of the day, Exchange is only software and it's only as good as the people who work with it. To do a good job, you need knowledge about the product and wisdom to know when that knowledge runs out and it's time to look elsewhere for help, whether that's seeking out a Web site or asking someone for an opinion. The key is to realize that best practice is never stagnant and evolves all the time. Assuming that knowledge gained two or three years ago represents the current best advice and guidance is a fool's position. Always be prepared to learn.

Tony Redmond
September 2010

The author can be contacted through his blog at
http://thoughtsofanidlemind.wordpress.com/.

Support for this book

Every effort has been made to ensure the accuracy of this book. If you do find an error, please report it on our Microsoft Press site

1. Go to *www.microsoftpressstore.com*.
2. In the Search box, type the ISBN for the book, and click Search.
3. Select the book from the search results, which will take you to the book's catalog page.
4. On the book's catalog page, find the Errata & Updates tab

If you have questions regarding the book or the companion content that are not answered by visiting the book's catalog page, please send them to Microsoft Press by sending an email message to *mspinput@microsoft.com*.

We want to hear from you

We welcome your feedback about this book. Please share your comments and ideas via the following short survey:

http://www.microsoft.com/learning/booksurvey

Your participation will help Microsoft Press create books that better meet your needs and your standards.

> **Note**
>
> **We hope that you will give us detailed feedback via our survey. If you have questions about our publishing program, upcoming titles, or Microsoft Press in general, we encourage you to interact with us via Twitter at *http://twitter.com/MicrosoftPress*. For support issues, use only the email address shown above.**

Your Companion eBook

The eBook edition of this book allows you to:

- Search the full text
- Print
- Copy and paste

To download your eBook, please see the instruction page at the back of this book.

CHAPTER 10

Clients

The Outlook question 557
Outlook Web App 574
OWA mailbox policies and feature segmentation 600
POP3 and IMAP4 clients 610
Exchange ActiveSync 618
Client throttling 637
Unified Messaging 641
Exchange 2010 APIs 647
A common connection point 650

Microsoft has pursued a multiple client access strategy for Exchange since it first introduced Web access and Post Office Protocol 3 (POP3) support in Microsoft Exchange 5.0. The first browser client was introduced at a time when the set of supported clients was very limited and largely centered on the existing "fat" client (the original Exchange Messaging Application Programming Interface [MAPI] viewer) and Microsoft Outlook 97. Although it could connect to Exchange, the browser interface was slow, only supported Internet Explorer, couldn't scale because it depended on MAPI, and lacked significant functionality, yet it proved that a Web-based interface was viable and laid the basis for a client that has become more and more popular in each new version. Microsoft established the third leg in their client access strategy with the introduction of server-based ActiveSync in Microsoft Exchange 2003 to allow Exchange to deal with an increasing demand for mobile access to mailbox, calendar, and contact information. Of course, RIM's BlackBerry had satisfied the same demand for some years beforehand, but somehow the requirement for mobile access seemed to be more legitimate (at least, for Exchange administrators) when Microsoft delivered ActiveSync. It also made it far cheaper to support mobile clients because ActiveSync is part of the base server and didn't require expensive software licenses or the deployment of additional servers.

Despite some early problems and a Windows Mobile client that continues to lag behind its competitors in terms of usability and features, the level of integration with Exchange that ActiveSync boasts, together with its unbeatable price point (zero extra cost), means that it has had a huge impact in driving mobile access for Exchange.

Microsoft's client access strategy supports the connection of a huge array of clients to Exchange 2010. Exchange Server 2010 supports a variety of different client types.

- Maximum functionality and features are available in Microsoft's own "fat" clients for Windows and Apple Mac that are part of the Office family. Microsoft doesn't support

connecting very old versions of Outlook to Exchange 2010, so you'll need to deploy at least Outlook 2003 before you can connect to Exchange 2010. If you don't like buying client software from Microsoft, you can use the Internet Message Access Protocol 4 (IMAP4) or POP3 protocols to connect anything from a free Microsoft client (like Windows Mail) to a mobile device that doesn't support ActiveSync.

- If you decide to use Outlook Web App (OWA), you have a range of supported Web browsers from Internet Explorer to Firefox, Safari, and Chrome. As you'd expect, versions 7 and 8 of Internet Explorer deliver maximum functionality, and you can get the same experience (OWA Premium) if you use Firefox or Safari. However, OWA Premium is supported only when Safari runs on Apple Mac OS X. Other browsers, including Opera, can use the downgraded OWA Basic or light version, which is still highly functional, if not quite as flashy as the premium edition. The full matrix of supported browsers for different versions of Exchange is available at *http://technet.microsoft.com/en-us/library/ff728623.aspx*.

- In the past, Microsoft's strategy for mobile clients has been centered on the partnership of server-based ActiveSync and Windows Mobile clients that run Outlook Mobile. Even today, you need to run Windows Mobile 6.5 or later for clients to enjoy the latest experience, but you can upgrade the Outlook Mobile application on Windows Mobile 6.1 to access the enhanced features delivered by Exchange 2010. I expect that Microsoft mobile clients will continue to deliver highly functional new versions of Outlook Mobile. However, the push to expand the set of available ActiveSync clients has gathered momentum over the last few years, and Microsoft has been very successful in licensing ActiveSync to companies that build mobile clients and mobile applications—from Apple to Google to Nokia to Palm—so it is not difficult to find suitable devices; in fact, restricting the number of device types that connect to Exchange is often a challenge for administrators. If you are among the millions of corporate email users who depend on their BlackBerry, you can continue to use the latest version of RIM's BlackBerry Enterprise Server (BES) to connect BlackBerry devices to Exchange 2010 (an upgrade for BES is necessary to deal with the new application programming interfaces [APIs] introduced in Exchange 2010).

The interesting thing about Microsoft's client access strategy is how much improvement has been made in the Web and mobile platforms in the last few releases. New APIs for browsers, general availability across an extremely wide range of mobile devices, smarter networking, and hard engineering effort has enabled Microsoft to get to a point where they can credibly claim to have delivered on "three screens."

The Outlook question

The perennial issue that comes to mind once Microsoft ships a new version of Exchange is what you should do with Outlook. In the past, Outlook and Exchange had a tenuous relationship. For whatever reasons in the depths of Microsoft politics, the two product groups didn't work together particularly well, and despite the fact that Exchange was easily the most functional and powerful mail server to which Outlook could connect, the focus of the Outlook development group seemed to be far more on Internet mail servers. In some respects, this was natural because far more people use Outlook as part of the Microsoft Office suite in non-Exchange environments (home, college, and connecting to other email systems, including Gmail and Lotus Notes), but it was puzzling at the same time, especially because Outlook's support for IMAP seems weaker than other clients such as Thunderbird or Eudora. Things began to improve in Outlook 2003 when Microsoft did the work to introduce cached Exchange mode and made many changes to improve Outlook's networking demands. Cached Exchange mode has proven to be fundamental for Exchange because without it Microsoft's foray into hosted Exchange online services would be much more difficult. It's also fair to say that the ability of cached Exchange mode to isolate users from network failures has greatly improved the user experience.

Further improvements occurred in Outlook 2007, which was released alongside Exchange 2007, and the two product groups seemed to share a common approach to solving the problems of large-scale enterprise-class deployments. Alas, the release of Exchange 2010 marks a divergence, as Outlook 2010 was released sometime after Exchange appeared to raise the inevitable issue of whether to wait to deploy the latest generations of server and client together or to go ahead with Exchange and deploy Outlook afterward. As we will see, the question isn't simply a matter of deployment timing, because some functionality in Exchange 2010 is dependent on client-side code incorporated in Outlook 2010 or simply works better with Outlook 2010.

Answering this question is easier for small companies than it is for large ones. The law of numbers conspires to create much greater complexity when a new application must be distributed to tens of thousands of desktops and issues such as user training, preparing the help desk to support the rollout, and the cost of new software licenses and potential hardware upgrades are considered. This is the reason so many companies continue to run Outlook 2003 or even earlier clients; they see no logic in going forward with an upgrade that promises great cost for new licenses and deployment while offering little obvious return in the form of user productivity, lower support costs, or anything else. The fact that the Exchange server CALs no longer include a license for Outlook will also make it harder for companies to justify an early upgrade.

However, there is no doubt that Outlook 2010 brings some interesting new functionality to the equation. Whether the new features are worthwhile enough to consider an upgrade is different for every company. To begin the debate, Table 10-1 provides a quick summary of the benefits included in the Outlook versions that you can deploy with Exchange 2010.

Table 10-1 Comparing different versions of Outlook

Outlook version	Major benefits for Exchange deployments
Outlook 2003	Introduction of cached Exchange mode and smarter networking to enable faster and more efficient synchronization between server folders and local replicas. Exchange 2010 requires Outlook 2003 SP2.
Outlook 2007	Introduction of AutoDiscover functionality to enable automatic configuration of user profiles. Movement away from public folders as the repository for shared data such as free/busy and Offline Address Book to use Web-based distribution instead. First implementation for managed mail and retention policies.
Outlook 2010	The first 64-bit version of Outlook (also available for 32-bit platforms). Supports features such as MailTips and message tracking from within Outlook. Far more developed and feature-complete version of messaging record management (document retention) policies. Supports cross-organization calendar sharing to help customers deploy in mixed on-premise/hosted deployments. Supports conversation view of email threads (also works with earlier versions of Exchange) as well as the ability to ignore threads you're not interested in. Outlook 2010 also supports personal archives located on Exchange 2010 servers and has the ability to open up to 15 Exchange mailboxes in addition to the primary mailbox.

Outlook 2010 is able to open up to 15 Exchange mailboxes concurrently, not all of which have to belong to the same Exchange organization. By default, Outlook imposes a limit of four mailboxes. This is deliberately set to prevent Outlook from taking up huge amounts of system resources, which would occur if someone attempted to open 10 or 20 mailboxes. However, you can increase the limit for concurrent open mailboxes to 15 by updating the value held in the system registry at HKCU\Software\Microsoft\Exchange\MaxNumExchange.

INSIDE OUT

How the new shared mailbox auto-mapping feature works

Exchange 2010 SP1 includes the ability to auto-map shared mailboxes for Outlook 2010 clients. If you assign Full Access permission to a mailbox using the EMC wizard or the Add-AdPermission cmdlet, Exchange updates the *MsExchDelegateListLink* attribute for the shared mailbox with the distinguished name of the mailbox that is now allowed to open the shared mailbox. When Outlook 2010 connects to Exchange, it receives details of the shared mailbox in the manifest provided by the Autodiscover service and is able to open the shared mailbox underneath the user's primary mailbox. This feature

to configure Outlook profiles to include shared mailboxes. However, it depends on the population of the *MsExchDelegateListLink* attribute and any permissions set before the deployment of Exchange 2010 SP1 will not be reflected in the attribute. You therefore have to remove and reassign Full Access permission to shared mailboxes before they automatically appear in Outlook.

Missing functionality when using earlier versions of Outlook

Outlook 2007 and Outlook 2003 are happy to connect to Exchange 2010 but were obviously designed and engineered to operate against previous versions of Exchange and therefore do not include the code necessary to deal with some of the enhancements incorporated in Outlook 2010. To illustrate the point, after you connect Outlook 2007 to Exchange 2010, a number of features are unavailable, including the following:

- No user interface is available to display the MailTips provided by the server.
- Conversation views. Outlook 2007 doesn't understand the internal identifiers that Exchange uses to connect related items into a conversation. The ability to clean up a conversation and remove obsolete items is also missing, as is the Ignore button.
- Integration with Exchange Control Panel (ECP) to access group information, newer Unified Messaging (UM) settings (such as call answering rules), and so on. However, users can still open ECP to access these options.
- Microsoft has announced that they will release code to allow Outlook 2007 to connect to personal archives. This code has not yet been made available to customers at the time of writing.
- Outlook 2007 is able to render voice mail previews as plain HTML in the message body but lacks the control used to play the voice content if you click part of the voice mail preview; Outlook 2007 also cannot process protected voice mail.
- You cannot send Short Message Service (SMS) messages from Outlook 2007.
- There is no user interface to support retention tags and polices. However, Exchange will apply the actions required by retention policies to user mailboxes even if they use Outlook 2007.

If they are configured to use encrypted remote procedure calls (RPCs), Outlook 2003 SP2 clients can connect to Exchange 2010. However, the elimination of User Datagram Protocol (UDP) support in Exchange 2010 causes a problem for Outlook 2003. Outlook 2003 depends on UDP packets for new mail notifications and to update folders in user

mailboxes. UDP notifications were an appropriate mechanism for this work when clients had to connect over a corporate network (or with a virtual private network [VPN]) to access mailboxes, but they are less useful as connectivity has moved toward a model where pervasive access across the Internet becomes the preferred model. Outlook 2003 supports a polling mechanism as a backup when UDP is not supported. The polling mechanism was provided to support the first Outlook clients that connected to Exchange 2003 servers using RPC over HTTP, but it does lead to a delay of up to one minute before the UDP notification fails and polling delivers notification that a new message has arrived. The problem is less noticeable when Outlook works in cached Exchange mode because of the asynchronous nature of operations, but it still exists. Thus, although you can connect Outlook 2003 SP2 clients to Exchange 2010, users might notice that notifications aren't as snappy as they were before. In order of attractiveness, the available options to address the issue are as follows:

1. Upgrade clients to Outlook 2007 or greater to remove UDP from the equation.

2. Reconfigure Outlook 2003 clients that work in online mode to work in cached Exchange mode (this is always recommended; there are many other advantages to be gained when clients are deployed in cached Exchange mode).

3. Change the polling interval so that notifications arrive faster. By default, Outlook 2003 clients poll every 60 seconds. You can reduce this interval to 10 seconds (Outlook ignores smaller intervals) by updating the system registry on Client Access Server (CAS) servers as described in *http://technet.microsoft.com/en-us/library/aa996515(EXCHG.80).aspx*. A reduced polling interval inevitably generates some increased load on the server, so this is not something to do on a whim.

There are other issues with Outlook 2003 that make this client a less than optimum client for Exchange 2010. The Exchange development group has described the most important issues that affect Outlook 2003 when it connects to Exchange 2010 at *http://msexchangeteam.com/archive/2010/04/23/454711.aspx*. The fact that such a long list of potential problems exists does not make Outlook 2003 bad software, because it was an excellent client when Microsoft first released it in 2003. However, the degree of change that has taken place since in databases, connectivity, and environments has made it difficult for Outlook 2003 to remain as usable as it once was, and it's probably time to refocus efforts on upgrading to a newer client in conjunction with Exchange 2010 deployments.

INSIDE OUT

What is the significance of the UDP problem?

The UDP problem or the list of features only available to Outlook 2010 clients is not mission critical, nor is it sufficient to justify an upgrade for thousands of desktops to Outlook 2010. On the other hand, this list does underscore the close development relationship between client and server and the fact that if you want to achieve maximum functionality from a server, you need to deploy a client that understands how to exploit all of the functionality that the server can offer.

Why new mail notifications seem slower on Outlook

As we've just discussed, Outlook began to transition from using UDP notifications after the introduction of RPC over HTTP. Outlook 2007 and Outlook 2010 use asynchronous RPC notifications because these notifications work through firewalls, whereas UDP usually does not.

Notifications tell Outlook that a change has occurred, such as the arrival of a new message in the mailbox on the server to which it is connected. Outlook still has to fetch details of the change to be able to display it to the user, but processing might be inefficient if Outlook leapt into action immediately. The nature of email is that several changes might occur rapidly at times of peak demand. For example, morning sessions are often marked by flurries of email as users come into work and process their Inboxes before setting out to address the other challenges of the day. If Outlook responded to a notification immediately, it would run the risk that several other new messages might arrive while it is processing the first, which would then force Outlook to engage in a back-and-forth conversation with the server. It is more efficient to batch changes and process them at the same time, which accounts for why you sometimes see several new messages appearing in your Inbox at once when other clients such as Outlook running on a Windows Mobile device display the arrival of individual messages.

When Outlook receives a notification, it sets off a 5-second timer. If no further notification occurs before the timer elapses, Outlook fetches and processes the change. If another notification arrives before the timer expires, Outlook resets the timer and waits again. If the second timer expires, Outlook batches the two notifications and processes them in one operation. However, if continuous changes are detected and the timer keeps being reset, Outlook waits for 60 seconds to let everything settle down on the server and then retrieves whatever is queued and processes these items.

Note

This mechanism is much more efficient in terms of bytes passing over the wire and in the use of system resources; it avoids a continual dialog between clients and servers during a time when the server is already busy (because it's dealing with a lot of new messages). The mechanism also works well over high-latency networks and is an appropriate way of dealing with the transient interruptions that these networks often experience.

The article at *http://technet.microsoft.com/en-us/library/cc179175.aspx* describes how to configure Outlook 2007 and Outlook 2010 clients to operate in cached mode, including how to alter the 5-second timer interval and the 60-second interval for batched changes. Reducing the timer to, say, 2 seconds will accelerate delivery of new mail at the expense of consuming more system resources to deal with additional synchronization requests. As pointed out earlier, this could have an impact at times of peak demand because you'll force the server to respond to additional requests from clients. For this reason, the wisdom of making a change in this area is unproven. In any case, the only people who are likely to realize that Outlook is slightly slower at announcing the arrival of new messages are (a) those who carry multiple devices and can measure the arrival of a new message on each device, and (b) personnel who insist on being able to access new mail within the nearest nanosecond of its arrival. In most cases, normal human beings don't care very much.

Forcing faster Outlook Anywhere connections

Outlook clients use RPCs to connect to Exchange. The RPCs can flow over TCP or HTTP. Clients seldom need to use HTTP in an environment where clients predominantly connect using an internal network (including VPNs), but an increasing number of connections now occur across the Internet in a mode referred to as Outlook Anywhere. This mode suits users who connect using wireless networks at home or in public places. If a deployment supports many clients who use Outlook Anywhere, you can configure Exchange to force Outlook 2010 clients to attempt to make HTTP connections before they use TCP. This is the reverse of the norm that has applied to date, and its value is that it avoids the need for Outlook to fail in an attempt to connect using TCP before it connects with HTTP, and thus speeds up the time before a client is online. Use this command to make HTTP connections the default mode:

```
Set-OutlookProvider EXPR -OutlookProviderFlags ServerExclusiveConnect
```

This command only affects Outlook 2010 clients. Outlook 2003 and Outlook 2007 clients will continue to operate as before.

To reverse the change:

```
Set-OutlookProvider EXPR -OutlookProviderFlags None
```

Conversation views

Earlier versions of email never included the text of previous messages in replies because doing so would add too much overhead to messages, an issue that was important in the days of dial-up connections and expensive disks. Incorporating all previous replies into messages only became common after PC clients such as Microsoft Mail introduced it as a "feature." Today, including the text of previous messages is default behavior for most email clients and it has become a blight. Although it is sometimes useful to understand the context of a conversation, this extra information is usually unwanted, unnecessary, and the occupant of millions of wasted gigabytes of data that have to be managed and backed up daily.

Most email conversations result in a series of messages with some new information being added in every response. The challenge is to see the valuable information while not being exposed to all the content that you've seen in previous contributions. Exchange 2010 (including OWA) and Outlook 2010 combine in a solution called conversation views. Previous versions of Outlook allow you to click the subject heading to build a primitive form of conversation views in that all of the messages that share a common subject are grouped together. Microsoft also developed some customized Outlook code in the past (used mostly internally within Microsoft) to implement better forms of conversation views, but this code never showed up in any released product.

The new solution compresses conversations into a view where the unique content from each message is shown in the reading pane as the message is selected in the conversation. To do this, new algorithms are used to detect and suppress redundant content from the view displayed to the user. Exchange uses some message properties to decide which messages are actually part of the same conversation. The message properties include the following:

- InternetMessageId
- In-Reply-To
- References
- Subject Prefix
- Normalized Subject

Exchange also maintains a new set of conversation-specific properties to track the items in a conversation. These properties are as follows:

- Conversation Topic
- Conversation Index
- Conversation Index Tracking
- Conversation Identity

For example, if you reply to a message, Exchange knows that the original message and the reply are linked and part of the same conversation and will note this fact, which means that the entire conversation can be viewed as a whole for as long as the items exist in any folder in the mailbox. Figure 10-1 illustrates a conversation that spans five items, one of which is in the Deleted Items folder (and indicated with strikethrough).

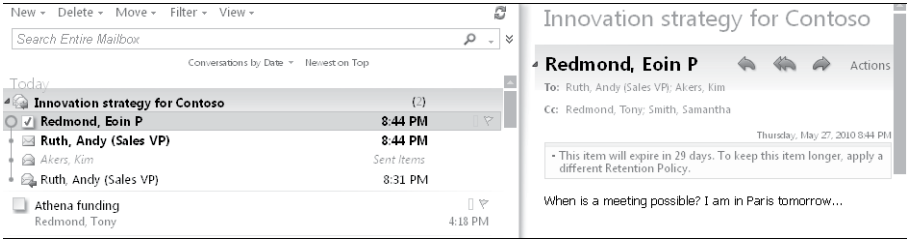


Figure 10-1 A threaded conversation shown in OWA.

INSIDE OUT

Conversation views on various platforms

The implementation of conversation views differs slightly across platforms to take account of the different storage and user interface capabilities of each platform. For example, Outlook 2010 is capable of including items that are stored in PSTs in conversations and can therefore do a good job of creating conversation views when connected to servers other than Exchange, including Gmail and Hotmail; OWA can only include items that are available within a mailbox (including the archive mailbox); and Windows Mobile can only show items from the currently selected folder. User interface differences across different client platforms are an inevitable fact of computer life. This is one reason you should be sure to check new platforms to detect possible opportunities for user confusion before introducing a new platform into production.

Older items that have been in mailboxes for a long time or those that originate in non-Exchange systems present a challenge for conversation views because all of their attributes might not be populated in the same way as are newly created items. To get around the problem, Exchange determines the items that form a conversation by using message subjects.

CAUTION

Sometimes this technique—associating conversations by using message subjects—can associate items that are not part of the same conversation together because they share the same subject. This is one reason to be careful with the Ignore feature in Outlook, which moves a selected item and all other current and future items in the same conversation to the Deleted Items folder.

Exchange attempts to avoid problems caused by older messages that share the same subject with a more recent message by only adding items to conversations if they are within 72 hours of each other, on the basis that the accuracy of including items in conversations based on message subject degrades over time. If you change the subject on a message before you send a reply, you effectively create a new conversation. There is one exception to this rule, and that's when a user starts off a conversation with a message with a blank subject and someone subsequently updates the subject.

Outlook 2010 includes additional code to allow it to implement conversation views when connected to older Exchange servers and non-Exchange servers. This code works, but because it has to function without any help from the server, it is slower and less accurate in terms of linking items together in a conversation than when Exchange 2010 and Outlook 2010 work together. In addition, the client has to process all actions for a conversation. For example, you can decide to "ignore" a conversation, which means that Outlook will automatically move any messages in that conversation into the Deleted Items folder. If Outlook 2010 is connected to a legacy Exchange server, it has to process new items as they arrive to decide whether they belong to the conversation that you've just ignored. Outlook then has to suppress the items that it determines to be in the conversation.

To improve efficiency of conversation processing, Exchange 2010 stores details of actions to apply to conversations in a hidden folder called Conversation Action Settings. The advantage of this approach is that the data are available to the server rather than being limited to Outlook. Thus, if you decide that you want to ignore a conversation, Exchange processes new items as they arrive into the mailbox and all clients see the same effect.

Because some limitations exist in applying conversation views to folders hosted on non-Exchange 2010 servers, Outlook 2010 allows you to suppress conversation views for

selected or all folders. From the View menu, select the Show As Conversations check box. Outlook then asks whether to suppress conversations for just the currently selected folder or for all folders. Later on, after you move the mailbox to an Exchange 2010 server, you can reverse the operation and re-enable conversations (Figure 10-2).

Figure 10-2 Enabling conversations in Outlook 2010.

Outlook doesn't provide an option to allow users to become engaged in a conversation after they ignore it, so if you make a mistake and decide that an ignored conversation really is important, the only way to reverse the action is to select an item from the conversation in the Deleted Items folder, open it, and then click the highlighted Ignore button. Outlook then prompts you to verify that you really want to stop ignoring the conversation (Figure 10-3). If you click Stop Ignoring Conversation, all of the items for the conversation will be moved back into the Inbox.

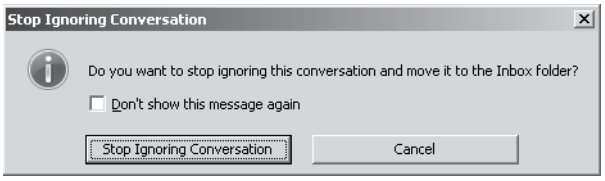


Figure 10-3 Option to stop ignoring a conversation.

INSIDE OUT

Control conversation size in Outlook 2010

You can expand and collapse the view to see the hidden content, but most of the time you'll want to see just what's new in the conversation. Outlook 2010 also includes a useful Clean Up Conversation option that removes redundant items from an email thread. These are great ways to keep a growing mailbox under control.

The user options available in Outlook 2010 and OWA to process conversations are shown in Figure 10-4. OWA doesn't support the ability to clean up a conversation, so its options are limited to how conversations are displayed in the user interface. Outlook 2010 allows you to control what messages are cleaned up from conversations and where the removed

messages are placed. It also supports a more sophisticated set of conversation views than OWA.

Of course, the notion of conversation views is not new and you might wonder why it has taken Microsoft more than 12 years of development to implement a feature that simply makes good sense, but at least it's available now.

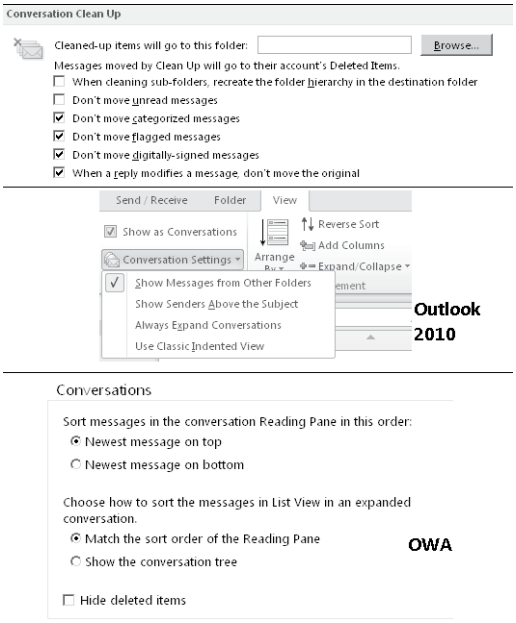


Figure 10-4 Outlook (top) and OWA (bottom) options to control conversation views.

Conflict resolution

Synchronization of items between clients and servers sometimes results in different versions of items in one place or the other. For example, a user works with Outlook offline and modifies an item in the cached version of her mailbox. Later on, she connects to Exchange online using OWA, reads the same item again, and takes an action to modify it, such as clearing an event flag. She then connects Outlook to Exchange and synchronizes, and a conflict results because different versions of the same item now exist. This is a somewhat convoluted example, but given the many ways that people interact with mailboxes through Outlook, OWA, BlackBerries, and Windows Mobile devices, such conflicts occur all the time, especially with calendar items.

All Outlook clients since Outlook 2003 use a conflict resolution engine designed to resolve conflicts automatically. The engine quickly resolves spurious conflicts (two versions exist, but they are identical) and presents a more elegant interface for users to track and resolve

conflicts that require user intervention. In these cases, the engine uses an algorithm to determine which copy of the item is most likely to be the version to keep, and Outlook retains this version in the original folder. Outlook moves the other versions into the Conflicts folder, which is a subfolder of the Sync Issues folder in the user's mailbox.

Note

You have to click the Folder List shortcut to have Outlook reveal the Sync Issues folder in the folder tree view.

You can review the contents of the Sync Issues folder from time to time to see what items Outlook has moved there. There are three subfolders: Conflicts, Local Failures, and Server Failures. Outlook uses the Conflicts folder to store all the items that Outlook believes to be in conflict. You can review these items individually and decide whether you want to keep the version in the Conflicts folder by replacing the version that Outlook has retained in the original folder, or you can delete the version in the Conflicts folder, which is an emphatic way of resolving the conflict. Alternatively, you can leave conflicts alone unless Outlook prompts you to resolve a conflict when you access an item. At this time, Outlook displays a conflict resolution band in the message header to present the options that a user can take. You can decide what version to keep and Outlook will update the folder with this version.

Local Failures and Server Failures folders

The Local Failures folder holds copies of items that Outlook was unable to synchronize with the server. Usually the problem that caused the failure is transient (such as a network interruption) and Outlook subsequently synchronized successfully. The Server Failures folder holds items that Exchange was unable to synchronize down to Outlook, and again, the failure condition is usually transient. You can delete the items that you find in the Local Failures and Server Failures folders—I do this on a regular basis to free up a small amount of space in my mailbox.

It is not a sign of good synchronization health if you find more than a few items in the Local Failures and Server Failures folders, and it could be an indication of an underlying problem that you need to address. If such a condition occurs, you can turn on email logging by selecting Tools, Options, Other, and then Advanced to force Outlook to begin logging more detailed results for synchronization operations, which it stores as items in the Sync Issues folder. Note that you have to restart Outlook before detailed logging begins. Some of the information that Outlook logs could help you understand why problems occur, but it is more likely that a Microsoft Customer Service and Support (CSS) specialist will be able to decipher the data, because the information is useful but a tad cryptic.

Listing client connections

You can see details of the clients that are currently connected to a mailbox server with this command:

```
Get-MailboxServer –Identity ServerName | Get-LogonStatistics | Format-List UserName,
ApplicationId, ClientVersion
```

The output will include information like this:

```
UserName          : SystemMailbox{dc877527-83e9-4c13-a50c-b4beda917ce3}


ApplicationId     : Client=EventBased
MSExchangeMailboxAssistants;Action=CalendarNotificationAssistant
ClientVersion     : 3585.0.32903.3

UserName          : Redmond, Eoin P.
ApplicationId     : Client=OWA
ClientVersion     : 3585.0.32903.3
UserName          : Redmond, Eoin P.
ApplicationId     : Client=WebServices;UserAgent=[NoUserAgent]
ClientVersion     : 3585.0.32903.3

UserName          : Clark, Molly (IT)
ApplicationId     : Client=MSExchangeRPC
ClientVersion     : 3585.0.32903.3
```

Some of the connections reported here will be internal server-side connections created by Exchange (for example, connections from the Store Driver to deliver messages to a mailbox), but it is easy to identify those which belong to real users. You'll see some interesting things:

- Outlook users are shown with an application identifier of *MSExchangeRPC*, which is the value used to identify connections handled by the RPC Client Access layer. Each Outlook client that is configured in cached Exchange mode generates at least five connections. Four of these are used by the threads that perform "drizzle-mode" background synchronization to maintain the folder replicas in the client OST; the other is used by the thread responsible for sending messages.

- OWA users have two types of connections. The connections with an application identifier of *OWA* are used to maintain connectivity with the server to perform tasks such as listing the contents of a folder or reading a message. The connections shown as *WebServices* are generated to send messages.

- Background Exchange tasks are identified with the mailbox assistant name. In the preceding example, the connection is for the calendar assistant that notifies users when calendar appointments are due.
- Connections created by the Store Driver to deliver messages from the transport system to mailboxes in a database show the database name as the user name.
- For Exchange 2010 SP1, the client version is the same for all connections. This can be regarded as a bug because it's obvious that client software comes in many different versions. Microsoft is aware of the issue and might address it in a future release.

Blocking client connections to a mailbox

Exchange allows you to disable any or all client connection protocols, including MAPI, on a per-user basis by amending the protocols that the mailbox can access on the Mailbox Features tab (Figure 10-5). If you disable MAPI, a user cannot use Outlook to connect to his mailbox. This option is often taken for mailboxes that use OWA exclusively and never need to use Outlook.

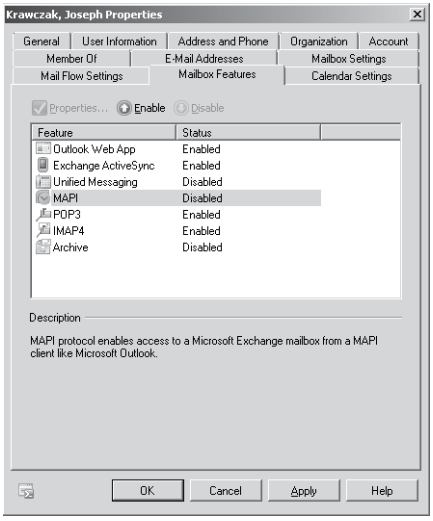


Figure 10-5 Disabling MAPI access for a user.

The Set-CASMailbox cmdlet supports a number of parameters to control how an individual mailbox can use MAPI to connect to a mailbox on an Exchange server:

- *–MAPIBlockOutlookRpcHTTP*: Allows you to determine whether you allow Outlook clients to connect over RPC over HTTP via Outlook Anywhere. Set the parameter to $True to block RPC over HTTP access and $False to allow access.

- *–MAPIBlockOutlookVersions*: Allows you to control what versions of Outlook can connect to Exchange. You might use this setting to force users to upgrade to a more modern version of Outlook such as Outlook 2007 because these clients are able to make efficient use of server resources. If a user attempts to use a blocked version of Outlook, the user will see the error message shown in Figure 10-6. Outlook clients configured for cached Exchange mode continue to work offline, but they cannot connect to the server until an administrator lifts the block.

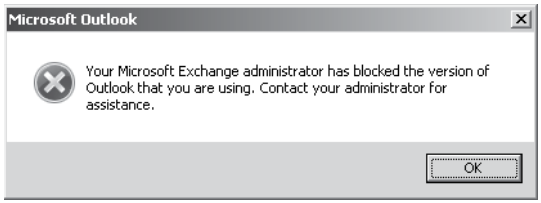


Figure 10-6 A user discovers that he can't use this version of Outlook to connect to Exchange.

- *–MAPIBlockOutlookNonCachedMode*: Allows you to determine whether you allow Outlook clients to connect in online mode to the server. Set this parameter to $True to allow online access and to $False to force clients to connect in cached Exchange mode. Somewhat confusingly, users blocked from online access also see the same message shown in Figure 10-6, followed by another error message to tell them that Outlook is unable to open their default email folders. Pointing to the version of Outlook rather than the need to use cached Exchange mode might confuse the help desk when users report their problem.

Microsoft identifies Outlook builds using a scheme of *major release, minor release, build number.* The *major release* number is shared across all the Office applications. The Office 12 suite includes Outlook 2007, Office 14 includes Outlook 2010, and so on (Microsoft did not produce an Office 13 suite). The *minor release* indicates whether the build is in the original RTM build, a service pack, or a cumulative update, and the *build number* is incremented daily to include code and fixes checked in by engineers. Here are the build numbers for some recent Outlook versions:

- Outlook 2007: 12.4518.1014
- Outlook 2007 SP1: 12.6425.1000
- Outlook 2010: 14.0.4760.1000

To discover the client version to specify in the *–MAPIBlockOutlookVersions* parameter, you can use the Help/About option with Outlook or check with the useful list of client versions that Microsoft maintains at *http://technet.microsoft.com/en-us/library/aa996848.aspx*.

As an example, here are two commands. The first restricts access so that the user must use Outlook 2003 or newer; the second restricts access to Outlook 2007 or later. In both cases, an explicit "allow" is included for version 6.0.0 to support Exchange server-side MAPI connections (server connections always use MAPI version 6.0).

```
Set-CASMailbox –Identity'Akers, Kim' –MAPIBlockOutlookVersions
'-6.0.0;10.0.0- 11.5603.0'
Set-CASMailbox –Identity'Akers, Kim' –MAPIBlockOutlookVersions
'-6.0.0;10.0.0-12.4406.0'
```

You have to wait up to 120 minutes for the cached information about the mailbox to expire from the Store's cache. Alternatively, you can restart the Information Store service, but apart from test situations, this is definitely not the best approach because it will affect all the other mailboxes connected to the server.

You can check to see if any restrictions are in place for any protocols on a server by using the Get-Mailbox cmdlet to examine the *ProtocolSettings* property of each mailbox. If a restriction is in place for a specific client version, you will see that version number listed. If an administrator has completely disabled MAPI access for the mailbox, you will see "MAPI" and no version number. For example:

```
Get-Mailbox –Server ExchServer1 | Where {$_.ProtocolSettings –ne $Null} | Select
Name, ProtocolSettings
```

```
Name                          ProtocolSettings
--------                      ----------------
Redmond, Tony                 {MAPI§§§§-6.0.0;10.0.0-11.5603.0§§§§}
Ruth, Andy                    {MAPI§0§§§§§§§}
Smith, John                   {OWA§1, IMAP4§0§§§§§§§§, POP3§0§§§§§...
```

You can also use the Get-CASMailbox cmdlet to check for MAPI blocks. On the one hand, Get-CASMailbox is more interesting because it also allows you to return the value of the *MAPIEnabled* property (this will be False if the user is completely blocked from using MAPI) and to see the details of all of the protocol settings that you can set on a mailbox. On the other hand, you cannot specify a server name to check against, so Get-CASMailbox will be less efficient, because it will scan the entire organization unless you restrict its scope by using a server-side filter to focus in on one server:

```
Get-CASMailbox –Filter {ServerName –eq'ExchServer1'} | Where {$_.ProtocolSettings
–ne $Null} | Select Name, ProtocolSettings, MapiEnabled
```

```
Name                ProtocolSettings                MAPIEnabled
--------            ----------------------          -----------------
Redmond, Tony       {MAPI§§§§-6.0.0;10.0.0-...      True
Ruth, Andy          {MAPI§0§§§§§§§}                 False
Smith, John         {OWA§1, IMAP4§0§§§§§§§§...      True
```

In addition to imposing blocks on MAPI connections, you can use the Set-CASMailbox cmdlet to disable client access to other protocols. For example:

- To disable access to POP3: Set-CASMailbox –Identity Bond –PopEnabled $False
- To disable access to IMAP4: Set-CASMailbox –Identity Bond –ImapEnabled $False
- To disable access to Outlook Web Access: Set-CASMailbox –Identity Bond –OWAEnabled $False
- To disable access to ActiveSync: Set-CASMailbox –Identity Bond –ActiveSyncEnabled $False

Blocking client access to a mailbox server

Implementing blocks on a mailbox basis is useful, but sometimes you want to block all access to a mailbox server. For example, you might want to update the server with some software or apply a patch without having users impose load on the server or potentially interfere with the upgrade. You could apply such a block with EMS by searching for all mailboxes hosted in active databases on the server and using the Set-CASMailbox cmdlet to disable MAPI access, but it is more convenient to be able to apply the block centrally. For all versions from Exchange 2000 to Exchange 2007, you can block MAPI clients from connecting to a mailbox server by configuring the Disable MAPI Clients key in the registry. This key is intended to allow administrators to require the deployment of a base-level version of Outlook. Put another way, it stops users from attempting to connect with earlier versions that might not meet your company's security requirements because the earlier software doesn't include recent anti-spam and antivirus features such as beacon blocking.

The registry key is set on a mailbox server so that it can be effective only if the mailbox server is responsible for handling client connections, which is the case from Exchange 2000 to Exchange 2007. The RPC Client Access layer running on CAS servers handles MAPI client connections for Exchange 2010, so the old registry key method doesn't work. In fact, because a CAS server can handle MAPI connections for multiple mailbox servers, no one-step mechanism exists in Exchange 2010 to block MAPI connections to a designated mailbox server. Two approaches can be taken if you need to block connections to a mailbox server.

1. Use the Set-RPCClientAccess cmdlet. This cmdlet allows you to block all MAPI connections coming from specific versions. For example, this command blocks access to any version of Outlook prior to Outlook 2007 (major release 12).

```
Set-RPCClientAccess -Server ExCAS01 -BlockedClientVersions
"0.0.0-5.65535.65535; 7.0.0-11.99999.99999"
```

The problem is that all connections to all mailbox servers supported by the CAS server will be blocked. This might be an effective method to use in small sites where you have just one CAS server and one mailbox server.

2. In larger sites that support multiple CAS and mailbox servers, you can set a per-mailbox block with the Set-CASMailbox cmdlet for every mailbox on the server that you want to maintain. For example:

```
Get-Mailbox –Server ExServer1 | Set-CASMailbox –MAPIBlockOutlookVersions
'-6.0.0;10.0.0-12.4406.0'
```

The RPC Client Access layer verifies whether a client can connect using MAPI by checking the server and mailbox blocks set by the Set-RPCClientAccess and Set-CASMailbox cmdlets before it allows a connection to pass from the CAS server to the mailbox server. Either mechanism is equally effective as a block. The choice between the two therefore comes down to whether you can block all connections flowing through a CAS server, no matter what mailbox server they are destined for, or you need to block connections to just one specific mailbox server.

Of course, if you run mailbox servers in a Database Availability Group (DAG), you can use the StartDAGServerMaintenance.ps1 script (see Chapter 8, "Exchange's Search for High Availability") to move all the active mailboxes off a server and block further activation. This step effectively prepares a mailbox server for maintenance operations.

Outlook Web App

After a shaky start when the browser interface could only be politely called clunky and slow, Microsoft has poured development effort into OWA with an eye to creating a browser client that is broadly equivalent in feature set and functionality to Outlook. "Broadly equivalent" is important because Outlook has many advantages due to its key position in the Office suite and the resources dedicated to its development since 1996. At the same time, other browser clients advance the state of the science and give Microsoft a challenge for each new release. Gmail moved away from the classic folder-centric user interface paradigm into a world of conversations; Zimbra and other Web-based email applications have developed snappy, well-built interfaces that are a pleasure to use. Even Microsoft's own free Hotmail browser interface has improved substantially in the last few years.

Even with the best intentions in the world, there are two basic reasons why OWA will never completely match Outlook.

- Microsoft is unlikely to stop development of Outlook, so there will always be a feature race where OWA will have to keep up with Outlook. Outlook 2010 sets a new bar for OWA 2010 to be judged against and there are some places where OWA misses a feature. For example, you can't create a reply and have OWA automatically insert the

text of the original message as you can with Outlook. Outlook supports an expanded set of the sort options available to OWA (such as sort by day). OWA doesn't have the same abilities to manage conversations that Outlook 2010 introduces, nor can it undo actions such as deletes or moves in the same way that Outlook can. Finally, there are some browser limitations that OWA has to cope with, such as the inability to save more than one attachment to the file system in a single operation. You can save multiple attachments, but you have to do them individually.

- Outlook's ability to work offline is unlikely to be matched by OWA unless Microsoft implements something similar to Google Gears (which allows Gmail and other Google applications to work offline). Interestingly, Google announced in February 2010 that they had stopped the development of Gears to focus on moving the Gears capabilities to an HTML 5–based solution. Microsoft could certainly build on the work done to provide a similar capability based on HTML 5 in Internet Explorer to add the ability to work offline. Of course, such a solution might not be appreciated in the Outlook development group, because it would remove one of the prime differentiators between the two clients.

Another factor that has to be taken into consideration is that OWA and Outlook are developed by different engineering groups that are under different market pressures. OWA is a client of Exchange and exclusively serves Exchange, so in that respect OWA will always be closer to Exchange. Outlook is regarded as the premier client for Exchange but only because it exposes most functionality. Outlook is part of the Microsoft Office suite and has to be a highly functional client for other email servers, including Hotmail (using an excellent Hotmail connector) and Gmail and other servers through IMAP4. Outlook cannot afford to ignore Exchange, but it cannot afford to be too focused on Exchange, either.

At the end of the day, if you compare the two clients on a feature-by-feature basis, OWA doesn't completely match Outlook, but it does a very good job of getting close. The question is whether the missing features are important to users and whether those missing features stop users from being productive. The answer is that OWA 2010 is highly usable and will meet the needs of the majority of those who use it. The SP1 version improves the responsiveness of OWA by caching and using data more intelligently to eliminate the slight sluggishness that some users experienced with the RTM version. Even so, there are some points that administrators need to know, and that's what we cover in this section.

A refresh for OWA provided by Exchange 2010 SP1

Although there is no doubt that the OWA application in Exchange 2010 is an improvement over its Exchange 2007 predecessor, Microsoft took the opportunity to smooth some rough edges and add some additional functionality in SP1. Part of this work was driven by user feedback received after Exchange 2010 moved into production; part came about simply because the developers had extra time to complete features for inclusion in SP1.

Figure 10-7 illustrates the major features of the new OWA interface in Exchange 2010 SP1. A number of visual differences are immediately apparent.

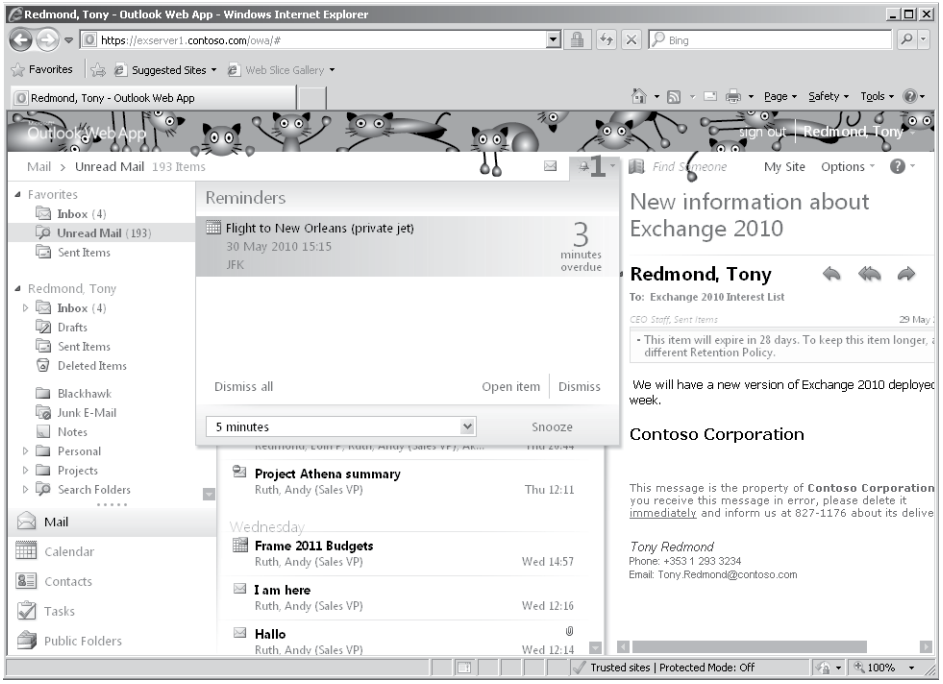


Figure 10-7 The new Outlook Web App user interface introduced in Exchange 2010 SP1.

- User-selectable themes have been reintroduced. My personal favorite (as illustrated in Figure 10-7) is the Herding Cats theme. The themes packaged with Exchange 2010 SP1 cover a reasonable variety of possible choices from the coolness of the Arctic theme to the splurge of colors in the Fingerpaints theme. I am less sure about the Cupcake theme. Microsoft also supports customers who want to customize OWA to apply corporate branding, colors, and icons.

- The typefaces used by OWA are larger to make information clearer and more accessible. The currently selected item is more obvious and now boasts a checkmark to indicate its status.

- Navigation is improved through what the Microsoft UI designers believe is a better use of screen real estate that reveals more content. Further improvement comes through the introduction of a breadcrumb trail to show users where they are and how they got there, plus redesigned icons for major options such as Reply (a change provoked because testing revealed that users made mistakes when they replied to

messages). Some unnecessary elements (like the Exchange icon at the top of the mailbox) are removed to declutter the interface.

- Pop-ups to notify users of situations such as an impending meeting are more obvious. Larger typefaces help to convey essential information—in this case, that the meeting is 3 minutes overdue.
- It's also easier to get to common options such as change password without having to navigate from OWA to ECP and back again (Figure 10-8).

Note

Although at first glance the changes to OWA's screen design might seem cosmetic, they have a real impact on PCs that have small screens, such as netbooks.

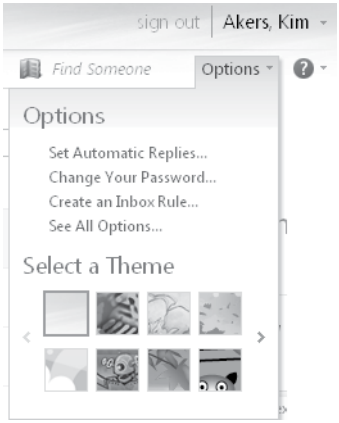


Figure 10-8 Enabling easier access to common user options.

Behind the scenes, Microsoft tweaked OWA performance to make the processing of large folders faster. Exchange 2010 moved to use "endless" views to better support folders that hold thousands of items by prefetching items as you scroll. This simulates the scrolling behavior of Outlook, where you get all messages in one continuous list, instead of the paged view used in OWA 2007. Navigation through views is now faster, especially when using conversation views. In SP1, the attention to performance focused on prefetching content to update views. In all previous versions of OWA, operations such as deleting items or marking an item as read are performed synchronously, and OWA waits for the server to confirm the operation as complete before it updates the view. In SP1, actions occur asynchronously and appear far more quickly because OWA updates the view without waiting for a response from the server.

> **Note**
>
> **There's an obvious danger here that a server problem could invalidate a client action, but this doesn't happen in the vast majority of cases, so it's a worthwhile shortcut to provide better user perception of performance.**

A further improvement is made in the way that OWA uploads attachments to make sure that an upload of even a very large file (more than 5 MB) doesn't block other actions. Notifications also receive a makeover to make them less obtrusive by using in-line notifications rather than modal pop-ups.

Apart from user-selectable themes, other features that make a reappearance in SP1 include OWA Web parts, the ability to print several different views of the calendar (a somewhat strange omission in Exchange 2010), and the option to use a reading pane at the bottom of the screen rather than at the side. Web-ready document viewing now supports documents protected with Active Directory Rights Management Services (AD RMS) on Internet Explorer, Firefox, and Chrome on Windows and Safari on a Mac. Many minor bugs in specific browsers—such as the inability to drag and drop between folders using Chrome—are fixed, although some restrictions remain, such as the need to load the optional Secure Multipurpose Internet Mail Extensions (S/MIME) control if you want to be able to drag and drop attachments into a message. Because the S/MIME control is only supported by Internet Explorer, it follows that other browsers can't perform this trick.

Overall, the Microsoft developers aimed to create an interface that is both rich and simple with the major features exposed and easily accessible. The changes made in SP1 are not a radical overhaul of the basic framework established for OWA in Exchange 2010. Instead, they are more like a tune-up to reveal the true potential of the application.

OWA functionality deprecated in Exchange 2010

OWA 2010 boasts a shiny new interface and includes many new features that we review in due course. On the downside, Microsoft has deprecated a number of features that appeared in previous versions because they didn't have the engineering time to upgrade the features to work with Exchange 2010, they felt that the feature wasn't used in the way or as much as they anticipated when they did the work, or for some other reason, including security concerns. Among the best examples of deprecated features are the following:

- Web parts: Exchange 2003 and Exchange 2007 allow you to specify a Web part in a URL to access that Web part directly. For example, you can go directly to a specific view of a folder in a user's mailbox, such as opening a user calendar in the week view. This feature could reappear in a service pack for Exchange 2010.

See *http://msexchangeteam.com/archive/2006/10/26/429362.aspx* for full details about Web parts in Exchange 2007.

- Document access: This feature was introduced in Exchange 2007 and allows OWA users to access documents in a Microsoft SharePoint site and to file shares pointed to through universal naming conventions (UNCs).

If a now-deprecated OWA feature is important to your deployment, you should work with your Microsoft account team to provide this feedback to the Exchange development team so that it can be taken into account when Microsoft draws up the feature list and work commitment for a future service pack or version of Exchange. The development team does listen, as is evident in the list of features that reappear in SP1.

Different browsers, different experiences

OWA is available in two versions: Premium and Light. The CAS server makes a decision about what version of OWA to provide to a client based on the value of the user agent string submitted by the browser when it connects to Exchange. The official Microsoft stance is that you must use Internet Explorer 7 (or later) or Firefox 3.0 (or later) on a supported operating system if you want to use the premium version of OWA with Exchange 2010. Earlier versions of Internet Explorer and Firefox cannot use the premium version and are automatically downgraded to the light version. Along the same lines, you can use OWA Premium with the Safari browser, but only when it is version 3 or later running on a system using the Leopard (10.5) or Snow Leopard (10.6) versions of the Mac X operating system. Other versions of Safari, such as those running on Windows, the iPad, or the iPhone, are only capable of supporting the light version of OWA.

It might seem strange that a browser is capable of supporting the premium version of OWA on one platform and not another, but a mixture of subtle and not-so-subtle rendering differences exist across platforms. Microsoft has to decide where to invest engineering resources to develop, test, and support a specific browser configuration, especially in situations where they depend on engineering groups in other companies to fix bugs and help address issues reported by customers. If Microsoft doesn't see sufficient customer demand to warrant the necessary investment to support the initial engineering for making a browser work well with OWA Premium, the testing to validate the engineering work, and the long-term sustaining support to fix any problems reported by customers and keep pace with new software releases from the browser vendor, they just don't do it. Each browser has its own unique challenges. Safari on iPad, for instance, is the first full-screen browser running on a tablet device that depends exclusively on a virtual keyboard for input (you can, of course, connect a hardware keyboard to an iPad).

The browser/operating system combinations listed in the Tier 1 and Tier 2 categories in Table 10-2 can run the premium version of OWA. For example, I commonly use Google's

Chrome browser with OWA (Figure 10-9). This screen shot used Chrome version 4.0.249.78 connected to Exchange 2010 SP1 complete with my favorite Herding Cats theme. Despite the fact that Microsoft doesn't test Chrome with OWA as thoroughly as it tests Internet Explorer and that Google's iterative development philosophy for their programs generates regular updates for Chrome, everything works pretty smoothly.

Table 10-2 Microsoft support for browser/operating system combinations

Microsoft Support Level	Browsers and Platform
Tier 1: Fully tested and supported by the product group	Internet Explorer 7+: Windows XP, Vista, Windows 7, Windows 2003, Windows 2008 Firefox 3+: Windows XP, Vista, Windows 7, Windows 2003, Windows 2008, Mac OS 10.5 and above Safari 3.1+: Mac OS 10.5 and above
Tier 2: Supported but with limited testing	Firefox 3.0 on Linux platforms Chrome on Windows Vista and Windows 7
Tier 3: Supported but only with OWA Light	Internet Explorer 6 Safari on Windows Other browser/operating system combinations (Opera, Safari for iPad, and so on)

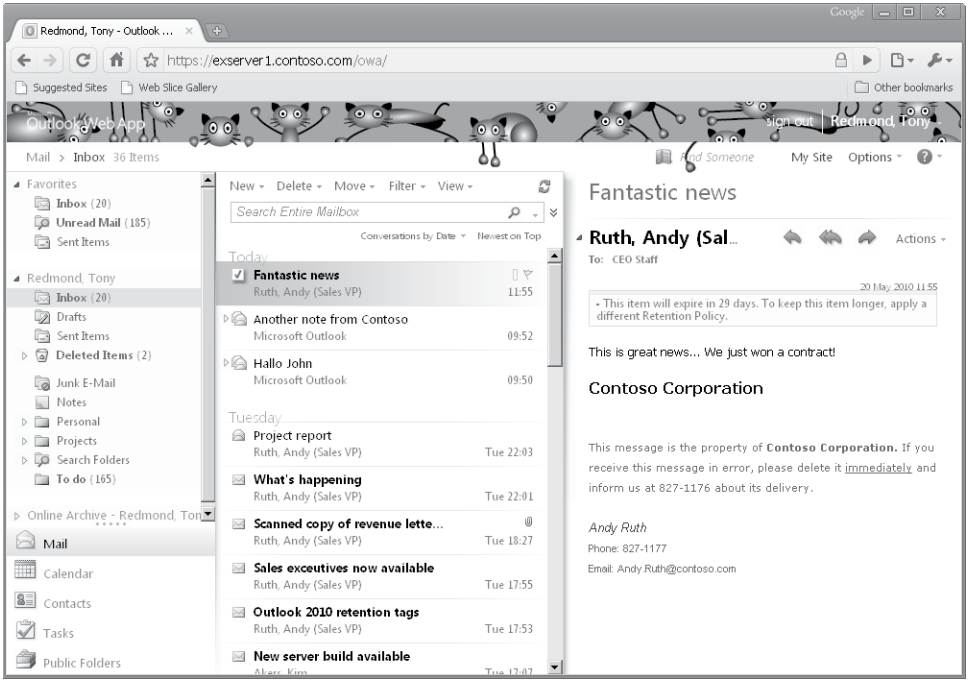


Figure 10-9 Using the premium version of OWA with the Chrome browser.

INSIDE OUT A couple of OWA features that don't work with Chrome

The only consistent problem that I have discovered with Chrome is that I cannot drag and drop items from one folder to another. Chrome stubbornly refuses to perform this operation, but it will allow you to drag and drop an item into a draft email to add it as an attachment. By comparison, Internet Explorer moves items between folders quite elegantly. Sometimes Chrome fails to signal notifications of new messages or upcoming appointments, but most of the time these work. Another small irritation is that the availability information for users isn't displayed when you browse the Global Address List (GAL) with Chrome where it is with Internet Explorer (Figure 10-10). All of this goes to prove what "supported but with limited testing" means in practice.

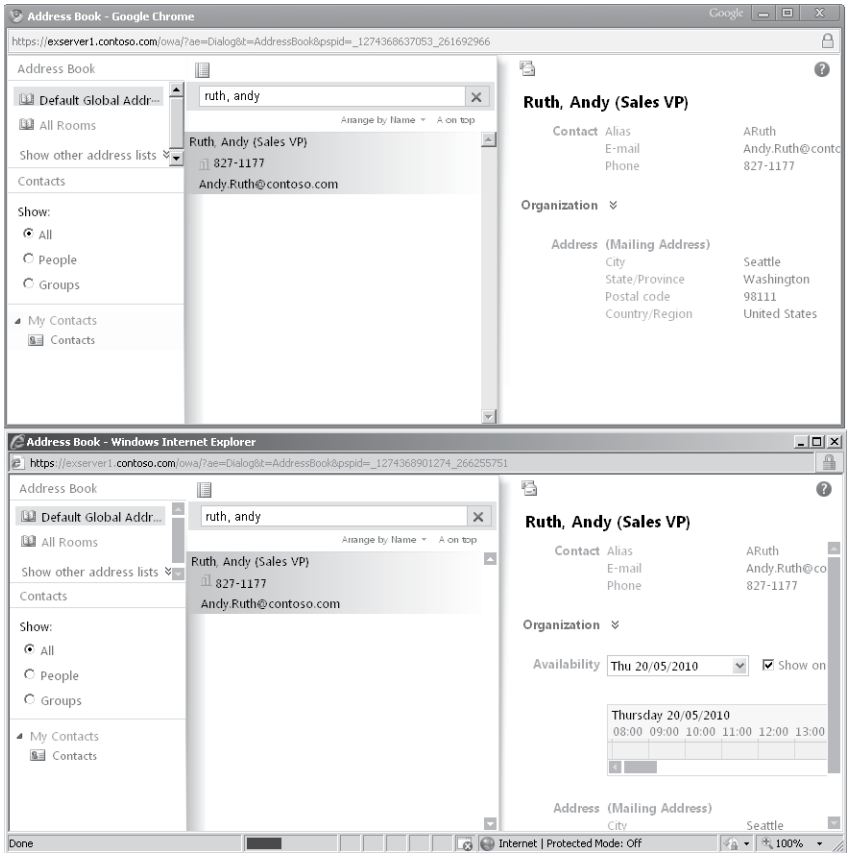


Figure 10-10 Spot the difference! Chrome and Internet Explorer display details of a user from the GAL.

OWA Light

OWA Light is designed to support many different browsers—from those that Microsoft doesn't test (like Opera) to earlier versions of those that they do test (like Firefox)—running on anything from Linux workstations to laptops. It is also designed to accommodate a wide range of screens, from netbooks with relatively low resolution to high-end workstations. Because of the range of capabilities found across different browsers and different versions of browsers, Microsoft limited the amount of "intelligence" in the form of code such as JavaScript that OWA Light runs on the browser. This ensures that OWA Light has a very good chance of working on any browser that it meets, but it does impose some limitations. For example, you might notice that column widths do not dynamically resize to match screen resolution; no matter what size screen you use, the column for message sender is always sized at 16 characters (see Figure 10-11). OWA Premium includes logic to detect the screen resolution and window width and resizes columns to display more or less information based on the current configuration.

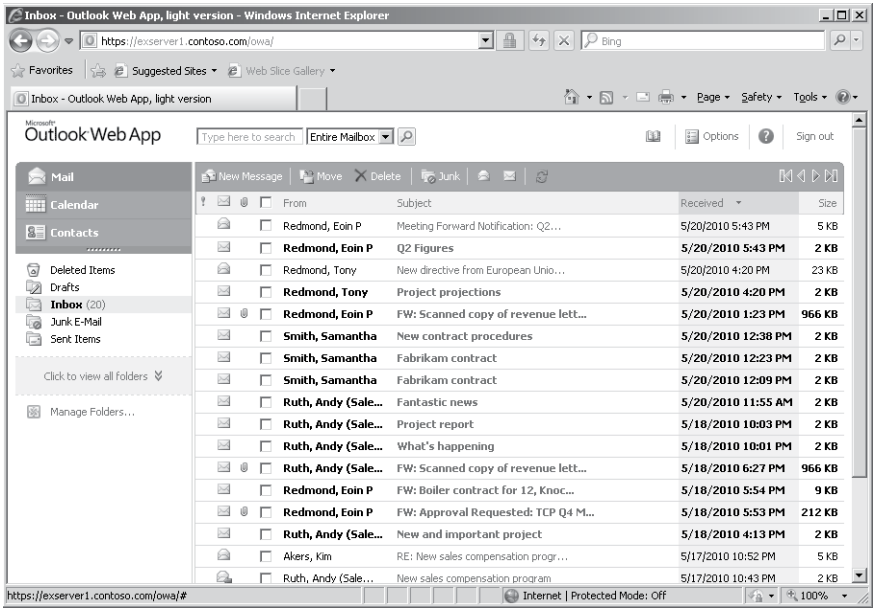


Figure 10-11 Using the OWA Light version.

Spell checking is another example of feature differences between browsers. OWA only supports the integrated spell checker for Internet Explorer. The logic here is that Firefox, Safari, and other browsers offer their own spell-checking capability and it is difficult to engineer

the same kind of spell check feature that Microsoft can include in Internet Explorer when they have full control over the interface. It's also fair to say that Internet Explorer doesn't come with an integrated spell checker, so Microsoft had to include one for OWA to achieve parity with Outlook. Taking all factors into account, Microsoft felt that users might be confused if they had to make a choice between the browser's own spell checker and one that they provided for OWA, so if a non–Internet Explorer browser is detected, OWA automatically disables its own spell checker, including the option for users to decide to spell check messages before sending.

INSIDE OUT **Multiple sessions**

Modern browsers support tabbed interfaces to allow users to move quickly between different Web sites. On the surface, you'd expect to be able to use the same facility to establish multiple connections to Exchange and be able to open multiple mailboxes with OWA. Unfortunately, this isn't possible because browsers share connections within the same process. OWA requires a relatively large number of connections for mailbox and directory information, so if you opened multiple mailboxes, the browser wouldn't be able support all of the necessary connections and one or both sessions would fail, specifically in the mechanism used to keep OWA updated with new information as the different sessions would block each other's connections.

The workaround is to use the File/New Session (in Internet Explorer 8) or File /New Window (in Internet Explorer 7) commands to create a brand new session complete with its own set of connections. You won't be able to use tabs to move between the sessions within a single browser instance, but you will be able to use Alt+Tab to navigate between the two windows to move between the two sessions.

OWA configuration file

Exchange stores many configuration settings for OWA in the Active Directory directory service. On the client side, OWA is an ASP.NET application that maintains another group of settings in an application configuration file called Web.config.xml that is located in the \ClientAccess\OWA folder under the Exchange root. These settings affect how OWA runs in a browser. You can edit the OWA configuration file with any text editor as long as you're careful to preserve the XML syntax. Each CAS server has its own OWA configuration file, so you need to apply any changes that you want to make separately on each server. You also need to check the settings after you apply a roll-up update or service pack for Exchange because there is no guarantee that Microsoft will not overwrite the configuration file during an update.

OWA also uses system registry entries to control some settings. In terms of administrator interest, the timeout for a session is probably the most popular of these settings. There are two values: One controls how long OWA will run without terminating a session when a user logs onto a public computer (or clicks the Public check box when they connect to OWA); the other controls the private timeout. The two values are as follows:

Public (15 minutes by default)

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\MSExchangeOWA
Name: PublicTimeout
Type: DWORD

Private (8 hours by default [640 minutes])

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\MSExchangeOWA
Name: PrivateTimeout
Type: DWORD

Missing favorites

Some users love to create Favorites folders, and some leave the default set alone (Inbox, Unread, Outbox). Users in the latter category won't care that OWA 2010 is not 100 percent compatible with any version of Outlook 2010 when it comes to Favorites folders. In effect, this means that you can create a new Favorites folder in Outlook 2007, but it might not turn up in the list of folders displayed by OWA. Likewise, you can create a new Favorites folder in OWA 2010 and not see it in Outlook 2007. Even stranger, sort orders vary between versions so that a set of folders enumerates differently in OWA 2010 than they do in Outlook 2007. The reason is that all versions of Outlook prior to Outlook 2010 save data about favorites locally, whereas Outlook 2010 saves the information in the user's mailbox on the server. For historical reasons, different clients store user data in a variety of places. MAPI profiles are in the system registry, whereas Outlook holds many of its settings in hidden items in the root folder of a user's mailbox as well as files such as the nickname cache (.nk2 file). Because OWA is designed for browsers that can run on many different computers, it normally stores its settings in the root of the user's mailbox.

The advantage of having all clients share a common repository is that they will show the same set of folders sorted in the same order. However, during the period when Exchange 2010 is used with older clients, some users might scratch their heads as they wonder where a Favorites folder has gone. Another difference between OWA and Outlook is that OWA does not display special folders that it doesn't need, whereas Outlook always does. For example, the Outbox folder is never used by OWA, so the premium version never displays it in the folder list (curiously, the light version of OWA does show the Outbox folder). The logic here is that you cannot send deferred mail with OWA in the same

way that you can with Outlook, so there is no need to show the Outbox (which is where Exchange holds deferred messages).

Forwarding meeting requests

Exchange 2007 introduced a feature to inform meeting organizers when an attendee forwarded the meeting request to another recipient. The Calendar Attendant generates these notifications after they process the meeting forward. Figure 10-12 shows a typical example of a meeting forward notification. In this case, probably no great harm to the organization is done when details of a meeting to celebrate someone's birthday are forwarded to another user; the situation might be very different for meetings to discuss sensitive topics such as budgets, corporate reorganizations and restructuring, and plans to introduce new products.

It might be interesting to know that someone else has been informed about a meeting that you've set up, but it can also be irritating to receive a whole batch of notifications, especially when you organize frequent meetings attended by lots of people. You can have Exchange delete the notifications automatically on a per-mailbox basis using the Set-CalendarProcessing cmdlet.

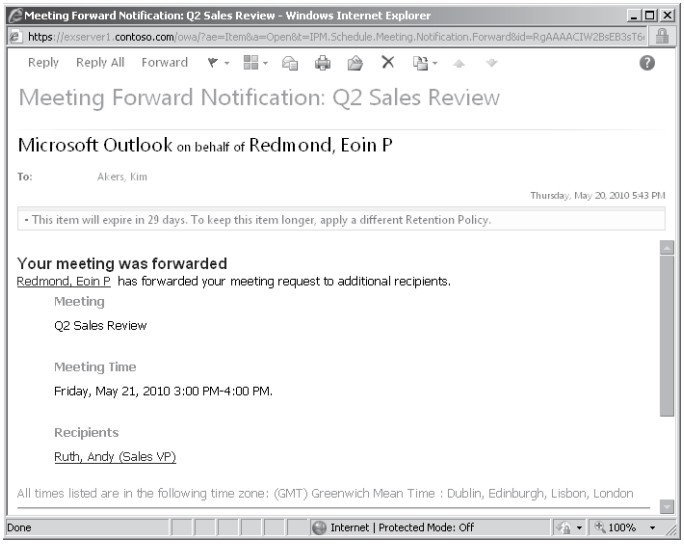


Figure 10-12 How a user knows that his meeting has been forwarded.

OWA also allows users to delete these notifications automatically by selecting the Delete Notifications About Forwarded Meetings check box in the Automatic Processing section of Calendar Options (Figure 10-13), which is an option that is not available in the Outlook UI.

In this example, we force Exchange to move any notification message to the Deleted Items folder for the nominated user:

```
Set-CalendarProcessing –id 'EPR' –RemoveForwardedMeetingNotifications $True
```

It's also possible to do the same thing with Exchange 2007, but you need to use the Set-MailboxCalendarSettings cmdlet, which is deprecated in Exchange 2010. It's also possible to suppress notifications going to external domains after meetings have been forwarded within your organization. This command blocks all meeting forward notifications to every domain. If you just want this to be done for a specific domain, you pass the identifier for that domain.

```
Set-RemoteDomain -MeetingForwardNotificationEnabled $False
```

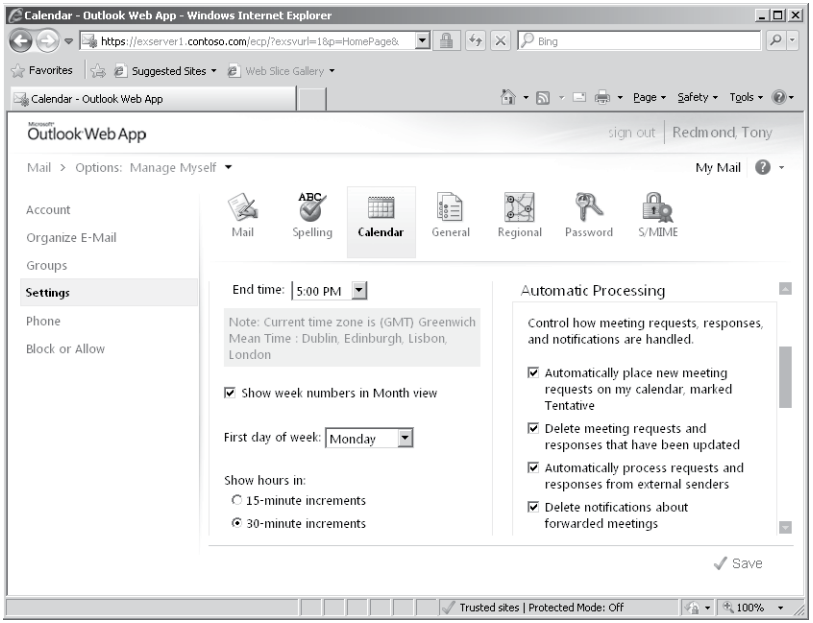


Figure 10-13 Opting to have Exchange automatically delete notifications about forwarded meetings.

OWA Web parts

Web parts refer to the different components used by OWA to assemble its UI. The value of making individual Web parts usable by other browser applications is that you can expose different parts of OWA functionality within those applications. This capability is not supported in Exchange 2010, but is supported with SP1.

The set of Web parts supported by Exchange 2010 and their command format is the same as for Exchange 2007 OWA and is described at *http://technet.microsoft.com/en-us/library/bb232199(EXCHG.80).aspx*. The only difference is that the *f* parameter used in Exchange 2007 to determine the folder to be displayed in the Web part is replaced by *fpath* in Exchange 2010.

Long signatures

If you're migrating from Exchange 2003, you might have configured a value called *SignatureMaxLength* in the system registry to increase the maximum size of signature text that OWA can apply to outgoing messages from the default 4 KB to an upper limit of 16 KB. This ability was often exploited by companies that wanted to append substantial multiparagraph disclaimers to protect themselves.

Exchange 2007 and Exchange 2010 increased the default size of an OWA signature to 8 KB but removed the ability to increase it any further. OWA flags the errors shown in Figure 10-14 if you attempt to input more text than fits into the 8 KB maximum.

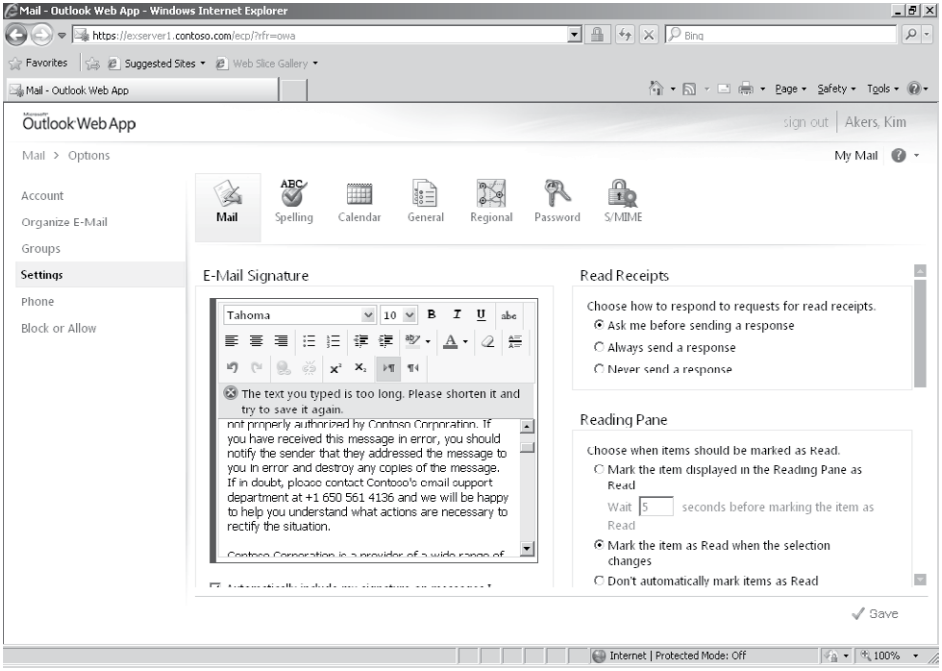


Figure 10-14 Problems adding a very long company disclaimer.

The reason you can't increase the size is that it's much more efficient to apply a company disclaimer to outgoing messages with a transport rule because a transport rule guarantees that a disclaimer will be applied, whereas asking users to configure a disclaimer on

an individual basis is prone to fail. The steps to create a transport rule to apply a company disclaimer are described in Chapter 16, "Rules and Journals." It's best to ask users to remove company disclaimer text from their signatures and restrain themselves to personal information that is not included in the disclaimer that the transport rule applies. Because transport rules can now fetch data such as names, telephone numbers, and email addresses from Active Directory to include in a disclaimer, the text that remains to be included in a personal signature is limited to items such as department-specific text or a personal "thought of the day." We'll describe how to create a transport rule that incorporates Active Directory data in a disclaimer in the section "Creating a corporate disclaimer" in Chapter 16.

Sharing calendars

The requirement to share calendars with co-workers is a common collaborative need. OWA allows you to share your calendar with other users and to add their calendars to your view. Everything works on the basis that a user sends an invitation to share her calendar to those with whom she wants to share. Go to the calendar and click the Share icon and then select Share Calendar. OWA creates a message (the top item in Figure 10-15) to inform recipients that you want to share your calendar.

INSIDE OUT

How much are you willing to share?

An important point here is the degree of sharing that you're willing to do. The options are to show everything, in which case people will see whatever you've entered into your calendar except for items that you mark as private; free and busy information, including subject and location, which means that people see the time slices that are taken up in your calendar together with basic information about what you're doing; and just the free and busy information, which displays just the time slots when you are occupied but provides no indication of whether you're on the golf course or engaged in something more productive. You also have the option of requesting reciprocal access to the other person's calendar.

The bottom item in Figure 10-15 shows an invitation to share someone else's calendar. The text giving instructions to go to Microsoft.com for instructions about how to view shared folders is inserted automatically by Exchange but isn't really necessary, because all the recipient has to do is to click the Add To Calendar icon to have OWA do the work to add the shared calendar.

The calendar sharing functionality in OWA is designed to allow users to view calendars belonging to others, so when you share a calendar with another user, Exchange assigns the

Reviewer permission to that user. This permission is sufficient to permit read-only access to your calendar. A user who holds *Reviewer* permission for a calendar cannot update events or add new events, which is the level of access that is usually required by users such as executive assistants. Behind the scenes, Exchange uses the Add-MailboxFolderPermission cmdlet to assign permissions, but there is no user interface provided in OWA to allow another user write access to the calendar. However, an administrator can run the Add-MailboxFolderPermission cmdlet to assign *Editor* permission for a calendar to a user to allow them write access or run the Set-MailboxFolderPermission cmdlet to upgrade an existing *Reviewer* permission to *Editor*.

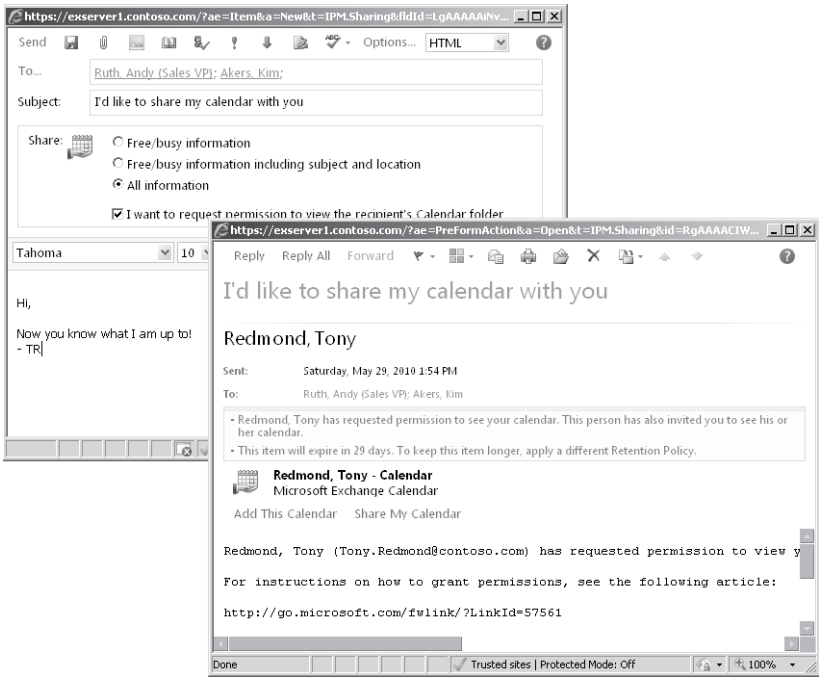


Figure 10-15 Sharing calendars with OWA.

These examples show how the cmdlets are used. The first command assigns the *Editor* permission for the calendar folder owned by Akers to another user called "Pelton, David". The second command upgrades the permission for the same calendar for a different user:

```
Add-MailboxFolderPermission -Identity 'Akers:\Calendar' -User 'Pelton, David'
-AccessRights Editor

Set-MailboxFolderPermission -Identity 'Akers:\Calendar' -User 'Smith' -AccessRights
Editor
```

You can add as many shared calendars as you like to your calendar list. However, all computer screens have limited real estate, and it gets very complex for developers to figure out how to squeeze information about all the calendars into the display. Some users run OWA on computers that have reasonably low-fidelity screens (think of a low-end notebook or netbook computer) that only support a screen definition of 1024×760, and OWA has to be able to cope with these situations as well as the extra-large screens that fill half a desk. There's always a trade-off, and in this case it's a limitation to be able to show a maximum of five calendars (your own calendar and four shared calendars). If you attempt to add another calendar, you'll run into the situation illustrated in Figure 10-16. There's no workaround or registry hack that forces OWA to fit more calendars onto the screen, even if the screen definition will support it, so you have to decide which calendars are most important to display and move others in and out of the display set as the need arises.

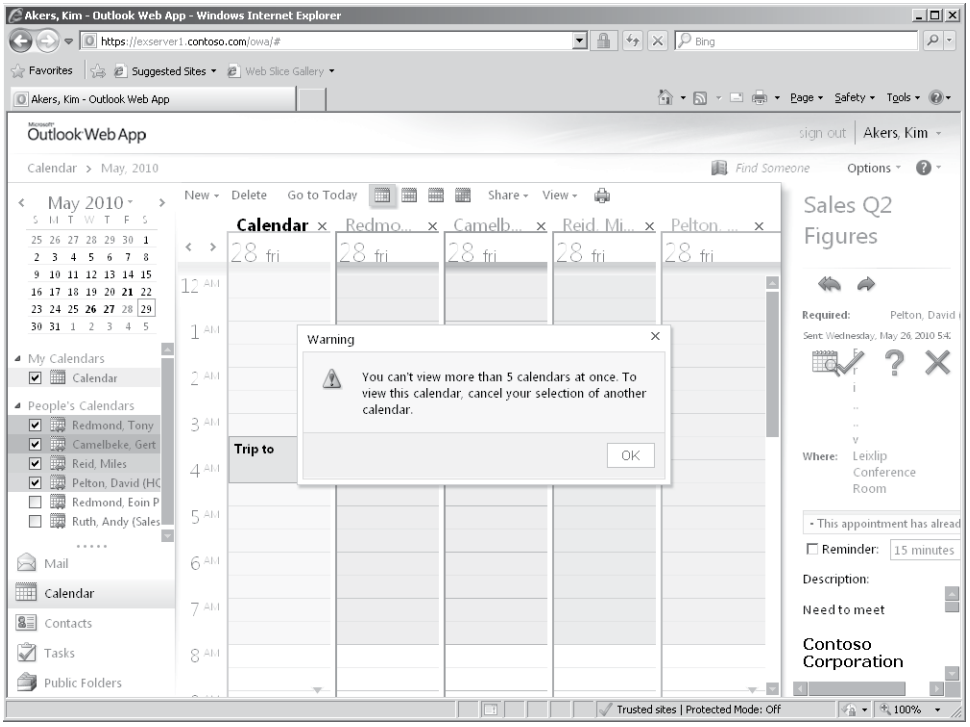


Figure 10-16 Viewing multiple calendars with OWA.

Sharing calendars with Internet users

Exchange 2010 SP1 introduces the ability to share or publish calendars to the Internet. This feature is intended to allow Exchange to offer the same facility that exists in other Web-based calendaring software such as Google Calendar and Yahoo! Calendar.

The feature leverages the work done to introduce federated calendar sharing between Exchange organizations while understanding that the intended target is quite different. Calendar sharing depends on a high level of trust between the participating organizations, whereas sharing calendars with Internet users is obviously a much looser arrangement, because no one manages the Internet and there's no notion of credentials being required to look at calendar data that a user has decided to publish. On the other hand, the Internet does encourage the development and implementation of standard protocols that can be used to share data, and in this case the protocol is iCal, or iCalendar.

Even though Exchange users can now share their calendars with Internet users, the operative word is "can." Sharing does not happen automatically. Exchange 2010 SP1 does not suddenly tear down the shutters and publish every user calendar as widely as possible. Instead, administrators and users alike must both take deliberate and planned actions to first create the conditions where calendar sharing is possible and then to make the decisions about with whom to share calendar data and what level of transparency or access to support.

The basis for calendar sharing is a vdir called /calendar that is underneath the /owavdir. To allow open sharing with the widest possible set of clients, the calendar vdir supports HTTP access. HTTPS connections will not be rejected, but HTTP is all that you need to share a calendar with an Exchange user.

> **Note**
>
> **The calendar vdir is serviced by a separate application pool to isolate it from OWA operations, so the fact that HTTP access is supported should not be a concern because there's no way for hackers to break into OWA just because they can get to a user's calendar.**

By default, Exchange 2010 does not allow calendar sharing, and the administrator must configure Exchange before users are allowed to share calendars. The following steps must be taken:

1. The OWA vdir must have an *ExternalURL* property set. Typically this is something like *https://mail.contoso.com/owa*.

2. The *InternetWebProxy* property of all of the mailbox servers that host mailboxes containing the calendars that will be shared with Internet users must be populated with the name of the CAS server in the Internet-facing site through which connections will be channelled. For example:

```
Set-ExchangeServer –Identity 'ExServer1' –InternetWebProxy
'http://ExCASInternet.contoso.com'
```

3. The OWA vdir of the CAS server must have its *CalendarPublishingEnabled* property set to $True:

```
Set-OWAVirtualDirectory –Identity "ExServer1\owa (default web site)"
–CalendarPublishingEnabled $True
```

4. A sharing policy must be configured to allow anonymous access to calendars. You can do this by amending the default sharing policy, or you can create a new sharing policy and apply that policy to the select group of mailboxes that you want to allow to share calendars. In this example, we set the default sharing policy to allow users to share calendar data at the level of free and busy information plus detail (the body of the appointment item) about appointments.

```
Set-SharingPolicy –Identity 'Default Sharing Policy' –Domains "Anonymous:
CalendarSharingFreeBusyDetail"
```

> **Note**
>
> **Attendee lists or attachments are never shared for meetings. Sharing policies are discussed in more detail in Chapter 5, "Exchange Management Console and Control Panel."**

If you create a new sharing policy, you will have to apply it to mailboxes using the Set-Mailbox cmdlet. For example:

```
Set-Mailbox –Identity 'Redmond, Tony' –SharingPolicy 'Internet Calendar Sharing
Policy'
```

Once the administrator has configured Exchange to allow calendar sharing to the Internet, users can publish calendars using the Publish This Calendar option available in the OWA calendar. Outlook doesn't support the same publishing feature directly, so when a user shares his calendar with an Internet correspondent through Outlook, he is redirected to the Web sharing page to execute the option.

Figure 10-17 shows the publication process to make calendars available to Internet users. On the left, the user selects the Publish This Calendar option. This causes OWA to display the dialog box shown in the middle to collect details about how the calendar should be published. The user controls:

- How much detail is revealed about appointments (the options are Availability Only, Limited Details, or Full Details). OWA will flag a warning if the user selects an option that is not allowed by the sharing policy that applies to the user's mailbox.
- How much calendar data are published. You can select to publish anything from one year to one day in advance of and before today.

- Whether the calendar is restricted or public. Restricted calendars have a GUID-based obfuscated URL that is extremely difficult to guess. A sample URL is:

  http://mail.contoso.com/owa/calendar/a6cc8807ab2e4e9385ced83564dc56c3@contoso.com/1f5d738f17aa4700a6469aa9428556b916453351635198862711/calendar.html

  In addition, Exchange does not allow restricted calendars to be indexed by search engines. Public calendars receive URLs of the type shown in the email at the bottom of Figure 10-17. You can see that the URL is reasonably straightforward and based on the user's alias. A user can switch between public and restricted by changing the publishing settings for the calendar. She can also stop publishing her calendar at any time.

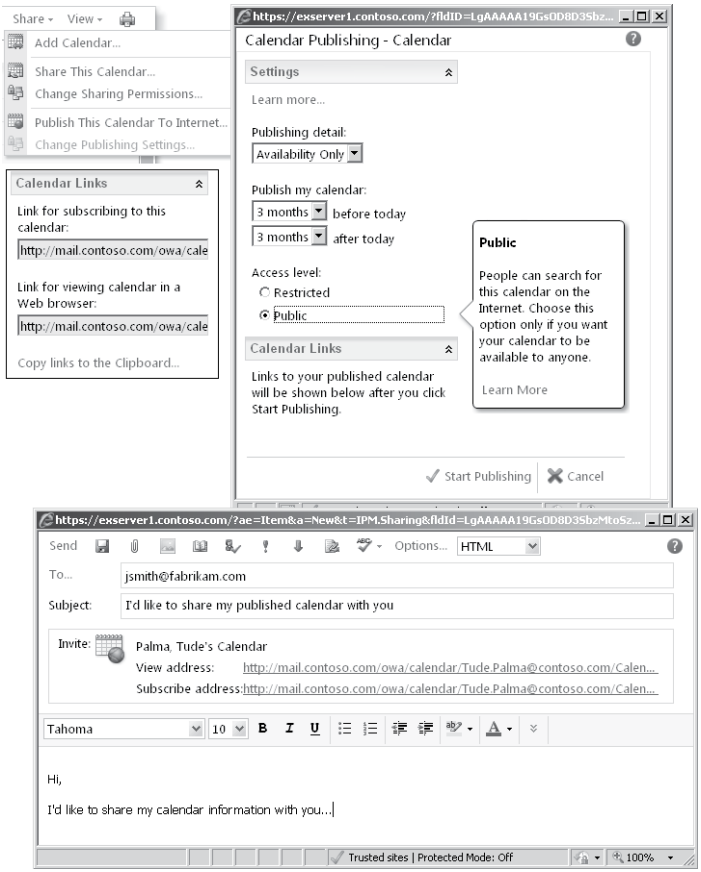


Figure 10-17 The OWA process to enable and then publish calendar information to Internet recipients.

- Who is advised about the availability of his calendar. A user could wait for his public calendar to be discovered by interested parties, but in most cases he will want to inform others that his calendar is now available from the Internet. OWA provides a Send Links To This Calendar option that generates an email for this purpose. Two URLs are included. The HTML link is to allow an Internet recipient to view details of your calendar as a simple Web page. The ICS (Internet Calendar Sharing) link is to allow others to add your calendar to a set of calendars (including presumably their own) that they need to reference on a frequent basis. The exact functionality that is enabled through the ICS link varies according to the client software used.

An administrator can also retrieve calendar settings for a mailbox with the Get-MailboxCalendarFolder cmdlet and configure calendar sharing with the Set-MailboxCalendarFolder cmdlet. For example, to retrieve the calendar settings for my mailbox, I'd use this command:

```
Get-MailboxCalendarFolder –Identity 'TRedmond:\Calendar'
```

```
Identity                : contoso.com/Exchange Users/Redmond, Tony:\calendar
PublishEnabled          : True
PublishDateRangeFrom    : ThreeMonths
PublishDateRangeTo      : ThreeMonths
DetailLevel             : LimitedDetails
SearchableUrlEnabled    : True
PublishedCalendarUrl    : http://mail.contoso.com/owa/calendar
/TRedmond@contoso.com/Calendar/calendar.html
PublishedICalUrl        : http://mail.contoso.com/owa/calendar
/TRedmond@contoso.com/Calendar/calendar.ics
IsValid                 : True
```

To change the published date range to one year before and after today's date and to change my calendar from public to restricted, I can use the following command:

```
Set-MailboxCalendarFolder –Identity 'Tredmond:\Calendar' –PublishDateRangeFrom
'OneYear' –PublishDateRangeTo 'OneYear' –SearchableUrlEnabled $False
```

Mailbox quota exceeded

OWA displays an indicator at the top of the folder list to show how much storage quota is available in the mailbox. Once the mailbox quota passes the warning limit, OWA changes the display to warn the user that she is running out of storage (Figure 10-18). She will also receive a message from the System Attendant to advise that items have to be removed from the mailbox before she can process more email. Separate storage quotas are maintained for the personal archive (if the mailbox has one), but OWA doesn't display any indication of how much space remains in the archive.

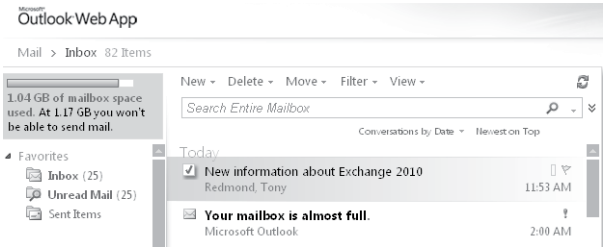


Figure 10-18 OWA signals that a mailbox is getting full.

All of this is very much what you'd expect Exchange to do when a mailbox is full, and you can leave the users to clean up their mailboxes to release some space. However, there are other consequences. When there is no space available, Exchange cannot update items in the mailbox—including hidden items that are used to hold mailbox settings. For example, if you go to Options and attempt to update any setting, OWA will display the error shown in Figure 10-19 when you attempt to save the mailbox settings. The solution is to free some space up in the mailbox to allow OWA to save items.

Figure 10-19 OWA can't save an item.

As discussed in Chapter 6, "Managing Mail-Enabled Recipients," another solution is for an administrator to assign additional quota to the affected mailbox. Because it caches information about mailbox settings for better performance, Exchange can take up to two hours to respect the new quota. However, once the change is effective, users will be able to create and update items in their mailboxes.

Handling attachments

A setting called *maxRequestLength* in Web.config.xml, the OWA configuration file, governs the maximum amount of data that a client can upload to a CAS. The default value for this setting is 30,000. The value is in kilobytes, and, in practical terms, it means that you can upload a 30 MB attachment to OWA or submit a message that is up to 30 MB (for instance, a message with several attachments that cumulatively amount to 30 MB). Of course, being able to attach a very large file and being able to send it are two very different things, and other limits placed on the organization, connectors, or a mailbox could interfere with the ability to actually deliver a large message.

INSIDE OUT Setting values and user expectations

Figure 10-20 illustrates how you can edit the OWA configuration file with Notepad. In this case, we change the maximum file size that the browser can upload to 4800, or 4.8 MB. The trick is to set the value so that it aligns with the maximum size supported by connectors so that users don't get into the situation where OWA allows them to create and upload a message only for it to be rejected by a connector. It's also important to inform users about the limitation so that they are not surprised if OWA refuses to upload a large file. Ideally, this should be communicated in the context of explaining how to use Exchange effectively and setting expectations about the size of messages that users can send and receive.

Figure 10-20 Editing the OWA configuration file.

When you attach a document to a message, OWA presents the Attach Files dialog box shown on the right side of Figure 10-21. This dialog box is reasonably effective, but it can be improved with the SP1 version of OWA by installing the latest version of Microsoft's Silverlight development platform on client computers. If OWA detects that Silverlight is available, it opens the normal file Open dialog box that is used elsewhere in Windows. This dialog box, shown on the left side of Figure 10-21, allows you to browse a directory and select multiple files in one operation. (In this case, I've selected some of the draft files for this book.) Due to its increasing popularity with Web developers, Silverlight is likely to be installed on client computers, but if not, you should add it to the list of updates that you consider adding to clients during deployment.

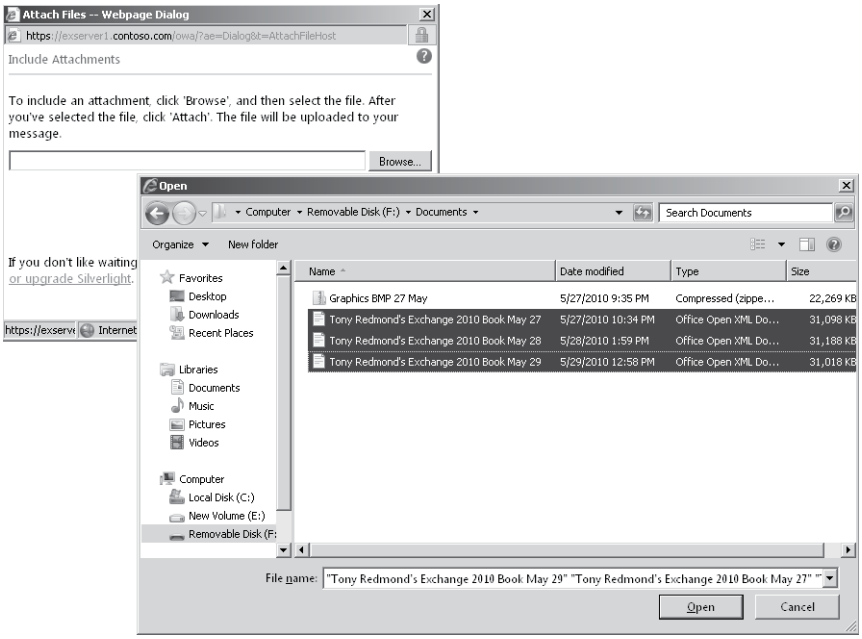


Figure 10-21 The difference that Silverlight makes to the OWA attachment dialog box.

OWA themes and customizations

A theme defines the color scheme and graphic elements used for OWA. Exchange 2007 supports customizable OWA themes and ships a number of different themes, including themes based on Microsoft Zune and Xbox products. You can also create your own theme and include corporate logos, color schemes, and so on. Users select from the range of themes through OWA options. Exchange 2010 still uses themes, but the RTM version doesn't support theme selection and everyone uses the same default theme. Exchange 2010 SP1 addresses the issue by providing a set of 27 themes from which users can select. Administrators don't have control over user choice and cannot impose a theme on users.

Creating a complete theme is a very extensive customization of the OWA UI. Many companies liked the idea of incorporating some aspect of their corporate identity into OWA without doing the work to create a theme. The complete source code of the OWA application is distributed with the Exchange kit, and the classic solution to the problem is to customize some of the files used for the default theme. However, because of the extensive upgrade that Microsoft has applied to the OWA UI in Exchange 2010, you cannot port changes made to the files used by Exchange 2007 directly to their Exchange 2010 equivalents. Instead, you have to redevelop any customization done for Exchange 2007 by reapplying the changes to the Exchange 2010 image and Cascading Style Sheets (CSS) files. The good

news is that Microsoft supports the customization of the OWA login page and documents what needs to be done to accomplish this task in the Exchange 2010 help file.

The most common customizations are applied to the OWA login pages to update the color scheme and logos to match corporate branding. TechNet contains a section dedicated to this topic that describes the names of the CSS and graphic files that OWA uses, how the components fit together to form the pages viewed by users, and how colors are assigned to the various text sections in the pages. Figure 10-22 shows a useful illustration from TechNet that shows the graphic files that OWA combines to present a customized log-in dialog box. You can use this information to develop the necessary customizations to comply with corporate branding.

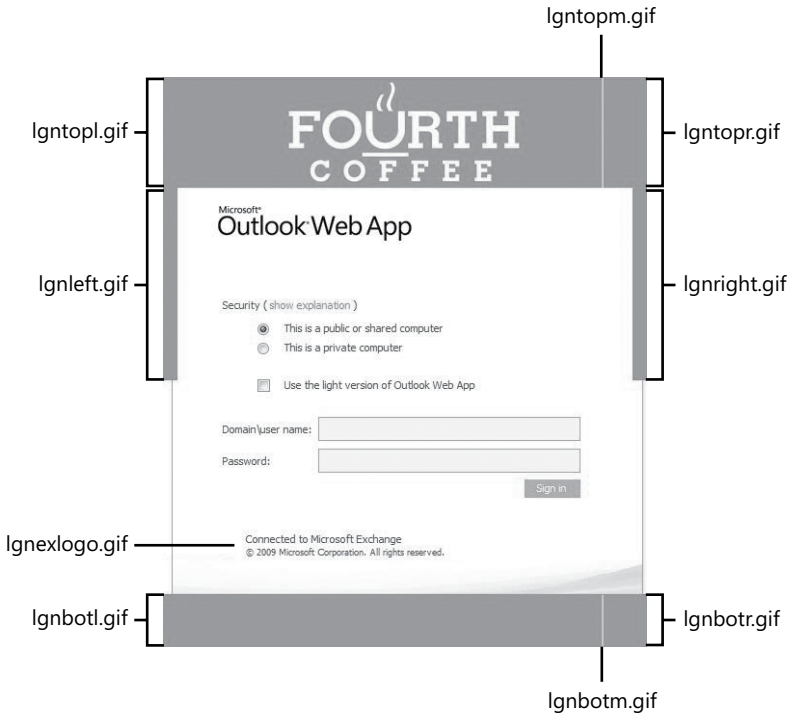


Figure 10-22 Components of OWA customization (source: TechNet).

An easy customization that you can accomplish in a couple of minutes is to add some text to the OWA log-in screen. This is commonly done to provide some guidance to users about how they should seek help in case of problems. Figure 10-23 shows how some text has been added for the contoso.com deployment and how it appears when a user logs into OWA.

To customize the log-in screen with some new text, you do the following:

1. Create a file containing the HTML formatting instructions for the text that you want to display. You can call the file anything you like as long as you store it in the \Program Files\Microsoft\Exchange Server\V14\Client Access Server\OWA\Auth directory. For the purpose of this explanation, let's call the file contoso-disclaimer.inc.

2. Copy the \Program Files\Microsoft\Exchange Server\V14\Client Access Server\OWA\Auth\Logon.aspx file. This is the file that contains all the instructions used by OWA when it logs on a user.

3. Open the Logon.aspx file with a text editor and search for the string "mid tblConn." Right below it, insert a line to instruct OWA to read and display the text contained in the contoso-disclaimer.inc file that we created in step 1. The code in Logon.aspx will then look something like this:

```
<table class="mid tblConn">
<!-- #include file="contoso-disclaimer.inc" -->
```

4. Save Logon.aspx and restart an OWA session (you don't have to restart Microsoft Internet Information Services [IIS] or any of the Exchange services). You should see your text displayed similar to Figure 10-23.

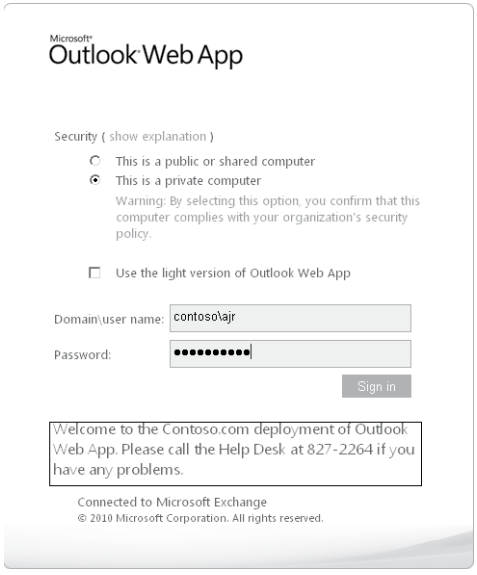


Figure 10-23 Updating the OWA log-in screen with some customized text.

5. The same approach can be taken to update Logoff.aspx if you want users to see a customized message when they sign off from OWA.

6. Once you are happy with the customization, you can apply it on all CAS servers by copying your modified files. There is no automatic mechanism to apply this kind of customization on every CAS server in an organization.

INSIDE OUT

Customizations could be overwritten by future product updates

The other thing to remember is that any customization to one of the OWA components is a candidate to be overwritten by a new Microsoft version of the component in a new service pack, hot fix, or roll-up update. That's why you keep careful documentation about any customization that you apply to OWA—to make it easier to apply it after you upgrade Exchange. You should also keep a copy of both the original and the customized versions of any file that you change so that you can review them in the future. It's also fair to say that there is no guarantee that Microsoft will not change the way that OWA works in a future version and render this method of customization—or any method of customization—invalid, so be prepared to build some time to test and perhaps do a little recoding for OWA customizations into every deployment plan.

OWA mailbox policies and feature segmentation

Exchange 2010 supports the ability to allocate different levels of functionality to OWA users through policies. Although Exchange 2010 includes a default OWA policy, it is not actually applied to mailboxes unless you explicitly select the mailbox and apply the policy to it. Otherwise access to OWA features is controlled by the segmentation properties defined for the OWA virtual directory on each CAS server (Figure 10-24). OWA mailbox policies didn't exist in Exchange 2007, and the only way that you could segment functionality was through the properties of the OWA Web site. The problem with this approach is that any change applies to all mailboxes that connect to that CAS. Using policies allows more granular control because you can apply different policies at the level of an individual mailbox. In addition to their ability to segment features presented through OWA, OWA mailbox policies control some of the user-controllable settings available through ECP.

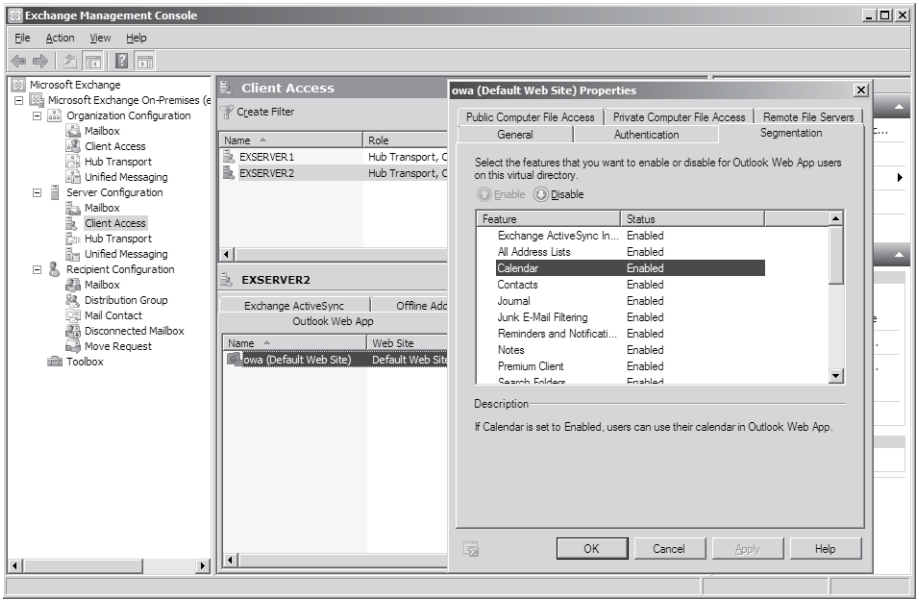


Figure 10-24 Viewing the segmentation properties of the default OWA Web site.

The easiest way to apply any OWA policy, including the default policy, to a set of mailboxes is with the Set-CASMailbox cmdlet. For example, this command fetches all the mailboxes that belong to the Exchange Users organizational unit (OU) and pipes them to Set-CASMailbox to apply the default OWA mailbox policy:

```
Get-Mailbox –OrganizationalUnit 'Exchange Users' | Set-CASMailbox
-OwaMailboxPolicy 'Default'
```

The default OWA policy typically duplicates the default out-of-the-box segmentation properties of the OWA default Web site as installed on a CAS server and permits access to all OWA features, including the premium client. To create a new policy, go to the Organization Configuration section of EMC, select Client Access, then on the Outlook Web App Mailbox Policies tab, and select the New Outlook Web App Mailbox Policy option in the action pane. A wizard then allows you to select which features you want users to access (Figure 10-25). In this case, we create a policy to restrict access to the OWA Light version that also selectively disables some OWA features.

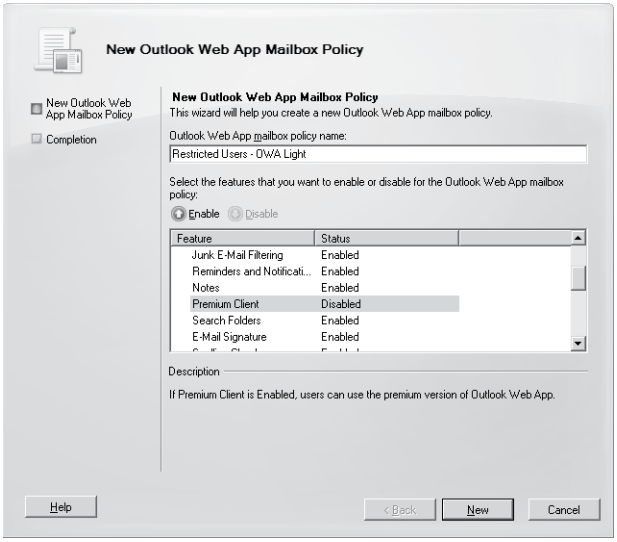


Figure 10-25 Creating a new OWA mailbox policy.

Table 10-3 lists the features that you can control in an OWA mailbox policy. Some of these features depend on other components (text messaging, public folders, and instant messaging), and others require a really good reason before you disable them. For example, it usually doesn't make much sense to disable the Change Password feature because handling user requests to change their passwords creates extra work for help desks.

Table 10-3 OWA features controllable through OWA mailbox policies

Feature	Meaning	Available through
Exchange ActiveSync Integration	If enabled, users can access details of the mobile devices that they have synchronized, including the ability to wipe devices if they are lost and retrieve logs containing details of synchronization operations. If disabled, the option is removed from ECP.	ECP
All Address Lists	If enabled, a user can see all defined address lists in the directory. If disabled, they can only see the GAL.	OWA
Calendar	If enabled, users can access the Calendar application. If disabled, the icon is removed from OWA.	OWA
Contacts	If enabled, users can access the Contacts application. If disabled, the icon is removed from OWA.	OWA
Journal	If enabled, users can see the Journal folder in their folder list. If disabled, OWA hides the folder.	OWA
Junk E-mail Filtering	If enabled, users can access the options to control junk mail processing such as blocked and safe user lists. If disabled, the option is removed from ECP.	ECP

Reminders and Notifications	If enabled, OWA will provide users with notifications of new messages, meeting reminders, and so on. If disabled, these notifications are suppressed.	OWA
Notes	If enabled, users can access the Notes application. If disabled, the icon is removed from OWA.	OWA
Premium Client	If enabled, users are able to use the premium client with a browser that supports this client. If disabled, users are forced to use the standard client no matter which browser they use.	OWA
Search Folders	If enabled, users can access search folders created by Outlook. If disabled, these folders are suppressed.	OWA
E-Mail Signature	If enabled, users can access the option to create or modify email signatures and apply them to outgoing messages. If disabled, the option is removed from ECP.	ECP
Spelling Checker	If enabled, users can spell check the content of messages. If disabled, the option is removed from OWA. Even when enabled, users do not have the option to customize their spelling dictionary.	OWA
Tasks	If enabled, users can create and manage tasks in OWA. If disabled, the option is suppressed.	OWA
Theme Selection	If enabled, users can select a theme other than the default and apply itto OWA and ECP. If disabled, the option is suppressed.	OWA/ECP
Unified Messaging Integration	If this feature is enabled and the mailbox is enabled for UM, users areable to access and manage their UM settings through ECP. If disabled, the option is removed.	ECP
Change Password	If this feature is enabled, users can change their account password from OWA. If disabled, OWA will not prompt users when their password is approaching its expiry date (prompts start 14 days in advance) and they will not be able to see the option to change their password in ECP.	OWA/ECP
Rules	If enabled, users will be able to create and modify rules through ECP. If disabled, the option is suppressed. However, any rules created with Outlook will continue to be respected by Exchange.	ECP
Public Folders	If enabled, users will be able to access and work with public folders. If disabled, the icon is removed from OWA.	OWA
S/MIME	If enabled, users can download the optional S/MIME control and then use it to apply digital signatures to messages and encrypt messages. If disabled, users are not able to create or read opaque-signed or encrypted messages. Messages that are clear-signed can be read (but not composed) and any digital signatures on the message will not be validated. Also, the option to download the S/MIME control is removed from ECP.	OWA/ECP
Recover Deleted Items	If enabled, users can recover deleted items. If disabled, users won't be able to recover deleted items with OWA, but Exchange will continue to preserve these items in the dumpster.	OWA

Instant Messaging	If enabled and an Instant Messaging (IM) integration is available, users will be able to access IM functionality from within OWA, including the ability to view presence information. If disabled, these features are unavailable.	OWA
Text Messaging	If enabled, users will be able to create and send text (SMS) messages from OWA. If disabled, this feature is removed.	OWA

A new policy can also be created with EMS. For whatever reason, this is a two-step process. First, you create the new policy with the New-OWAMailboxPolicy cmdlet, and then you use the Set-OWAMailboxPolicy cmdlet to define what features are enabled or disabled by the policy. For example, here's a policy that allows users to use the premium client while removing some of the more esoteric features:

```
New-OWAMailboxPolicy -Name 'Limited OWA features'
Set-OWAMailboxPolicy -Identity 'Limited OWA features'
-ActiveSyncIntegrationEnabled $True -AllAddressListsEnabled $True
-CalendarEnabled $True -ContactsEnabled $True -JournalEnabled $True
-JunkEmailEnabled $True -RemindersAndNotificationsEnabled $True
-NotesEnabled $True -PremiumClientEnabled $True -SearchFoldersEnabled $False
-SignaturesEnabled $True -SpellCheckerEnabled $True -TasksEnabled $True
-ThemeSelectionEnabled $False -UMIntegrationEnabled $False
-ChangePasswordEnabled $True -RulesEnabled $True -PublicFoldersEnabled $False
-SMimeEnabled $True -RecoverDeletedItemsEnabled $True
-InstantMessagingEnabled $False -TextMessagingEnabled $False
```


More than just segmentation

Although feature segmentation is the most obvious use of OWA mailbox policies and receives the most attention, you can also control other aspects of how users work with OWA through these policies. After you create a new OWA mailbox policy, you are able to define rules for file access and download when OWA is run on private and public computers. Click the policy with which you want to work and then select Properties. You can then access the properties that control feature segmentation and two other tabs for Public Computer File Access and Private Computer File Access (Figure 10-26).

> **Note**
>
> **In this context, Public and Private refer to the access mode chosen by the user when he starts OWA and indicate whether the browser runs on a public computer such as a kiosk or a private computer such as his own laptop.**

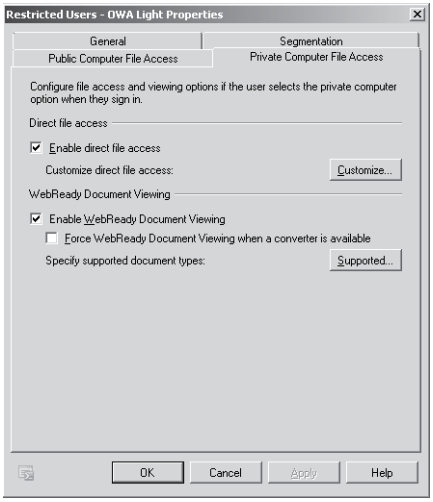


Figure 10-26 OWA Policy File Access options.

The Direct File Access settings (Figure 10-27) allow you to control how various file types are opened by users through OWA. The default option for both public and private computers is to allow direct access, meaning that users are able to open files. However, all types of files are not treated equally, as there are some file types that pose a potential risk of infection because they are often used as threat vectors by hackers who wish to infiltrate a computer. Files are therefore grouped into four categories:

- Always Allow: These files are deemed to be innocuous and safe to open on the client computer. The list includes types such as Word documents (.doc and .docx extensions) and Windows bitmaps (.bmp extension) that you can be reasonably sure will not contain malicious code.
- Always Block: These files pose a significant risk to a computer when they are opened by a user because they contain executable code. These files include types such as Windows batch files (.bat extension) and Windows command files (.cmd extension).
- Force Save: These are files that users cannot open directly and must save to disk before they can access the content. These types include Windows compiled help files (.chm extension).
- All others (unknown files that are not included in the other lists): The policy states what should be done if an unknown file type is detected. The default is to force a save to disk.

The priority given to action is from top to bottom. In other words, if a file type is on both the Always Block and the Force Save lists, it will be blocked.

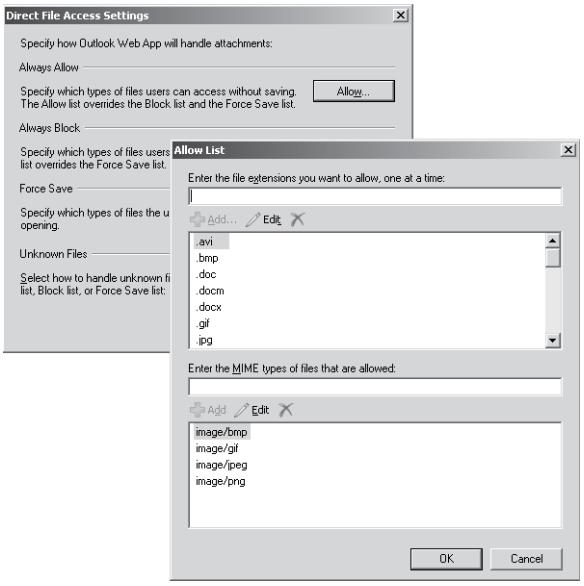


Figure 10-27 Configuring the file access allow list for an OWA mailbox policy.

If you prefer to have users open a viewer to access files rather than running the native application, you can select the Force WebReady Document Viewing When A Converter Is Available option. The effect is to force OWA to check documents as they are opened to see whether a WebReady converter is available and, if so, to always use the converter to open the file rather than calling the application. The idea is to eliminate any potential risk from macros or other code that could be carried around in the common file formats supported by WebReady, such as Microsoft Word and Microsoft Excel. In truth, the antivirus software that runs on today's PCs will usually catch any malicious code, so forcing WebReady viewing for OWA when it is run on a private computer could be considered overkill. Figure 10-28 shows how to access the list of file formats supported by WebReady converters. This list has been augmented over the last few years and supports a reasonably full set of the most common file formats that users will need to open in office environments.

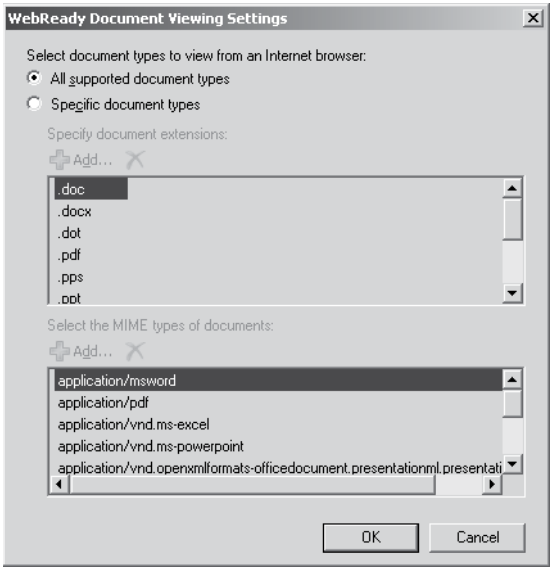


Figure 10-28 Viewing the list of WebReady supported document types.

It might be safe to allow users to open documents with applications on private computers, but it's a different matter on computers that are used for public access. In this scenario, it is reasonably common to block access to attachments to avoid the risk that users might download and leave sensitive files on a computer that can be accessed by an unauthorized individual. You can do this by clearing the option through EMC or by running the Set-OWAMailboxPolicy cmdlet. Settings applied through an OWA mailbox policy override those set through the properties of the OWA virtual directory. For example:

```
Set-OWAMailboxPolicy –id 'Restricted Users – OWA Light'
-DirectFileAccessOnPublicComputersEnabled $False
-ForceWebReadyDocumentViewingFirstOnPublicComputers $True
```

When this policy is applied, users will not be able to open or download and save files on public computers, but they will be able to access the content if a WebReady viewer is available. Web links that are included in messages are still active. Exchange 2010 includes viewers for Microsoft Office documents (see Figure 10-29), RTF, and PDF files.

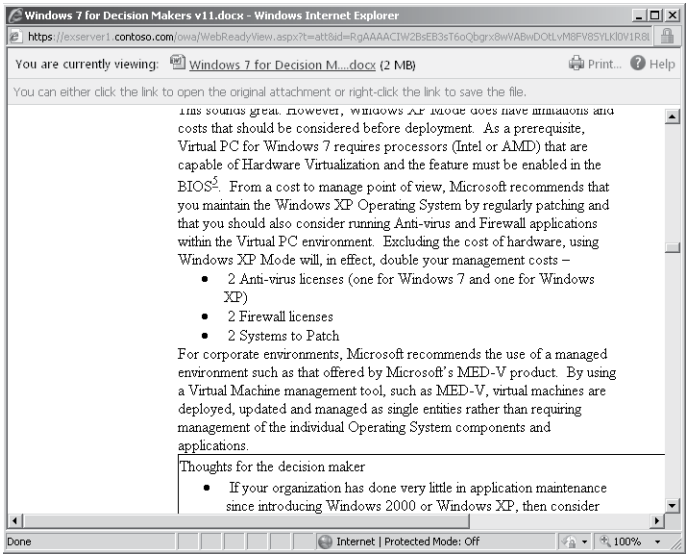


Figure 10-29 Using a WebReady viewer to read a Word document.

INSIDE OUT **A note on document access**

Exchange 2007 introduced the ability for OWA users to access documents through UNC paths (typically to a file share) or on SharePoint sites. This feature is deprecated in Exchange 2010, possibly because it is not used extensively but also because of the potential security concerns in accessing documents from unknown or untrusted locations.

Attachment processing

Administrators control how OWA handles attachments by creating a list of attachment types and marking each as blocked, allowed, or "force to save." Obviously, *blocked* means that users cannot open or download an attachment of this type to their PC, normally because the file type is likely to contain a virus or some other dangerous content. *Allowed* means the opposite, as there is a high degree of confidence that these attachments are safe.

OWA performs special processing for attachments marked as *force to save*. This means that the user has to save the attachment to his local disk before he can view its contents. As OWA downloads the attachment from the server, it checks to see whether it is XML or

HTML. In this case, OWA runs some code called Safe HTML to strip out any malicious XML or HTML code. If the attachment is another type, OWA examines the content to see if it actually contains XML or HTML code. This check is performed to ensure that no attachment is ever downloaded that contains malicious code, which could introduce a virus or another dangerous program onto the PC. If hidden XML or HTML code is detected, OWA strips the attachment and replaces it with a text file to tell the user that the attachment was removed.

INSIDE OUT

A high level of protection on OWA prevents the receipt of code in text files

The level of protection supplied by OWA is very high and reflects the experience of PCs being infected by attachments. However, it also means that you cannot send HTML code in a text file to an OWA user because it will be stripped.

Applying an OWA mailbox policy

After the new policy is created, to apply it, you switch to Recipient Configuration and select one or more mailboxes and then Properties from the action pane. Click the Mailbox Features tab, select Outlook Web App, and then select Properties. You can then select an Outlook Web App mailbox policy and apply it to the mailbox (Figure 10-30).

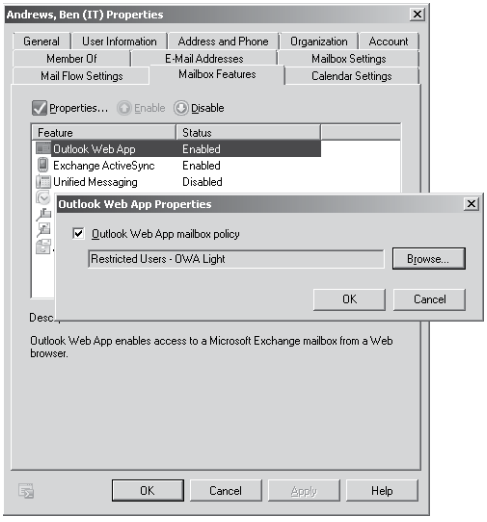


Figure 10-30 Applying an Outlook Web App mailbox policy.

Exchange enforces the new policy the next time that the user logs into her mailbox. If everything works as expected, the user will be presented with a restricted version of OWA Light. Of course, you can also apply an OWA mailbox policy to a mailbox with EMS:

```
Set-CASMailbox –Identity 'Andrews, Ben (IT)' –OWAMailboxPolicy 'Restricted Users
–OWA Light'
```

INSIDE OUT

Integrating OWA and OCS

One small glitch might creep in with the instant messaging section of the policy. OWA 2010 supports a nice integration with Office Communications Server (OCS), but if you want to create the link between the two products, you have to ensure that the OWA mailbox policy that is applied to mailboxes that want to use OCS specifies "OCS" in the *InstantMessagingType* attribute. For example:

```
Set-OWAMailboxPolicy –Identity 'OCS Integration Enabled' –InstantMessagingType
'OCS'
–InstantMessagingEnabled $True
Set-CASMailbox –Identity 'Akers, Kim' –OWAMailboxPolicy 'OCS Integration
Enabled'
```

You can find more information about the integration between OWA and OCS, including some important changes made in Exchange 2010 SP1 to improve how the integration is accomplished, at *http://msexchangeteam.com/archive/2010/09/27/456446.aspx*.

POP3 and IMAP4 clients

POP3 and IMAP4 are Internet email protocols that are supported by a wide variety of clients and servers. Fans of these protocols love the lightweight nature of their connections, which is one of the reasons they are the protocols of choice for most consumer free email services such as Hotmail and Gmail (Hotmail supports POP3; Gmail supports both protocols). POP3 is the older and less functional protocol. IMAP4 is more functional than POP3, but less functional than MAPI. Nevertheless, modern IMAP4 clients, including Outlook, can build a rich range of features around the rudimentary but superefficient communications to download messages from a server. Of course, unlike MAPI, POP3 and IMAP4 are both protocols that clients use to retrieve messages from a server. Apart from age, the fundamental differences between the two protocols are as follows:

- POP3 downloads messages to a client and removes them from the server.
- POP3 supports a very limited set of folders on the server (essentially, the Inbox).
- IMAP4 can leave copies of downloaded messages on the server.

- IMAP4 can access any folder that a server exposes and download messages from the folders to client-side replicas.
- IMAP4 allows a "live sync" mode in which the client holds open a connection to the server; this provides a more Outlook-like sync experience in which messages trickle in to the Inbox as they arrive instead of arriving in batches when a POP3 connection is made.

The vast bulk of clients that connect to Exchange 2010 via POP3 and IMAP4 belong to four categories:

- Users in an educational establishment such as a university.
- Users who access an Exchange mailbox running as a hosted service when Outlook is not provided to restrict costs.
- Users who consider Outlook to be overfeatured, bloated software. Often, these users have used a client such as Eudora or Thunderbird for many years and don't see a reason to change.
- Users who run an operating system that doesn't support the premium version of OWA or who simply prefer to use IMAP. Many Linux users are in this category.

The attraction of using free POP3 or IMAP4 clients is the avoidance of Outlook license fees. This is less of an issue in large corporations that conclude enterprise licensing agreements with Microsoft that include the entire Office application suite. For this reason, relatively few users in large corporate deployments use POP3 or IMAP4 clients. OWA is available if they don't want to use Outlook, and it's easier for the help desk if a limited number of clients are in use. Another reason is that POP3 and IMAP4 clients are purposely designed to work across any server that supports these protocols. They therefore do not support features that are specific to Exchange, such as:

- Display of MailTips
- Organization of message threads into conversation views
- Display of retention tag information
- Display of protected content such as items that require licenses from Active Directory Rights Management Services

For the remainder of this discussion, I focus on setting up the Exchange 2010 IMAP4 server and configuring clients to connect to the IMAP4 server. The steps to set up and configure POP3 access are similar in concept if different in detail but not covered here. Copious detail on this topic is available on the Internet.

Configuring the IMAP4 server

When you install the CAS role on a server, the setup program creates the Microsoft Exchange POP3 and Microsoft Exchange IMAP4 services to support client connections via these protocols, but does not start the services. Therefore, the first step to support POP3 or IMAP4 clients is to start these services. In addition, you should change the startup state for the services from Manual to Automatic so that Windows will start them every time the server is booted. These Windows PowerShell commands serve the purpose:

```
Set-Service msExchangeImap4 -StartupType Automatic
Start-Service -Service msExchangeImap4
```

Once the IMAP4 service is started, the CAS server runs a virtual IMAP4 server that monitors incoming client connections on port 143 (Transport Layer Security [TLS] or unencrypted connections) and 993 (Secure Sockets Layer [SSL] secured connections). Figure 10-31 shows the properties that are usually of most interest to administrators when they configure IMAP connectivity for Exchange 2010. The *Binding* properties define the ports that the IMAP4 server monitors and the set of IP addresses that clients can use to connect to the IMAP4 server. The *Connection* properties define various limits that the server uses to control client connections. See TechNet for detailed information on these settings.

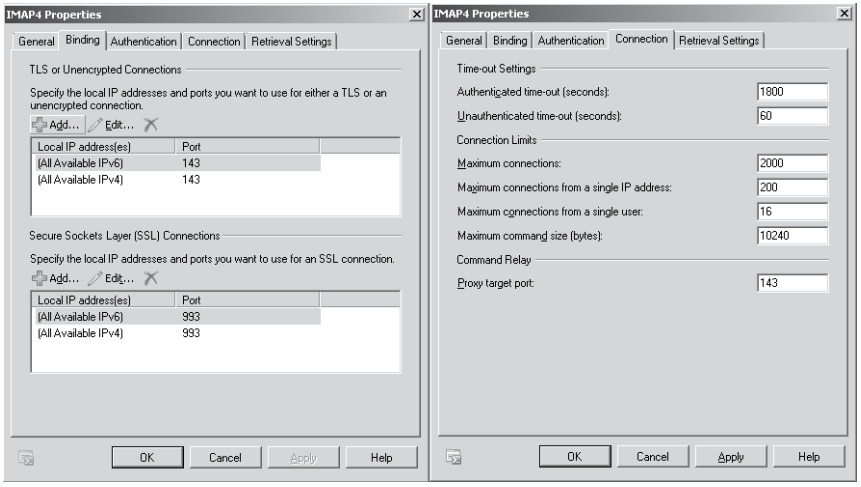


Figure 10-31 Viewing the properties of the Exchange IMAP4 server.

Exchange 2010 SP1 allows you to use basic authentication, Integrated Windows Authentication, and TLS-secured logons (the default) to connect POP3 or IMAP4 clients. As shown in Figure 10-32, you can select the authentication method that you want to use through the Authentication tab of the IMAP4 server Properties dialog box.

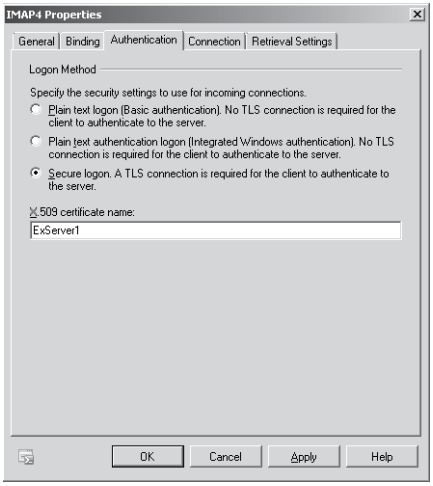


Figure 10-32 Authentication properties for the IMAP4 server.

The Get-IMAPSettings and Set-IMAPSettings cmdlets are used to retrieve and set configuration settings for the IMAP4 server. The equivalent cmdlets for POP3 are Get-POPSettings and Set-POPSettings. For example, to retrieve the current configuration for a CAS server called ExServer1, we use the following command:

```
Get-IMAPSettings –Server ExServer1Protocol
```

```
Name                                : IMAP4
Name                                : 1
MaxCommandSize                      : 10240
ShowHiddenFoldersEnabled            : False
UnencryptedOrTLSBindings            : {:::143, 0.0.0.0:143}
SSLBindings                         : {:::993, 0.0.0.0:993}
InternalConnectionSettings          : {ExServer1.contoso.com:993:SSL, ExServer1.contoso.com:143:TLS}
ExternalConnectionSettings          : {}
X509CertificateName                 : ExServer1
Banner                              : The Microsoft Exchange IMAP4 service is ready.
LoginType                           : PlainTextAuthentication
AuthenticatedConnectionTimeout      : 00:30:00
PreAuthenticatedConnectionTimeout   : 00:01:00
MaxConnections                      : 2000
MaxConnectionFromSingleIP           : 2000
MaxConnectionsPerUser               : 16
MessageRetrievalMimeFormat          : BestBodyFormat
ProxyTargetPort                     : 143
CalendarItemRetrievalOption         : iCalendar
OwaServerUrl                        :
EnableExactRFC822Size               : False
LiveIdBasicAuthReplacement          : False
```

```
SuppressReadReceipt                 : False
ProtocolLogEnabled                  : False
EnforceCertificateErrors            : False
LogFileLocation                     : C:\LOGS\IMAP4
LogFileRollOverSettings             : Daily
LogPerFileSizeQuota                 : unlimited
Server                              : EXSERVER1
Identity                            : EXSERVER1\1
```

Note that the *ExternalConnectionSettings* property listed has a blank value. This property is displayed by the Account Information page of ECP when a user clicks Settings For POP, IMAP, And SMTP Access. The intention is to provide users with the information that they need to input into clients that use these protocols so that they can connect to Exchange. Unfortunately, if you leave the default settings in place, users will see blank values when they access the page. Users need to know the name of the CAS server to which to connect, the port number for the protocol, and the encryption type security setting. We can set these values with the Set-POPSettings and Set-IMAPSettings cmdlets. To provide the information, you have to take these steps:

- Provide details for POP3 access:

  Set-POPSettings –ExternalConnectionSettings ExServer1.contoso.com:995:SSL –Server ExServer1

- Provide details for IMAP4 access:

  Set-IMAPSettings –ExternalConnectionSettings ExServer1.contoso.com:993:SSL –Server ExServer1

- Publish information about the connector:

Most companies provide a receive connector on a hub transport server that POP3 and IMAP4 clients can use to relay their outgoing messages. To make users aware of which server to use, you publish information about the connector by setting its *AdvertiseClientSetting* property to $True. In this case, I'm telling users to connect to the default receive connector on a hub transport server, but in production circumstances you'd be more likely to create a specific receive connector that is dedicated and configured to act as a client relay.

```
Set-ReceiveConnector –Identity 'ExServer1\Default ExServer1' –AdvertiseClientSettings
$True
```

Remember that the POP3 and IMAP4 settings are server-specific, so you have to set the values on each of the CAS servers that you want to use for this purpose. A restart of IIS on the CAS server might be necessary to make the new values available to clients.

If you change any of the configuration settings for the IMAP4 server, you have to restart the Microsoft Exchange IMAP4 service. It's common to find that you want to turn on protocol logging to help debug connections from a particular client. To enable protocol logging for IMAP4 clients, you need to enable logging and tell Exchange where it should create the log. Enabling logging in Exchange 2007 requires you to edit a configuration file, but Exchange 2010 allows you to enable logging with the Set-IMAPSettings cmdlet. For example:

```
Set-IMAPSettings –Server ExServer1 –ProtocolLogEnabled $True –LogFileLocation
'C:\Logs\'
```

Logging generates a mass of data on the server, some of which is fairly obtuse if you are not familiar with debugging IMAP connections. Clients can also generate logs, and if you need to provide data to help a support representative solve a problem, you should generate server and client logs to ensure that they have full knowledge of what the client is sending and how the server is responding.

Configuring IMAP4 client access

From a user perspective, it is easy to configure a POP3 or IMAP4 client to connect to Exchange 2010. For my example, I chose the Thunderbird free IMAP4 client that you can download from *http://www.mozillamessaging.com/en-US/thunderbird/*. Two separate connections must be configured before an IMAP4 client can download and send messages.

1. An IMAP4 server hosted by a CAS server must be ready to accept client connections so that IMAP4 clients can access mailboxes and download folders and items.

2. A Simple Mail Transfer Protocol (SMTP) receive connector hosted by a hub transport server must be ready to accept client connections to allow IMAP4 clients to relay outgoing messages via SMTP.

Note that the CAS blocks POP3 and IMAP4 from the Anonymous and Guest accounts. Additionally, the Administrator account cannot be used to connect to Exchange via these protocols either. To access the Administrator mailbox, you must use Outlook or OWA.

The steps required to configure the client to connect to Exchange 2010 are as follows:

1. Set the authentication setting to Basic for the IMAP4 server on the CAS to which you want to connect the client. This is sufficient for testing purposes because it ensures that just about any IMAP4 client will be able to connect. Once you have established that connections work freely, you can increase the level of security by moving to Integrated Windows Authentication or Secure Logon, depending on what authentication mechanisms are supported by the client.

2. Restart the IMAP4 server to effect the change in the authentication setting.

3. Configure the client with the name of the CAS server and the user name in domain name\account name format. Figure 10-33 shows the server settings as input into Thunderbird.

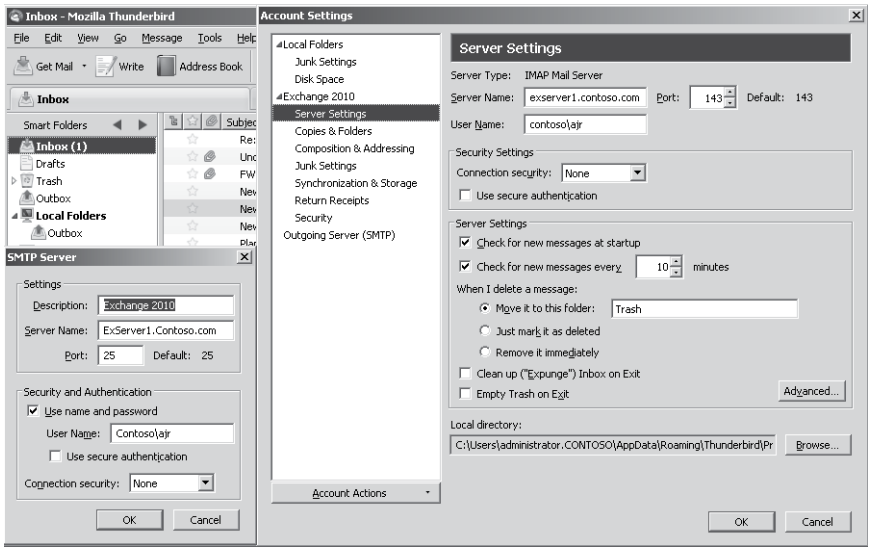


Figure 10-33 Connecting the Thunderbird IMAP4 email client to Exchange 2010.

4. Connect the client to prove that messages can be downloaded.

5. Check that the Permission Groups assigned to the default client receive connector on the hub transport server that you want to use for sending outbound messages allows anonymous connections. Again, this is the easiest setting to use to test outgoing message connectivity and should ensure that all types of clients are able to connect to send messages. Once you know that messages are flowing, you can increase the security. As you can see in Figure 10-33, the Thunderbird client supports STARTTLS security with username and password credentials, so this means that the receive connector doesn't need to allow anonymous connections because these connections will be regarded as "Exchange users."

Once messages are being downloaded and sent freely, the next step is to configure Lightweight Directory Access Protocol (LDAP) access to Active Directory so that we can use Active Directory as an address book. The details of how to configure a connection to Active Directory vary from client to client, as does the ability of the client to use the data fetched from Active Directory. Some clients are only able to browse Active Directory, whereas others, like Thunderbird, are able to validate email addresses against Active Directory as they are entered into message headers. Figure 10-34 shows the settings that I used to define Active Directory as an address book for Thunderbird.

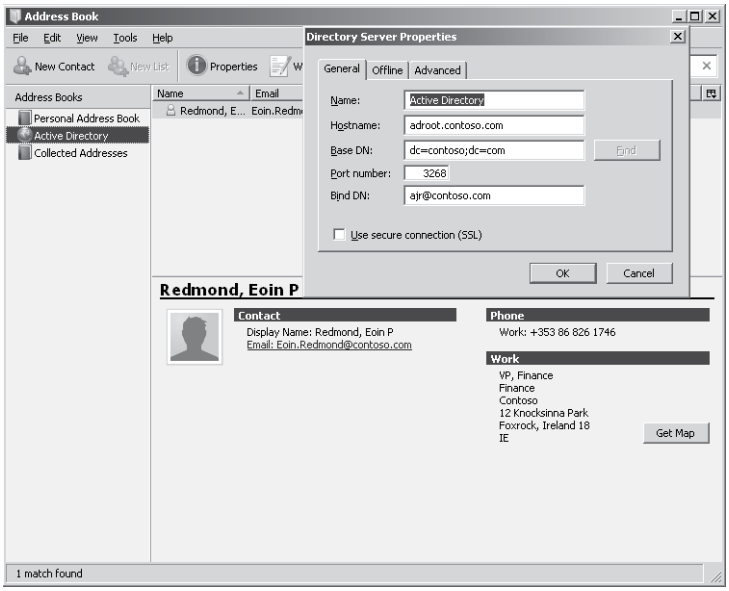


Figure 10-34 Configuring the Thunderbird address book to connect to Active Directory.

The following settings are used:

1. The name is set to Active Directory, but it can be anything that you like, because this serves purely as an illustrative name and has no other function.
2. The hostname is set to the fully qualified domain name (FQDN) of a global catalog server that provides access to Active Directory. Ideally, this should be a global catalog server in the same site as the CAS server.
3. The base DN provides a starting point for LDAP searches in the directory. In this instance, we provide the root of the directory to ensure that searches can find any mail-enabled object in contoso.com.
4. The port number is set to 3268 rather than the standard port (389) used by LDAP.
5. The bind DN is set to my SMTP address.

To test the connection, open the client address book and attempt to search for some mailboxes that you know exist. You should be able to see mailboxes, contacts, and distribution groups.

INSIDE OUT **Only minor issues**

Two small issues are the following:

1. **The LDAP searches executed by a client might ignore Exchange-specific filters. For example, if you select the Hide From Exchange Address Lists check box for an object, it stops Outlook and OWA users from seeing that object through the GAL. However, this block means nothing to other clients, and the hidden objects will probably be revealed to users.**
2. **Along the same lines, an LDAP search against Active Directory doesn't impose any filters to eliminate objects that are not mail-enabled, so you'll probably be able to see security groups such as Enterprise Admins. However, you won't be able to send email to these objects because they don't have email addresses.**

These are small hiccups along the road, and the fact that users have read-only access to directory information that reveals some objects that other clients don't show isn't really very serious.

Exchange ActiveSync

Microsoft added server-based ActiveSync to Exchange with the release of Exchange 2003 SP2. Exchange 2007 marked a major upgrade for ActiveSync. The version of ActiveSync provided with Exchange 2010 is much less of a change because it's mostly a case of tweaking existing functionality to improve support for clients. It might be the case that ActiveSync is now "competitive enough," and Microsoft is concentrating on licensing ActiveSync as widely as possible. Certainly the licensing activity is progressing in leaps and bounds, because all major smartphone vendors now license ActiveSync, including Apple for iPhone and Google for Android-based devices. The attractiveness of ActiveSync is strongly linked to the success of Exchange as the de facto corporate email platform. Although Windows Mobile has not evolved to match the capabilities of other mobile platforms, Microsoft has experienced increasing success due to the growing number of Exchange mailboxes allied to greater availability of ActiveSync-enabled devices.

Among the major functionality changes in Exchange 2010 ActiveSync that upgraded clients can use are the following:

- Like Outlook and OWA 2010, messages can be grouped in conversations so that all of the items relating to a topic can be handled at a single time. You can deal with a conversation by reading the latest item, deleting one item or the complete conversation, or moving everything in one operation. The most interesting aspect is the way that

Exchange maintains metadata on the server to handle ongoing processing of conversations as new items arrive. The metadata lets Exchange take care of operations such as "delete any further items in this conversation" or "move items as they arrive into this folder."

- You have the ability to fetch free/busy information from the server for contacts to determine if they are available at a certain time. This is a great feature when you attempt to set up a meeting on a mobile device.

- If you deploy Exchange 2010 Unified Messaging, voice mail messages are now played directly within Outlook Mobile (because they are encoded as MP3 files) rather than having to export the voice attachment to be played through Windows Media Player or another audio player. In addition, voice mails are delivered with a transcription of the message in the body so you can get an idea of what the message is about without having to open the voice mail attachment.

- Outlook Mobile stores details of recent email addresses in a nicknames cache in a hidden item in the mailbox root that it shares with OWA and Outlook so that you can see a suggested list of recipients as you enter a new address.

- You can use your phone to send and receive Short Message Service or text messages. You can use OWA or Outlook to compose and send new messages and ActiveSync then synchronizes the SMS messages down to the mobile device, which then transmits the message like any other SMS. The reverse also works, and incoming SMS messages are downloaded to the device and then synchronized by ActiveSync into your mailbox where they can be processed using OWA or Outlook. Text messages are stored like any other message in the Inbox and Sent Items folders and can be operated on like any other item. For example, you can flag a text message for follow-up or forward it to another user.

> **Note**
>
> **Anyone who despairs at the limited keyboards available on many mobile devices will delight in the ability to create and send SMS messages using their computer keyboard. Using ActiveSync to synchronize text messages with Exchange is a simple, brilliant idea. The only complaint that you could make is to ask why this feature hasn't appeared before now, and the answer is probably that although text messaging has been enormously popular in Europe and Asia for years, it's only recently that SMS communications have become popular in the United States.**

- Similar to Outlook, an icon to indicate that you have already replied to or forwarded a message is now shown.

Phones running Windows Mobile 6.1 can install a CAB update to upgrade Outlook Mobile to access the new features available in ActiveSync in Exchange 2010. Exchange 2010 users whose mailboxes are enabled for ActiveSync and who access their mailboxes with a suitable device will receive a message inviting them to download an over-the-air (OTA) upgrade to apply to their device (Figure 10-35). The upgrade is downloaded from Microsoft's Web site. When applied, it adds Outlook Mobile to the device and allows the user to access features such as SMS synchronization and conversations. Phones running Windows Mobile 6.5 or greater already have the code necessary to access these features. This is a long-overdue improvement to the situation that existed where companies were essentially forced to upgrade Windows Mobile devices to be able to utilize any features introduced in a new version of ActiveSync, and it might represent a realization within Microsoft that enterprises don't discard mobile devices used for business purposes quite as quickly as consumers do when they pursue the latest fashion device.

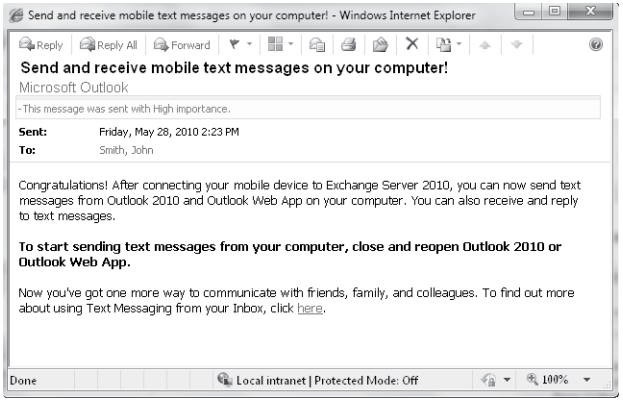


Figure 10-35 The invitation to use integrated text messaging.

Setting ActiveSync policies

Unless you specify a different policy, Exchange applies the default ActiveSync policy to mailboxes when they are enabled for ActiveSync. Policies defined on Exchange 2007 are upgraded to be used by Exchange 2010 by the addition of a new capability to allow or block specific applications on a mobile device. This capability is revealed on the Other tab of a policy's Properties dialog box (Figure 10-36). Note that use of this capability requires an upgrade to the enterprise Client Access License (CAL). The properties revealed through the other tabs remain the same, because Exchange 2007 and control settings such as whether a password is required to access the device and what type of password should be used, what size of messages and attachments are to be synchronized, whether nonprovisionable devices (those that do not respect policies) are allowed, and what device capabilities (camera, removable storage, Wi-Fi, and Bluetooth) are supported.

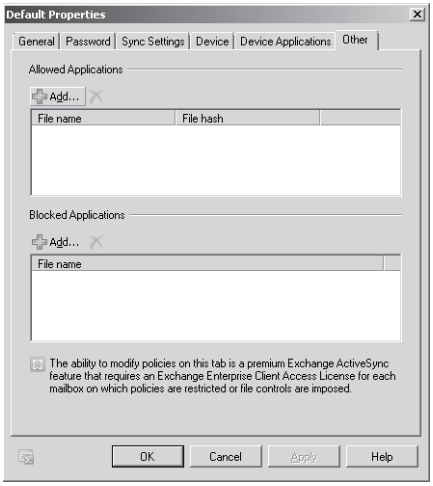


Figure 10-36 Editing an ActiveSync policy to allow or block applications for mobile devices.

INSIDE OUT

Mailbox policy and ActiveSync client–server partnership

Although you can collect all these settings into a policy and apply that policy to mailboxes, it is important to realize that ActiveSync is based on a client–server partnership. The server might insist on some settings, but the client can blithely ignore the demands of the server if the client operating system has never implemented the code to apply the policy fully. ActiveSync makes a clear differentiation among fully compliant devices, partially compliant devices, and noncompliant devices. A *fully compliant* device is one that implements all of the settings specified by a policy. A *partially compliant* device applies some but not all of the settings in a policy, probably because the necessary code to apply one or more settings is not implemented in the operating system. A *noncompliant* device can synchronize with Exchange but essentially ignores policy settings because it does not recognize or will not accept a policy provided to it by the server. For example, you can create a policy that blocks the use of a camera in mobile devices, and this policy will be effective for Windows Mobile devices where the operating system understands how to disable the camera after the policy is applied. However, an Apple iPhone will ignore the policy and the camera can still be used. The reason is that the iPhone includes a partial implementation of ActiveSync, certainly enough to do the basics of email, calendar, and contact synchronization, but missing some important management details.

You can change the policy that is applied to a mailbox at any time. For example:

```
Set-CASMailbox -Identity 'Ruth, Andy' -ActiveSyncMailboxPolicy "Executives"
```

Exchange 2010 SP1 allows you to create and maintain ActiveSync policies through the Phone And Voice section of ECP. Exactly the same parameters are exposed as in EMC. Figure 10-37 illustrates how the creation of a new ActiveSync policy appears in ECP.

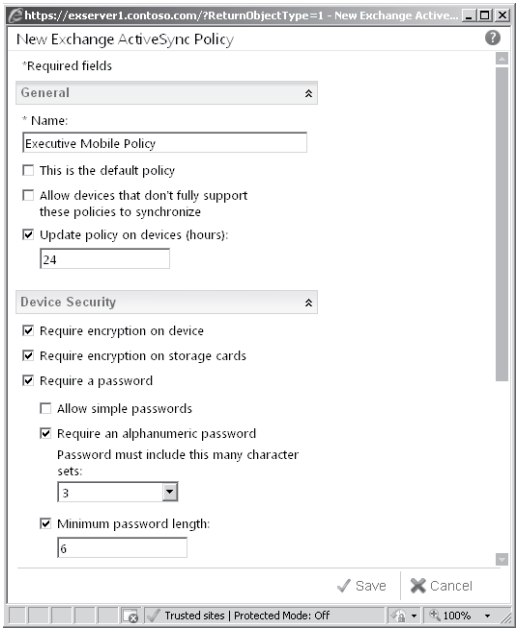


Figure 10-37 Creating a new ActiveSync policy with ECP.

Generating ActiveSync reports

All of the transactional information about communications between ActiveSync clients and Exchange is recorded in IIS logs. Here you'll find information such as the protocol version of the devices that communicate with Exchange, the type of synchronization performed (mail, calendar, contacts, everything), statistics about the number of operations performed (adds, changes, deletes), and specific operations such as out-of-office (OOF) message creation. Although all this is available, it is buried in the mass of other data recorded in the IIS logs, and some help is therefore required to retrieve the correct data and interpret it in a human-friendly manner. The Export-ActiveSyncLog cmdlet is designed for just this purpose.

Export-ActiveSyncLog is run on a CAS server. The cmdlet scans the IIS logs to filter data relating to ActiveSync operations and uses these data to generate a set of reports. These are:

- Usage report (Users.csv): A summary of the total items sent and received (number of items and bytes) broken down by item type (email, calendar, etc.).
- Servers report (Servers.csv): A summary of the servers hosting the mailboxes that are associated with ActiveSync requests.
- Hits report (Hourly.csv): The total number of synchronization requests processed per hour plus the total number of unique devices that initiate synchronization requests each hour.
- HTTP status report (StatusCodes.csv): A summary of the different HTTP error response codes and the percentage of the time that each code was encountered. The intention of the report is to give an indication of the overall performance of the server.
- Policy Compliance report (PolicyCompliance.csv): A summary of the number of devices that are compliant, partially compliant, and noncompliant.
- User Agent list (UserAgents.csv): A summary of the total number of unique users who have connected to ActiveSync during the report period, organized by mobile device operating system (different versions of Windows Mobile, iPhone, Android, and so on).

Because the output files are in CSV format, you can open and interpret them to meet your own needs. You can open the files with Microsoft Excel or Microsoft Access or import them into a database to allow more sophisticated analysis and reporting based on data collated over an extended period.

Reporting synchronized devices

The Get-ActiveSyncDeviceStatistics cmdlet provides information about the synchronization status for a mailbox. For example, here's an edited version of the information reported for a mailbox:

```
Get-ActiveSyncDeviceStatistics –id JSmith
```

```
FirstSyncTime                 : 2/24/2010 7:00:01 PM
LastPolicyUpdateTime          : 3/3/2010 7:30:22 PM
LastSyncAttemptTime           : 3/4/2010 6:06:18 PM
LastSuccessSync               : 3/4/2010 6:06:18 PM
DeviceType                    : PocketPC
DeviceID                      : 5ECE2DBB684616DD07FB173DF09254B5
DeviceUserAgent               : MSFT-PPC/5.2.5070
DeviceWipeSentTime            :
```

```
DeviceWipeRequestTime          :
DeviceWipeAckTime              :
LastPingHeartbeat              :
RecoveryPassword               : ********
DeviceModel                    : HP_KB1
DeviceImei                     : ******
DeviceFriendlyName             : HP_iPAQ_Glisten
DeviceOS                       : Windows CE 5.2.21871
DeviceOSLanguage               : English
DevicePhoneNumber              : *******7701
MailboxLogReport               :
DeviceEnableOutboundSMS        : True
DeviceMobileOperator           : AT&T
Identity                       : contoso.com/Exchange users/JSmith
/ExchangeActiveSyncDevices/PocketPC§5ECE2DBB684616DD07FB173DF09254B5
Guid                           : 8fb8848c-8c65-4f43-bb1e-450e582e1622
IsRemoteWipeSupported          : True
Status                         : DeviceOk
StatusNote                     :
DeviceAccessState              : Allowed
DeviceAccessStateReason        : Global
DeviceAccessControlRule        :
DevicePolicyApplied            : Mobile Policy (default)
DevicePolicyApplicationStatus  : AppliedInFull
LastDeviceWipeRequestor        :
DeviceActiveSyncVersion        : 14.0
NumberOfFoldersSynced          : 7
SyncStateUpgradeTime           :
```


INSIDE OUT

It's actually useful data

Some interesting data are revealed here because you can see which devices have synchronized with the mailbox, when they synchronized, and even the mobile operator. These data can provide the basis of some management reports, such as the number of mailboxes that use mobile devices, an analysis of the devices being used, and the distribution of users across mobile operators. You could use these data for multiple purposes such as negotiating a better corporate deal with a mobile operator or planning a replacement strategy for old devices.

The problem with the Get-ActiveSyncDeviceStatistics cmdlet is that it functions on the level of a mailbox, and there's no cmdlet available to provide aggregate data of the type that is

useful for analysis. Clearly you don't want to review synchronization data for thousands of mailboxes to gain some understanding of what's happening on a server, so some code is required to fetch the necessary data and store them in a format that permits analysis.

This code uses the Get-CASMailbox cmdlet to fetch information about any mailbox with an ActiveSync partnership that's connected to an Exchange 2010 mailbox server. The Get-ActiveSyncDeviceStatistics cmdlet then extracts statistics for each device (remember, someone can create partnerships with several mobile devices) and writes out these data to a variable. Eventually, after data have been fetched from all the mailboxes, the aggregated data are written into a CSV format file that can be used for later analysis.

```
$Devices = $Null
$Mbx = Get-CASMailbox –ResultSize Unlimited |
Where {$_.HasActiveSyncDevicePartnership -eq $True -and
$_.ExchangeVersion.ExchangeBuild -ilike "14*"}

ForEach ($m in $Mbx)
{
$Devices += Get-ActiveSyncDeviceStatistics -Mailbox $m.Identity
}

$Devices | Export-CSV ExServer1ActiveSync.csv
```

The code works, but it's really only appropriate for use in small deployments of less than 1,000 mailboxes where you won't run into problems processing a lot of data in memory after it's fetched from mailboxes. Another way of doing much the same thing is to use two separate loops. The first processes the list of mailboxes that have ActiveSync partnerships, and the second fetches information for each device that has synchronized with the mailbox. The resulting data are written out in a less verbose manner into a simple text file. The code has been used to generate monthly management reports in organizations supporting more than 30,000 mobile devices:

```
$Date = Get-Date -uformat "%Y%m%d"
$Logfile = "C:\Logs\ActiveSync-all-$date.txt"

$Lst = Get-CASMailbox -ResultSize Unlimited | Where
{$_.HasActiveSyncDevicePartnership -eq $True}

ForEach ($CASMbx in $lst) {
$Devices=$Null
$Devices= @Get-ActiveSyncDeviceStatistics -Mailbox $CASMbx.name)
ForEach ($device in $devices) {
$DeviceModel = $Device.DeviceModel
$DeviceType = $Device.DeviceType
$LastSyncTime = $Device.LastSuccessSync
$PhoneNumber = $Device.DevicePhonenumber
```

```
$UserAgent = $Device.DeviceUserAgent
Add-Content -path $Logfile "$casmbx.name
|$DeviceModel|$DeviceType|$UserAgent|$LastSyncTime|$PhoneNumber|"

   }
}
```

Whatever approach to reporting you take, the important point is that you capture data that make sense for your organization and use them to build a solid and practical ActiveSync policy for your company.

Blocking types of mobile devices

New mobile devices appear all the time, and users are tempted to buy these devices and then attempt to connect the devices to their mailboxes. Often, these connections occur without the knowledge or the intervention of an administrator. This isn't a problem if everything works and the device connects the first time and continues to synchronize mailbox contents perfectly, but it can become a problem when a user attempts to introduce a new device that doesn't comply with corporate security guidelines or runs an operating system that the help desk isn't able to support. For example, the original ActiveSync implementation on the Palm Pre did not enforce the PIN locking feature, which is a pretty big security issue for many companies, so users were told not to use these devices until Palm fixed the problem. The original Apple iPhone also caused some heartburn for some companies because its implementation of all ActiveSync security features was not as complete as found in Windows Mobile. If you enable support for "nonprovisionable devices" by allowing an open connect policy, essentially you allow any device that supports ActiveSync to connect to Exchange and run the risk that the policies to enforce desired security behavior will never reach the device (or that the device will ignore them altogether).

To solve the problem, Exchange 2010 introduces the Set-ActiveSyncOrganizationSettings cmdlet to allow you to exert more control over what happens when users attempt to synchronize new mobile devices for the first time. In this context, a new mobile device is a device with a type for which Exchange has not defined an access policy. You can opt to block synchronization completely, quarantine the device, or allow it. When a device is quarantined, its information is sent to a set of nominated administrators who can decide whether to allow the device to synchronize or continue to block its access to Exchange. For example, the following code sets the default access level to Blocked and sends a note to the HelpDeskAgents@contoso.com email address any time a user attempts to connect a new device. It's more convenient when this address points to a distribution group as this usually ensures a faster response. The *–UserMailInsert* parameter specifies a text string that Exchange includes in the message sent to the user to inform him that his device has been quarantined.

```
Set-ActiveSyncOrganizationSettings –AdminMailRecipients 'HelpDeskAgents@contoso.com'
–DefaultAccessLevel 'Quarantine' –UserMailInsert 'Device quarantined. Please call the
Help Desk to unblock the device'
```

The Get-ActiveSyncOrganizationSettings cmdlet reveals the current policy for new device connections. From this output, we can see that the default ActiveSync policy for the organization allows any device to connect:

```
Get-ActiveSyncOrganizationSettings
```

```
DefaultAccessLevel          : Allow
UserMailInsert              :
AdminMailRecipients         : {}
Name                        : Mobile Mailbox Settings
OtherWellKnownObjects       : {}
AdminDisplayName            :
ExchangeVersion             : 0.10 (14.0.100.0)
Identity                    : Mobile Mailbox Settings
```

Let's assume that the help desk receives notification that a user has attempted to synchronize with a new device called Whiz-Bang01. The administrators should now make the decision about how to deal with these devices. They can allow, block, or continue to quarantine. In view of the fact that users are already attempting to connect, the decision really lies between block and allow. Once the decision is made, it is implemented with a new ActiveSync device access rule. These rules give Exchange the ability to block devices selectively by device type or model. If we want to allow access, we can do this with an access rule like this:

```
New-ActiveSyncDeviceAccessRule -QueryString 'Whiz-Bang01' -Characteristic DeviceModel
-AccessLevel Allow
```

The question now arises about determining the correct values to use to identify a new mobile device when you create a new ActiveSync access rule. One simple way is to examine the characteristics reported in ECP for an ActiveSync partnership established with a mailbox. Go to the Phone section of ECP, select Mobile Phones, and then select the device that you want to check from the list of known devices that have synchronized with the mailbox. Click the Details icon and you'll see the kind of information shown in Figure 10-38.

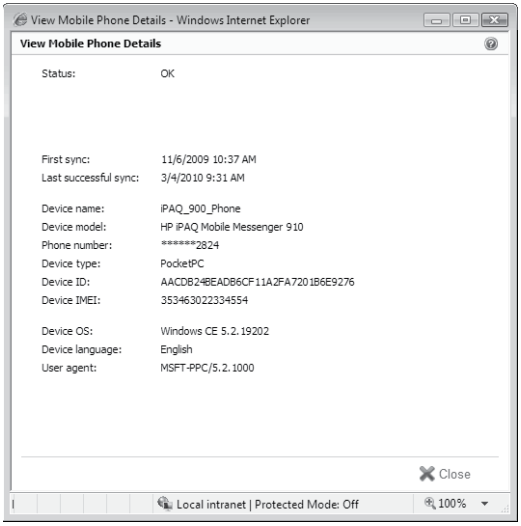


Figure 10-38 Viewing mobile device characteristics reported by ECP.

You can create access rules based on the device type, model, user agent, or operating system. In this example, we see that the characteristics of the device are as follows:

- Device name: iPAQ_900_Phone
- Device model: HP iPAQ Mobile Messenger 910
- Device type: PocketPC
- Device operating system: Windows CE 5.2.19202

Clearly some of these characteristics are broader than others. For example, if you create an access rule that allows any device of type PocketPC, you allow for a huge variety of Pocket PC devices from many different manufacturers.

Another way to discover the characteristics of devices that connect to ActiveSync is to scan the ActiveSync entries in the IIS logs on the CAS server. The entries are pretty obvious. For example, here's one that shows a user connected with a PocketPC device.

```
2010-03-04 17:27:39 16.234.42.1 POST /Microsoft-Server-ActiveSync/default.eas
Cmd=Sync&DeviceId=5ECE2DBB684616DD07FB173DF09254B5&DeviceType=PocketPC
```

And here's an iPhone entry:

```
2010-03-04 00:03:49 16.234.42.1 POST /Microsoft-Server-ActiveSync/default.eas
User=blackadder&DeviceId=Appl87945RZ53NP&DeviceType=iPhone&Cmd=Sync&Log
```

Android phones show up with a device type of Android.

To block iPhones, we could use the following access rule:

```
New-ActiveSyncDeviceAccessRule -QueryString 'iPhone' -Characteristic DeviceType
-AccessLevel Block
```

After creating the necessary ActiveSync device access rules, we can check them with the Get-ActiveSyncDeviceAccessRule cmdlet. For example, this output shows that the organization has three device access rules in place. All Pocket PCs are allowed to connect as are the Whiz-Bang01 devices, but SmartPhones are blocked.

```
Get-ActiveSyncDeviceAccessRule | Format-Table Name, Characteristic, QueryString,
AccessLevel -AutoSize
```

```
Name                          Characteristic     QueryString       AccessLevel
----------------              -------------      ------------      -----------
Whiz-Bang01 (DeviceModel)      DeviceModel       Whiz-Bang01       Allow
PocketPC (DeviceType)          DeviceType        PocketPC          Allow
SmartPhone (DeviceType)        DeviceType        SmartPhone        Block
```

Once a user's device is blocked by a device access rule, the device will not be allowed to synchronize. It might also be the case that a new device access rule causes a device to be temporarily quarantined while Exchange awaits a human's decision on whether to permit the device to synchronize. In either case, the user will receive an email to tell her what the problem is and what she should do. The information contained in the message will help an administrator to understand what the issue is if the user seeks help. Of course, the user will have to read the message using another client and might decide not to contact the administrator if he realizes that he's using an unapproved device.

> **Note**
>
> **Temporary device quarantine can take up to 90 minutes to resolve for servers running the original version of Exchange 2010; this period is reduced to about 15 minutes for SP1 servers.**

An example of text that a user might receive during a period of temporary quarantine is as follows:

```
Your phone can't synchronize with the server via Exchange ActiveSync until it's
identified and its compliance with the access policies is verified.
You may see synchronization errors on your phone while your phone is being
recognized. If you see this sort of error, select mail as the only content to
synchronize with Exchange and start synchronization from your mobile phone.

Information about your mobile phone:

Device type                    : iPhone
Device ID                      : Appl5K2373BX7TR
Device user agent              : Apple-iPhone2C1/801.26000002
Exchange ActiveSync version    : 14.0
Device access state            : DeviceDiscovery
Device access state reason     : DeviceRule

Sent at 5/10/2010 11:18:53 AM to TRedmond@contoso.com.
```

Exchange 2010 SP1 allows you to manage ActiveSync device access rules through ECP. Figure 10-39 shows the set of rules that we've been working on together with an additional rule to block iPhone access. Apple iPhones have taken an increasing share of the mobile device market since they were first released in 2007. I happen to like the iPhone and use one on a daily basis, so there's no bias against the device here. Apple licenses ActiveSync, which means that you can readily connect an iPhone to Exchange. However, the exact details of the implementation of ActiveSync on a device are entirely the prerogative of the device manufacturer, and the Apple implementation for the iPhone does not create quite as secure an infrastructure as can be attained with Windows Mobile devices, which is why some companies choose to block iPhone access.

In mitigation, Apple offers its Enterprise Deployment Guide to help companies deploy iPhones, and there are other sources that you can consult to find out how companies are managing iPhones. For example, Exchange MVP Jeff Guillet has written an excellent discussion of the strategy used to achieve secure iPhone connectivity to Exchange in one project; see *http://www.expta.com/2010/02/how-to-securely-deploy-iphones-with.html*. There are other examples in blogs and Web sites that you can review to determine whether any of the suggestions and tactics described offer some value to your project.

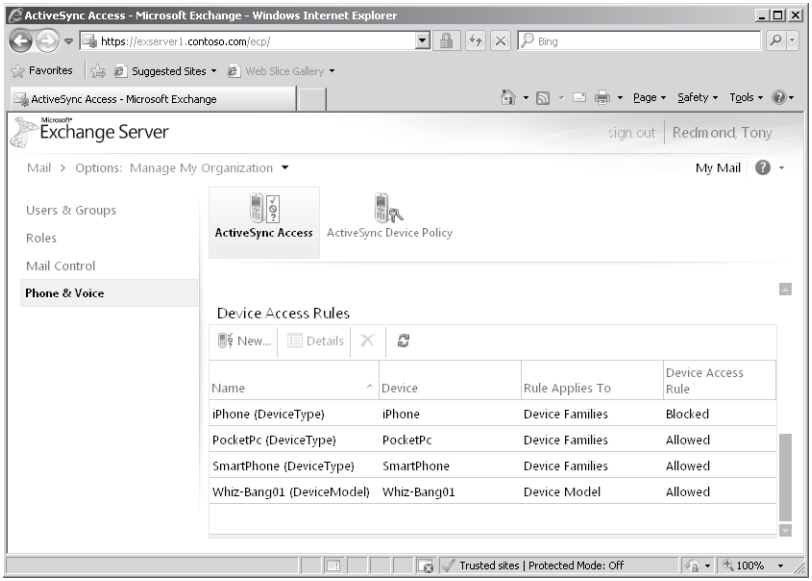


Figure 10-39 Viewing ActiveSync device access rules with ECP.

Blocking devices on a per-user basis

In addition to creating device access rules to control the types of devices that can be connected to ActiveSync, you can use the Set-CASMailbox cmdlet to set the *–ActiveSyncAllowedDeviceIDs* parameter with a list of device identifiers that are allowed to connect to a mailbox. The default value for this parameter is null, meaning that any device can synchronize with a mailbox.

The first step in blocking devices on a per-user basis is to determine the device identifier. The easiest way to discover a device's identifier is to connect it to Exchange. Afterward, you can use the Get-ActiveSyncDeviceStatistics cmdlet to retrieve details about users' ActiveSync activity, including the identifiers for each mobile device that they have connected to Exchange. For example:

```
Get-ActiveSyncDeviceStatistics –Mailbox 'Pelton, David'
```

You can add multiple device identifiers to the list, separating each identifier with a semi-colon. For example, this command allows just one specific device to synchronize with John Smith's mailbox.

```
Set-CASMailbox -Identity 'Pelton, David' –ActiveSyncAllowedDeviceIDs
'4B9207650054671AD0AEE83A424BCD7F'
```

To clear the device identifier to allow any device to connect to the mailbox:

```
Set-CASMailbox –Identity 'Pelton, David' –ActiveSyncAllowedDeviceIDs $Null
```

If we have a list of devices and only want to remove a single device from the list, we can do it by exporting the list to a variable, updating the list in the variable, and then writing it back with Set-CASMailbox:

```
$Devices = Get-CASMailbox –Identity 'Pelton, David'
$Devices.ActiveSyncAllowedDeviceIDs -= '4B9207650054671AD0AEE83A424BCD7F'
Set-CASMailbox –Identity 'Pelton, David' –ActiveSyncAllowedDeviceIDs
$Devices.ActiveSyncAllowedDeviceIDs
```

The same techniques can be used to block devices. In this case, we update the *–ActiveSyncBlockedDeviceIds* parameter with Set-CASMailbox. For example, you might make a corporate decision to block Android-powered devices and then discover that someone is using one of these devices to access Exchange. A quick retrieval of the device identifier followed by input to Set-CASMailbox will block further synchronization.

Wiping lost devices

It is the nature of mobile devices that some will be lost in airports, taxis, shops, and other places. In the same way, it's almost inevitable that some devices will never be recovered. Being able to wipe the device through an over-the-air command is therefore a necessity to protect the data held on the device. Administrators can wipe a mobile device by selecting the mailbox in EMC and then selecting the Manage Mobile Device option or by running the Clear-ActiveSyncDevice cmdlet. Users can wipe a device by selecting it through the Mobile Phones option of ECP (Figure 10-40). Naturally, these options are only available if a user has first synchronized a device with her mailbox to make the device known to Exchange.

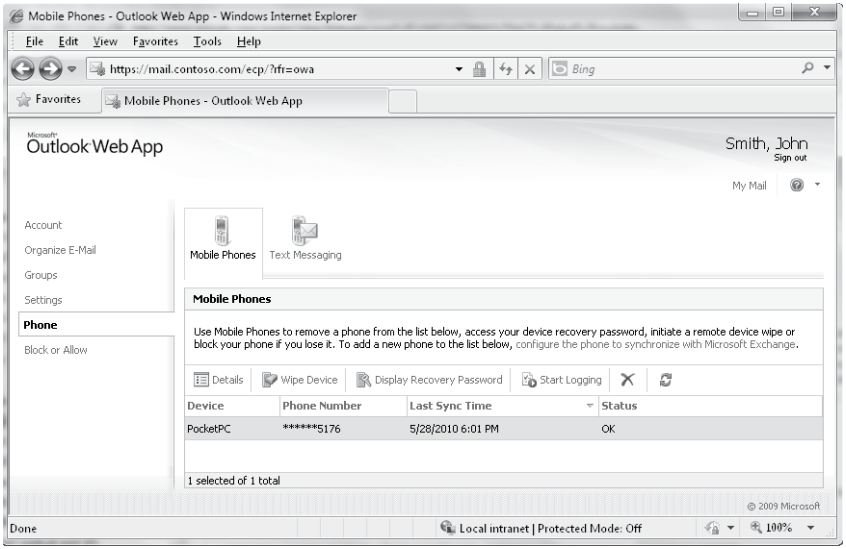


Figure 10-40 ECP option to wipe a mobile device.

When a remote wipe is initiated by an administrator or user, ActiveSync sends a wipe command to the device, which then executes the appropriate command locally. The client then acknowledges the wipe command back to the server together with an indication of success or failure. The following steps happen:

1. Client connects to ActiveSync.
2. Client provides its DeviceID.
3. Exchange queries device for account credentials.
4. Client authenticates with username and password.
5. Exchange checks its ActiveSync block lists to see whether DeviceID is allowed to connect (other things happen at this stage, including a check to see whether the policy that applies to the device has to be refreshed).
6. Exchange checks whether a wipe command is queued for the device and issues command if found.
7. Device acknowledges wipe command and reports status.

Note that if you send a remote wipe request to a device that doesn't support the wipe function, the device will not be able to execute the request and the data will remain. However, synchronization of future data will fail, so you can at least prevent any more sensitive data going to the device. The different degrees of support offered by various device types for remote wipe functionality are a good reason for you to test this feature before approving a device type for deployment. ActiveSync issues a confirmation message when a device acknowledges a wipe request. If a user issues the command through OWA, he receives the confirmation message, and if an administrator issues the request, the administrator and the user associated with the device both receive confirmation messages.

CAUTION!

It's important to realize that a remote wipe command will not erase data on any storage card in the mobile device—the only data removed by a "wipe device" command is that known to ActiveSync. Another thing to think about is that a device must authenticate before Exchange is able to send it a wipe command. In other words, there's no way that you can wipe a device that is stolen unless the thief attempts to connect to Exchange to download new mail. This is one reason why strong passwords and maybe even encryption should be used to protect sensitive corporate information that's stored on mobile devices.

The fact that a device must authenticate before any ActiveSync communication concerning policies (including device wipe commands) is possible introduces the issue of how to deal with people who leave the company. Most companies have well-developed procedures that the IT department uses when they are notified that someone is leaving, and the usual first step that IT takes is to disable the employee's account or change her password so that she can no longer log on to any corporate system. This is a good way to protect confidential information that's held in corporate systems, but it does nothing to remove the information that the employee has on her mobile device. The departure process might require the employee to hand in her mobile device on the day that her employment finishes, but what happens if the device is owned by the employee? The immediate answer is that you should issue a wipe command to remove at least all the email from the device, but this step is impossible if you disable the account, because the device will never be able to authenticate. The lesson here, therefore, is that you need to issue the wipe command and make sure that it is acknowledged before the employee's account is disabled.

Debugging ActiveSync

Understanding what's happening as a mobile device synchronizes with an Exchange mailbox is simple in concept and complex in execution. Much can go wrong between the point where a message is created in Outlook Mobile and the time it is submitted for sending on the server. When problems arise and the user calls the help desk, an administrator can work though the basic problem-solving steps with the user to ensure that the connection is working and nothing else obvious is awry. After that, it can be a challenge to understand what might be going wrong.

To address the problem, Exchange maintains ActiveSync logs to capture details of the interaction between mobile device and Exchange. The contents of these logs describe the events that occur as the device connects and retrieves information. Logging is off by default, so if a problem occurs with a device, the user must click Start Logging in the Mobile Phones options to instruct Exchange to begin to capture the events. Once he has worked through the steps to re-create the problem, the user can stop logging and Exchange sends him the ActiveSync log as an attachment to a message (Figure 10-41).

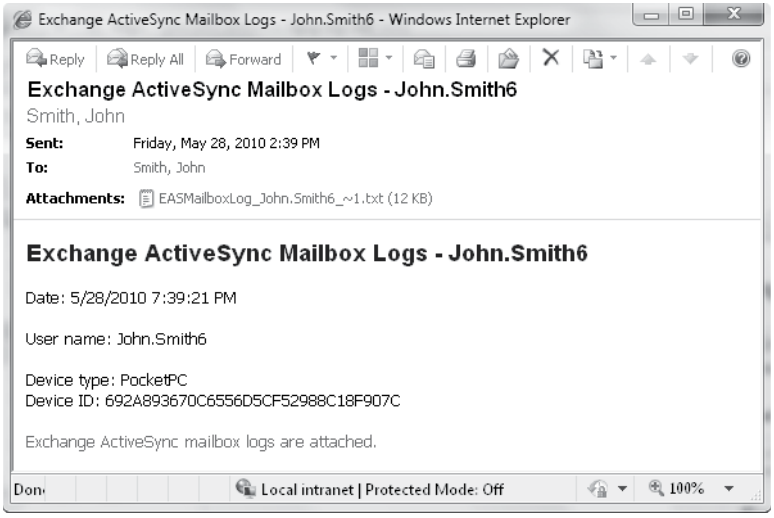


Figure 10-41 Exchange delivers an ActiveSync mailbox log for a mobile device.

ActiveSync logs are in XML format (Figure 10-42), and their contents take some time to interpret. However, these logs are not really designed to be used by administrators. Instead, they are a diagnostic tool for Microsoft support personnel who have access to all of the coded information that might appear in a log and who can therefore figure out what happened between a device and Exchange during a connection.

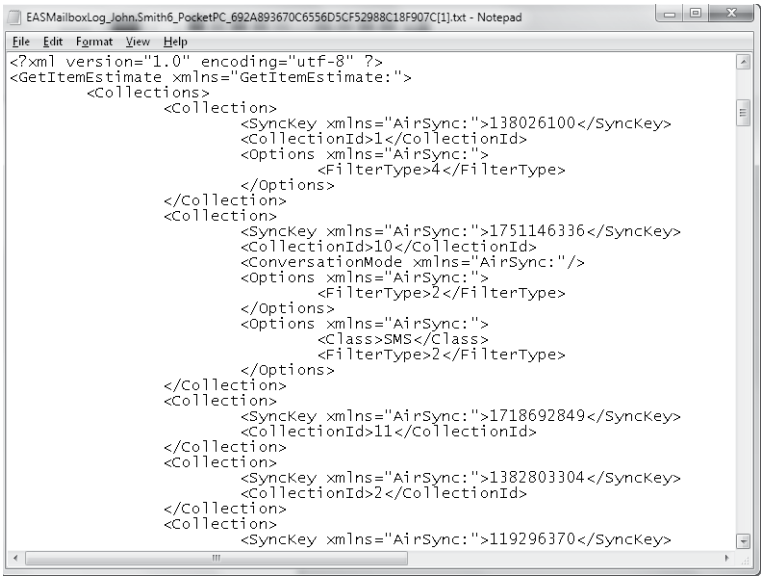


Figure 10-42 Viewing an ActiveSync mailbox log.

Testing mobile connectivity

Microsoft provides a set of emulator images for Windows Mobile 6.1 and Windows Mobile 6.5 devices that you can download to test connectivity for mobile devices (search for "Windows Mobile Emulator Images"). An emulator is not yet available for Windows Phone 7 (released to manufacturing in September 2010), but should appear in due course. These emulators can be loaded into Microsoft Virtual PC and connected to your network to test that ActiveSync works as expected. If you run a large ActiveSync deployment, it's a good idea to set up these emulators to test against new roll-up releases and service packs of Exchange to ensure that no problem is introduced with the new server software.

ActiveSync for BlackBerry

The traditional method to connect BlackBerry mobile devices to Exchange is to deploy RIM's BES. Although there is no doubt that BES works, it is usually expensive to license and requires a separate server infrastructure. In addition, BES defaults to using MAPI to access Exchange mailboxes and directory information and can generate a significant load on a mailbox server when it connects to fetch and send messages. In some scenarios, the load generated by a BlackBerry user can be three times or more that of an Outlook user. (Newer versions of BES support using Exchange Web Services instead, which helps performance at the cost of a bit more setup work for the BES administrator.)

Technology changes all the time, and solutions are now available (for example, *www.astrasync.com*) that use the ActiveSync protocol to connect BlackBerry devices to Exchange. ActiveSync is included in Exchange, so there is no need to license additional BES software or to deploy more hardware. The performance profile of an ActiveSync client is less than a BlackBerry, so if performance is an issue, a transition to ActiveSync might address the problem. Another advantage is that replacing BES with ActiveSync removes complexity from the infrastructure because all mobile devices will then use a single protocol that can be managed through Exchange. For example, you can enable or disable a mailbox to use ActiveSync through EMS or EMC, whereas a separate management utility is required for BES. The potential savings are somewhat offset by the need to buy and deploy ActiveSync client software, but these costs are usually dwarfed by the savings.

INSIDE OUT **Do the cost–benefit analysis for an upgrade**

No one should move away from a working solution without good reason. Administrators who know how BES works and have the necessary skills to manage the full end-to-end client-to-server communications, perhaps using tools such as Zenprise's MobileManager, might be unwilling to consider a move when faced with the challenge of upgrading to Exchange 2010. However, given that a new version of BES must be deployed to support Exchange 2010, and most users refresh their mobile devices regularly, it is a good idea to use the opportunity presented by an Exchange 2010 upgrade project to run a cost–benefit exercise that captures all of the pros and cons of a transition to ActiveSync.

Client throttling

Clients can occasionally create an excessive load on an Exchange server. The reasons this happens are many and varied but usually involve some form of software bug that causes the client to communicate in an unpredictable manner and so creates an out-of-the-ordinary load. The usual corrective action taken in previous versions is to first identify the errant client with the Exchange User Monitor (ExMon) utility and then terminate its connection to relieve the strain on the server and restore normal levels of responsiveness to other clients. You can then figure out what action the client was taking to cause the problem and resolve the situation.

ExMon is one of those invaluable utilities developed by Microsoft engineers that should really be included in the formal product as one of the utilities presented in the Toolbox. For now, you have to download it from Microsoft's Web site. ExMon works against

Exchange 2003, Exchange 2007, and Exchange 2010 servers and allows administrators to view details of the clients that are currently connected to the server, including IP address and software versions.

Exchange 2010 introduces the concept of client throttling to allow administrators to take a more proactive approach to resource management by defining and applying policies to users to control the resources that they can consume for these categories:

- ActiveSync (EAS)
- RPC Client Access(RCA; used for MAPI clients such as Outlook)
- Outlook Web App (OWA)
- POP3
- IMAP4
- Exchange Web Services (EWS; this category includes Unified Messaging users and users running Entourage or Outlook for Mac OS X)
- Windows PowerShell

A default policy is automatically created and enforced within the organization when you install Exchange 2010. The policy comes into effect when the percentage of CPU utilization by Exchange exceeds the threshold defined in the *CPUStartPercent* property of the default policy. This setting is applied on a per-service basis. The default value for *CPUStartPercent* is 75, so when one of the Exchange services monitored for client throttling reaches this threshold, Exchange begins to apply any throttling restrictions that are defined in the default policy or on a per-mailbox basis to ensure that the server can continue to provide a reasonably smooth service to all clients.

Throttling policies can only be managed through EMS. You can view details of the default policy with this command:

```
Get-ThrottlingPolicy | Where {$_.IsDefault -eq $True} | Format-Table
```

A lot of data are output when you examine the attributes of a throttling policy. However, you can break them down into the categories listed earlier. The first six categories correspond to client protocols and are identified with the prefix shown in parentheses. Thus, we can retrieve the settings that govern MAPI clients with:

```
Get-ThrottlingPolicy | Select RCA* | Format-List
```

```
RCAMaxConcurrency                 : 20
RCAPercentTimeInAD                :
```

```
RCAPercentTimeInCAS                :
RCAPercentTimeInMailboxRPC         :
```

The output for the RCA parameters indicates that only one threshold currently is in place to control user workload within the RCA layer: The maximum concurrency for any user is set to 20 (the range is from 0 to 100), meaning that a user can have up to 20 active sessions with a CAS. A connection is maintained from the time a request is made to establish it until the connection is closed or otherwise disconnected by a user action (logging off). If a user attempts to establish more than the allowed maximum, that connection attempt will fail. The other limits are set to null, indicating that no limit is in place. Essentially, this means that a user can continue to use these resources until they are exhausted.

These settings control the amount of time a client can use to execute LDAP requests against Active Directory, run CAS code, and execute mailbox RPC requests. The values can range from 0 to 100 and represent the percentage of a one-minute window that a client can spend in the mode. It is possible to exceed 100 percent because a client can issue concurrent requests, each of which makes a heavy demand. The cumulative load would force Exchange to throttle the load for this client. Because mailbox RPC and LDAP requests flow through the CAS and therefore consume CAS resources, it follows that the CAS setting (*PercentTimeinCAS*) is an overlapping superset of the mailbox and Active Directory settings, so the value for the CAS setting should always be larger than the mailbox and Active Directory settings.

Similar groups of settings are available for the other client categories. For example, you can find those applying to Exchange Web Services with:

```
Get-ThrottlingPolicy | Select EWS* | Format-List
```


TROUBLESHOOTING

Exchange is throttling BES activities

Introducing client throttling had an unfortunate side effect on some applications that impose heavy demands on Exchange. The BES provided the best example, because the account that it uses essentially mimics a hyperactive user that accesses multiple mailboxes to fetch and send messages to mobile devices. The usual problem was that Exchange throttled BES activities because it exceeded the RCA maximum concurrency threshold. The solution was to create a new throttling policy that set the value of the *–RCAMaxConcurrency* parameter to $Null and then assign the new policy to the BES account. This is a step that the administrator can perform after installing BES.

A number of specific parameters are available to control workload generated through Windows PowerShell:

- *–PowerShellMaxConcurrency* (default value 18): This constraint is applied in two different ways. It defines the maximum number of remote Windows PowerShell sessions that a user can have open on a server at one time. It also defines the maximum number of cmdlets that EMS can execute concurrently.
- *–PowerShellMaxCmdlets* (default no limit): Sets the number of cmdlets that a user can execute within the time period specified by *–PowerShellMaxCmdletsTimePeriod*. After the value is exceeded, no future cmdlets can be run until the period expires.
- *–PowerShellMaxCmdletsTimePeriod* (default no limit): The period in seconds that Exchange uses to determine whether the maximum number of cmdlets constraint has been exceeded.
- *–ExchangeMaxCmdlets* (default no limit): Specifies the number of cmdlets that a user can execute within the time period set by *–PowerShellMaxCmdletsTimePeriod*. After the constraint is exceeded, Exchange slows down the execution of other cmdlets.
- *–PowerShellMaxCmdletQueueDepth* (default no limit): Specifies the number of operations that Exchange will allow a user to execute. Operations are consumed by cmdlets as they run. They are also consumed by internal operations (for example, the *–PowerShellMaxCurrency* operation uses two operations). Microsoft recommends that, if set, the value of *–PowerShellMaxCmdletQueueDepth* is set to three times the value of *–PowerShellMaxConcurrency*. Exchange does not apply this constraint to the code run by ECP or EWS.

Three additional settings can be used to constrain the consumption of general resources:

- *–MessageRateLimit* (default no limit): Governs the number of messages per minute that a user can submit to the transport system for processing. Messages over the limit are placed in the user's Outbox until the server is able to accept them. The exception is for clients such as POP3 and IMAP4 that submit directly to the transport system using SMTP. If these clients attempt to submit too many messages, their request is declined and they will be forced to reattempt later.
- *–RecipientRateLimit* (default no limit): Specifies the number of recipients that can be addressed in a 24-hour period. For example, if this value is set to 1,000, it means that the user is allowed to address messages to up to 1,000 recipients daily. Messages that exceed this limit are rejected.
- *–ForwardeeLimit* (default no limit): Specifies a limit for the number of recipients that can be configured in Inbox Rules for the forward or redirect action.

INSIDE OUT **Storing the default throttling identifier in a variable**

You'll note that the default throttling policy has a value like *DefaultThrottlingPolicy_dade6c60-e9cc-4692-bc6a-71771158a82f* given to its name and identifier. I suspect that this is a joke played on us by the Microsoft engineers, because no sensible human being could think that such a name is understandable. If you plan to work with a policy, you might want to store the identifier in a variable so that you can use it to refer to the policy that you want to work with. For example:

```
$TP = (Get-ThrottlingPolicy).Identifier
Set-ThrottlingPolicy –Identity $TP -EWSPercentTimeInCAS 80
```

If you create a new policy with the New-ThrottlingPolicy cmdlet, the values from the default policy are inherited. All you have to do is to state values for the settings that you want to change. Thus, we can do:

```
New-ThrottlingPolicy –Name 'Restricted CAS Access' –RCAMaxConcurrency 10
```

To apply the new policy, we can either make it the default:

```
Set-ThrottlingPolicy –Identity 'Restricted CAS Access' –IsDefault $True
```

Or, we can apply it selectively to users:

```
Set-Mailbox –Identity 'David Jones' –ThrottlingPolicy 'Restricted CAS Access'
```

Exchange 2010 SP1 improves the way that client throttling works based on feedback from production deployments. In RTM, if a client is throttled, the result is one or more failed requests for server data, which can lead to a bad user experience because something will not work properly or as expected. In SP1, requests that the server throttles are added to a queue to delay but not fail processing. The requests are backed off by being forced to wait for a few milliseconds before the server attempts to process them again. Hopefully, when the wait time elapses, the need to throttle will have passed and the client requests can be processed normally.

Unified Messaging

UM is a topic that deserves a complete book in its own right to cover the many issues that have to be considered in implementing Exchange-based voice mail as a replacement for a traditional voice mail system that is tightly associated with a private branch exchange (PBX). A great deal of negotiation and coordination normally has to occur between the teams responsible for telecommunications and messaging within the company to ensure that

issues such as dial plans are addressed. Companies such as Nortel and Cisco had offered voice mail integration for Exchange in previous versions, but Exchange 2007 marked Microsoft's first integrated version purpose-built to work with Exchange, Outlook, and OWA and to fit into an overall Unified Communications strategy. Microsoft's software-driven approach is different from the hardware-centric approach taken by traditional telecommunications vendors and is centered on TCP/IP communications and open standards. The result is lower cost and a more open platform than has been possible for voice mail systems in the past.

Microsoft has upgraded many aspects of UM in Exchange 2010 to take into account customer experience in deploying Exchange 2007–based UM. Two major advances are also incorporated. First, Exchange 2007 Unified Messaging supports three different audio codecs: WMA, GSM 06.10, and GSM G.711. You could pick a codec for individual users, but none of the codecs were a great solution for non-Windows Mobile devices. Exchange 2010 updates the range of audio codecs to support MP3 and uses MP3 as the default codec for voice messages. The idea behind the change is that MP3 is supported by so many different devices and applications that using MP3 as the default will improve the ability of users to work with voice messages across multiple devices.

The second major enhancement in Exchange 2010 is the introduction of voice mail preview. This is an exciting attempt to make voice content more accessible to users when they cannot replay a message. The challenge of transcribing voice messages into text poses challenges in computer science, language, and culture. It is therefore worthy of comment.

Voice mail preview

Voice mail preview is the ability to transcribe voice messages to text so that the messages can be read on a screen. This feature is especially useful to read voice messages on a mobile device to understand whether you have to respond to the sender quickly. The only problem is that the transcription algorithms sometimes generate text that doesn't quite convey the meaning or intent of the message. All systems that integrate voice face the challenge of comprehending the meaning of voices that share a common language but use different tones, accents, and word patterns, and even mix in words or phrases from other languages. Names can often be a particular challenge, especially if they are "foreign" to the expected language. The code does attempt to clean up the spoken word by eliminating pauses and terms like "ugh" that people often use when they are speaking.

Clearly there's no point in shipping voice preview if every second word is not recognized. However, different languages vary in how recognizable machine-generated text is in terms of the recipient being able to understand the nature and content of the message. English is full of slang; English terms and slang are finding their way into daily use for more languages, so moving the spoken form of a language from a point where 50 percent of words are recognized by automated transcribers (at this level a somewhat accurate version of

the message is generated) to 80 percent or higher (where an almost accurate message is generated) relies heavily on testing the results of real-life messages that users attempt to understand. User perception of accuracy varies with the usability of the text generated by the computer; if the text is sufficiently accurate for the user to understand the meaning and importance of its content, the user will probably think that the transcription is pretty accurate. On the other hand, once important words are dropped or mangled, user perception diminishes and even an 85 percent accuracy rate (measured in the number of words that are accurately transcribed) might be considered utterly unusable.

> **Note**
>
> **The word "preview" in the feature name is important; it is unreasonable for a general-purpose email system to provide an absolutely accurate rendition of message contents. Voice mail preview provides a quick way for a user to understand the gist of a message without having to listen to the whole thing.**

It's no excuse for technology, but it is true that it is often difficult for humans to understand the content of a voice mail received from someone that has a strange accent or who uses unfamiliar slang or technical terms—imagine how difficult it must be to build code to interpret exactly what someone says. Another difficulty is that there's no way that a general-purpose computer running Windows can determine the language used by the person who leaves a voice mail. An assumption can be made, but it might not be accurate.

Before a computer has any chance of understanding what a human has said in a voice message, a huge amount of analysis and interpretation has to be performed against a vast number of actual messages to comprehend how human beings communicate with each other. The more messages and the more topics that these messages cover, the higher the accuracy of the voice mail preview will be. Getting the required volume and variety of messages to be able to transcribe the contents and use them to validate the computer-generated output is the key limiting factor that prevented Microsoft from supporting more languages for voice preview. A terrific amount of testing and validation must occur before voice preview can support a language. The following factors influence Microsoft's decision to ship a language:

- Has sufficient testing been done to assure a high rate of accuracy for voice transcription?
- How usable is the text generated in the voice preview?
- How sensitive is a culture or language to machine transcription?

The following languages are supported for voice mail preview in Exchange 2010 RTM:

- English (U.S. English and Canadian English, but not Australian English or other local dialects)
- French
- Italian
- Portuguese
- Polish

Exchange 2010 SP1 adds Spanish (as spoken in Spain rather than in Latin America).

Assuming that your language is on the supported list for voice mail preview, you will probably find that the algorithm used by Exchange to interpret voice messages generates text with a reasonably high degree of fidelity. Your experience will vary from the high 90 percentile accuracy downward to much lower, depending on how clearly users speak when they leave voice mail, the devices that they use, whether there's wind or other noise in the background, and a myriad of other influences. Sometimes the text generated is comical, but most of the time it's acceptable even if a couple of errors creep in. No algorithm can process Klingon or other esoteric languages, but you will get good results if you guide people to be as clear as they can when they leave voice mail. Of course, if people ignore the advice, you can run a competition to find the funniest voice mail preview generated by Exchange. If you really cannot cope with the feature, UM-enabled users can turn off voice mail preview through the options available through OWA.

Voice mail previews are generated as messages arrive. Creating the preview at this point instead of later on through background processing delivers a number of advantages. The preview is immediately available when a message arrives in a user mailbox and can be viewed through a number of interfaces, including mobile devices, to allow users to decide whether the message is important. Microsoft believes that some 90 percent of the value provided by previews is delivered because the preview is immediately available. Because the preview becomes part of the inbound message, its text can be used in notifications and processed by rules. Background processing to locate voice mail and generate previews would work, but this would impose a huge additional load on mailbox servers.

The computational load to generate voice mail previews in real time forces some compromises in how transcription occurs. Transcription is a CPU-intense activity that requires roughly one second of CPU core processing for every second of spoken content. To stop an individual message from soaking up excessive processor resources, Exchange won't attempt to transcribe a message that is longer than 75 seconds. It therefore pays for senders to keep their messages brief and to the point.

Taking the rule of one second of core processing per second of voice mail, we can therefore calculate that a four-core server is capable of transcribing four 60-second messages per minute over a sustained period. During peak periods of user activity during the day, it is likely that more voice mails will arrive for processing than Exchange can transcribe to create previews, so Exchange throttles back its transcription activities whenever a UM server comes under heavy load to avoid the creation of backlogs and to ensure that users get their voice mail as quickly as possible. For example, Microsoft IT runs four UM servers that have 24 cores among them. During one five-day week, the Exchange team found that some 236,269 voice mails were processed by the servers. Taking an eight-hour working day, this works out at about 100 incoming voice mails per minute or around four messages for each of the available cores. Within Microsoft, an average voice mail is around 30 seconds, so only two of these can be transcribed, leaving two to be delivered without preview. Of course, these are average values, and all voice mails are delivered complete with preview during some hours, whereas fewer than half will arrive with preview at peak times.

The settings used to control transcription are contained in the MSExchangeUM.config configuration file, where you'll find values such as *TranscriptionMaximumMessageLength* (the longest message that UM will transcribe) and *TranscriptionMaximumBacklogPerCore* (the maximum number of messages that Exchange will allow to be queued for transcription; the default is 5). You could update the values in the configuration file to alter the way that UM works, but you should understand the following:

- Microsoft doesn't support changes, so if you do alter the values and encounter problems, you'll have to revert to the original values to see if the problem still occurs before you can call Microsoft for support.
- There is no replication of configuration updates between UM servers, so any changes have to be applied manually to all UM servers.
- The next upgrade of Exchange is likely to overwrite the changed values.
- The changes might affect the performance of the UM servers in unpredictable ways (see the previous discussion).
- Microsoft has done a lot of testing to determine the optimal settings. For example, a typical voice mail lasts between 25 and 30 seconds, so one that lasts 75 seconds is likely to be rambling, possibly incoherent, and probably includes some non-business-critical information such as an enquiry about the health of the recipient's partner. Do you therefore need to encourage users to waste computer resources to process and store verbal outpourings of no great import?

Some companies won't care that some messages arrive without previews. If you're in this category, you can deploy sufficient capacity to handle the average UM load. On the other hand, if you want to ensure that every voice mail arrives with a preview, you have to deploy sufficient processor capacity to handle the expected peak load plus a percentage to allow for higher peaks and growth. The computer-intense nature of transcription makes Exchange 2010 UM servers a poor choice for virtualization, which is why Microsoft recommends that you use standard computers for these servers.

After transcription, Exchange stores voice mail preview content as an XML property for the message. The XML content contains information such as timing (the length of the voice mail), the confidence level for the transcription (how accurate Exchange thinks the preview content is), and phone numbers. If Exchange can recognize the sender, it captures information about the sender for Outlook to display.

Figure 10-43 shows a good example of how voice mail preview appears for a user. This shows a real message that I left for a co-worker after listening to a call run by the Exchange development team to discuss some technical background about voice mail transcription. I attempted to convey some important points about the meeting, but some of the words were lost in the transcription, possibly due to my accent (an Irish accent moderated by spending a lot of time in the United States is a real challenge for the U.S. English transcriber). For example, "Unified Messaging people at Microsoft" becomes "you know five messaging people at Microsoft," which isn't quite what I meant. Nevertheless, the text is good enough to attract the attention of the reader and perhaps get him to call me back, which is what I want to accomplish as the caller.

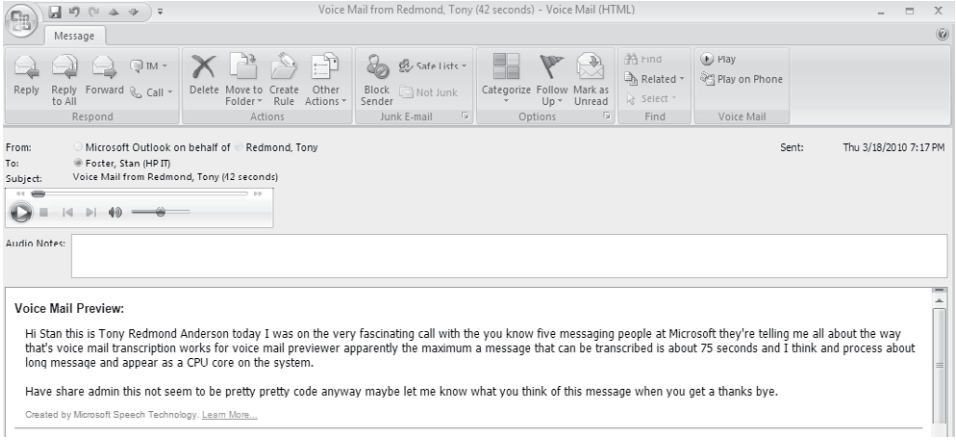


Figure 10-43 Voice mail preview in action.

Fax integration

Unlike Exchange 2007, there is no support for the ability to create new messages from incoming fax calls in Exchange 2010, because Microsoft determined that this was a feature that wasn't used by many customers and so decided to leave this area of functionality to third parties. If Exchange 2007 UM was used, its fax configuration properties are retained and the server is still capable of responding to fax tones for incoming calls. If a fax tone is detected, Exchange looks for the value of the *FaxServerURI* property on the UM Mailbox policy to determine whether a third-party fax solution is available. This value provides a path to the third-party solution and allows Exchange UM to hand off the call in progress to that product, which is then responsible for establishing a session with the sender, creating a new fax message, and sending the message to the user's mailbox. Fax messages in the Inbox still look the same as they did in Exchange 2007 UM; the sole difference is that Exchange UM has no part in their creation or processing. The bottom line, therefore, is that customers who want to integrate fax into Exchange 2010 UM (for example, by allowing a single number to be used for incoming fax and voice calls) need to buy and integrate a suitable third-party solution.

Exchange 2010 APIs

Exchange's history with APIs is checkered at best and poor at worst. APIs have appeared in one release and disappeared in the next, and developers have been unable to depend on an API lasting more than a couple of releases, which is not a good basis on which to plan for a project that wishes to leverage Exchange data. MAPI is the ever-present API in both server and client versions, but it's a difficult API to master as the documentation is sparse and most of the expertise exists within Microsoft. Other APIs such as Web Distributed Authoring and Versioning (WebDAV) were hailed with great anticipation as the "next great thing" when they appeared and have slowly decreased in importance, and others such as Exchange Routing Objects were launched, were rejected by the development community, and sank without trace.

Having inconsistent or incomplete APIs after so many releases is an embarrassment for Exchange—especially when email is the foundation of a collaboration platform—and Lotus Domino, Exchange's major rival, has always excelled in this domain. Exchange 2010 sets out a new beginning that focuses on three APIs. The hope is that these represent the future development platform for the product. Time will tell. Table 10-4 lists the major APIs used by recent Exchange versions and the option presented by Exchange 2010. In most cases, the focus is firmly on EWS, a Simple Object Access Protocol (SOAP)-based interface. Note that some of these APIs (like WebDAV) do not exist in the Exchange 2010 code base, whereas others are outside the direct control of Exchange (Windows Management Instrumentation [WMI]) and will persist for a while longer. Of course, anything that Microsoft decides to deprecate is a bad choice for you to invest in as a development option.

Table 10-4 Old Exchange APIs and Exchange 2010 options

API	Used for	Exchange 2010 option
CDOEX	Mailbox access	EWS
WebDAV	Remote Mailbox access	EWS
ExOLEDB	Mailbox access	EWS
OWAURLs	Free/busy access and name resolution	EWS
Store events	Asynchronous and synchronous events	Transport delivery events and EWS
WMI	Management	Windows PowerShell
Collaboration Data Objects (CDO 1.2.1)	Access to Outlook objects through a COM-based interface to MAPI	Will continue to work

In addition, Microsoft has announced that Collaboration Data Objects (CDO) version 1.2.x is going into extended support, meaning that it is now in life support and is no longer a good option for future projects.

You might ask why Microsoft is changing the API landscape again. There is some logic in asserting that the product has changed dramatically in Exchange 2007 and Exchange 2010 and it's time to refocus on a set of APIs that developers can use with confidence. Microsoft wants developers to leverage the .NET Framework, and some of the older APIs (CDOEX, ExOLEDB, MAPI) are difficult to use from managed code. Some of these APIs also depend on redundant code that Microsoft wanted to remove from the code base to reduce maintenance costs and the possibility that a security breach might arise through old code that's subjected to a new style of attack. In addition, Microsoft began a fundamental shift away from monolithic product architecture in Exchange 2007 by encapsulating business logic in a set of Windows PowerShell commands that are called from multiple interfaces. This process continues in Exchange 2010 to create a more scalable and robust product, so it's logical that older interfaces that don't leverage the common set of business logic are removed.

Exchange Web Services

Apart from Windows PowerShell, the big focus for most developers is now on EWS, which was first introduced in Exchange 2007. EWS is an API that is hosted on the CAS server role and exposed as a Windows Communication Foundation (WCF) Web service. Platforms that cannot use the managed WCF API can use the raw SOAP-based functions, which is how EWS is used for platforms such as the Apple Macintosh, iPhone/iPad, and Linux.

You can download a copy of EWS from the Microsoft Web site, and its license allows third-party developers to distribute EWS along with their own code. However, you should resist the temptation to install EWS into the Global Assembly Cache (GAC). Because Microsoft allows developers to include EWS with their code, the danger of installing any particular copy of EWS into the GAC is that you might create a condition similar to "DLL hell" when a function in the version of EWS in the GAC doesn't support a third-party application. You can download a copy of EWS from MSDN.

EWS allows developers to create code to leverage Exchange core functionality (create and send messages, update mailboxes, delete items, and perform searches) while working with a complete set of Store items from messages to appointments to tasks. EWS can work with the GAL, expand distribution lists, access free and busy data, and handle delegate access to mailboxes. It can access search and public folders and manipulate the permissions on folders. The Exchange 2010 version of EWS can access the dumpster and deal with attachments and personal distribution lists, and it can even impersonate users for role-based access control (RBAC) purposes. Windows developers use EWS through a client-side .NET API that provides object-oriented access to the items and structures that they need to deal with. Because EWS is "fully baked" in Exchange 2010, Microsoft has moved away from WebDAV, the API that underpinned recent versions of their Entourage client for Exchange so that the latest version of Entourage is now totally based on EWS.

If you're looking for a good example of how to approach the application of EWS, the simple Exchange mailbox client built by Exchange MVP Glen Scales (*http://gsexdev.blogspot.com/2009/06/simple-exchange-email-client-for.html*) provides an excellent starting point because it's written in the form of a Windows PowerShell script. The script (Figure 10-44) demonstrates how to use AutoDiscover to locate a mailbox, connect to a mailbox, enumerate the folders in the mailbox, create and send messages, and view the contents of message bodies and headers. It's a great example of EWS in action that you can use to discover how to apply EWS to access many Exchange functions.

Although Microsoft positions EWS as the API of choice for developers, it is not used within the product. Instead, Microsoft builds many of the Exchange components that need to work with mailboxes, including EWS, RCA, ActiveSync, and UM, on top of an internal API called XSO. Server-side MAPI is still used within the product, but XSO has taken center stage recently.

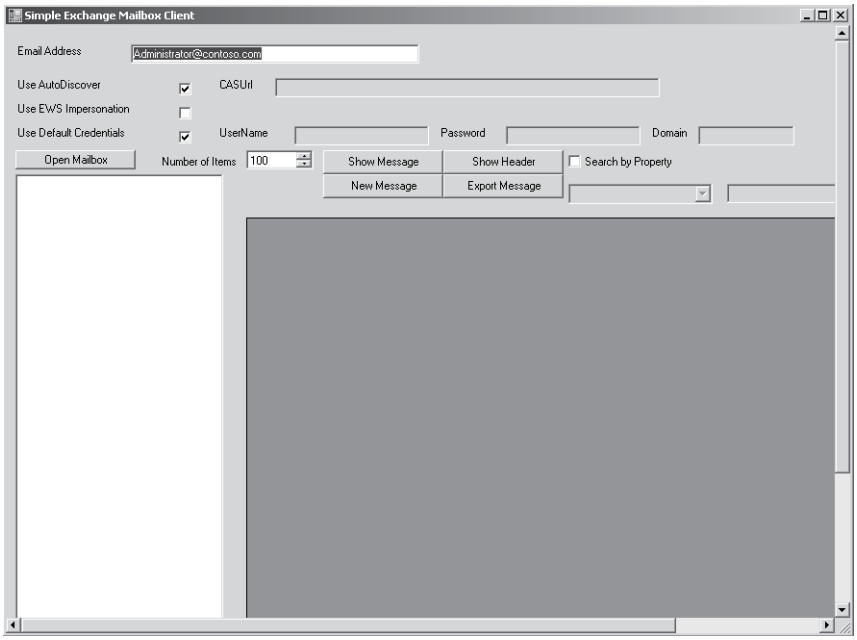


Figure 10-44 The Simple Exchange Mailbox client written in Windows PowerShell.

Windows PowerShell remains the top choice for any server-focused management or administrative process that you need to automate, including when you need to invoke a Windows PowerShell cmdlet from within a .NET language. Remote Windows PowerShell allows administrators to execute cmdlets from workstations that have no Exchange code installed (naturally, the latest version of Windows PowerShell must be installed) and complies with the RBAC model so that administrators can only manage the objects that they have been granted permission to manage.

A common connection point

Having clients is all very well, but we've got to connect them to Exchange to be able to do some work. The CAS is the common connection point where everything comes together in Exchange 2010, so logically it's where our story takes us next, even if, as we discuss, deploying an Exchange 2010 CAS is one of the first steps in many deployments.

CHAPTER 15

Compliance

The joy of legal discovery 974
Personal archives 976
Messaging records management 989
How the Managed Folder Assistant implements retention policies 1018
Putting a mailbox on retention hold 1021
Putting a mailbox on litigation hold 1022
The very valuable dumpster 1025
Discovery searches 1033
Auditing administrator actions 1049
Auditing mailbox access 1057
Message classifications 1064
Protecting content 1070
Outlook Protection Rules 1080
Rules help compliance, too 1082

THE need to achieve compliance with legal and regulatory requirements is a fact of corporate life today. Legislation such as the Sarbanes-Oxley Act in the United States has influenced many other countries to introduce similar requirements to keep records that show when something was done and by whom. Microsoft started on the process to build records management capability into Microsoft Exchange Server 2007 with the introduction of managed folders. However, users (and many administrators) didn't really understand the purpose of managed folders and compliance was weak. To address these issues and to provide a true basis for compliance, Microsoft Exchange Server 2010 introduces its own features, including the following:

- The ability to audit administrator actions
- The provision of archive mailboxes
- The ability to create and apply retention policies to items and folders
- The ability to recover items even if a user has deleted them from the dumpster
- The ability to place mailboxes on retention hold or litigation hold

There is a big difference between retention hold and litigation hold (also referred to as legal hold) that we need to be clear about when we discuss compliance. Retention hold means that any retention policies that are in force on a mailbox are suspended, normally when the mailbox owner is unable to process messages for a period. Litigation hold is a totally different feature that is designed to capture all edit and delete operations for a mailbox inside the dumpster so that a user cannot affect or eliminate data that might be required by a discovery action. If you put a mailbox on litigation hold, you nullify the effect

of retention hold because the effect of any retention policies is suspended. Wise administrators take advice and guidance from the company's legal department before they place a mailbox on retention or litigation hold to ensure that the action complies with any legal requirements that are in force and does not compromise any document retention policies that are in effect. There's no point in enabling a feature that collects unneeded or unwanted data.

The features just listed are not exhaustive, as other Exchange features can be associated with compliance. For example, transport rules allow a disclaimer to be appended to every outgoing message that can limit liability by complying with rules that say that messages from a company must contain specific contact or other information about the company. The point is that compliance is an area that continues to evolve. There is no doubt that Microsoft will update the Exchange feature set over time to satisfy the broadest possible set of requirements. However, it is unreasonable to expect any software to deliver a complete answer. For example, although you can place mailboxes on litigation hold or establish an extended deleted item retention period to ensure that important information is not deleted, you also need to figure out the administrative procedures to handle situations such as mailbox retention following the death of an employee. Handling situations like this is reasonably straightforward (disable the account, hide the user's mailbox from the Global Address List [GAL], and keep it and any archives—including PSTs that you recover from the user's PC—until any legal hold period is passed) as long as you think everything through.

With that cheerful thought in mind, let's review the new compliance features in Exchange 2010 and explore the updated functionality available for older compliance features such as transport rules and journaling.

The joy of legal discovery

Legal discovery actions have been around for centuries. Over the last two decades, we have seen the focus of discovery or searches for information pertinent to a legal case begin to shift from paper evidence to electronic evidence. This shift reflects the different manner in which organizations store data today. We still have filing cabinets stuffed full of paper, but much of the correspondence that companies conducted by letter, fax, and telex is now sent by email, so the focus for discovery has to accommodate both paper and electronic media.

Discovery actions for email systems first began in the mid-1980s. At that time messages were recovered from backup tapes and printed for lawyers to review. The process was dreadfully expensive and time consuming. The only mitigating factor was that it was much easier to determine who might have sent an incriminating message because relatively few people in a company had email and the overall volume of email was low. Messages were text only and tended to be short. It was therefore possible to satisfy a judge's

order to retrieve all messages for ten specific users over a month without running up an extraordinarily high bill.

Today's environment is obviously different. Many more users are typically hosted on each server, they send and receive an ever-increasing volume of messages, and those messages contain many different types of attachments, including video and audio files. The result of living in the age of electronic communication is that the cost of legal discovery is higher because there is more information to process. In March 2009, Fortune Magazine reported that the court-appointed trustee of bankrupt Lehman Brothers, Inc. had captured 3.2 billion email and instant messages occupying 1.4 TB. This isn't an unusual amount, as the FBI investigation of Enron in 2001 reviewed 31 TB of data and ended up using 4 TB as evidence. Email is a critical means of business communication that has replaced telexes, faxes, and written letters in many respects, so legal discovery of email has moved from an out-of-the-ordinary situation to something that is extremely common, whether it is to satisfy a legal or regulatory requirement, respond to a subpoena, or deal with an internal matter concerning employee ethics, harassment, or discipline.

The first generation of Exchange offered no way to keep mail around after it was deleted, which meant that you had to restore a database from a backup if you wanted to recover a message, whether it was needed to satisfy a legal order or because a user had deleted it in error. Gradually Microsoft began to add new features to Exchange to help. The original version of the "dumpster" as implemented in Exchange 2000 through Exchange 2007 provides a two-phase delete process where messages are marked as deleted but kept in the database until their retention period expires, at which time they are removed. The initial operation is a "soft delete," the latter is a "hard delete." Note that folder structures are not respected in the dumpster, as messages are "flattened" into a single repository. In other words, if you delete ten folders, each of which holds 2,000 items, and then realize that you should not have deleted one of the folders, you will have to recover the 2,000 items for that folder from the 20,000 items that are put into the dumpster. As we will see in the section "Dumpster 2.0 arrives" later in this chapter, Exchange 2010 includes an enhanced dumpster with some useful new features.

Journaling made its appearance in Exchange 2003 and was upgraded in Exchange 2007. However, the functionality offered by Exchange was basic, and most companies that invested in products to capture and archive messages went for purpose-designed products such as Symantec's Enterprise Vault, Mimosa Systems' (a division of Iron Mountain) NearPoint, or the HP Information Archive. Microsoft added managed folders in Exchange 2007 with the idea that administrators could create folders that are distributed to mailboxes for users to store important items. The contents of these folders are managed through policies, and it is possible to create procedures to harvest information from these folders on a regular basis. Not many companies used managed folders, and it is an example of a reasonable idea with a good purpose that collapsed when it was exposed to the acid usability test of real-life deployment outside Microsoft.

The compliance features in Exchange 2007 were a start and provided useful feedback from the companies that deployed managed folders. However, the overall experience was not compelling enough to generate widespread usage of the compliance features, which then led Microsoft to deliver a new set of features in Exchange 2010 and then further enhance the features in SP1. At TechEd and other events, Microsoft presenters acknowledge that many vendors have been actively selling archive solutions for Exchange for nearly a decade and that some offer much more developed functionality than Exchange 2010, especially in areas such as workflow, their ability to archive information taken from other sources, and the experience that companies have with these products in integrating the archival process with regulations. They go on to characterize the target market for Exchange 2010 archiving as the vast majority of the installed base that:

- Does not use archiving today.
- Depends on PSTs as a "relief valve" for restrictive mailbox quotas.
- Relies on tape/disk backups to respond to requests to recover data from users or to respond to discovery actions; this is obviously a very costly and time-intensive method.

Microsoft has to convince customers that having integrated archiving and search incorporated into an email server is a better solution than dedicated archiving and search applications that have been in use and developed over many years. It can be argued that cost is one key Microsoft advantage because archiving is available at the price of an enterprise Client Access License (CAL) that might be already acquired. The cost of an enterprise CAL for each user will often be lower than the cost of dedicated archiving software plus any additional hardware that is required to run the archiving software. This argument works if the functionality available in Exchange 2010 meets your requirements but fails if it doesn't. Microsoft makes the point that they work closely with third-party software developers to ensure that the widest possible choice is available to customers, and it will be interesting to see how vendors such as Symantec and Mimosa cooperate and compete with Microsoft in this area over the next few years.

Personal archives

A personal archive is an extension of a user's primary mailbox that provides an online archive facility. It is also referred to as an archive mailbox. The name might cause some confusion with the personal archives that users create with PST files for Outlook. The big difference is that the Exchange archive is integrated into the Information Store, and the data held in the archive are therefore accessible using all the features available to mailboxes, including discovery searches. By comparison, PST archives are confined to a PC, and the data that they contain are inaccessible to server-based processing. Indeed, Outlook's AutoArchive feature can be argued to conflict with the archiving functionality now available

in Exchange because it focuses on moving items from server-based folders into PSTs, whereas a central point of Exchange-based archiving is the elimination, whenever possible, of PSTs. For that reason, you might want to consider using group policy settings to disable the use of AutoArchive.

Exchange 2010 originally restricted the location of the personal archive to the same database that hosts the primary mailbox. From Exchange 2010 SP1 onward you can elect to have the archive in a separate database that can be on a completely different server, provided that the database is located on a server in the same Active Directory site as the primary mailbox. The archive can be in a database managed by a different Database Availability Group (DAG) if the Active Directory site supports multiple DAGs.

> **Note**
>
> **The archive can also be on a mailbox server that is not part of a DAG, although in this case you should be concerned about data protection because only one copy exists of the database that holds the personal archive.**

Finally, if you use the Microsoft Exchange Online service (part of the Office 365 suite), the personal archive can be stored "in the cloud," an option that might prove increasingly attractive as companies gain more experience and confidence with cloud-based services (this feature will be made available after Microsoft upgrades its Exchange Online service to use Exchange 2010 , expected in early 2011). It is attractive to be able to hive off personal archives to a cloud-based service because this allows you to remain focused on the care and maintenance of production mailboxes while the hosting provider takes care of the archives. Whatever option is chosen, a mailbox can have just one personal archive, and each personal archive requires that the mailbox has an enterprise CAL.

Microsoft views personal archives as the natural replacement for PSTs, which were never designed to function as user archives. The growth of messages and the reluctance of administrators to increase mailbox quotas—coupled with the inability of Exchange and its clients to deal elegantly with very large mailboxes (5 GB and upward)—meant that most organizations were forced to use PSTs to offload data from the online store. Users do like to behave like human pack rats and keep messages, even if they never look at them again (some estimate that a message filed into a PST has a 99 percent chance of never being looked at again after six months). Other problems with PST management typically cited in corporate messaging deployments include the following.

- **Reduced security** PSTs are personal stores, but users keep just about anything in them, including sensitive and usually unencrypted corporate information ranging from budgets to presentations about new products to performance reviews. If

someone loses a laptop—or even a USB device that has a PST on it—that information is immediately exposed and potentially available to anyone who finds the device and accesses it. Even if protected by a password, the PST file structure is insecure and can be quickly cracked using utilities commonly available on the Internet. Once the password is bypassed, a PST can be opened using any Microsoft Outlook client.

- **Inability to respond to discovery actions** Information held on a PST is usually invisible to searches that a company performs to respond to discovery requests. This is fine if the information is personal or irrelevant to the discovery request, but it could be very expensive if required information is not disclosed to a court and is subsequently discovered.

- **Inability to apply policy** Many companies have a data retention policy that requires users to delete documents and messages after a certain period. The period may vary depending on the type of information contained in different items. In any case, the company loses any ability to apply policy centrally once a user moves an item from his mailbox into a PST.

- **Exposure to data loss** Laptop disks are notoriously prone to failure. If users don't back up their data, any disk crash exposes them to potential data loss, and that information might be important.

The alternative solution to increasing disk quota for mailboxes in previous versions of Exchange was to buy and deploy a dedicated third-party archiving solution such as Symantec Enterprise Vault. Using PSTs is obviously far cheaper for a company. It's also easier for users because they control how many PSTs they create and how they use them. Some create a separate PST for each year; some create a PST for each major project. However, the big downside is that PSTs then expose the company to the risks previously described. Even so, it will take time to pry user fingers from their beloved PSTs.

Exchange personal archives are not perfect, and a number of limitations exist that could hinder deployment, including the following.

- Exchange 2010 does not support delegate access to a personal archive. Users can delegate access to their primary mailbox, but the same delegation does not carry through to the archive. This is an issue for assistants who support executives. You can impose retention policies to force items to move into the archive, but the mailbox owner is the only person who can manage the items afterward (and few executives will have the time or interest for this work). Exchange 2010 SP1 supports delegate access to the personal archive; when you enable delegate access to a mailbox, delegates are automatically granted access to the mailbox's personal archive.

- You cannot transfer an archive to another mailbox. If a user leaves and you delete her mailbox, the archive disappears, too. You can save data by exporting items from the archive (and the primary mailbox) to a PST and then importing them back into the personal archive of another user, but it would be more elegant to be able to transfer the archive intact.

- You cannot copy or move sections of the archive to transfer it to another user. For example, a user who wants to transfer responsibility for a project to another user has to extract and provide the folders and other items relating to the project from his archive and provide them to the other user. Again, the workaround is to export selected folders from the personal archive to a PST and provide the PST to the other user (or import the PST into her archive).

- You cannot assign permissions on a folder level within the archive to allow users to give access to parts of their archive to other users. In fact, there is no permissions model for the archive yet.

These are examples of areas where Microsoft will doubtless consider enhancements in the future. It's likely that they will wait to see how archives are used in practical terms within customer deployments before they plan how archives will evolve in future releases of Exchange.

Enabling a personal archive

Before you can create and use personal archives with Exchange 2010, you have to deploy clients that support the feature. When first introduced, Microsoft Outlook 2010 (Figure 15-1) and Outlook Web App were the only clients that supported personal archives. The need to deploy a new version of Outlook proved to be a significant deployment blocker for many companies, so Microsoft announced their intention to provide an upgrade for Outlook 2007 with the code necessary to detect that a mailbox had an associated personal archive and then display it in the list of available mailbox resources. At the time of writing, Microsoft has not yet released the upgraded code for Outlook 2007, but it is expected to work in much the same way as Outlook 2010 interacts with personal archives.

The easiest way to assign a personal archive to a mailbox is when you create the mailbox (Figure 15-2). The SP1 version of Exchange Management Console (EMC) allows you to select a different database to host the personal archive, providing that the database is not mounted on an Exchange 2010 RTM server. Interestingly, if you place the personal mailbox in a different database, Exchange automatically transfers the dumpster to the personal archive to minimize the size of the primary mailbox.

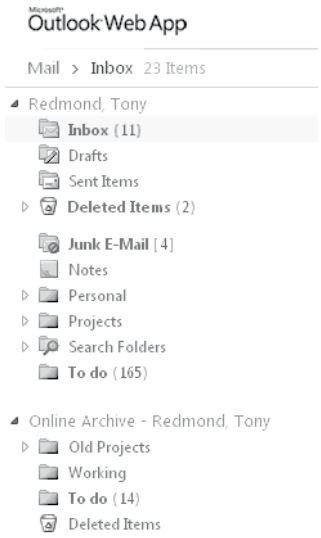


Figure 15-1 Archive mailbox in Outlook Web App.

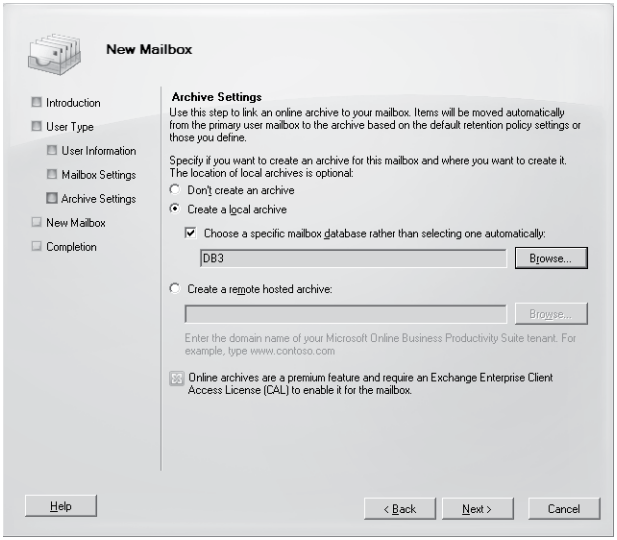


Figure 15-2 Creating a personal archive with a new mailbox.

INSIDE OUT

You won't lose access to personal archives if there is a server failure

EMC restricts the databases that you can choose to hold the personal archive to those that are mounted on servers in the same site. However, Exchange supports an exception to this rule when a database transfers to a server in another site following a failure. In this case, the CAS will redirect clients to the personal archive in the database in the other site using a cross-site connection for as long as the database is hosted in that site. You won't want to use cross-site connections for an extended period, and normal connections will resume after you switch the database that contains the personal archive back to a server in its original site. You can also see the option displayed by EMC to allow the personal archive to be hosted by Office 365 when this feature is supported by Microsoft for their Exchange Online service.

To enable an archive when you create a mailbox with EMS, you simply add the *–Archive* parameter to the New-Mailbox cmdlet. See Chapter 6, "Managing Mail-Enabled Recipients," for a full discussion about how to create new mailboxes.

You can also enable a personal archive for existing mailboxes by selecting a mailbox in EMC and then selecting the Enable Archive option in the action pane. EMC warns you that enabling this feature requires an enterprise CAL, and if you click OK, the mailbox is enabled. You can also enable a personal archive for an existing mailbox with Exchange Management Shell (EMS). For example:

```
Enable-Mailbox –Identity 'Tony Redmond' –Archive
```

You cannot archive mailboxes that use managed folders

You can enable an archive for room and equipment mailboxes, which seems a little strange. However, you cannot enable an archive for a mailbox that has been assigned a managed folders policy (Figure 15-3). Managed folders provide the basis for messaging records management in Exchange 2007, but they are superseded by retention policies in Exchange 2010. Retention policies work closely with archive mailboxes and are the future basis for Exchange messaging records management. Given these facts, the developers decided not to support archives for mailboxes that use managed folders.

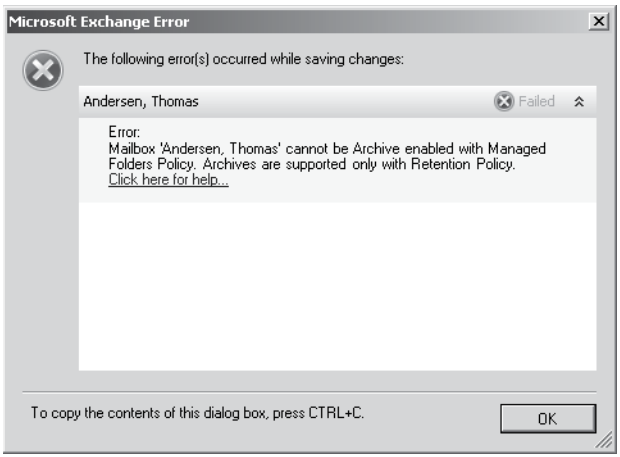


Figure 15-3 Exchange can't enable a personal archive because the mailbox uses managed folders.

Scanning mailboxes that are in managed folders

As part of your preparation for the deployment of personal archives, you can scan for mailboxes that are assigned a managed folders policy. On the surface, you'd expect that using some code to look for any mailbox that doesn't have a null value in its managed folders mailbox policy property would do the trick. For example:

```
Get-Mailbox –Filter {ManagedFolderMailboxPolicy –ne $Null} | Select Name,
ManagedFolderMailboxPolicy
```

This code returns a list of mailboxes, but it's flawed because it includes any mailbox that was assigned a managed folders mailbox policy in the past, even if the managed folders policy was subsequently removed from the mailbox and replaced by an archive mailbox. Better code that produces the right results by filtering out archive-enabled mailboxes is:

```
Get-Mailbox –Filter {ManagedFolderMailboxPolicy –ne $Null –and ArchiveName –eq $Null}
| Select Name, ManagedFolderMailboxPolicy
```

You can remove the MRM 1.0 policy from a mailbox with a command like this:

```
Set-Mailbox –Identity 'Andersen, Thomas' –ManagedFolderMailboxPolicy $Null
```

Note

It is possible to enable a personal archive for a room or resource mailbox. I cannot quite think of why anyone might want to maintain an archive for a room or resource mailbox, but I'm sure that someone will come up with a good reason in time.

Filtering for archived mailboxes

EMC includes a canned filter to allow you to see the mailboxes that already have archive mailboxes (Figure 15-4). The filter is changed slightly in SP1 from the one used in the original release of Exchange 2010 (Has Archive = Yes) because SP1 can host personal archives on an on-premise or hosted service.

You can also use the Get-Recipient or Get-Mailbox cmdlets to search for mailboxes that have an archive. For example:

```
Get-Mailbox –Filter {ArchiveName –ne $Null} | Select Name, ArchiveName
```

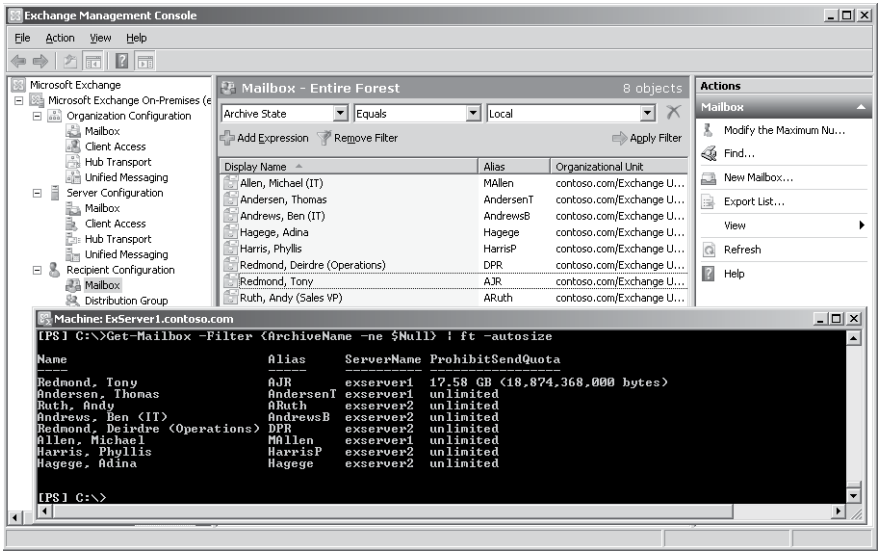


Figure 15-4 Displaying the list of personal archives in EMC and EMS.

Enabling the archive and its properties

Behind the scenes, EMC calls the Enable-Mailbox cmdlet to enable an archive. These commands first enable the personal archive for a mailbox and then retrieve the properties that Exchange maintains for an archive.

```
Enable-Mailbox –Identity 'Andy.Ruth@contoso.com' –Archive
Get-Mailbox –Identity 'Andy.Ruth@contoso.com' | Select Name, Arch*
```

```
Name                : Ruth, Andy
ArchiveGuid         : f7552939-8185-4634-824e-d4cd6241d674
ArchiveName         : {Online Archive –Ruth, Andy)}
ArchiveQuota        : unlimited
```

```
ArchiveWarningQuota : unlimited
ArchiveDomain       :
ArchiveDatabase     :
ArchiveStatus       : none
```

The first four properties listed here are always present for a mailbox after its archive is enabled. The globally unique identifier (GUID) identifies the archive mailbox within the database where it is stored. The default name for the archive is derived from the prefix "Online Archive" plus the mailbox's display name and can be changed afterward to whatever name you prefer. The archive quotas are inherited from the default values set for the database and reflect the values that Exchange uses to limit the amount of information in the archive and the point when it starts to issue warning messages.

You can alter these values with the Set-Mailbox cmdlet. For example:

```
Set-Mailbox –Identity 'Andy.Ruth@contoso.com' –ArchiveName "Andy's Splendid Online
Archive" –ArchiveQuota 2GB –ArchiveWarningQuota 1.9GB
```

The last three of the archive properties listed for the mailbox are introduced in Exchange 2010 SP1.

- *ArchiveDomain* is only used if the personal archive is stored on an Exchange Online server (Office 365). If used, the property holds the Simple Mail Transfer Protocol (SMTP) name of the hosted domain.
- *ArchiveStatus* contains a status value to indicate whether the personal archive has been created on an Exchange Online server.
- *ArchiveDatabase* is blank if the personal archive is stored in the same mailbox database as the primary mailbox; otherwise the property contains the name of the mailbox database that holds the archive.

Checking space usage

The amount of space used in an archive mailbox can be checked with the Get-MailboxStatistics cmdlet, which supports the *–Archive* parameter to tell it to report details of the archive mailbox rather than the primary mailbox. For example:

```
Get-MailboxStatistics –Identity 'John Smith' –Archive | Select DisplayName,
ItemCount, TotalItemSize, LastLogonTime
```

```
DisplayName                    ItemCount  TotalItemSize                LastLogonTime
-------------                  --------   -------------                -------------
Online Archive - Smith, John...  128        31.51 MB (33,037,293 bytes)   4/14/2010 3:30:26
AM
```

Updating the name of an archive mailbox

You can also update the name of the archive mailbox through EMC. To do this, select the mailbox, click Properties, select Mailbox Features, and then select Archive from the list of mailbox features. However, although you can update the name (Figure 15-5), you can't update archive quotas through EMC, nor can you view details of the items stored or quota used in the archive mailbox.

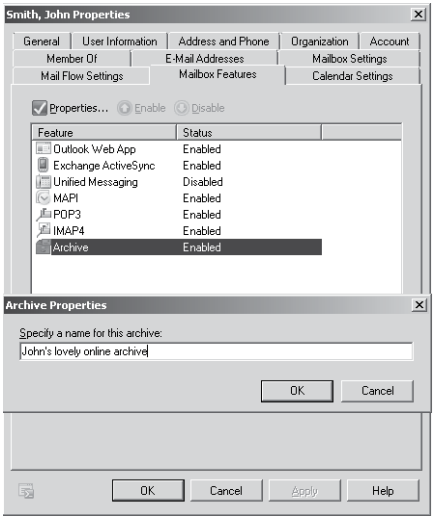


Figure 15-5 Updating the name of a personal archive through EMC.

Default archive policy

When you enable a personal archive for a mailbox, Exchange assigns a retention policy called Default Archive and Retention Policy to the mailbox to help the mailbox's owner use the archive by automatically moving items from the primary mailbox into the archive as their retention period expires. The retention period applied by the default tag in the policy is two years, so the effect of applying the policy is that any item that is not stamped with another tag will be moved into the archive after it is two years old. The retention policy assigned to the mailbox becomes effective the next time the Managed Folder Assistant processes the mailbox. The default policy is not assigned if the mailbox is already under the control of another retention policy. We discuss how to manipulate retention policies and tags in the "Messaging Records Management" section later in this chapter.

Originally, the RTM version of Exchange 2010 applied a different retention policy called Default Archive Policy that only contained archive tags. You'll find both policies are available, but Exchange 2010 SP1 now only uses the Default Archive and Retention Policy. As

the name implies, the big difference between the two retention policies is that the Default Archive and Retention Policy contains both retention tags (those that affect how long an item is kept by Exchange) and archive tags (those that affect when an item is archived). Table 15-1 describes the retention and archive tags that are included in the default archive policy. You can add or delete retention and archive tags to the default archive and retention policy if required. You do not need to delete the older default archive policy. However, after upgrading an organization to Exchange 2010 SP1, you can check for mailboxes that have the older policy assigned to them and replace these assignments with the new default archive and retention policy.

Table 15-1 Tags included in default archive policy

Tag name	Type	Purpose
Default 2 year move to archive	Default	Automatically move items to the personal archive when they are two years old. This tag is applied to any item in the mailbox that does not have an explicit tag applied by the user or is inherited when an item moves into a folder that has a default policy.
Personal 1 year move to archive	Personal	Tag that the user can apply to items to instruct the Managed Folder Assistant to move the items into the personal archive after they are one year (365 days) old.
Personal 5 year move to archive	Personal	Tag that the user can apply to items to instruct the Managed Folder Assistant to move items into the personal archive after they are five years (1,825 days) old.
Personal never move to archive	Personal	Tag that the user can apply to items to block the Managed Folder Assistant from ever moving the items into the personal archive.
Recoverable Items 14 days move to archive	Recoverable Items folder	Move items placed in the Recoverable Items folder to the personal archive after 14 days.
1 Month Delete	Personal	Move items into the Recoverable Items folder after one month.
1 Week Delete	Personal	Move items into the Recoverable Items folder after one week.
6 Month Delete	Personal	Move items into the Recoverable Items folder after six months.
1 Year Delete	Personal	Move items into the Recoverable Items folder after one year.
5 Year Delete	Personal	Move items into the Recoverable Items folder after five years.
Never Delete	Personal	Disabled tag that prevents the Managed Folder Assistant from processing the item; the effect is to stop the item from ever being deleted.

The major impact of the application of the Default Archive and Retention Policy is that the Managed Folder Assistant will begin to move items into the personal archive after they are two years old. This leads to the "disappearing items" syndrome where users log problem reports that their mailbox is missing items. In the vast majority of cases, the missing items are found safe and sound in folders in the archive mailbox. It just takes time for users to realize that Exchange will move items automatically after they reach a certain age, so this

underlines the importance of communication with the user community as you implement archive mailboxes.

INSIDE OUT

Don't delete the default policy! Change it or create a custom retention policy.

You should not delete the Default Archive and Retention policy, because this will impact the processing performed by the Managed Folder Assistant for the mailboxes to which the policy is assigned. It's a better idea to create a custom archive and retention policy tailored to the needs of the company or to different groups of users and apply that policy to their mailboxes. In this case, the custom retention policy replaces the default archive policy. We discuss how to manipulate retention policies and tags later in this chapter.

Disabling a personal archive

You can disable an archive with the Disable-Mailbox cmdlet. For example:

```
Disable-Mailbox –Identity 'Smith, John' –Archive
```

EMS prompts for a confirmation before it proceeds unless you add the *–Confirm:$False* parameter. This is not a good idea unless you are absolutely sure that you want to disable the archive. When it disables an archive mailbox, the Store disconnects it from the primary mailbox and keeps it in the database until the deleted mailbox retention period expires.

Using a personal archive

Assuming that a personal archive is in place and a suitable client is at hand, working with items in a personal archive is just like working with items in the primary mailbox. You can create new items, reply to messages, move items around, and so on. After the archive mailbox is created, it is up to the user to populate it, most likely by using drag and drop to move folders or items from his primary mailbox. Administrators can import the complete contents of PSTs into a mailbox, but there are some limitations with this approach, as we discussed previously.

Exchange doesn't support offline access for data held in personal archives. In other words, when Outlook is configured to use cached Exchange mode, it has access only to the offline copies of the folders from the primary mailbox that are stored in the OST and uses background synchronization to keep those folders updated. This arrangement allows Outlook to continue to work through transient network interruptions. Outlook has to be able to connect to the server before it can work with a personal archive.

TROUBLESHOOTING

I can't access my personal archive when I'm offline.

If you want something to be available offline, you have to store it in the primary mailbox. The personal archive is designed to hold information that isn't always required immediately and you can wait until you can get back online to access it, so if you need something from the archive and know that you have to work offline (for example, on a road trip), then you have to plan ahead and move the desired items from the personal archive into the primary mailbox beforehand.

INSIDE OUT Some personal archive issues

Exchange 2010 marks Microsoft's first venture into archive mailboxes, and it's inevitable that there will be some issues that implementers have to understand as they plan deployments.

The first and most obvious issue is the need for archive-aware clients to gain full advantage of the archive. If you have a large population of Outlook 2003 or Outlook 2007 clients, you need to plan for client upgrades to Outlook 2010 or decide whether Outlook Web App is a viable workaround for users who want to access an archive. As mentioned earlier, Microsoft has announced their intention to provide code that allows Outlook 2007 clients to access archive mailboxes, but nothing will be done for Outlook 2003.

Archive mailboxes require enterprise CALs. This might not be a problem if you use other features that require the enterprise CAL, but it can be an additional and unexpected cost if you only use standard CALs today. As noted in Chapter 5, "Exchange Management Console and Control Panel," you can use the Get-OrganizationConfig cmdlet or the EMC option to collect Organization Health information to report on the number of enterprise CALs that you require.

Outlook 2010 is the only client that can perform a client-side search across the contents of both primary and archive mailboxes. Both Outlook and Outlook Web App limit the user to searching in either the primary or archive mailbox. This is somewhat more understandable in the case of Outlook that performs searches on the PC, especially when the client is configured in cached Exchange mode, as the contents of the archive mailbox are not replicated to the OST and are therefore not available when the user works offline. However, I cannot understand why a client like

Outlook Web App, which works online all the time and has access to the Exchange content indexes, cannot perform a search across both repositories. The logic might be that such a search could require access to two different servers if the primary and archive mailbox are located in separate databases, but it's not something that will make much sense to users.

The RTM version of Exchange 2010 does not support delegate access to archive mailboxes. This issue is addressed in SP1.

Messaging records management

Exchange 2007 introduced the messaging records management (MRM) system as its "business email" strategy to help users comply with regulatory and legal requirements. The idea is to provide a method for users to retain messages and attachments that are required business records. Another way of thinking about MRM is that it helps users keep control over mailboxes by automating the retention process; marked items are kept as long as required, whereas others can be automatically discarded when their retention period (otherwise known as the expiration limit) expires.

The key to success for any scheme that aims to alter user behavior is to make it as simple as possible while achieving maximum functionality. Exchange 2007 didn't quite meet this goal. Its version of MRM uses a set of managed folders that have policies attached to them. The folders can be one of the default mail folders (Inbox, Sent Items, and so on) or a specially created folder that can be used to store items that the business wishes to control. The Managed Folder Assistant (MFA) is responsible for the application of the policies attached to the folders. The MFA runs on a regular basis to process items in managed folders. Items are stamped with the retention policy that applies to the folder, and this dictates what happens to the items in the future. For example, if a policy is set on the Inbox to delete any item older than 60 days, the MFA will move items older than this limit to the Deleted Items folder.

Exchange 2007 MRM works if users are disciplined in their filing habits and understand the concept of managed folders. Some people like the structure imposed by managed folders because it creates a structured approach to work. The problem is that the vast majority of Exchange users are relatively undisciplined when it comes to filing, and they do not wish to spend time moving items around unless it's necessary to delete items or move them into a PST to get their mailbox size under quota so that they can send or receive new messages. Indeed, the radically better search facilities that are available in recent versions of Exchange and Outlook encourage users never to refile anything because they can always search for

an item when required. In addition, Exchange is able to handle very large folders that hold tens of thousands of items, so the imperative to refile items to achieve acceptable performance does not exist. The combination of human nature and better software conspired to make Exchange 2007 MRM ineffective in real terms. Microsoft therefore needed to change its tactics to provide a workable implementation of MRM for Exchange 2010.

The new approach to messaging records management in Exchange 2010

Managed folders persist in Exchange 2010, but only for backward compatibility. The future of Exchange-based MRM lies in a structure created by retention tags and policies. Retention tags can be applied to any item in any folder to specify what action Exchange should take for the item when its retention period expires. Supported actions include the hard (permanent) or soft (recoverable) deletion of the item, moving the item to a personal archive, or flagging the item for user attention. Retention policies group retention tags together in a convenient manner to allow administrators to apply policies to mailboxes rather than having to assign individual retention tags to folders. Retention tags and policies are organization-wide objects that are stored in Active Directory and can therefore be applied to any mailbox in the organization after they are created. Just like Exchange 2007, the MFA is responsible for checking mailbox contents against policy and taking whatever action is determined by policy for items that exceed their retention period.

This all sounds like a workable solution. The only issue is that Microsoft didn't ship any GUI to allow administrators to set up and manage retention tags and policies in the RTM release of Exchange 2010. Instead, you have to perform all management of retention tags and policies through EMS until you deploy Exchange 2010 SP1, which includes the necessary GUI in its version of EMC. In addition, Exchange 2010 SP1 includes a set of retention and archive tags such as "1 Month Delete" (items stamped with this tag are moved into the Deleted Items folder after one month) that you can use as a starting point to develop your own retention policies.

Types of retention tags

Table 15-2 describes the three types of retention tags supported by Exchange 2010. The "type" shown in the third column is a value passed to the *–Type* parameter when you create a new tag with the New-RetentionPolicyTag cmdlet. Exchange uses this value to understand the scope of the items in a user mailbox to which it can apply the tag.

Table 15-2 **Types of retention tags**

Tag type	Context	Type
Retention policy tags (RPT)	Administrators can apply these tags to default mailbox folders such as the Inbox, Sent Items, and Deleted Items. In Exchange 2010 SP1, tags cannot be applied to the Tasks and Contacts folders. If an RPT is assigned to a default folder, all items in the folder automatically come under the control of the tag unless the user applies a personal tag to the item. Only one RPT can be assigned per default folder.	DeletedItems Drafts Inbox JunkMail Journal Notes Outbox SentItems All
Default policy tags (DPT)	A catch-all tag that the MFA applies to any item that does not inherit a tag from its parent folder or has not had a tag explicitly applied to it by the user. In other words, if no other tag applies to an item, Exchange will respect the instructions contained in the default tag. A retention policy only includes a single DPT that is used to delete items; you can specify another to control the default movement of items into the personal archive. It's logical but sometimes overlooked that if you specify two DPTs in a policy, the tag that moves items into the archive must have a shorter retention period than the tag that deletes items.	All
Personal tags	Users can apply these tags to nondefault folders and individual items in a mailbox. Personal tags that move items into the archive can also be applied to default folders. Personal tags mark an item with an explicit retention, usually to comply with a business requirement. For example, you might use an "Audit" tag to mark items that users are compelled to retain for audit purposes. A retention policy can include many different personal tags.	Personal

You'll notice that some default folders that you expect to find in every mailbox are excluded from the list of folders that support retention policy tags. The RTM version of Exchange 2010 doesn't allow retention policy tags to be applied to the Calendar, Contacts, Journal, Notes, and Tasks folders. However, items in these folders were covered by the default retention policy and so could be removed unexpectedly.

INSIDE OUT

There is a way to create retention policy tags for calendars

Microsoft's thinking on the subject evolved in Exchange 2010 SP1, and you can now use the New Retention Policy Tag Wizard to create retention policy tags for almost all of the default folders. Although the wizard interface allows you to select the Contacts folder and create a tag, this is really a bug, and the MFA ignores any tag placed on

Contacts. Somewhat bizarrely, the wizard interface does not allow you to select the Calendar folder to create a retention policy tag for it, but you can create a retention policy tag for the Calendar with the New-RetentionPolicyTag cmdlet. Perhaps the logic here is that you should be careful when applying a retention policy to user calendars because users often want to keep the items stored in these folders for longer than items in other folders, so you are forced to do more work to create a retention policy tag for the calendar. After all, no one will thank you if you clean out the CEO's calendar after 120 days! Once you create the retention policy tag for the calendar with EMS, you can manage it as normal with EMC, and you can include it in retention policies and apply it to mailboxes.

Overall, the set of default folders covered by Exchange 2010 SP1 is much broader than before and includes those where items often accumulate without users noticing, such as the Sync Issues and RSS Feeds folders, so you can now create and apply retention tags that clear out these folders for users automatically. You can see the list of folders for which you can create retention policy tags in Exchange 2010 SP1 in Figure 15-6. The All Other Folders In The Mailbox choice is used when you create a default policy tag.

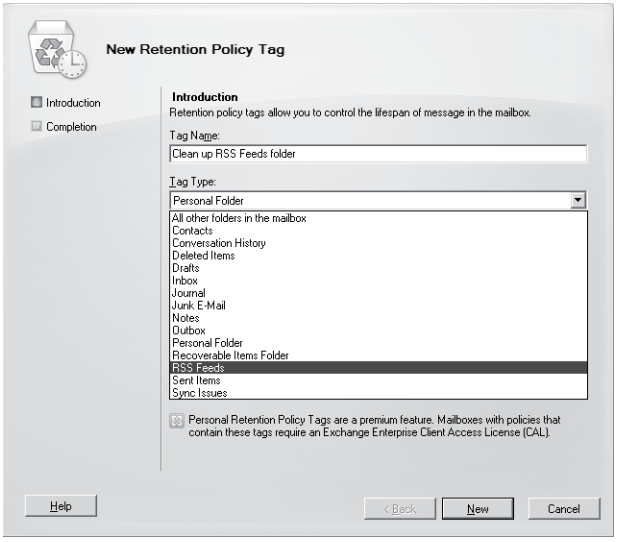


Figure 15-6 Listing of folders for which you can create retention policy tags.

INSIDE OUT Some items are timeless

Items in some of these folders tend to be more "timeless" than general-purpose messages, so you should be careful to think through the potential consequences when you create retention policy tags for folders such the Journal or RSS Feeds. For example, how long do users reasonably want to keep items in the Drafts folder? Some users like to keep drafts for a long time because it's their practice to create a message and compose it through multiple edits over time before they decide whether they want to send it. Others view the Drafts folder as strictly transient and never keep items there for longer than a day or so. For this reason, it might be best to create retention policy tags that mark some items to "never expire." This is done by setting the tag to be disabled rather than giving it a retention period. When an item is stamped with a disabled retention tag, it tells the MFA that it should ignore the item and the item will therefore never be processed.

Retention tags cannot be applied to items directly. They first have to be assigned to a retention policy and the retention policy assigned, in turn, to the mailboxes that you want to manage. A retention tag can be reused several times in different policies. Although there is no practical limit to the number of retention tags that you can define for an organization, it makes sense to create a set of tags that can be shared and reused between retention policies rather than creating separate tags for each policy.

Exchange can only apply one retention policy tag and one archive tag to an item. Two simple rules are used when Exchange evaluates policies that it can apply to an item. The first rule states that the policy with the longest retention period always wins and is intended to ensure that Exchange never deletes an item before its time truly expires. The second rule is that an explicit policy is always respected ahead of an implicit or default policy. In other words, if you apply a personal tag to an item to retain it for six years and the default retention policy for the folder requires deletion after 12 months, the item will be kept for six years. Retention tags can be placed on items, conversations, or complete folders, and they are transferred with items if you move them between folders.

Note

When you apply a tag to a conversation, you really just apply the tag to the items that make up the conversation at that point in time. Exchange knows that the items are part of the conversation and is able to apply the tag, but it won't look for and tag new items as they arrive and join the conversation. This is because a conversation is not a real storage container within a mailbox and therefore cannot be tagged permanently. In short, tags only exist in a persistent manner for folders and individual items.

Of course, to make any sense of retention policies, you also need to deploy clients that include the necessary intelligence and user interface. At the time of writing, the only clients in this category are Outlook 2010 and Outlook Web App. As we'll see when we review how retention policies function from a user perspective, Outlook's user interface provides the richest views of retention policies and tags. Outlook Web App is less capable, but still highly usable.

System tags

Exchange 2010 supports two types of retention tags: system tags and nonsystem tags. System tags are used by Exchange for its own purposes and are not shown when you run the Get-RetentionPolicyTag cmdlet unless you specify the *–IncludeSystemTags* parameter. By default, Get-RetentionPolicyTag only lists nonsystem tags (those created to be used with normal retention policies). To see the system tags defined in an organization, you can execute this command (nonsystem tags will be listed afterward):

```
Get-RetentionPolicyTag –IncludeSystemTags | Format-Table Name, Type, SystemTag
```

```
Name                                Type                SystemTag
-------                             ------              ---------
AutoGroup                           Personal            True
ModeratedRecipients                 Personal            True
Personal 1 Year move to archive     Personal            False
Default 2 year move to archive      All                 False
Personal 5 year move to archive     Personal            False
Personal never move to archive      Personal            False
```

The first two entries (AutoGroup and ModeratedRecipients) are system tags that are used by Exchange to prevent items from accumulating in arbitration mailboxes. The tags instruct the MFA to clean out these mailboxes as items expire. To see details of the retention policy used for arbitration mailboxes and its links to the two system tags, run these commands:

```
Get-RetentionPolicy –Identity 'ArbitrationMailbox'
Get-RetentionPolicyTag –Identity 'AutoGroup'
Get-RetentionPolicyTag –Identity 'ModeratedRecipients'
```

The last four entries are nonsystem tags that belong to the Default Archive and Retention Policy. Exchange automatically applies this policy after a personal archive is enabled for a mailbox. The idea is to provide a set of tags to allow users to control how items are moved into the archive. The tags are revealed by clients after the user's mailbox is processed by the next run of the MFA. The default archive policy is replaced when another retention policy is applied to a mailbox.

CAUTION!

You cannot add system tags to a retention policy that's applied to user mailboxes. Deleting a system tag is also a bad thing as you have no idea of what potential consequences might follow from this event.

Designing a retention policy

Many different retention policy tags can exist within an organization. This allows great flexibility in creating appropriate policies for different groups that work within a company. For example, the finance department might want Exchange to permanently delete everything in the Deleted Items folder more than three days old (the shred principle), whereas users in other departments are not concerned if items survive in the Deleted Items folder for 30 days or more. You can apply a retention policy to members of the finance department that includes a retention policy tag for the Deleted Items folder that instructs the MFA to remove items after three days. The same policy might include a personal tag that allows members of the finance department to mark items that have to be archived for audit purposes after a month in the primary mailbox. The MFA will move items with this tag to the archive mailbox when it processes the mailbox.

Why are you creating this retention policy?

Before you rush to create a retention policy for anyone—even the finance department—you should sit down and determine the why, when, and how for the policy:

- **Why you are implementing the policy? What business need will the policy serve?**
- **When will you implement the policy? What mailboxes will the policy be applied to? How will you communicate the policy to end users so that they understand the purpose of the policy and how it will affect the contents of their mailboxes?**
- **How will you implement the policy? What tags and types of tags are required? What actions will you enforce through tags and what retention periods are used? Do any restrictions exist as a result of other aspects of your deployment? For example, if you use an archiving product from another vendor, you cannot deploy tags to move items into an archive mailbox after a designated period.**

The design for a retention policy might be captured in a simple table format that makes it clear what tags are included in the policy, their purpose, and the folders that are processed

by the MFA. Apart from its other advantages, capturing the design like this makes it easier to communicate the policy to users. Table 15-3 lays out a simple policy that could be applied to help managers cope with overloaded mailboxes.

Table 15-3 Laying out a retention policy

Retention Policy Name:	***Management retention policy***		
Applies to:	***Mailboxes with CustomAttribute7 = "Management"***		
General purpose:	*Automatic clean-out of Inbox and Sent Items folders to encourage users to keep these folders tidy. Items in all other folders can remain in place for a year. Removal of items from the Deleted Items folder after a week and permanent removal of anything filed into the Junk Mail folder after two days. A tag is provided to allow users to mark items for retention for five years.*		
Tag name	*Tag type*	*Applies to*	*Action*
RPT-Inbox	RPT	Inbox folder	Move items to Recoverable Items after 30 days
RPT-SentItems	RPT	Sent Items	Move items to Recoverable Items after 30 days
RPT-Deleted	RPT	Deleted Items	Permanently remove items after 7 days
RPT-JunkMail	RPT	Junk Mail	Permanently remove items after 3 days
DPT-General	DPT	All folders	Move items to Recoverable Items after 365 days
PER-Retain	PER	All folders	Move items to Recoverable Items after 1,825 days (5 years)

Logically, you can only have a single RPT for each default folder within a retention policy. It would be very confusing to have two retention policies compete within a single folder! In addition, a retention policy can only have one default retention policy that applies to all folders.

INSIDE OUT Keep it simple

Exchange allows you to create and apply as many retention policies as you want, but the question of long-term supportability arises. You should also consider the question of how many retention policies are really required for the organization as a whole and attempt to restrict the number to the minimum necessary to meet business needs. A couple of well-designed, logical policies that satisfy the vast bulk of requirements will be easier to create, deploy, and manage on an ongoing basis than a mass of granular policies generated to meet the specific needs of a department or other business group that might disappear following the next corporate reorganization. The more policies that exist, the more potential there is to confuse administrators and users alike.

Exchange uses the date and time when an item is created in a user's mailbox as the baseline to calculate the age of the item for retention purposes, so an age limit of 30 days for the Inbox default retention tag essentially means that items become eligible for processing by the MFA 30 days after they are delivered into the Inbox. The creation date is used for retention purposes even for modifiable items such as posts. You can create a tag to mark items never to be processed by the MFA. Such a tag will have no value set for its *AgeLimitForRetention* property, and its *RetentionEnabled* property will be set to $False.

The MFA is responsible for implementing the actions specified in retention and archive tags when it processes a mailbox. For example, if the retention period for the Inbox is 30 days, the MFA will tag any item aged up to 30 days and take the specified action for items aged 30 days. Therefore, before you implement a policy that potentially will affect thousands of items in user mailboxes, it is critical to clearly communicate what is going to happen, when it will happen, and how users can prepare for the implementation of the retention policy and respond to its actions afterward. You might have to communicate several times before the retention policies are implemented to avoid a deluge of calls to the help desk the morning after the MFA runs.

Naming retention tags

The tags described in the example management retention policy that we created follow a specific naming scheme. Retention policy tags are prefixed with "RPT," default policy tags are prefixed with "DPT," and personal tags are prefixed with "PER." The tag name is then completed with some text to convey its meaning and to associate it with the retention policy where it is used. Thus, DPT-General makes sense in an administrative sense because the name conveys that the tag is a default policy tag used generally across the organization. Of course, the last sentence means nothing to end users, especially if they have never coded and have not been exposed to the cryptic (but always logical) naming schemes beloved by programmers.

The problem that has to be solved when you determine a tag naming scheme is that the retention policy menu displayed by Outlook 2010 and Outlook Web App lists tag names and their retention period (such as "6 months") to end users but doesn't display any other detail such as the action that will be taken when the tag's retention period expires. Tags can have a variety of associated actions, from permanent deletion to merely warning that the retention period has expired. Outlook 2010 users can view the actions for the default tag on a folder by viewing the folder properties, but this information is not available to Outlook Web App, and they are the only two clients that expose retention tags today. It can be argued that the tags used in an archive policy and displayed in the archive menu are an exception because users should know that the purpose of these tags is to move items into the personal archive when their retention period expires, but that's still no reason to use cryptic tag names.

The question, therefore, has to be asked whether you should use a more user-friendly naming scheme for retention tags. For example, would "RPT-Inbox" be better named "Inbox retention policy" and should "PER-Retain" be called "Retain for five years"? Some prefer the structure of the first approach, but users probably find the second approach easier to understand.

Another approach that is often taken is to use names that give clear business directives for retention tags. For example, you might use names such as these:

- Business Critical
- Partner Negotiations
- Legal Retention

Tags named like this are usually more specific to departments or groups than more generic names such as "Keep for five years" or "Required for Annual Audit," so you might end up defining a set of retention tags for each department to match their work practices.

It's impossible to give a definitive answer about a naming convention that is suitable for all deployments. Some organizations are happy with cryptic tags because they are a standard that is valid no matter what language is used to connect to Exchange; others will elect to use more user-friendly names because it's easier to communicate the purpose of a retention policy to users and they feel that this will both ease the introduction of retention policies within the organization and avoid some calls to the help desk. The important thing is to make a decision before you start to design and implement retention policies, as changing the names of tags halfway through a deployment is guaranteed to cause maximum confusion.

Creating retention tags

Retention tags can only be created using EMS with Exchange 2010 RTM. With SP1, you have the choice of working with EMC or EMS. EMC is easier to deal with, so we'll begin with it. Under Organization Configuration, go to the Mailbox section and select the New Retention Tag option to launch the New Retention Policy Tag Wizard. As you can see from Figure 15-7, this wizard is very straightforward, and all we need to do is input the settings laid out for each tag in the policy described previously in Table 15-3.

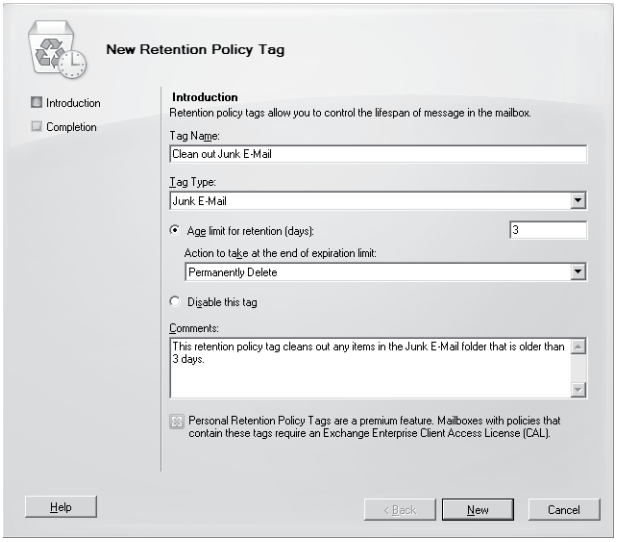


Figure 15-7 Creating a new retention policy tag.

After creating all of the retention policy tags that we need, we should end up with something like the situation illustrated in Figure 15-8. This is the complete set of the retention policy tags defined for the organization, and you can immediately see the advantage of following a well-thought-out naming convention for tags, as this set contains both structured and free-form names. In addition, you can see how it is possible to quickly accumulate a large number of tags that are used by different retention policies in an organization. With some forethought, it is possible to reduce the total number of tags by designing some utility tags that are included in every policy, which then means that the only additional tags that you need to define are those specifically required by a policy. For example, you can probably define utility tags to clean out folders such as Junk E-Mail and RSS Feeds that apply the same retention period and action for every policy. You might not be as successful in defining utility tags for default folders, such as the Inbox or Sent Items, as different sets of users might need to keep items in these folders for different periods.

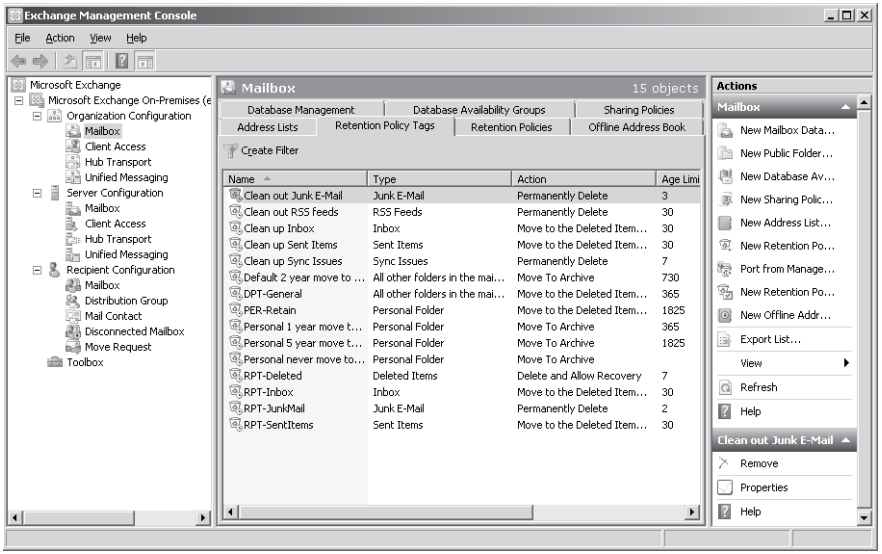


Figure 15-8 Viewing the set of retention policy tags defined for the organization.

As an example of how to accomplish the same task with EMS, let's create the four retention policy tags for the Inbox, Sent Items, Junk Mail, and Deleted Items folders.

```
New-RetentionPolicyTag –Name 'RPT-Inbox' –RetentionAction DeleteAndAllowRecovery
–AgeLimitForRetention
30 –Type Inbox –Comment 'Inbox items are automatically deleted after 30 days'
–RetentionEnabled $True

New-RetentionPolicyTag –Name 'RPT-SentItems' –RetentionAction DeleteAndAllowRecovery
–AgeLimitForRetention 30 –Type SentItems –Comment 'Sent Items are deleted
after 30 days' –RetentionEnabled $True

New-RetentionPolicyTag –Name 'RPT-JunkMail' –RetentionAction PermanentlyDelete
–AgeLimitForRetention 2 –Type JunkEmail –Comment 'All junk mail is permanently
removed after two days'
 –RetentionEnabled $True

New-RetentionPolicyTag –Name 'RPT-Deleted' –RetentionAction DeleteAndAllowRecovery
–AgeLimitForRetention 7 –Type DeletedItems –Comment 'Deleted Items are removed after
7 days; they can be recovered if necessary'     –RetentionEnabled $True
```

We can check the properties of our new retention tags with the Get-RetentionPolicyTag cmdlet. For example:

```
Get-RetentionPolicyTag -Identity 'RPT-Inbox' | Format-List
```

```
IsPrimary                          : False
MessageClassDisplayName            : All Mailbox Content
MessageClass                       : *
Description                        : Managed Content Settings
RetentionEnabled                   : True
RetentionAction                    : DeleteAndAllowRecovery
AgeLimitForRetention               : 30.00:00:00
MoveToDestinationFolder            :
TriggerForRetention                : WhenDelivered
MessageFormatForJournaling         : UseTnef
JournalingEnabled                  : False
AddressForJournaling               :
LabelForJournaling                 :
Type                               : Inbox
SystemTag                          : False
LocalizedRetentionPolicyTagName    : {}
Comment                            : Inbox items are automatically deleted after 30 days
LocalizedComment                   : {}
MustDisplayCommentEnabled          : False
LegacyManagedFolder                :
AdminDisplayName                   :
Name                               : RPT-Inbox
Identity                           : RPT-Inbox
```

The output from Get-RetentionPolicyTag confirms that the tag can cover any class of item (MessageClass = *, the default for Exchange 2010), that it is for the Inbox folder (Type = Inbox), and that items tagged with this RPT will be moved to the Deleted Items folder after 30 days (indicated in the *RetentionAction* and *AgeLimitForRetention* properties). In fact, unlike managed folders, retention tags don't accommodate the notion of item segregation. In other words, you cannot build a retention tag that only applies to items of a certain class in a folder (such as apply the policy to items of class *IPM.Note* but ignore those of class *IPM.Contact*).

INSIDE OUT

Voice mail is an exception

Voice mail is a noted exception to this rule because you can create a specific tag for voice mail. Along the same lines, you can't define different actions for different item types such as moving expired messages to an archive folder while deleting any other item type. Some observers consider these shortcomings to be a retrograde step in messaging records management.

Let's now create the other tags that are required for the management retention policy. This time a personal tag (type = personal) is needed to allow users to mark items to be kept in their mailbox for five years (1,825 days), after which the items will be automatically moved into the Recoverable Items folder. Exchange gives the action and retention period defined in a personal tag priority if a user applies it to an item in a folder that's already under the control of a retention policy tag. In other words, if a user applies the PER-Retain tag on an item in the Inbox, Exchange will not move it to the Recoverable Items folder after 30 days as called for by the retention policy tag associated with the Inbox. Instead, Exchange will respect the action and retention period defined in the personal tag because the rule is that an explicit policy always trumps an implicit policy. In addition, you should also remember that Exchange will keep a personal tag on an item even if the item moves to another folder that has an associated retention policy tag.

We create the new personal tag with the following command:

```
New-RetentionPolicyTag –Name 'PER-Retain' –RetentionAction PermanentlyDelete
–RetentionEnabled $True –AgeLimitForRetention 1825 –Type Personal –Comment 'Item to
be kept forfive years before it is moved to Recoverable Items'
```

Setting the *Type* parameter to Personal is the critical thing here because it makes the tag personal and explicit rather than the implicit tags applied to all items in a folder. To create a personal tag with EMC, select Personal Folder as the tag type.

TROUBLESHOOTING

I created a retention tag with the wrong type. What do I do?

If you make a mistake and create a retention tag of the wrong type, you aren't able to change the type with the Set-RetentionPolicyTag cmdlet. Instead, you will have to delete the tag with the Remove-RetentionPolicyTag cmdlet and then re-create it afterward with the correct type.

To complete the design, the policy needs to provide managers with a default retention tag that forces any items older than a year (365 days) to be moved into the Recoverable Items folder. As you'll recall, a default tag is used when no other tag has been applied to an item.

```
New-RetentionPolicyTag –Name 'DPT-General' –RetentionAction DeleteAndAllowRecovery
–RetentionEnabled $True –AgeLimitForRetention 365 –Type All –Comment 'Items older
than a year are moved to Recoverable Items unless otherwise tagged'
```

You'll have noticed that all of the tags that we created specify *–RetentionEnabled $True*. This means that the tag is active and should be processed by the MFA. To disable a tag, you set *–RetentionEnabled $False*. A tag in this state is ignored by the MFA.

Most of the tags that we have created so far use the DeleteAndAllowRecovery action. The other actions are as follows:

- ***PermanentlyDelete*** Immediately deletes the item in such a way that it cannot be seen using the Recover Deleted Items option. If the mailbox is on retention or litigation hold, the item is retained and still available to discovery searches.
- ***MoveToArchive*** Moves the item to a folder of the same name in an archive mailbox. Clearly this action is only possible if the mailbox has a personal archive. If not, the action is ignored. Moving to the archive is analogous to the Outlook Auto-Archive option that moves items into a PST on a regular schedule to help keep a mailbox under quota. The big difference is that users don't get to vote whether they want to use the option, as Exchange moves items into the personal archive automatically without asking for user opinion. Policies that move items into an archive mailbox are known as archive policies. Exchange will ignore the archive tags if you create a retention policy that includes tags to move items into the archive and apply it to a mailbox that doesn't have a personal archive. If the mailbox is subsequently assigned a personal archive, the MFA will apply the archive tags for the mailbox the next time that it runs.

To check that we have all of the required tags in place to build the retention policy, we can review the set of tags through EMC or execute the following EMS command:

```
Get-RetentionPolicyTag | Format-Table Name, Type, RetentionAction, RetentionEnabled,
AgeLimitForRetention -AutoSize
```

```
Name            Type         RetentionAction          RetentionEnabled    AgeLimitForReten-
tion
--------        ------       --------------           --------------      -----------------
RPT-Inbox       Inbox         DeleteAndAllowRecovery  True                30.00:00:00
RPT-SentItems   SentItems     DeleteAndAllowRecovery                      30.00:00:00
RPT-JunkMail    JunkEmail     PermanentlyDelete       True                2.00.00.00
RPT-Deleted     DeletedItems  PermanentlyDelete       True                7.00:00:00
```

```
PER-Retain      Personal      DeleteAndAllowRecovery  True              1825.00:00:00
DPT-General     All           DeleteAndAllowRecovery  True              365.00:00:00
```

All seems correct in this case, but if you make a mistake, you can remove a retention policy tag with the Remove-RetentionPolicyTag cmdlet. If the tag has already been applied to mailbox items, the MFA will clean up by removing any reference to the removed tag from items as it processes mailboxes.

Creating a retention policy

Now that we have created the necessary retention tags to help managers impose order on their mailboxes, we can proceed to create a new retention policy. In EMC, under Organization Configuration in the Mailbox section, select the New Retention Policy option to launch a wizard to help guide us through the process. We have two tasks to accomplish:

1. Select and add the retention tags that we want to include in the new policy. Figure 15-9 shows that the six tags that we created have been selected for inclusion in the new policy.

2. Select the mailboxes that we want to apply the new retention policy to after it is created (Figure 15-10). This is an optional step, and you are not required to select any mailboxes now. A retention policy can be added to mailboxes at any time after it is defined.

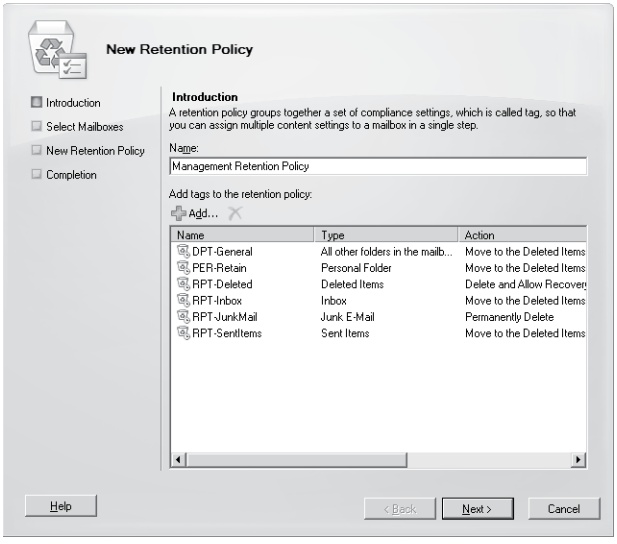


Figure 15-9 Creating a new retention policy with EMC.

Figure 15-9 illustrates a retention policy containing a number of tags that specify actions that have been removed in Exchange 2010 SP1. Microsoft no longer supports the use of the Move to the Deleted Items and Mark as Past Retention Limit actions. Although these actions are still respected by the MFA, they might cease to work in a future release of Exchange. The policy shown here was created with Exchange 2010 RTM and needs to be updated to use the set of retention actions available in SP1. It is possible that Microsoft will reintroduce a more expansive set of retention actions in a future version of Exchange.

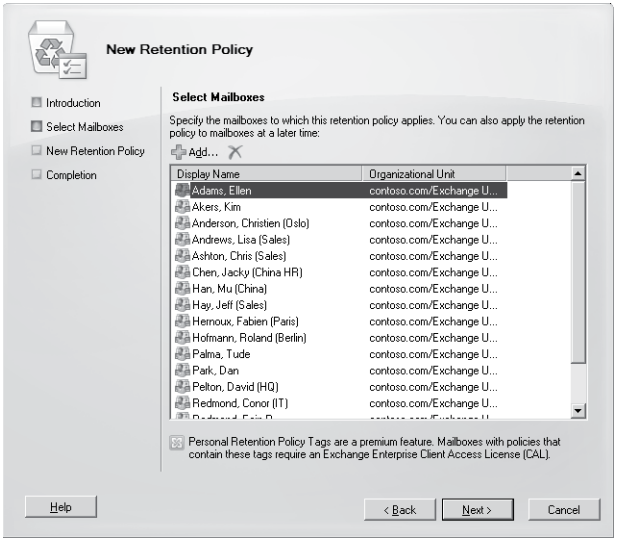


Figure 15-10 Adding mailboxes to a new retention policy.

When you click Finish to complete the wizard, Exchange first reviews the set of retention policy tags that you have assigned to the policy to validate that you are not including multiple tags for the same folder.

TROUBLESHOOTING

I created multiple tags for the same folder.

This is a common error, and if it is detected, Exchange flags the error as shown in Figure 15-11. Fixing the error is easy: Note the problem tag reported by Exchange and then check it against the set of tags that you have included in the policy to determine which of the duplicate tags you want to retain in the policy. Once you've addressed the problem, go through the wizard pages again to finish and create the new retention policy.

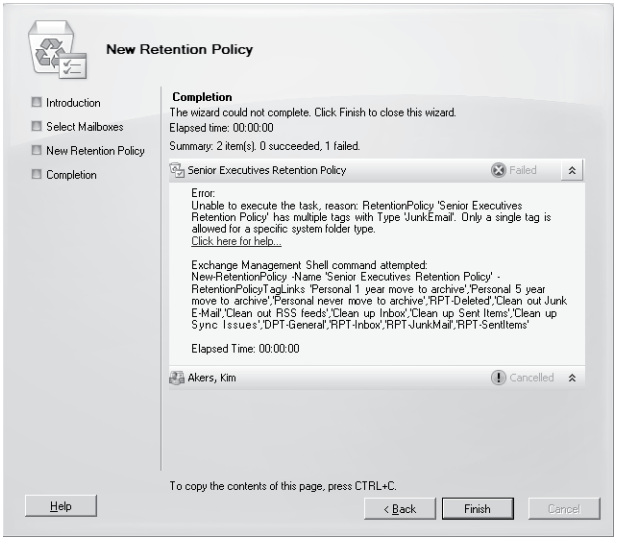


Figure 15-11 Encountering an error in creating a new retention policy.

Assuming that everything goes according to plan, Exchange will create the new retention policy and then apply it to any mailboxes that you selected. The MFA will stamp the tags defined in the retention policy on items in the mailboxes the next time that it processes the mailboxes.

Exchange 2010 only allows you to create retention policies through EMS using the New-RetentionPolicy cmdlet. In this command, we create the policy and associate the six tags that we want to use with the new policy.

```
New-RetentionPolicy –Name 'Management retention policy'
–RetentionPolicyTagLinks 'RPT-Inbox', 'RPT-SentItems', 'RPT-Deleted',
'PER-JunkMail','PER-Retain', 'DPT-General'
```

We can examine details of the new retention policy with the Get-RetentionPolicy cmdlet:

```
Get-RetentionPolicy –id 'Management retention policy'
```

```
RetentionPolicyTagLinks : {DPT-General, Per-retain, RPT-Deleted, RPT-SentItems, RPT-Inbox}
AdminDisplayName        :
ExchangeVersion         : 1.0 (0.0.0.0)
Name                    : Management retention policy
```

```
DistinguishedName       : CN=Management retention policy,CN=Retention Policies Container,
CN=contoso,CN=Microsoft Exchange,CN=Services,CN=Configuration,DC=contoso,DC=com
Identity                : Management retention policy
ObjectClass             : {top, msExchRecipientTemplate, msExchMailboxRecipientTemplate}
```

A six-tag policy is a reasonably simple retention policy, as other policies can incorporate a lot more tags to create a very exact retention environment for a user to operate within. Obviously, you might have more tags than this if you decide to include a retention policy tag for every default folder. Using retention policy tags to clean out items that otherwise accumulate and are never cleared out in default folders such as Sync Issues, Junk E-Mail, and RSS Feeds is a good example of where you can gain real value from a well-designed retention policy. Figure 15-7, shown previously, is a good example of such a "good folder health" retention policy tag.

INSIDE OUT

Good reasons to limit the number of tags in a policy

Microsoft recommends that you have no more than 10 personal tags in a policy, as otherwise you might confuse users with too much choice. This is reasonable advice, but as with most advice, there will be edge cases where you need to incorporate more tags in a policy to meet specialized business needs. A more sophisticated policy for a department might have separate retention tags for many of the default folders, a set of personal tags developed specifically to suit the retention needs of the department, and a default retention tag for everything else. User interface constraints are another good reason for limiting the number of tags in a policy. If you have 10 tags or fewer in a policy, there's a reasonable guarantee that Outlook Web App and Outlook will be able to display all the tags in their user interface. On the other hand, if you have 20 tags in a policy, you'll find that Outlook Web App is unable to list all of the tags and some will simply drop off the end of the available list. Outlook has a separate dialog box that users can navigate to if they want to discover all of the tags available in a policy, but this requires many separate clicks and additional knowledge of where to go to find the tags. For all these reasons, it's just a bad idea to go crazy and create a tag-filled policy.

Applying a retention policy to mailboxes

The Set-Mailbox cmdlet is used to apply a retention policy to an existing mailbox. The New Mailbox Wizard available in the original release of Exchange 2010 does not allow you to set a retention policy on a mailbox when it is created, but this problem is addressed in

the SP1 version. The policy becomes active the next time the MFA processes the mailbox. SP1 also allows you to select a retention policy from the Mailbox section of Organization Configuration and add one or more mailboxes to it. Because you can use the mailbox picker to browse and select from all the mailboxes in the organization, this is the easiest way to add a large number of mailboxes to a retention policy in one operation.

INSIDE OUT **Only one retention policy—ever**

A mailbox can only ever have one retention policy, so when you assign a retention policy to a mailbox, the action overwrites any policy that might already be in place. You can change retention policies multiple times on a mailbox, but this isn't a good idea unless you really need to switch policies because the effect of the different policies might confuse users, as the MFA responds to different retention settings in the different policies. Setting a value for the *RetentionURL* parameter is not compulsory, but it is a useful way to communicate where a user might go to find additional details about the company's retention policy. This URL is only visible through Outlook 2010 and isn't displayed by earlier clients or Outlook Web App.


```
Set-Mailbox –Identity 'JSmith' –RetentionPolicy'Management retention policy'
–RetentionComment 'Management retention policy applies to this mailbox'
-RetentionURL'http://Intranet.contoso.com/RetentionPolicies.html'
```

Exchange will warn you that clients earlier than Outlook 2007 don't support retention policies. More correctly, this should be Outlook 2010. Of course, if you're setting a policy for a group of users, you'll probably do it in one operation by selecting the mailboxes with the Get-Mailbox cmdlet and piping the results to Set-Mailbox. For example:

```
Get-Mailbox –Filter {CustomAttribute7 –eq'Management'} | Set-Mailbox
–RetentionPolicy 'Management retention policy'
–RetentionComment 'Management retention policy applies to this mailbox'
```

The new policy will be applied to the mailboxes the next time that the MFA processes the mailboxes. See the section "How the Managed Folder Assistant implements retention policies" later in this chapter for more information about the processing performed by the MFA.

To discover the set of mailboxes that have retention policies in place, you can use a command like this:

```
Get-Mailbox –Filter {RetentionPolicy –ne $Null} | Format-Table Name, RetentionPolicy
–AutoSize
```

The value of $Null

When you want to remove a retention policy from a mailbox, you simply set the policy to *$Null*. For completeness, it's a good idea to set the other properties associated with retention policies to null as well. Here's the command:

```
Set-Mailbox –Identity 'JSmith' –RetentionPolicy $Null –RetentionComment $Null
–RetentionURL $Null
```

Of course, after you begin to deploy retention policies to mailboxes, the question arises of how to integrate the assignment of retention policies with any user provisioning process that your company has in place. Exchange doesn't have a default retention policy that can be assigned automatically, so an explicit administrative action is always required to allocate a retention policy to a mailbox. This action is not difficult to code with EMS, but it is something that needs to be considered as part of your deployment plan.

Modifying a retention policy

Policies can evolve over time by the addition or removal of tags. As we've discussed, you add multiple retention tags to a policy by separating the entry for each tag with a comma. You can add new tags to the policy afterward with the Set-RetentionPolicy cmdlet. To add a new tag, you need to include it in the full list of tags submitted to Set-RetentionPolicy in the *–RetentionPolicyTagLinks* parameter. It is not sufficient to merely specify the new tag on its own, as this will update the policy to only include the new tag.

You can use two approaches to including a new tag in a retention policy. The first approach is best for simple policies that only include a few tags and requires you to write the complete list of tags into the policy to overwrite the existing list. For example:

```
Set-RetentionPolicy –Identity 'Audit Department'
–RetentionPolicyTagLinks 'RPT-Audit-Inbox', 'RPT-Audit-SentItems'
Get-RetentionPolicy –Identity 'Audit Department'
```

The second approach is best when dealing with complex policies that have six or more tags, and the potential exists that you might forget to input one of the tags. *RetentionPolicyTags* is a multivalued property, so to add a new tag to an existing list, you first extract the existing tags into a variable, then add the new tag to the variable, and finally write the new set back into the policy. Here's code that updates a complex policy with a new tag:

```
$TagList = (Get-RetentionPolicy –Identity
'Management Retention Policy').RetentionPolicyTagLinks
$NewTag = Get-RetentionPolicyTag –Identity 'Per-New-ArchivePolicy')
$TagList += $NewTag
Set-RetentionPolicy –Identity 'Management Retention Policy' –RetentionPolicyTagLinks
$TagList
```

```
Get-RetentionPolicy –Identity 'Management RetentionPolicy' | Select Name,
RetentionPolicyTagLinks
```

The second approach requires a little more typing on the part of the administrator, but it absolutely guarantees that all existing tags are preserved.

To remove a tag from a policy, you have to write a replacement list into the policy as in the first approach previously described. If you remove a tag from a policy, users covered by the policy cannot apply the tag to any items to their mailbox, but existing items that have been stamped with the tag continue in place and will be processed by the MFA.

Changing a retention tag: An exception to the rule

Changing the retention period in a tag is similar to removing a tag. All items stamped with the retention period up to the point where you made the change will continue to use that retention period; items that are stamped with the updated policy will have the new retention period. The exception to this rule is when you set the *RetentionEnabled* property of a tag to $False, as this value instructs the MFA to leave the tags in place but ignore them when it processes items. For example:

```
Set-RetentionPolicyTag –Identity 'Keep Items Forever' –RetentionEnabled $False
```

This situation continues until the user explicitly assigns a replacement tag to an item or you remove the tag from Active Directory using the Remove-RetentionPolicyTag cmdlet. When this happens, the next time that the MFA runs, it will remove the deleted tag from any items where it was used.

Customizing retention policies for specific mailboxes

You can tailor the retention policy for a specific user by assigning personal tags on a per-mailbox basis. This can only be done if a retention policy already applies to the user's mailbox. For example, let's assume that you want to assign a new personal tag to a user to allow him to mark an item to be moved into the archive after a year. You can do this as follows:

```
Set-RetentionPolicyTag –Mailbox JSmith –OptionalInMailbox 'Per-Move-Archive'
```

Exchange adds the optional tag to the set of tags covered in the retention policy that already applies to the mailbox and makes the expanded set available the next time that the user connects. Unfortunately, no cmdlet is available to report whether a mailbox has been assigned optional tags. If you examine a mailbox with Get-Mailbox, it tells you if a retention policy is assigned, but nothing else. Therefore, if you want to change the list of optional tags assigned to a mailbox, you have to write the complete list with Set-RetentionPolicyTag.

For example, to add an additional tag to the one that has already been assigned, we use this command:

```
Set-RetentionPolicyTag –Mailbox JSmith –OptionalInMailbox 'Per-Move-Archive',
'Per-Keep-LongTime'
```

EMS doesn't validate that the tags that you assign to a mailbox will be effective. For example, you can assign a new archive tag to a mailbox that doesn't have a personal archive. This is really a null operation because neither Outlook Web App nor Outlook displays archive tags if the mailbox doesn't have a personal archive.

To remove all optional retention tags from a mailbox, you set the list to *$Null* as follows:

```
Set-RetentionPolicyTag –Mailbox JSmith –OptionalInMailbox $Null
```

INSIDE OUT Accessing personal tags in Exchange 2010 SP1

Exchange 2010 SP1 makes the process of accessing personal tags easier by allowing users to see a list of available personal tags through Exchange Control Panel (ECP) to decide what personal tags they would like to use. Figure 15-12 shows the option exposed through the Organize Email section of ECP.

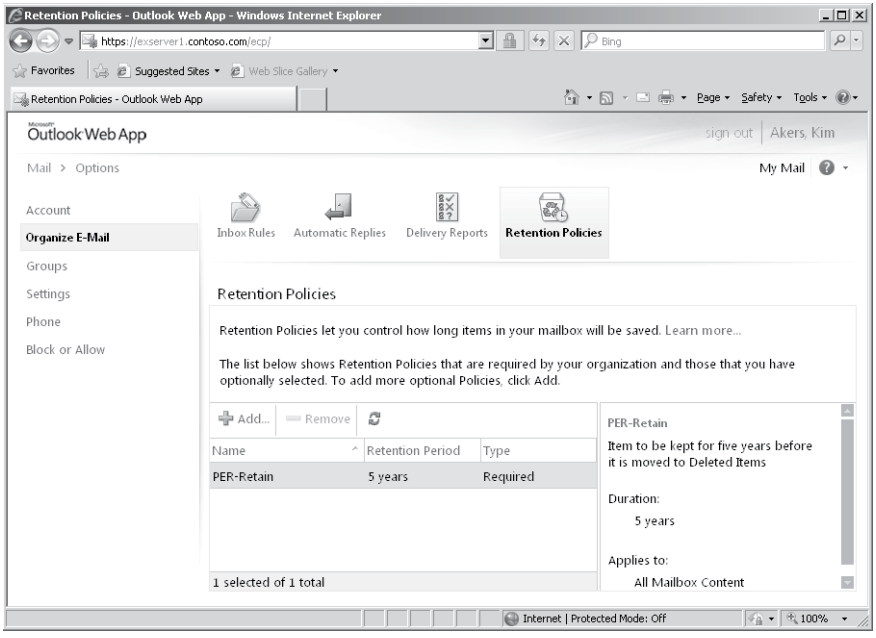


Figure 15-12 User access to personal retention tags through ECP.

A retention policy must be in effect for a user's mailbox, and the MyRetentionPolicies policy must be included in the role assignment policy for the mailbox before ECP reveals personal tags. If shown, the user sees the personal tags that she can already use because they are included in the retention policy (these tags are listed as Required) and the other personal tags that are defined for the organization that she can choose to use (these tags are listed as Optional). The user cannot remove any of the Required tags because their presence is mandated by the retention policy that is applied to the mailbox. A user can begin to apply personal retention tags to items immediately after adding the tags to her mailbox.

User interaction with retention policies

The first evidence that users see that their mailbox has been assigned a retention policy is when they see indications in message headers that start to appear 30 days before an item expires. These warnings are visible when a message is opened or shown in the message preview. Figure 15-13 shows how Outlook Web App advises that a message has 27 days before it expires as the result of a retention policy tag placed on the Inbox. The user now has the choice to either leave the message to expire, in which case the MFA will process whatever action is defined in the tag (Move to Deleted Items, Permanently Delete, and so on) or apply a different tag to the item.

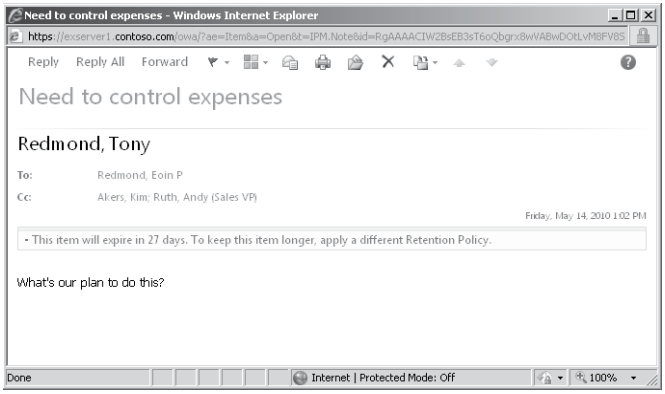


Figure 15-13 Outlook Web App warns that an item is approaching its expiry deadline.

Users have two options to apply a different tag to an item in their Inbox. First, they can move the item to a different folder and so remove it from the influence of the retention policy tag that applies to Inbox items. After it is moved, the item is governed by the default policy tag defined in the retention policy that applies to the mailbox, if one exists, or by an explicit policy that is applied to the folder and therefore inherited by all items that are

added to the folder. If neither of these conditions exists, the item is left untagged and therefore will not be subject to processing by the MFA.

The second option is to place an explicit tag on the item. Users can choose from any of the personal tags defined in the retention policy applied to their mailbox by right-clicking an item and then selecting the personal tag to apply. Figure 15-14 shows how Outlook 2010 (left) and Outlook Web App (right) display retention and archive tags included in a single retention policy in the list of options that can be taken for a message. If a tag specifies *MoveToArchive* as its action, clients list it under Archive Policy rather than Retention Policy. Logically, archive tags can only be used with mailboxes that have personal archives. Outlook provides a richer set of options, even if you can argue that Outlook Web App's user interface is less confusing for the novice user. You won't see the user interface for retention policies unless a policy is applied to your mailbox.

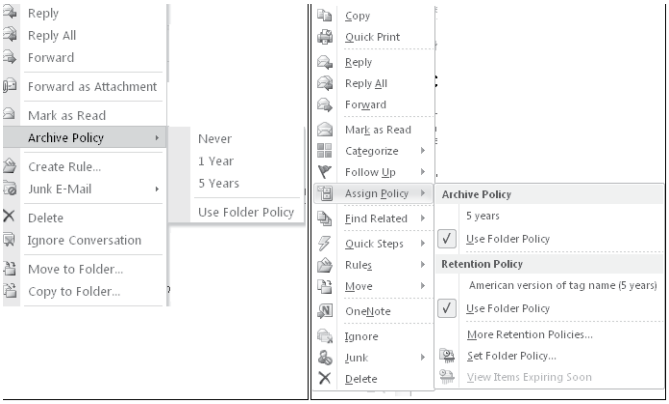


Figure 15-14 How Outlook and Outlook Web App display the list of available retention and archive tags.

After a personal tag has been applied to an item, the item is no longer subject to the provisions of either the folder policy or the default policy, as an explicit tag always takes precedence over a tag placed on a folder. The personal tag also remains with the item if it is moved to another folder or into the personal archive. If users want to impose a different retention policy on the item, they will have to replace the existing tag with a new personal tag.

Outlook keeps users updated about the retention policy that applies to an item by displaying details as part of the message header. The retention information displayed by Outlook 2010 when a message is read is shown in Figure 15-15. Users can see quite clearly how long it will be before the item expires, and they can also see details of the retention policy that is applied to the item.

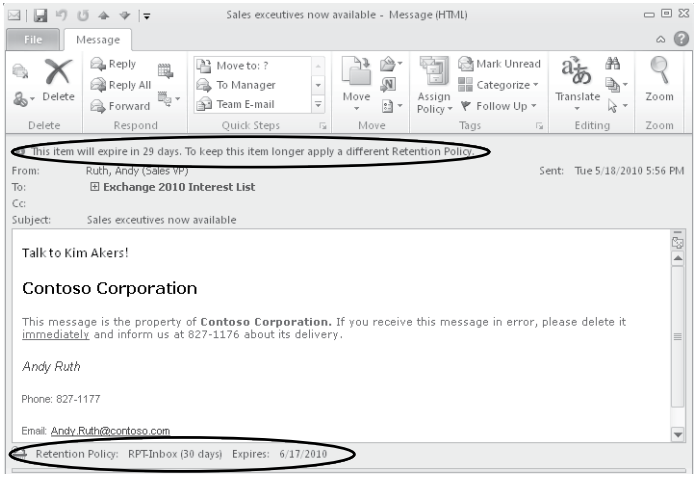


Figure 15-15 Outlook 2010 displays retention information.

Managed Folder Assistant automatically applies the retention policy

Although most retention policies will include a default policy tag to provide retention instructions for items held in nondefault folders, Outlook and Outlook Web App support the use of personal retention tags to set a different retention policy on a folder. In effect, this means that Exchange will apply the policy defined in the personal retention tag to items held in the folder, much in the same way that it applies the retention policy tags placed on default folders such as the Inbox. In some respects, you can use this approach to create a roughly equivalent situation to the functionality provided by Exchange 2007 managed folders. However, the big difference is that you have to create the folders and apply the retention policies manually, whereas the MFA does the work to push out new folders to user mailboxes and apply the retention policy automatically for managed folders.

To set a new default policy for a folder with Outlook, select the folder and click Assign Policy on the toolbar, then select Set Retention Policy from the drop-down menu. Outlook then displays the folder properties positioned on the Policy tab (Figure 15-16). You can select any personal tag to use as the new default retention policy for the folder, and items subsequently created in the folder will inherit the default tag. The same inheritance occurs when an item is moved into the folder unless an explicit tag has already been applied to the item, in which case the existing tag is retained. To set a default policy on a folder with Outlook Web App, select the folder from the folder list under the mailbox root, right-click

to select the Retention Policy option, and then select the retention policy to apply to the folder.

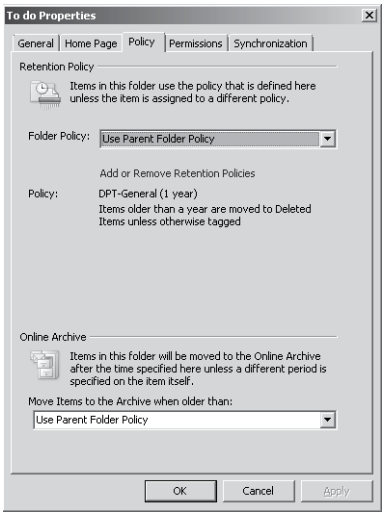


Figure 15-16 Changing the default retention policy for a folder.

Figure 15-17 shows how Outlook 2010 is able to display some information about retention policy in its backstage area. In this example, if we look at the mailbox, we'll see that its properties are as follows:

```
Get-Mailbox –Identity 'Ruth, Andy' | Select Retent*
```

```
RetentionComment   : The management retention policy applies to this mailbox
RetentionUrl       : <a href="http://intranet.contoso.com/retentionpolicies.html"> Retention
Policy Information</a>
RetentionPolicy    : Management retention policy
```

The *RetentionComment* property provides the text that you can see beside "Account Settings," and the *RetentionUrl* property is used to provide a URL to a Web site where the additional information resides. These properties are usually set when you place a mailbox on retention or litigation hold. We will come to these topics shortly. For now, although you don't have to set these properties to impose an effective retention regime, they are helpful to communicate information to users about what's going on in their mailbox. Experience of many projects demonstrates that anything that assists in effective communications with users is likely to reduce help desk calls. Apart from the two properties that we can set on a mailbox, Outlook tells the user that the default archive policy for the mailbox will move items out of the primary mailbox after they are two years old.

Figure 15-17 Viewing retention information in Outlook's backstage area.

Figure 15-18 shows that you can provide localized versions of retention tags that Outlook will display to users based on the client language setting.

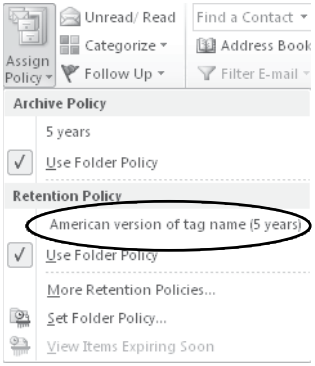


Figure 15-18 Viewing a localized retention policy tag.

Note

Clearly you don't have to go to the trouble of creating local language versions of tags if you operate a mono-language environment, but once again this is a useful thing to do when you have to support users who operate in different languages. Just be sure that you get accurate translations that clearly convey the meaning of the tag, and don't be tempted to cut corners and use school-quality translations or even those that you might be able to procure free of charge from the Internet.

If we examine the tag that generates the output, we can see this output:

```
Get-RetentionPolicyTag –Identity 'Per-Retain' | Select Local*, Comment
```

```
LocalizedRetentionPolicyTagName : {En-IE: Irish version of tag, En-US: American
version of tag name}
LocalizedComment                : {En-US: General five year tag (US), En-IE:
General five year tag (IRL)}
Comment                         : Item to be kept for five years before it is
moved to Deleted Items
```

Notice that the *LocalizedRetentionPolicyTagName* property has two values for "En-IE" (Ireland variant of English) and "En-Us" (U.S. variant of English). The screen shows that Outlook displays the U.S. version, so you know that Outlook is running in that language version. An example of the command to provide localized text for a retention tag is:

```
Set-RetentionPolicyTag –Identity 'Per-Retain' –LocalizedRetentionPolicyTagName
'EN-US: American version of tag name', 'EN-IE: Irish version of tag name'
–LocalizedComment 'EN-US: US text comment', 'EN-IE: Irish text comment'
```

Removing a retention policy

The Remove-RetentionPolicy cmdlet is used to remove a retention policy from the organization. For example:

```
Remove-RetentionPolicy –Identity 'Retention Policy - PR Department'
```

Removing a retention policy has the effect of removing the policy from any mailboxes to which it is currently applied. If any mailboxes are associated with the policy, EMS will prompt you to confirm its removal. If you proceed, Exchange removes the reference to the now-deleted policy from the mailboxes. Exchange can't decide what retention should replace the one that has just been removed, so no policy is applied. Locating the mailboxes to which a retention policy is applied is therefore a proactive step that you should take before you remove the policy. You can scan mailboxes to discover where a retention policy is applied with a command like this:

```
Get-Mailbox | Where {$_.RetentionPolicy –eq "Retention Policy - Audit Department"} |
Select Name
```

A similar set of commands can be run to locate mailboxes with a specific retention policy and assign a new retention policy to the mailboxes. For example:

```
Get-Mailbox | Where {$_.RetentionPolicy –eq "Retention Policy – Audit Department"} |
Set-Mailbox –RetentionPolicy 'New Retention Policy for Auditors'
```

Upgrading from managed folders

You can upgrade a managed folder to a retention tag by using it as the template to create a new tag. For example, let's assume that you have a managed folder called Never Delete that acts as a repository for items that users never want to have removed from a mailbox because they are so important. You could argue the case that these items could be equally stored in an archive mailbox. However, archive mailboxes didn't exist in Exchange 2007, and it takes time for people to change their behavior. We can use a command like the one shown here to create a new retention policy tag from the Never Delete managed folder:

```
New-RetentionPolicyTag –Name 'Mark item to never expire' –ManagedFolderToUpgrade
'Never Delete' –Comment 'Tag created from old Never Delete managed folder'
```

Of course, to complete the process, we have to associate the new tag with a retention policy and assign it to a user. After this is done, the user will be able to apply the new tag on any item in his mailbox rather than just the items placed in the managed folder.

How the Managed Folder Assistant implements retention policies

After you apply a retention policy to a mailbox, you can either wait for the next scheduled run of the MFA or start it manually so that the new policy is applied immediately. When the MFA runs, it performs the following tasks:

- It applies the tags specified in retention policies to the mailboxes covered by these policies and stamps the items in the various folders covered by the policies with the appropriate tag name and expiration date.
- It populates new managed folders into mailboxes that are under the control of a managed folder policy.
- If a policy defines a retention or expiry period for items, it stamps a Messaging Application Programming Interface (MAPI) property (*ElcMoveDate*) on the items indicating the date and time from which the retention period will start. A future run of the assistant can then use this date and time to calculate when to delete an item or mark it as expired.
- It locates items in folders that are past their expiration date and takes whatever action is defined in the policy (delete, age out, move to another folder).
- If required by policy, it journals new items that have been placed in managed folders. In this context, journaling is different than that performed by transport rules because items are only processed when the MFA is active rather than immediately when they arrive into the folder. The MFA does not use the transport engine to journal items

because there is no guarantee that the transport role is installed on the mailbox server that hosts the managed folders.

The default schedule for the MFA on Exchange 2010 mailbox servers extends from 1 A.M. to 9 A.M. daily. On small servers that host a few hundred mailboxes, the MFA invariably has plenty of time to complete processing of all mailboxes during its scheduled timeslot. On large servers where several thousand mailboxes might need to be processed, a run of the MFA might not complete during its timeslot, especially if this is the first time that policies are applied and many items have to be deleted or moved into an archive.

Behind the scenes: When a timeslot expires before processing is complete

The work done by the MFA to process mailboxes, stamp items with retention tags, and action items whose retention period has expired is resource-intensive in terms of server resources and might also create a lot of network traffic within a DAG to replicate all of the store transactions created as items are processed. If the timeslot expires before the MFA completes processing, it will stop and will resume processing at the point where it was forced to stop when the next timeslot becomes available. You can alter the default schedule on the Messaging Records Management tab of the server Properties dialog box, but only on Exchange 2010 servers, as the tab doesn't exist on Exchange 2010 SP1. When the job starts, the MFA begins multithreaded (concurrent) processing of all of the databases on a server.

As discussed in Chapter 12, "Mailbox Support Services," Exchange 2010 SP1 introduces a new method to schedule and perform the work done by mailbox assistants, including the MFA. When the scheduled window for the MFA opens on an Exchange 2010 RTM server, the Assistant begins to process all mailboxes one after another as quickly as possible. In effect, the MFA sprints through all its work in an attempt to reach the finish line as quickly as possible. This creates a high processing load on the server, and this could occur at the same time that other housekeeping activities happen, such as background maintenance and backup jobs. The fact that the MFA does more work than ever before to stamp new items and process items according to the conditions specified in retention policies is also of concern, as this drives additional server load.

Instead of sprinting to the finish, the Exchange 2010 SP1 version of the MFA assesses the expected workload in terms of the number of mailboxes that it has to process and then spreads out its processing across the complete window. For example, if 600 mailboxes are to be processed over three hours, the MFA will create its own internal schedule to process 200 mailboxes per hour, or roughly three mailboxes per minute. In addition, there is a checkpoint defined for the work cycle, at which time the MFA will look for new mailboxes

that should be added to its list for processing. The default values for the work cycle and checkpoint are both one day, meaning that the MFA will attempt to process every mailbox in its list daily and will check for new mailboxes daily. Overall, the work cycle mechanism makes more effective use of server resources in an easy and relaxed manner throughout the day and doesn't create potential spikes in demand.

You might find that you want to run the MFA immediately, perhaps to apply a policy to a group of users for the first time. To force a nonscheduled run of the MFA on an Exchange 2010 server, connect to the server that hosts the database where the mailboxes are located, start EMS, and enter this command:

```
Start-ManagedFolderAssistant
```

Exchange 2010 SP1 will still process the mailboxes if you force an immediate run, but the mailboxes will be processed as described earlier.

Note

The *–Identity* parameter is no longer used by the Start-ManagedFolderAssistant cmdlet to refer to a server in SP1. Instead, it replaces the previous use of the *–Mailbox* parameter and is used to identify a mailbox that you want the Assistant to process immediately.

Forcing immediate execution for a selected mailbox is a useful thing to do when you start to apply policies to mailboxes and want to gauge the effect of the policy by examining the output of a log file, which might be easier than asking users what happened to the contents of their mailboxes (especially if you've made a mistake with the policy and just removed half of the items from the mailbox). To force processing for a selected mailbox, we specify its name with the *–Identity* parameter:

```
Start-ManagedFolderAssistant –Identity 'Akers, Kim'
```

To process a group of mailboxes, we either provide a set of mailbox identifiers as input or use the Get-Mailbox cmdlet with a filter to retrieve a set of mailboxes and pipe it as input to Start-ManagedFolderAssistant. In the first example, two mailbox identifiers are provided as input. In the second, we process all the mailboxes in a database, and in the third, we use a filter to find all the mailboxes from a particular office.

```
"Redmond, Tony", "Akers, Kim" | Start-ManagedFolderAssistant
Get-Mailbox –Database 'VIP Data' | Start-ManagedFolderAssistant
Get-Mailbox –Filter {Office –eq 'Dublin'} | Start-ManagedFolderAssistant
```

The time required for the MFA to complete its run depends on the number of mailboxes and the number of items to which it has to apply retention policies. A run on a small server

that hosts a few hundred mailboxes will complete in a couple of minutes unless the mailboxes hold thousands of items. On the other hand, processing 7,000 mailboxes, each of which holds an average of 20,000 items, could take several hours, especially if the server is loaded with other tasks or the policies cause a heavy I/O load because many items are permanently removed or moved from primary to archive mailboxes. You should monitor the first runs of the MFA on a server to gauge the scope of the activity and how long a "normal" run takes to complete. Equipped with this information, you'll be able to quickly assess whether future runs are progressing as expected.

After the MFA has applied a new policy to a mailbox, the next time that the user connects to the mailbox with a client that supports retention policies, she will see that retention tags are shown on items and the retention policy options are visible. Another important point that you should understand is that if you apply a retention policy that contains a default policy tag, the MFA will stamp the default tag on every item in the mailbox. This action will force Outlook to download the complete contents of the mailbox the next time the client connects and synchronizes with Exchange. Clearly, such a massive synchronization has the potential to flood a network and keep clients fully occupied for a long time. Including a default archive tag in a policy does not have the same effect, as the MFA does not stamp every item with this tag.

Putting a mailbox on retention hold

When you put a mailbox on retention hold, you tell Exchange to suspend the processing of any retention policies that apply to the mailbox. For example, if a user is away for an extended period and will not be able to process the items in his mailbox, you could put his mailbox on retention hold to prevent Exchange moving from items to his archive mailbox. You can set retention hold on a mailbox through EMC or EMS. To do this with EMC, select the mailbox and view its properties. Click the Mailbox Settings tab and select the Messaging Records Management option. You can then select the start and end date for the retention hold period (Figure 15-19). Setting any hold on a mailbox—retention or litigation—could take up to 60 minutes to become effective because the hold is respected after Exchange refreshes the cache that it uses to hold account information.

The equivalent command to set retention hold on a mailbox as executed through EMS is shown next. You'll see that we have also added a retention comment in this command. The retention comment does not appear in versions of Outlook before Outlook 2010 as there is no user interface exposed for this purpose. Outlook Web App does not display the retention comment either, for the same reason. The retention comment will appear in Outlook after the MFA next runs and processes the mailbox.

```
Set-Mailbox –Identity 'Andrews, Lisa (Sales)' –RetentionHoldEnabled $True
-StartDateForRetentionHold '7/20/2010 8:00:00 AM' -EndDateForRetentionHold
'8/11/2010 8:00:00 AM'
```

```
-RetentionComment 'This mailbox is on retention hold while the user is on vacation
between July 20 and August 11, 2010'
```

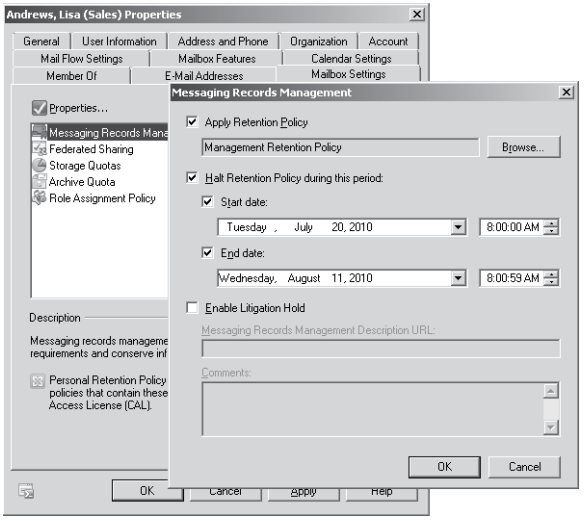


Figure 15-19 Setting retention hold through EMC.

To remove the retention hold and restore the normal processing of retention policies, set the property to $False:

```
Set-Mailbox -Identity 'Andrews, Lisa (Sales)' -RetentionHoldEnabled $False
-RetentionComment $Null
```

Putting a mailbox on litigation hold

When you place a mailbox on litigation hold (sometimes referred to as "legal hold"), Exchange stops removing items when their deleted items retention period expires, and any attempts by the user to delete or change items are retained in the dumpster. Items remain in the dumpster indefinitely until the litigation hold is released and are not subject to any quotas. Because items are retained, they remain available to be indexed and can be retrieved by discovery searches (see the section "Discovery searches" later in this chapter).

Exchange 2010 RTM only supports placing a mailbox on litigation hold using the Set-Mailbox cmdlet. For example:

```
Set-Mailbox -Identity 'Ruth, Andy (VP Sales)' -LitigationHoldEnabled $True
-RetentionComment 'Mailbox placed on litigation hold on 16 May 2010'
-RetentionURL 'http://intranet.contoso.com/LegalHold.html'
-LitigationHoldDate '4/1/2011 09:00'
-LitigationHoldOwner 'Legal Department'
```

With Exchange 2010 SP1, you can set litigation hold on a mailbox with EMC in much the same way as you set retention hold on a mailbox (Figure 15-20).

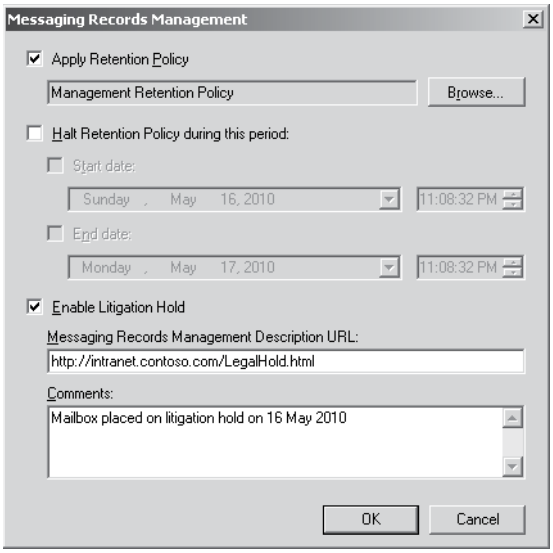


Figure 15-20 Setting litigation hold on a mailbox with EMC.

The *RetentionComment* and *RetentionURL* properties are used to populate the Account Settings section of Outlook 2010's backstage area to inform users that their mailbox has been placed on hold. The *–LitigationHoldDate* and *–LitigationHoldOwner* parameters are only available with Exchange 2010 SP1 and are used to hold the date and time when the hold was enforced and the account that enforced the hold. Exchange completes these details automatically when you place a mailbox on litigation hold using EMC or ECP, so if you put a mailbox on hold with EMS you should also provide these details.

Litigation hold: What about the user?

Exchange doesn't automatically inform users that their mailbox has been placed on litigation hold, and unless they visit the backstage area and notice the retention comment, they will be unaware that their mailbox is in a hold status. Indeed, if they don't use Outlook 2010, users might never be aware of this fact. For this reason, you might want to incorporate a step in the hold process where whoever authorizes the litigation hold is responsible for sending users an email notification to inform them why the hold is being placed on their mailbox and provide some information about what being on litigation hold means for a mailbox.

Releasing the mailbox from litigation hold is done by reversing the process:

```
Set-Mailbox –Identity 'Akers, Kim' –LitigationHoldEnabled $False
–RetentionComment $Null
```

To set litigation hold through ECP, select the mailbox and scroll down to the Mailbox Features section to reveal the option (Figure 15-21). You can then enable the hold and input an appropriate retention comment and URL for users to access more information about what this new status means for them. After you save the new setting, Exchange updates the mailbox properties (as previously) and advises you that it might take up to 60 minutes before the hold becomes effective.

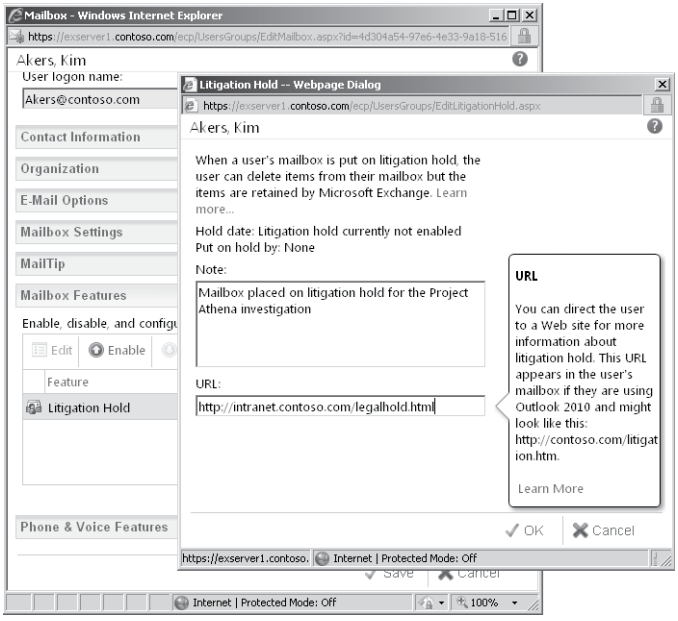


Figure 15-21 Putting a mailbox on litigation hold with ECP.

The exact delay depends on your Active Directory infrastructure and how quickly updated mailbox settings are replicated. Two influences are in play. First, Active Directory must replicate the updated litigation hold property to all global catalog servers before you can be assured that the setting applies across the forest. Second, the Store caches Active Directory data about mailbox properties for performance reasons and therefore will not know that the litigation hold setting has changed for the mailbox until the next time that the Store refreshes its cache. The updated litigation hold setting will be fetched from Active Directory and become effective the next time that the Store refreshes its cache. The complete cycle of Active Directory replication and Store cache refreshes could take up to an hour. For this

reason, it is a good idea to implement litigation holds, if possible, at a time when users are not actively using their mailboxes.

The very valuable dumpster

Two dumpsters operate in Exchange. The transport dumpster functions on hub transport servers and acts as a repository for in-transit messages that might have to be replayed after a server outage, so although it functions on an ongoing basis, an administrator shouldn't have to rely on the transport dumpster too often. The Store dumpster is much more useful on a day-to-day basis because it works for every mailbox and saves administrators from the need to restore mailbox databases to recover items that users have deleted in error. Of course, not every user can justify the expense of going through a full database recovery—and even the most important users probably can't justify the expense for every item that they delete in error—but mailbox restores for item recovery were quite a common practice before the dumpster first appeared in Exchange 2000. The elimination of these restore operations is of huge value to administrators, and that's why the dumpster is one of the high-value, low-cost features in Exchange.

Dumpster basics

Before we consider the changes that have occurred in Exchange 2010, we should review some dumpster basics. By default, the dumpster holds an item for a retention period after a user deletes it from a folder. The default retention period is 7 days in Exchange 2003 and 14 days in Exchange 2007 and Exchange 2010. Items in the dumpster are "soft deleted" in that users have deleted them from their original folder and emptied the Deleted Items folder. The items still remain in the database. In previous versions of Exchange, soft deleted items are kept in the Deleted Items folder but are hidden from the user's view. In Exchange 2010, when users empty their Deleted Items folder, Exchange moves the items into a new subfolder under Recoverable Items called Deletions. The items in this folder are what Outlook and Outlook Web App show when a user selects the Recover Deleted Items option. Non-MAPI clients that use protocols such as Post Office Protocol 3 (POP3) and Internet Message Access Protocol 4 (IMAP4) do not include the user interface or the basic support in the protocols to enable recovery from the dumpster, and that's why you don't find the feature in these clients.

When you select the Recover Deleted Items option, Outlook displays a list of all of the items in the dumpster. For example, Figure 15-22 shows a list of deleted items from my mailbox as displayed through Outlook Web App. (Outlook Web App is a little more functional than Outlook because it allows you greater flexibility about where to recover items, including the ability to create a new folder for the purpose.) The list includes items deleted from all folders, including those that have transited through the Deleted Items folder and those placed directly into the dumpster by being hard deleted with the Shift+Delete key combination.

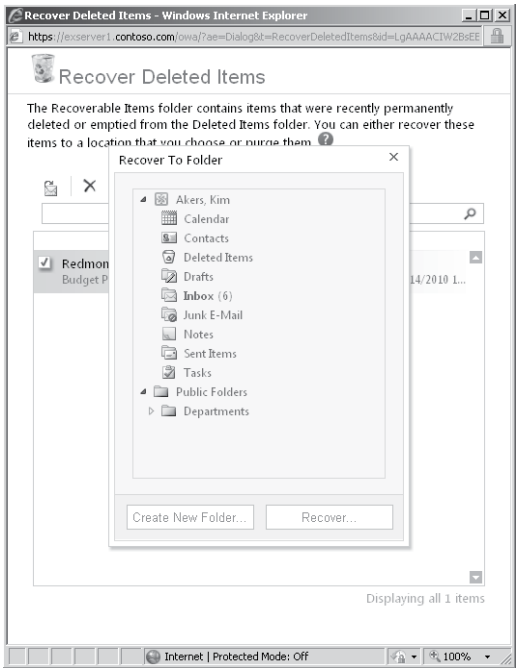


Figure 15-22 Recovering deleted items from the dumpster.

Recoverable items expire when the dumpster's retention period passes, at which time the items will be hard deleted or permanently removed from the database. Items can also be hard deleted immediately as the *PermanentlyDeleted* action required by a retention tag. In this case, the MFA is responsible for removing the items stamped with the tag from the database and these items are deleted the next time that the MFA runs after the item's retention period expires. Deleted items that are still recoverable don't count against the user's normal mailbox quota, so an extended retention period won't make any difference to users, except that they can recover items at any time up until the retention period passes.

Experience says: Use longer deleted items retention periods

By contrast, once removed from the database, items can only be made available again by an administrator after considerable effort to restore the database that contains the mailbox from a backup and then exporting the recovered items to a PST to provide to the user. It is for this reason that experienced Exchange administrators prefer to use longer rather than shorter deleted items retention periods. The default 14-day period is a good starting point, but 28 days might be even better. After all, if someone can't remember that she made a mistake in deleting an item within 28 days, maybe the item isn't important enough to warrant administrator intervention.

By default, Outlook 2003 only supports recovery of items that were originally removed from the Deleted Items folder, whereas Outlook 2007 allows recovery of items from any folder, including those that have been hard deleted. You can force Outlook 2003 to support recovery of deleted items from any folder by inserting a new DWORD value set to 1 in the following location:

HKEY_LOCAL_MACHINE\Software\Microsoft\Exchange\Client\Options\DumpsterAlwaysOn

Dumpster 2.0 arrives

From a user perspective, the experience of working with the Exchange 2010 dumpster is similar to previous versions. Behind the scenes, the introduction of new features to meet legal and regulatory requirements has had a massive influence on the dumpster. If users are under litigation hold, it's obvious that they shouldn't be able to affect the content in the dumpster, as deleted items could form part of the legal record. This requirement prompted Microsoft to look at how the dumpster works and led to the design of an enhanced dumpster ("dumpster 2.0") for Exchange 2010. Among the major changes are the following:

- Items in the dumpster are included when you move a mailbox from one database to another rather than being purged as in previous versions of Exchange. This prevents the loss of any item that should be retained as a consequence of normal rebalancing of server load.
- Items in the dumpster are indexed so that they can be discovered by searches.
- The Store maintains a quota for the dumpster.
- Versions are maintained of changes made to items in the mailbox.
- Users cannot purge data by using the Recover Deleted Items option to view the contents of the dumpster, selecting one or more items, and then deleting them (with the "X" option shown previously in Figure 15-22).
- The dumpster is maintained on a per-mailbox rather than a per-folder basis.

The Exchange 2007 version of the dumpster is based on a hidden view maintained on a per-folder basis. As items are deleted, the Store sets a MAPI flag (ptagDeletedOnFlag) and starts the retention time countdown. The deleted items stay in the folder but are hidden from clients. They cannot be indexed or searched because Outlook and other clients don't see these items. This mechanism works, but it can't accommodate the requirements for indexing to allow discovery searches. The new dumpster is implemented as a hidden folder called Recoverable Items located in the non-IPM subtree of user mailboxes. No client

interface ever exposes this subtree, so the folder remains invisible to users. The Recoverable Items folder contains three subfolders called Deletions, Versions, and Purges used to hold items at different points in their journey toward eventual removal from the Store. Using folders instead of views enables indexing and searching and also makes sure that dumpster data are moved along with the rest of the mailbox.

The Deletions folder replaces the MAPI flag and hidden view. When a user deletes an item as normal (soft delete) by emptying the Deleted Items folder, the Store moves the item into the Recoverable Items\Deletions folder. The Recover Deleted Items option accesses this folder, even when accessed by older Outlook clients, as the RPC Client Access Layer interprets client requests that were previously satisfied using the hidden view through items retrieved from the Deletions folder. If the user has a personal archive, a separate set of dumpster folders is maintained in the archive to handle the deletions that occur for archived items.

The Dumpster does not preserve the folder context for deleted items. In other words, you don't see the folder from which an item was deleted when you view the contents of the dumpster. This isn't usually an issue unless you have a very large number of items in the dumpster and therefore have to peruse a long list to find the right item to recover or a user deletes a complete large folder by accident and is faced with the need to find and recover all of the items from the deleted folder from among the mass of other items in the dumpster.

Some observers have commented that this situation happens often enough to keep them busy restoring deleted folders from backup copies, in turn meaning that the thought of ever going to a "no backup" regime for Exchange is impossible until the dumpster captures folder information, too. For now, the default sort order used in the dumpster is the date and time when an item is hard deleted (removed from the Deleted Items folder). Sometimes it is easier to find items if you click the appropriate column heading to sort by message subject or author. Microsoft is considering how best to improve matters in future versions of Exchange, perhaps by supporting the preservation of the folder structure for deleted items, which potentially would allow you to find items based on the folder from which they were originally soft deleted.

If you enable auditing for a mailbox (see the section "Auditing mailbox access" later in this chapter), Exchange stores the audit data in the Audit subfolder of the Recoverable Items folder in the dumpster. Audit entries age out after 90 days by default, so this folder can contain many items if a high degree of audit settings is enabled on the mailbox.

Note

The dumpster does not retain information if you delete a mailbox. The dumpster only handles deleted items from active mailboxes, so if you delete a mailbox, its content is no longer visible to Exchange for functions such as discovery searches. You can't suspend the final removal of a mailbox and retain it indefinitely or for a specified period, either, as the Store will remove a deleted mailbox permanently from its database when the deleted mailbox retention period expires. If you need to retain information for mailboxes used by people who leave the organization so that the items in the mailboxes can be found by discovery searches, you can disable the mailbox and keep it for as long as the items might be required. You could also import any PSTs that you want to retain into the mailbox to make them available to searches.

Single item recovery

From a user perspective, everything discussed so far works as in previous versions. The foundation is different but the effect remains the same. The Versions and Purges subfolders, which are never exposed to clients, provide additional functionality by allowing Exchange to preserve data even if a user attempts to change or purge deleted items. Microsoft calls this feature Single Item Recovery.

As we know, deleted items are now held in the Recoverable Items\Deletions folder and remain there until the deleted items retention period elapses, at which time the MFA permanently removes them from the database. Mailboxes that are placed on litigation hold do not respect the deleted items retention period, and items will remain in the folder until the litigation hold is released. Note that calendar items are always retained for 120 days, the logic being that the calendars of those under investigation are usually highly interesting to the teams working on legal discovery.

It is possible that a user might seek to change or remove items while they are in the Deletions folder. For example, if users receive a message containing some incriminating information, they can delete it with Shift+Delete to force the item into the Deletions folder. They can then use the Recover Deleted Items option to view the items in the Deletions folder, select the offending item, and delete it. In previous versions of Exchange, the item would be immediately removed from the database and a database restore would be required to retrieve it thereafter. Administrators will probably not be aware that the item was deleted, the database recovery will probably never be performed, and the item is lost for good.

However, for Exchange 2010 mailboxes that are enabled for single item recovery, the Store moves the item into the Recoverable Item\Purges folder. Users are unaware of this fact

because the Purges folder is invisible to any client, and they probably don't know that their mailbox is enabled for single item recovery. As far as the users are concerned, the evidence has been buried, but the Purges folder is indexed so a discovery search performed by an administrator will locate the item in the Purges folder as long as it is within the deleted items retention period. The items identified by a search are extracted and placed into the selected discovery mailbox in a folder named after the user and the date and time of the search. The administrator or other authorized user with access to the discovery search mailbox can then export the discovered items to provide the evidence to the legal team.

The Versions folder comes into play if users attempt to alter an item. Let's assume that a user is worried about a document attached to a message in the Inbox. She opens the message, removes the attachment, and saves the change. To the user's eyes, the item no longer has an attachment, but in reality the Store has saved a copy of the original message complete with the attachment in the Versions folder. Technically speaking, any action that changes an item generates a new version through a copy-on-write operation. Changes to subjects, message bodies, attachments, sender and recipient details, and date information are all examples of actions that generate a new version. Table 15-4 lists the actions that cause Exchange to retain a new version of an item in the Versions subfolder.

Table 15-4 Actions that cause item versions to be generated

Item	Actions that cause versions to be retained
Messages and posts	Updates to: Subject Item body Attachments Sender or recipient data Send or received dates
Other item types	Changes to any property visible to a client except: The folder in which the item is stored Item read status Retention tag status

Draft items are the only exception to dumpster processing. A draft item is one that has the "unsent" bit set in the MAPI message flags. If this bit is set, the dumpster does not capture updates in the versions folder. There are two reasons for this: First, a draft item typically goes through multiple revisions that might be captured by a client's auto-save process before it is eventually sent. Second, a draft item is not really interesting for discovery until it is sent and becomes a full-fledged communication to another person. After all, it's not a problem if a user thinks about doing something wrong, such as making an illegal recommendation to someone else to engage in insider trading, and captures the thought in a draft message. The thought only becomes a problem and consequently of interest for discovery purposes if the user sends the message to another person.

INSIDE OUT

Keeping items indefinitely through a litigation hold

Items captured through single item recovery remain in the Purges and Versions folders until their normal retention period expires. When this happens, the MFA removes the items as normal. If you need to keep items for an indefinite period, you should enable litigation hold for the mailbox as this instructs Exchange to keep everything until the hold is eventually lifted.

Knowing what's in the dumpster

A user can view the items in the dumpster at any time by using the Recover Deleted Items option in Outlook or Outlook Web App. Administrators can't access items in the dumpster unless they open a user's mailbox, but they can get a sense of how much data are held there and what type of data they are by using the Get-MailboxFolderStatistics cmdlet and pointing it at the dumpster with the *–FolderScope* parameter. For example:

```
Get-MailboxFolderStatistics –Identity TR –FolderScopeRecoverableItems |
Select Identity, ItemsInFolder, FolderSize, FolderType
```

```
Identity            ItemsInFolder       FolderSize                  FolderType
---------           -----------         ----------                  ----------
tr\Recover          6                  15.1 KB (15,466 bytes)       RecoverableItemsRoot
tr\Deletions        75                 6.905 MB (7,240,761 bytes)   RecoverableItemsDele-
tions
tr\Purges           9                  40.59 KB (41,562 bytes)      RecoverableItemsPurges
tr\Versions         3                  269.7 KB (276,174 bytes)     RecoverableItemsVersions
```

You can see references to the different types of items captured by the dumpster. As we know from the description in the previous section:

- The Root folder holds stripped versions of calendar items.
- The Deletions folder stores any soft deleted items.
- The Purges folder stores any hard deleted items.
- The Versions folder stores any previous versions of deleted items that have been edited.

Calendar items are held in the dumpster for 120 days. "Stripped" versions of calendar items have no attachments. Exchange creates these copies from calendar items that are purged

or updated in the dumpster to use as logs to track these changes. The stripped items are stored in the Root folder of the dumpster. Full copies of items that are changed or purged are also stored in the Versions or Purges folders, respectively.

Managing dumpster parameters

Single item recovery is not enabled for any mailbox by default. You can enable a mailbox as follows:

```
Set-Mailbox –Identity 'John Smith' –SingleItemRecoveryEnabled $True
```

In a scenario where executives or other users who need to use single item recovery are gathered into a single database, you can enable all the mailboxes by piping a list of mailboxes into the Set-Mailbox cmdlet. In this example, we also set the period for the Store to maintain the deleted items to 28 days (the default is 14).

```
Get-Mailbox -Database 'DB1' | Set-Mailbox -SingleItemRecoveryEnabled $True
–RetainDeletedItemsFor 28
```


Exchange doesn't provide a method to configure every mailbox in an organization for single item recovery through a global setting. It's easy (but slow in large organizations) to find all mailboxes with the Get-Mailbox cmdlet and set single item recovery for each, but you then face the task of ensuring that any new mailboxes are enabled for single item recovery thereafter, meaning that you have to run jobs to locate mailboxes that haven't been enabled regularly. This is an annoyance that Microsoft is aware of and one that they are considering addressing in a future release.

Moving on from single item recovery, you can set a default deleted items retention period for a database with the Set-MailboxDatabase cmdlet:

```
Set-MailboxDatabase –Identity 'DB1' –DeletedItemRetention 15
```

Like many other settings, these are held in Active Directory and cached by the Store for faster access. The exact time when a mailbox is enabled for single item recovery depends on how quickly Active Directory replicates the new setting around the organization and when the Store cache is refreshed after replication.

Items in these folders do not count against the mailbox quota. You can set separate quotas for the dumpster folders on a per-mailbox or per-database level. For example:

```
Set-Mailbox –Identity 'John Smith' –RecoverableItemsWarningQuota 1GB
–RecoverableItemsQuota 1.5GB
```

In this case, we set a warning point at 1 GB and an absolute quota at 1.5 GB for the dumpster folders in the specified mailbox. The default values used if these parameters are not explicitly set are 20 GB and 30 GB, which seems excessive unless a mailbox is placed under

litigation hold and needs to maintain a large amount of deleted items for a significant period. Increasing the retention quota does not affect the mailbox quota, but it does have an impact on the calculation of database size as the Store keeps the deleted items in the same database.

When the total size of the folders reaches the warning point, the Store issues warnings in the event log. Litigation hold disables expiry of items from the dumpster, so they will be held until the litigation hold is released. More important, Store background maintenance will begin to delete the oldest items in the Deletions folder to free up space and to accommodate newly deleted items. If the absolute quota is reached, the Store flags the error and will not preserve newly deleted items until some quota is released.

Table 15-5 summarizes the different data retention states that an Exchange 2010 mailbox can be in from the default position where the mailbox essentially operates much like Exchange 2003 or Exchange 2007—through the enabling of single item recovery—which provokes the use of the Purges and Versions folders, to a point where litigation hold freezes the mailbox and prevents any data from being removed until the hold is released.

Table 15-5 The different data retention states for Exchange 2010 mailboxes

Mailbox status	Deleted items kept in dumpster	Versions and purges kept in dumpster	User can delete items from the dumpster	When message management removes items from the dumpster
Default—Single item recovery not enabled on a mailbox	Yes	No	Yes	After deletion item retention period expires (120 days for calendar items)
Single item recovery enabled	Yes	Yes	No	After deletion item retention period expires (120 days for calendar items)
Litigation hold enabled for mailbox	Yes	Yes	No	Items retained until litigation hold is released

Initial measurement of the effect of single item recovery on mailbox sizes indicates a growth of 3 to 5 percent for a 14-day retention period. It's not usual to enable every mailbox in a database for single item recovery unless you have dedicated databases for VIPs or other users affected by legal proceedings, so the overall effect on a mailbox database is usually less.

Discovery searches

Discovery searches are performed through ECP by users who hold the Discovery Management role. ECP reveals the options to initiate mailbox searches under the Reporting node (Figure 15-23) to mailboxes that hold the Discovery Management role. This feature is a good example of functionality that is available through ECP and doesn't appear in EMC.

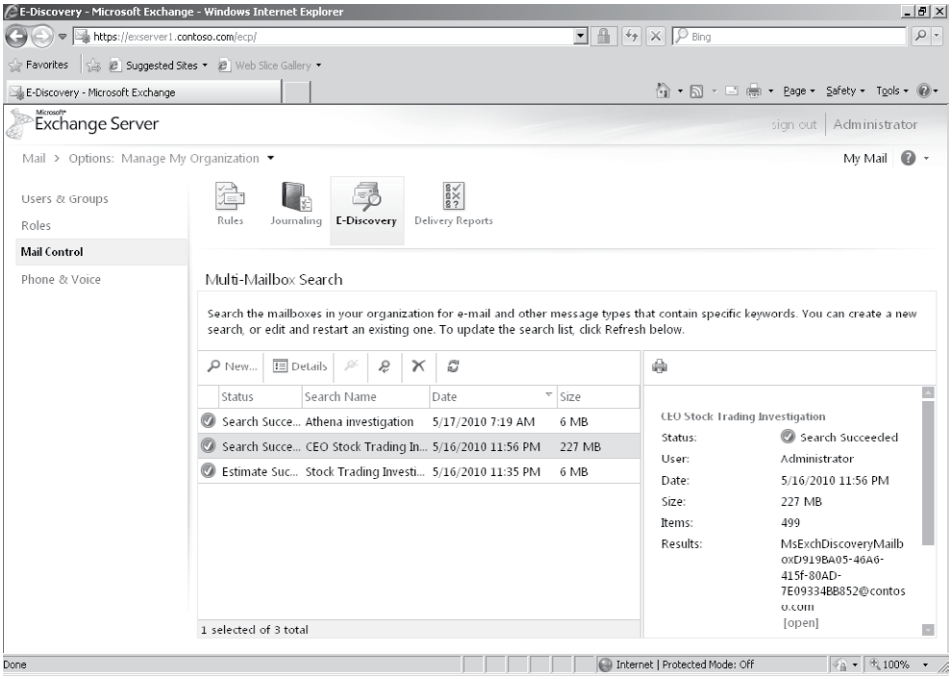


Figure 15-23 Viewing the ECP options for mailbox searches.

It is a quirk of Microsoft licensing policy that you need an enterprise CAL to be able to use the Discovery options included in ECP, but a standard CAL suffices if you conduct searches using the Search-Mailbox cmdlet as described later on in this section. You can therefore save a few dollars by executing all searches through EMS, which seems to be a strange situation!

Exchange is able to search across items stored in primary mailboxes, archive mailboxes, and the dumpster. It is not able to search through mailboxes that are deleted, even if the mailbox content is still in the database, because it hasn't exceeded the deleted mailbox retention period.

INSIDE OUT

Discovery search policy for exiting employees

If your company commonly needs to conduct discovery searches, it's wise to have a policy of disabling mailboxes for a month or so after someone leaves the company rather than rushing to delete mailboxes. This is especially true for mailboxes that belong to company officers or other executives, where you might need to keep a mailbox for up to a year after an employee leaves the company.

The content index catalogs that are maintained for mailbox databases are critical to Exchange's ability to perform searches: If the catalogs are unhealthy or not fully populated, then search results will be unpredictable or incomplete. Exchange uses the same content indexes for searches by clients, including Outlook Web App and Outlook. However, Outlook only uses the Exchange content indexes when it is configured to work in online mode. Most Outlook clients are now configured in cached Exchange mode, in which case they use local search indexes created with Windows Desktop Search to be able to conduct searches even when they are not connected to a server.

Microsoft introduced the current system of content indexing in Exchange 2007 and improved the performance and throughput of the content indexing component in Exchange 2010 in the following areas:

- Content indexing uses fewer system resources such as CPU, memory, I/O, and disk space.
- Items are typically indexed within 10 seconds of their creation on a server. Query results are much faster.
- You don't need to configure Exchange Search. It is automatically installed and configured on all mailbox servers.
- Attachments are indexed (see the "Unsearchable items" discussion next).
- Indexing throttles back automatically in periods when mailbox servers experience heavy load. Again, administrators don't have to take any action for this to happen.

Administrators tend to forget about content indexing because it hums away in the background and doesn't make their lives difficult.

Unsearchable items

As shown in the left screen in Figure 15-24, all item types are discoverable, including voice messages, drafts, attached documents of various formats, and IM conversations (if stored in mailboxes). Before Exchange can include a document, usually an attachment to a message, in its content indexes, it must be able to extract the content. Exchange includes a set of content filters for this purpose. Unlike the RTM version of Exchange 2010, Exchange 2010 SP1 registers the IFilters for Office 2010 with Exchange Search. However, if you want to use other IFilters, such as the one for Adobe PDF, you have to install them separately.

See *http://technet.microsoft.com/en-us/library/ff622320.aspx* for more information.

You can see the list of default filters installed on mailbox servers by looking in the system registry at HKLM\SOFTWARE\Microsoft\ExchangeServer\v14\MSSearch\Filters. The list

includes formats that you would expect, such as Microsoft Office, text attachments, HTML files, and so on.

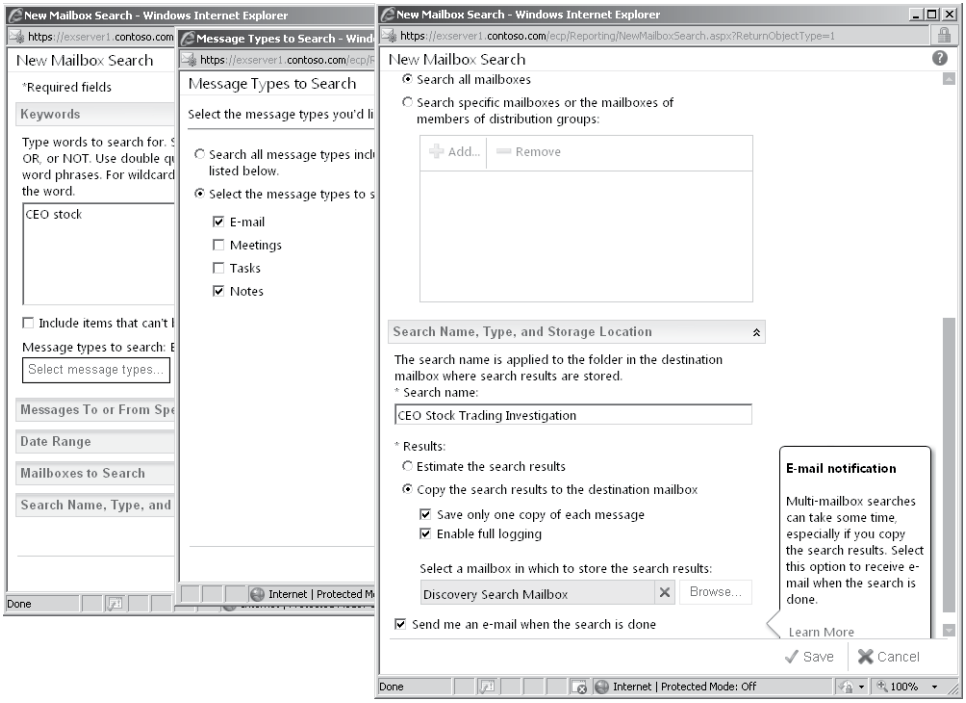


Figure 15-24 Determining the options for a multimailbox search.

If Exchange meets content that it doesn't understand, it marks the item as unsearchable. For example, if you use an application that generates files of a type that are only understood if the application is installed on a client workstation, the content indexing agent running on a mailbox server won't be able to open and index the files. Other items that Exchange deems as unsearchable include items encrypted with Secure Multipurpose Internet Mail Extensions (S/MIME). However, messages protected with Active Directory Rights Management Services remain searchable for discovery purposes.

You can see a list of unsearchable items with the Get-FailedContentIndexDocuments cmdlet. When you run the cmdlet, you can pass it the name of a server to see all items on a server, or just a mailbox database to see the unsearchable items in the content index for that database. For example, here's how to run the cmdlet followed by an extract of the information returned for an unsearchable item (pipe the results to the Format-List cmdlet to see this information). As you can see, the item shown couldn't be indexed because a filter wasn't found for the attachment type.

```
Get-FailedContentIndexDocuments –MailboxDatabase DB1
```

```
RunspaceId       : 5de022fc-bd60-4b22-8f5e-e983550a4f8a
DocID            : 21847
Database         : DB3
MailboxGuid      : ab83c57b-d51c-4527-8f99-5609e0ee96c8
Mailbox          : Ruth, Andy
SmtpAddress      : Andy.Ruth@contoso.com
EntryID          : 000000002BA4E1B5193C7441BCD9110F91902C5A0700A0EAF17663EEB9429D-
934943C3A2409300000000000200005036EA2334225B46B46EA1623061B2A40000017803030000
Subject          : Designing Secure Multi-Tenancy into Virtualized data center (Secure Cloud
Architecture)
ErrorCode        : 2147749142
Description      : Filter not found
IsPartialIndexed : True
Identity         :
IsValid          : True
```

Should you be worried if many unsearchable items exist for your database? The answer is, "It depends." First, it depends on the percentage of unsearchable items. If 0.0002 percent of items are unsearchable, then it's probably acceptable because any search has a very high chance of discovering information that's required. Second, it depends on the items that are failing to be indexed. If they are all of the same type and a filter is available, you can install that filter to solve the problem. However, if the items are of a type for which a filter is not available or known to be unsearchable (such as S/MIME encrypted items), then you might have to live with the situation.

See *http://technet.microsoft.com/en-us/library/ff354976(EXCHG.80).aspx* for information about how to install a new filter.

Normally, a relatively small number of items turn out to be unsearchable. In addition, you should remember that item properties (sender, recipients, subject, and so on) and message bodies are always indexed and searchable, so the fact that a small percentage of attachments can't be searched is probably not going to be of great concern in a legal search. After all, if people are doing something that they shouldn't, it's likely that they will leave some trace of their activity in a searchable property that will be discovered. After this happens, the next step is often for investigators to take a complete copy of the suspect's mailbox to conduct a detailed search to discover what it contains, and any lurking unsearchable items can be reviewed at that time.

Creating and executing a multimailbox search

A mailbox search can cover every mailbox in the organization (rightmost screen in Figure 15-24) or a select set formed of individual mailboxes or the members of specific distribution groups (but not dynamic distribution groups). You can include other search

criteria such as date ranges and specific words or phrases in message bodies and subjects. You can update the search criteria multiple times; each time you do, Exchange will restart the search and discard any items found using the previous criteria. However, if you want to change the criteria for a search while it is being processed, you have to stop the search before you can make any changes.

After you input all the search criteria and click Save, Exchange stores the criteria as search metadata in the default discovery mailbox. At this point, the original version of Exchange 2010 proceeds to execute a full search and will copy any results that it finds into the selected discovery mailbox. Exchange 2010 SP1 offers you the option of running either an estimate or the search. A search estimate is a scan of the content indexes to determine how many hits are likely if you initiate a search with the criteria as provided. You can see these options revealed under the Search Name And Storage Location section of the right-most screen in Figure 15-24.

An estimate does not actually retrieve the located items and copy them into the discovery mailbox, so it runs much faster than a "copy" search. After Exchange completes its scan to determine the estimate, the number of hits and the mailboxes that contain items are shown in the results pane (Figure 15-25). Because estimates run faster, you can afford to run a number of estimates to refine the search criteria to meet your exact needs. You don't have to run an estimate before you conduct a full search. If you want to search and copy items without an estimate, just select the Copy Results To The Selected Mailbox option and save the search. However, it makes sense to run an estimate to see just how much data might be found and test the efficiency of the search criteria.

Figure 15-25 Viewing a search estimate.

As you refine a search, you'll probably experiment with the query that lies at the heart of the search. The query is passed in Advanced Query Syntax (AQS) format. Table 15-6 lists the most important query terms that you are likely to use in discovery searches.

http://www.microsoft.com/windows/products/winfamily/desktopsearch/technicalresources/advquery.mspx **provides further information about how to construct AQS queries.**

Table 15-6 **AQS terms that can be used in search queries**

Property	Example	Search results
Attachments	Attachment:BadReport.ppt Attachment:Bad*.pp*	Return any items that have an attachment called BadReport.ppt. The second example shows how to use wildcards to conduct a less specific search.
CC	CC: Joe Healy CC: JoeH CC: JoeHealy@contoso.com	Return any message with Joe Healy listed as a CC recipient in the message header. The second and third examples show how to specify a search for an alias or an SMTP address.
From	From: Tony Redmond From: Tony.Redmond@contoso.com	Return any message sent by Tony Redmond using different forms of his address.
Keywords	RetentionPolicy:Critical	Returns any item that has the Critical retention tag applied to it.
Expiration	Expires: 10/10/2010	Returns any item that expires on October 10, 2010.
Search message recipients	Person: Tony Redmond Person: TR@contoso.com	Returns any item that has the recipient included as a To:, CC:, or BCC: in the message header. You can pass the display name, alias, or SMTP address.
Sent	Sent: yesterday	Returns messages sent yesterday.
Subject	Subject: "Trading Tip"	Returns all items that include the words "Trading Tip" in the subject.
To	To: Tony Redmond To: TR@contoso.com	Returns any message that has Tony Redmond listed as a To: recipient. You can use the display name, alias, or SMTP address.

You can run as many search estimates as you want. To change the search criteria, click the Details icon and amend details such as the mailboxes to search or the phrases for which you are looking. It's entirely possible that your first attempt to create a search could result in an estimate of thousands of hits across hundreds of mailboxes when you expect to find just a few items. This might force you to narrow (or sometimes widen) the search criteria. For example, you might exclude some mailboxes from the search or include some new terms that you think will help to locate just the right information. On the other hand, you might be happy that you've found so much data. From Exchange 2010 SP1 onward, ECP and EMS both default to deduplicated searches.

Once you're happy that the search will find the data that you're interested in, you can click the Details icon to change the type of search and instruct Exchange to copy the matching items. You also need to decide whether Exchange should save a copy of each matching item in the discovery mailbox or if it should reduce the number of items that it copies by only capturing the first copy found. After you save the updated search parameters,

Exchange will then conduct the full search and copy any items that it locates into the discovery mailbox.

Mailbox searches are performed in the background. You can wait for the search to be complete or have Exchange notify you with an email (Figure 15-26). The next step is then to access the discovery mailbox where Exchange has copied the items found by the search.

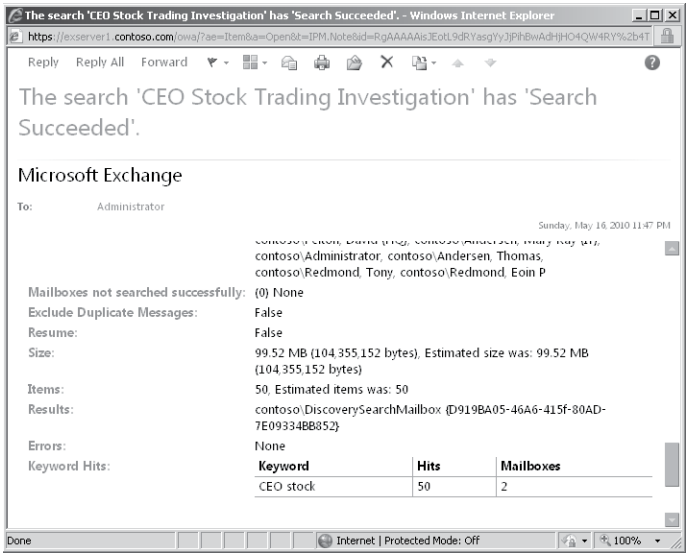


Figure 15-26 Notification message after a successful search.

Accessing search results

By default, there is a single discovery mailbox in an Exchange organization. As described in Chapter 6, you can create additional discovery mailboxes to use to hold search results, and you have to select a discovery mailbox to use when you create a new search. The results of the search, including copies of all items that match the search criteria, will be placed in the selected discovery mailbox. If you have the necessary permission to access the discovery mailbox, you can enter its name in the in the Switch Mailbox control (Figure 15-27). The name of the default discovery mailbox is long, but you can enter the first few characters and then press Ctrl+K to have Outlook Web App validate the mailbox name. Thereafter, Outlook Web App will remember the mailbox and you can select it easily from a drop-down list of mailboxes if you need to access the discovery mailbox again. Of course, you can also use Outlook to open the discovery mailbox by configuring a suitable profile.

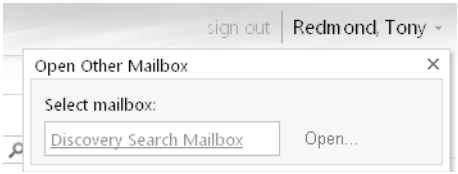


Figure 15-27 Switching to the discovery mailbox.

Within the discovery mailbox, Exchange inserts the items located by a search into a set of folders called after the name that you gave to the search. For example, if you call the search "Illegal stock trading investigation," Exchange will create a root folder of this name in the discovery mailbox and then create a child folder underneath for each mailbox where a matching item was found. The date and time of the search (the date and time of the server rather than the client workstation that starts the search) is appended to the mailbox name to clearly identify different searches that have occurred and to provide a solid time line for when evidence is gathered for an investigation. If you open the folder for a mailbox (Figure 15-28), you see all of the folders from which items have been copied in both the primary mailbox and the personal archive (if the mailbox has one). You can then click on the items to review their content and decide whether they are of real interest to your investigation. Incriminating evidence can be retained and any useless thoughts of idle minds discarded.

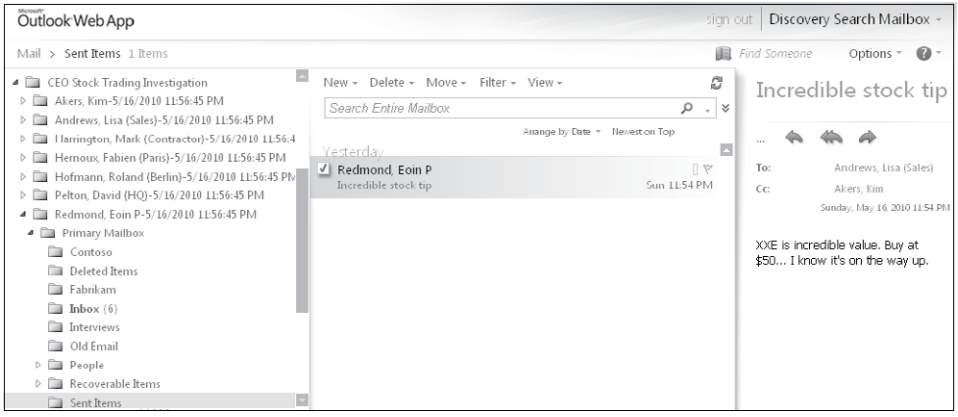


Figure 15-28 Viewing search results in a discovery mailbox.

If you use Active Directory Rights Management Services (see the section "Protecting content" later in this chapter), searches might uncover items that are protected because a user has applied an Information Rights Management (IRM) template to them. When an item is protected, its content can only be read by the sender, the intended recipients,

and members of the Active Directory Rights Management Services (AD RMS) Super Users group; the team that is reviewing the contents copied into the discovery mailbox won't be able to see anything but the message header data (Figure 15-29). This information might be enough to eliminate an item from the list of those that an investigator wants to see, but more often it's an indication that makes an item even more interesting to an investigator.

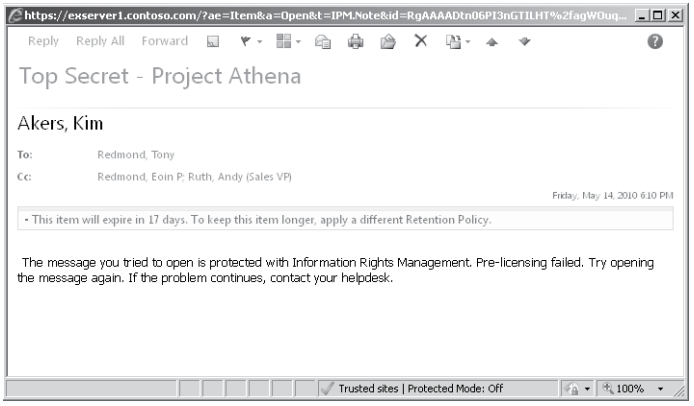


Figure 15-29 Viewing a protected message uncovered by a mailbox search.

Typically, the AD RMS Super Users group only contains the federated system mailbox, as its membership allows Exchange to decrypt protected messages as they pass through the transport system and apply transport and journal rules as required. In the RTM version of Exchange 2010, to allow investigators to view protected content, we therefore have to make the discovery mailbox a member of the AD RMS Super Users group for as long as the investigators need to review items uncovered by the search. The discovery mailbox uses a disabled account, and this also has to be enabled. These actions will allow the AD RMS server to provide the necessary credentials to the discovery mailbox to reveal the hidden content to the investigators. It seems strange to insist that the discovery mailbox account must be enabled to allow access to protected content, but AD RMS can only provide credentials to enabled accounts. The act of enabling the discovery mailbox should be approved and audited by some authority within the company because enabling the account creates a higher risk that someone could have unauthorized access to its contents.

Enabling accounts that should remain disabled is clearly an unacceptable workaround to a problem that should be fixed in software. Microsoft addressed the issue in Exchange 2010 SP1 by introducing a new parameter for the IRM configuration cmdlet to instruct Rights Management to allow access to protected content for legal investigators. To make everything work, you have to run the Set-IRMConfiguration cmdlet as follows:

```
Set-IRMConfiguration –EdiscoverySuperUserEnabled $True
```

Search access to dumpster content

All searches launched by ECP automatically examine the contents of the dumpster (you can exclude the dumpster contents with a search created with EMS) to ensure that any items that are of interest are captured even if a user has attempted to remove all traces of their existence. As shown in Figure 15-30, if an item is found in the dumpster, it will be shown under the Recoverable Items folder within the user's primary mailbox or personal archive.

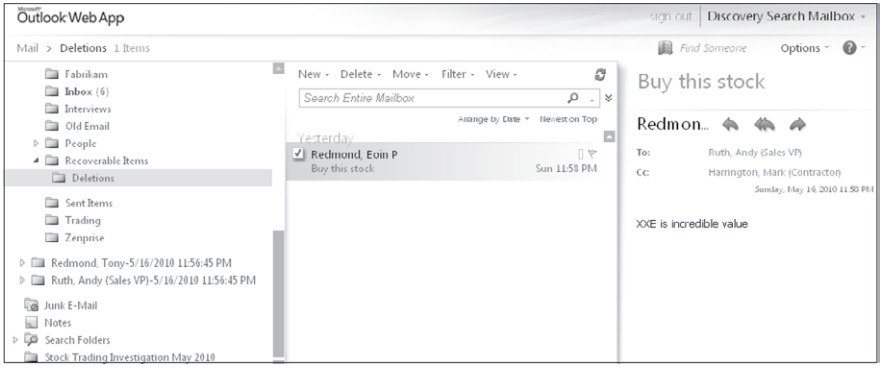


Figure 15-30 Dumpster items retrieved in a discovery mailbox.

Deduplication of search results

An item that you are searching for might exist in multiple mailboxes. You don't necessarily want to copy every single occurrence of the message from every mailbox in which Exchange finds it. Apart from the system overhead that is incurred to copy and store every instance of a message found in the searched mailboxes, providing extra copies of messages will drive up the cost of responding to legal discovery actions if the lawyers or other individuals who review the search results are paid on a per-item basis. Deduplication is therefore a very useful feature, with the only drawback being that storing the first discovered copy of an item sent to a distribution group does not prove that an individual received the item. You'd need to find the item in their mailbox to prove this.

Note

Because Exchange does not expand the membership of a group into a message header, seeing that a message was delivered to a group doesn't tell you whether someone received a copy because there's no way of proving what the membership of the group was at the point when the transport service expanded the group membership and delivered the message.

You instruct Exchange to deduplicate search results by selecting the Copy Only One Instance Of The Message option under the Search Name And Storage Location section. When Exchange copies items for a deduplicated search, it places a single copy of each unique item in a single folder called "Results" and the date and time of the search under a root folder for the search name (Figure 15-31). The message identifier, which is a unique value established when items are first created, is used as the basis of deduplication.

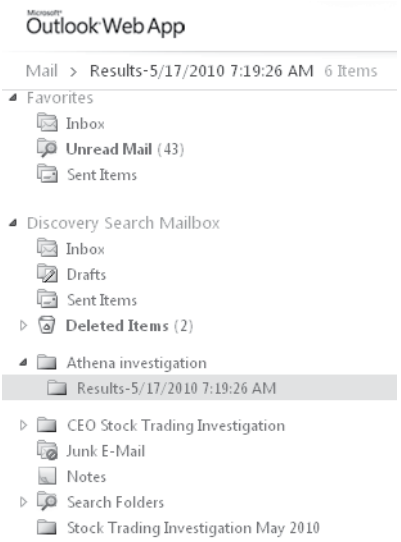


Figure 15-31 The folder structure created by a deduplicate search.

The users who perform discovery searches are not necessarily those who can access the results of the searches that are placed in discovery mailboxes. As discussed in Chapter 6, you need to assign full access permission to the discovery mailbox to a user before he will be able to open it to access the search results. By default, members of the Discovery Management role group should be able to access the default discovery mailbox, but you have to explicitly grant full access to any other discovery mailboxes that you create for use in mailbox searches.

A clear separation therefore exists between the following:

- Membership of the Discovery Management role group, which is required to be able to create and execute mailbox searches.
- Full access to the discovery mailbox used for a mailbox search, which is required to be able to open the discovery mailbox and review the items copied there by the mailbox search.

The separation in the two requirements allows for a division of responsibilities between those who are responsible for responding to requests for information (often the IT department) and those who will review the retrieved information forensically to look for evidence or other information that is important to an investigation (often the legal department). You might therefore create discovery mailboxes to hold information retrieved for different types of searches so that you can restrict access to those mailboxes to ensure that confidential material is always treated in a correct and legally defendable manner. Some discovery mailboxes might be used for straightforward legal discovery actions and be under the control of the legal department, whereas others might be used for the pursuit of internal complaints against an employee for something like sexual harassment and be restricted to selected members of the HR department.

CAUTION

Access to content held in discovery mailboxes should be carefully controlled so that only the people who need to be able to review and work with the data have access. You also need to be sure that the users do not interfere with the search results in an unauthorized manner. For example, it would not be a good situation if someone attempted to cover up illegal activities by appearing to conduct a search for suspicious items and then deleted a selected group of the discovered items to remove evidence. To address this situation, you can enable auditing for discovery mailboxes to force Exchange to capture information about the actions that these users take when they work with items. More information about how to set up mailbox auditing is available in the section "Auditing mailbox access" later in this chapter.

Search logging

Exchange generates a log for every mailbox search unless you suppress it by setting the *–LogLevel* parameter for the search to Suppress. By default, the search log captures basic information about the parameters used for the search as well as the results of the search. You can also increase the logging level to Full, in which case Exchange captures information about the items that are captured by the search in an attachment. The search report and any attachment are stored in the top-level folder created for the search in the discovery mailbox. Figure 15-32 shows a typical search report. You can see that it has an attachment, so this indicated that the logging level was set to full (you can also see this in the search parameters).

You can conduct a search and copy results multiple times. However, if you do this without creating a new search, Exchange removes the previous search results from the discovery

mailbox before it copies items as a result of the new search. Therefore, you have to create and execute a new search if you want to keep the results of a previous search.

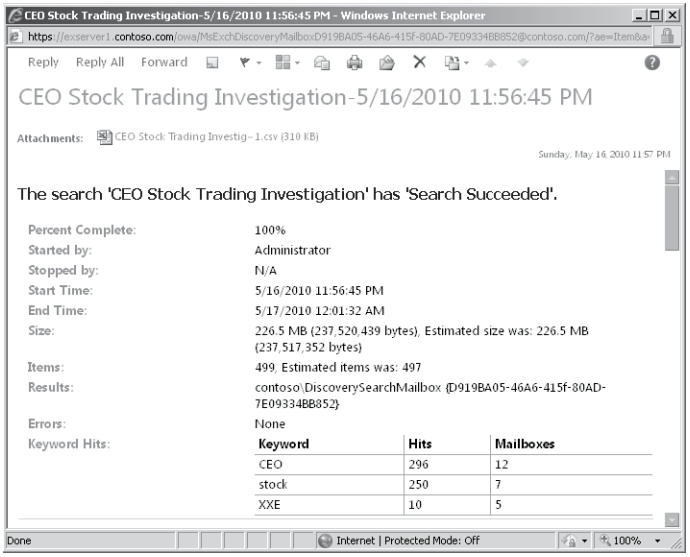


Figure 15-32 Viewing a search log report.

Search annotation

The ability to annotate search results is a new feature in Exchange 2010 SP1. Basically, the idea is that the people who look through search results should be able to mark the items that are of interest. Exchange accomplishes this through some special user interface that Outlook Web App exposes whenever a user logs into a discovery mailbox.

Figure 15-33 shows how annotation works. The Open Message Annotation option is exposed in the shortcut menu. This opens a simple text box to allow users to input whatever text they deem fit to mark the item. For example, they might mark items with a case reference or other indicator. Later on, they can search the mailbox for the marked items to see the collection of items of interest. There's no feature provided to export annotated items from the discovery mailbox if you need to provide copies for use elsewhere, but it's easy to copy the items to a folder and then use the New-MailboxExportRequest cmdlet to export the folder to a PST. Alternatively, you can open the discovery mailbox with Outlook and drag and drop the copied items into a PST.

The annotation is only visible through Outlook Web App and can't be accessed with other clients.

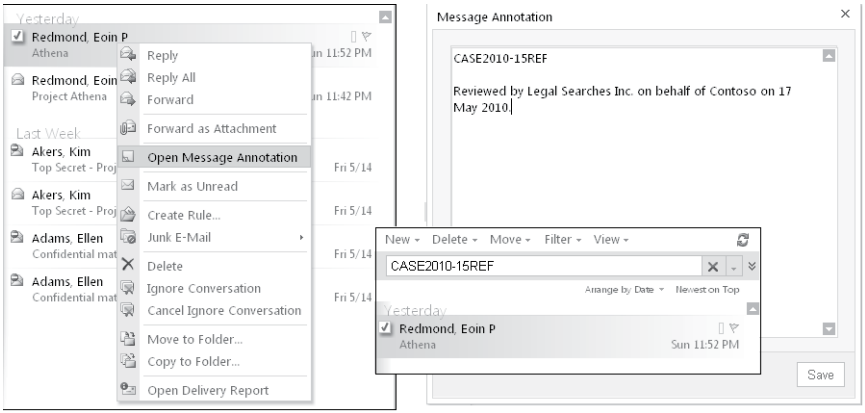


Figure 15-33 Annotation of search results.

Executing searches with EMS

ECP is a very convenient interface to create and initiate searches, but you can also do the same through EMS using a set of cmdlets that are only exposed if you are a member of the Discovery Management role group. These cmdlets are as follows:

- **New-MailboxSearch** Creates and initiates a new mailbox search.
- **Get-MailboxSearch** Retrieves details of a mailbox search.
- **Set-MailboxSearch** Changes the search criteria for a search that has already been created.
- **Start-MailboxSearch** Restarts a mailbox search.
- **Remove-MailboxSearch** Removes a mailbox search. This action also removes all of the items found by a search from the discovery mailbox.

For example, a new search to look for information about potential illegal stock trading by company officers could be initiated with this command:

```
New-MailboxSearch -Name "Stock Trading Discovery 2" -SourceMailboxes 'Company
Officers' -TargetMailbox 'DiscoveryMailbox@contoso.com' -StartDate '10/01/2010'
-EndDate '11/30/2010' -SearchQuery "XXE Stock tip"
-StatusMailRecipients 'LegalSearch@contoso.com' -SearchDumpster -DoNotIncludeArchive
-EstimateOnly
-IncludeUnsearchableItems -ExcludeDuplicateMessages:$False -LogLevel Full
```

Table 15-7 lists the most important parameters that you are likely to use with the New-MailboxSearch cmdlet and their meaning.

Table 15-7 **Important parameters for the New-MailboxSearch cmdlet**

Parameter	Meaning
Name	A unique identifier for the search that should be something meaningful, such as "Illegal stock trading review."
SourceMailboxes	Specifies the mailboxes that Exchange will search. If you have more than a few mailboxes to search, it is more convenient (and probably more accurate) to create a distribution group to identify the mailboxes to include in the search. If you don't specify the –*SourceMailboxes* parameter, Exchange searches all mailboxes.
TargetMailbox	Specifies the SMTP email address of the discovery mailbox where you want to store the search results. The default discovery mailbox has a rather long and complicated email address so I usually assign a new and shorter secondary email address to the mailbox to make it easier to type. In fact, this mailbox doesn't have to be a discovery mailbox, as Exchange is happy to place search results in any mailbox that you select.
SearchQuery	An AQS-format query that Exchange will execute to locate items in the target mailboxes. In the example shown, Exchange will match any of the words in the search query. This search query is a very simple one and some trial and error is probably required to arrive at the best query. If you omit the search query, Exchange will find every item in every mailbox that you include in the search and store copies of all those items in the discovery mailbox. This kind of search can swamp a server with work.
StatusMailRecipients	Tells Exchange the recipients who should be notified by email after the search is complete. No message is sent if you don't provide a value for this parameter. You can provide one or more recipient SMTP addresses to receive notifications, separating each address with a comma. It's often more convenient to use a distribution group for this purpose.
SearchDumpster	Forces Exchange to include the contents of the dumpster in the search. All searches executed through ECP include this parameter. As shown in Figure 15-30, any items from the Dumpster that are found by a search are placed in the Recoverable Items folder in the discovery mailbox.
DoNotIncludeArchive	Instructs Exchange to ignore items stored in any personal archives that are assigned to mailboxes.
EstimateOnly	Tells Exchange that it is to run a search estimate only rather than to copy items that match the search criteria to the discovery mailbox.
ExcludeDuplicateMessages	Tells Exchange how to deal with duplicate items that it encounters in mailboxes. Set the parameter to $True to force Exchange to deduplicate (only copy a single instance of an item) or $False to copy every copy of an item that it finds.
LoaLevel	Dictates the level of logging that Exchange performs for the search. Valid options are Suppress, Basic (default), and Full. If Basic or Full are chosen, Exchange creates a search report in the root folder for the search in the discovery mailbox.

The Get-MailboxSearch cmdlet tells us what happened to a search. All known searches are revealed. For example:

```
Get-MailboxSearch | Format-Table Name, Status, PercentComplete, ResultSize,
ResultNumber -AutoSize
```

```
Name                                Status      PercentComplete  ResultSize             ResultNumber
-------                             ---------   ---------------  ----------             ------------
Review Dumpster content             InProgress  39 112.8 MB      (118,262,783 bytes)    395
Deduplicated search                 Failed      9 3.061 MB       (3,209,944 bytes)      20
XXE Investigation March 2010        InProgress  87 1.132 GB      (1,214,974,519 bytes)  730
CEO Discovery                       Succeeded   100 136.1 MB     (142,687,323 bytes)    161
XXE Investigation Feb 2010          Succeeded   100 134.8 MB     (141,344,252 bytes)    156
Stock Trading Discovery 3           Succeeded   10020.42 MB      (21,413,083 bytes)     536
Stock Trading Discovery 2           Succeeded   1005.269 KB      (5,395 bytes)          2
Illegal stock trading investigation Succeeded   100 9.008 KB     (9,224 bytes)          2
```

The information we are interested in here is the status (this will be Estimate Succeeded, Succeeded, InProgress, or Failed) and the number of items found by the search. The size of the items is interesting if we expect to find a large attachment. As you can see from the search called "XXE Investigation March 2010," a search can generate a lot of information. In this case, the search located a number of very large objects (730 objects for 1.132 GB at 87 percent complete), so it will be interesting to check the contents of the discovery mailbox to find out just what these objects are.

INSIDE OUT

Running concurrent searches

You can run concurrent searches as long as each search has a different name. The searches proceed a little slower because of contention when writing found items into the discovery mailbox. If you need to run concurrent searches on an ongoing basis, it would be a good idea to spread the load by creating several discovery mailboxes and locating them in different databases.

Auditing administrator actions

As even a brief reading of this book reveals, an administrator has the ability to change many settings or create many new objects that influence the way Exchange operates. Up to Exchange 2010, there was no facility available to be able to track who did what and when at an administrative level. The addition of Windows PowerShell and its ability to affect

many objects with relatively simple commands reinforced the need to be able to log what happens within an Exchange organization.

Exchange 2010 includes the ability to audit actions taken by administrators in EMC and EMS. This is intended to allow organizations to maintain records of who did what and when to execute the cmdlets used to manage Exchange. Apart from providing definitive proof about what account was used to add a mailbox, change properties on a connector, set up a new domain, or any of the myriad day-to-day operations that occur in an Exchange organization, maintaining an audit log can help satisfy legislative requirements by demonstrating that strict controls are imposed on the work that administrators do with Exchange.

Some administrators will not welcome this development and will view it as yet another example of big brother looking over their shoulder as they struggle to keep the email system up and running. Others will consider increased oversight as part of modern life, much in the same way that we all seem to be under the eyes of video surveillance wherever we go. Auditing is not enabled by default. The Set-AdminAuditLogConfig cmdlet controls how administration logging functions across the organization.


```
Set-AdminAuditLogConfig -AdminAuditLogEnabled $True
```

When logging is enabled, administrators see no indication that their actions are being captured in the audit log unless they search the audit log.

Enabling administrative logging instructs the Admin Audit Log agent, one of the standard set of cmdlet extension agents shipped with Exchange 2010, to begin capturing details of administrative events. The admin audit agent runs on all Exchange 2010 servers to monitor the running of all cmdlets and record details of the cmdlets that you configure to be audited. As described in Chapter 3, "The Exchange Management Shell," the execution of all business logic in Exchange 2010 flows through cmdlets so the agent is able to monitor all administrative operations.

INSIDE OUT

Disabling administrative auditing

To disable administrative auditing, you run the same command and set the parameter to $False. Remember that this setting has to be replicated across the organization before it is effective on all servers, so it might take an hour or so before you can be sure that all administrative actions are being captured.

A number of other configuration settings are used to control the finer details of administrator audit logging. You can view the current audit configuration settings for the organization with the Get-AdminAuditLogConfig cmdlet. For example:

```
Get-AdminAuditLogConfig | Format-List
```

```
AdminAuditLogEnabled     : True
TestCmdletLoggingEnabled : False
AdminAuditLogCmdlets     : {*}
AdminAuditLogParameters  : {*}
AdminAuditLogAgeLimit    : 90.00:00:00
AdminDisplayName         :
ExchangeVersion          : 0.10 (14.0.100.0)
Name                     : Admin Audit Log Settings
DistinguishedName        : CN=Admin Audit Log Settings,CN=Global
Settings,CN=contoso,CN=Microsoft Exchange,CN=Services,CN=Configuration,DC=contoso,DC=com
Identity                 : Admin Audit Log Settings
```

Auditing is performed on a cmdlet basis and can be further refined to select specific parameters to audit. By default, the use of every cmdlet and every parameter is audited, so the preceding configuration has values of "{*}" (asterisk). In fact, Exchange ignores auditing for cmdlets beginning with "Get," "Search," and "Test" so as not to clutter up the audit log with entries for cmdlets that simply retrieve or read information. The purpose here is to audit operations that create or manipulate objects. Enabling the audit of every cmdlet is a bad idea because it will generate a huge mass of audit entries, including entries for actions that are probably not of great concern such as a user updating her OOF.

To focus on a specific set of cmdlets, we define the cmdlets in a list passed in the *–AdminAuditLogCmdlets* parameter for the Set-AdminAuditLogConfig cmdlet. For example, this command tells Exchange that we want to audit any use of the New-Mailbox and New-DistributionGroup cmdlets and any cmdlet that has "Transport" in its name.

```
Set-AdminAuditLogConfig –AdminAuditLogCmdlets 'New-Mailbox, New-DistributionGroup,
*Transport'
```

Exchange now captures details about any creation of new mailboxes and distribution groups plus any action taken with the cmdlets used to manage the transport service. Let's assume that you only want to capture certain details about new mailboxes. This command captures details of the name, display name, and custom attributes for new mailboxes.

```
Set-AdminAuditLogConfig -AdminAuditLogParameters Name, DisplayName, *Custom'
```

Clearly, you have to arrive at a balance to capture the required auditing data but not so much as to make it difficult to find an instance when necessary. It is likely that some trial and error will be required to settle on the right list of cmdlets and parameters to audit.

The audit mailbox

On Exchange 2010 RTM servers, you have to create and configure a mailbox to act as the repository for the audit reports that Exchange creates every time that one of the cmdlets within the audit scope is run. Exchange 2010 SP1 removes the requirement to configure an audit mailbox. Instead, SP1 uses a folder called AdminAuditLogs in the special arbitration mailbox with the display name "Microsoft Exchange" as the repository for audit reports. This is a more secure location because administrators can't simply grant themselves access to the audit mailbox and log on to remove any audit reports that they don't want others to see. On the other hand, the change in audit mailbox location means that any audit reports collected in the audit mailbox that you define for Exchange 2010 are ignored for the purposes of the audit reports that you can view through ECP. This shouldn't be a real problem in practice and you can delete the original audit mailbox after you have deployed Exchange 2010 SP1 throughout the organization.

In addition to the change of audit mailbox, SP1 introduces an aging mechanism for audit reports to prevent an unwanted accumulation of data. By default, audit reports are held for 90 days and the MFA removes audit logs after their retention period expires. If you want to change the retention period, you can update it with the Set-AdminAuditLogConfig cmdlet. This command sets the audit log retention period to 182 days (approximately six months):

```
Set-AdminAuditLogConfig –AdminAuditLogAgeLimit 182.00:00:00
```

The audit configuration applies to administrator activity on an organization-wide basis so all of the reports generated across the entire organization go to the one mailbox. As you can imagine, it is all too easy to fill this mailbox with audit reports if you use settings that enable auditing of an extensive set of cmdlets and keep the reports for extended periods.

How administrator auditing happens

Auditing is performed by the Admin Audit Log agent, which evaluates cmdlets as they are run against the audit configuration to decide whether the use of the cmdlet needs to be logged. If so, the agent creates an item containing details of the cmdlet in the Inbox of the audit mailbox. Table 15-8 lists the data that are captured in the audit reports.

Table 15-8 Data captured in audit reports

Field	Description
CmdletName	The name of the cmdlet that was executed
ObjectModified	The object that the cmdlet was used to access (for example, a mailbox)
CmdletParameters	The parameters and values specified for the cmdlet
ModifiedProperties	The properties that were modified by the cmdlet

Caller	The user account that ran the cmdlet
Succeeded	True or False to indicate whether the cmdlet succeeded
Error	Details of any error message that was generated
RunDate	Date and time when the cmdlet was executed in UTC format

The audit agent creates separate reports for each object if you execute an action that is performed against several objects. For example, if you use Get-Mailbox to fetch a list of mailboxes from a database and then use Set-Mailbox to set a new storage quota for each mailbox, the audit agent creates a separate report for each mailbox as it is updated.

You can also write your own entries into the audit log. For example, if you wanted to document a script being run or to take note of a particular administrative operation that you performed to solve a problem, you can capture it with the Write-AdminAuditLog cmdlet as shown here. You can insert up to 500 characters of text into the comment parameter, which is captured in the *CmdletParameters* property of the log entry:

```
Write-AdminAuditLog –Comment 'Server acting up; cleared by increasing HeapSize to
30000'
```

Only one audit mailbox is used for the organization. This can pose some difficulties in widely distributed organizations where actions performed in one part of the network might have difficulty being registered in the arbitration mailbox. Even in highly centralized environments where a small set of administrators perform all actions for the organization, it is still possible to see errors caused by the unavailability of the database that hosts the arbitration mailbox. For example, Figure 15-34 shows what happens when the database containing the audit mailbox is unavailable. Exchange is unable to capture audit entries until the database becomes available again. While the database containing the audit mailbox is unavailable, Exchange writes event 5000 from the msExchange Management Application into the application event for each instance when it is unable to log an audit entry.

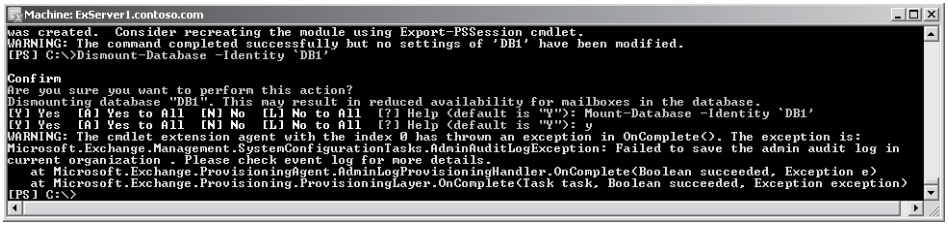


Figure 15-34 EMS flags an error accessing the audit mailbox.

Exchange 2010 doesn't provide facilities to interrogate the audit reports and create analysis of the management activity within an organization. Exchange 2010 SP1 addresses the lack in two ways.

- A set of precanned audit reports that cover the most common needs for an organization (as determined by the Exchange developers) are available through ECP (Figure 15-35). The reports cover both mailbox audit data and administrator audit data.
- The new Search-AdminAuditLog cmdlet is provided to allow administrators to create their own analysis of the administrator audit logs.

The reports provided through ECP depend on the Search-AdminAuditLog and other cmdlets to generate their data. For example, the litigation hold report is created by examining the ligitation hold property set on mailboxes that are on hold and the role groups report examines changes made to role groups to grant permission to users to perform different administrative operations. I'll discuss what happens with mailbox audits and how you can search audit data in the next section.

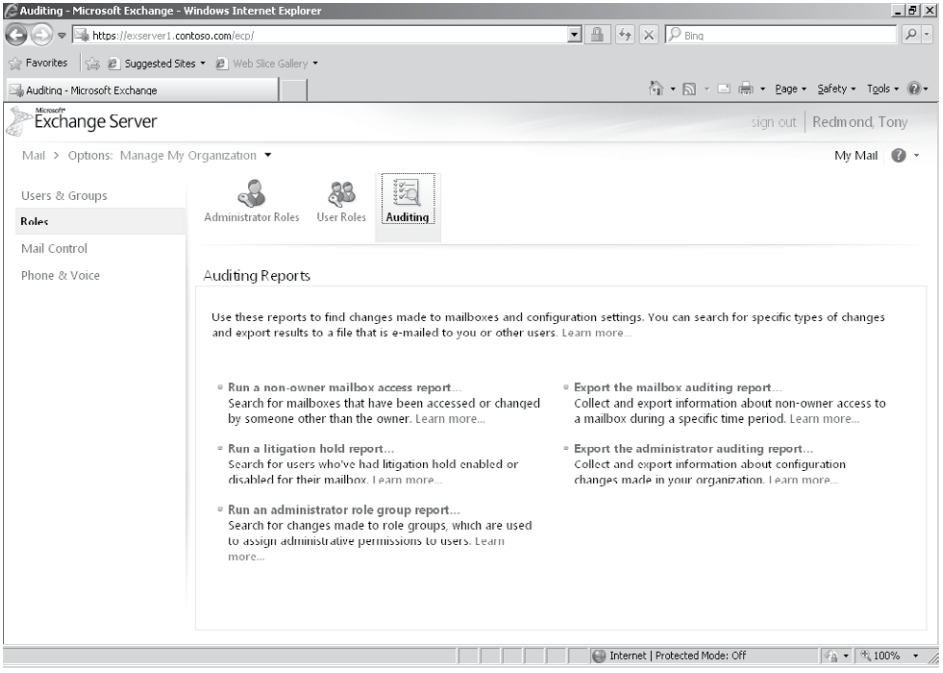


Figure 15-35 Audit reports available in ECP.

Interrogating the audit reports with the Search-AdminAuditLog cmdlet is straightforward. Here are a few examples that illustrate the possibilities that exist to discover just what administrators are doing within the organization:

1. Search for actions performed by one or more users. Each of the users that you want to look for is identified by alias, email address, display name, or distinguished name. Separate the names with commas:

```
Search-AdminAuditLog –UserIds Administrator, 'Tony.Redmond@contoso.com' |
Format-Table Rundate, Caller, CmdletName
```

```
RunDate              Caller                          CmdletName
-----------          --------                        ----------------
4/5/2010 7:45:56 PM  contoso.com/Users/Administrator  Set-Group
4/5/2010 10:18:11 PM contoso.com/Users/Administrator  Set-OrganizationConfig
4/6/2010 1:50:38 AM  contoso.com/Users/Administrator  Set-Mailbox
4/6/2010 1:55:07 AM  contoso.com/Users/Administrator  New-MailboxImportRequest
4/6/2010 1:55:55 AM  contoso.com/Exchange             Users/Redmond, Tony Set-CalendarProcessing
4/6/2010 2:22:21 AM  contoso.com/Users/Administrator  Set-Mailbox
4/6/2010 2:56:57 AM  contoso.com/Users/Administrator  Start-ManagedFolderAssistant
4/8/2010 3:10:16 PM  contoso.com/Users/Administrator  New-Mailbox
4/8/2010 3:10:17 PM  contoso.com/Users/Administrator  Set-User
4/8/2010 4:23:17 PM  contoso.com/Exchange             Users/Redmond, Tony Remove-MailboxSearch
```

2. Search for specific actions. In this example we want to know who has recently mounted or dismounted mailbox databases. To locate the audit records, you specify the cmdlets that are used for these purposes. The *ObjectModified* property returned in each audit log tells you the name of the database that was operated on.

```
Search-AdminAuditLog –Cmdlets Dismount-Database, Mount-Database | Format-Table
RunDate, Caller, CmdletName, ObjectModified -AutoSize
```

```
RunDate               Caller                          CmdletName          ObjectModi-
fied
-----------           --------                        ----------------    -----------
--
3/19/2010 6:06:47 PM  contoso.com/Users/Administrator  Mount-Database      Managers
3/19/2010 6:07:49 PM  contoso.com/Users/Administrator  Dismount-Database   Managers
3/19/2010 6:07:58 PM  contoso.com/Users/Administrator  Mount-Database      VIP Data
```

3. Search for audit records within a particular date range. In this case we want to find out who has been creating new mailboxes over a specified period. Note that we include the *Succeeded* property in the output because it is possible that some attempts to run the New-Mailbox cmdlet were unsuccessful, which is, in fact, what we can see in the results.

```
Search-AdminAuditLog -StartDate "04/01/2010 00:00" -EndDate "04/15/2010 23:59"
-Cmdlets New-Mailbox | Format-Table RunDate, Caller, ObjectModified, Succeeded
```

```
RunDate              Caller                          ObjectModified                    Succeeded
-------              ------                          --------------                    ---------
4/8/2010 3:10:16 PM  contoso.com/Users/Administrator  contoso.com/Users/extest_3a8609984 True
4/8/2010 4:24:20 PM  contoso.com/Users/Administrator  contoso.com/Users/Legal Searches   True
4/9/2010 9:47:33 AM  contoso.com/Users/Administrator  contoso.com/Users/Leitrim Room     False
4/9/2010 9:47:58 AM  contoso.com/Users/Administrator  contoso.com/Users/Leitrim Room     False
4/9/2010 9:48:21 AM  contoso.com/Users/Administrator  contoso.com/Users/Leitrim Room     False
4/9/2010 9:48:34 AM  contoso.com/Users/Administrator  contoso.com/Users/Leitrim Room     True
4/9/2010 3:58:16 PM  contoso.com/Users/Administrator  contoso.com/EMEA/Pelton, David     True
4/9/2010 3:59:01 PM  contoso.com/Users/Administrator  contoso.com/EMEA/Ruth, Andy        True
```

4. Analyze the cmdlets that are being run by administrators. These data are really just for interest's sake, but they do reveal what are the most commonly used cmdlets.

```
Search-AdminAuditLog | Sort CmdletName | Group CmdletName | Format-Table Count,
Name -AutoSize
```

```
Count            Name
--------         -------
26               Add-DistributionGroupMember
2                Add-MailboxPermission
1                Dismount-Database
1                Enable-Mailbox
2                Mount-Database
1                Move-DatabasePath
1                Move-OfflineAddressBook
4                New-ActiveSyncDeviceAccessRule
1                New-ActiveSyncMailboxPolicy
9                New-DistributionGroup
```


INSIDE OUT

Accessing comments from the audit log

One small quirk with the Search-AdminAuditLog cmdlet is that it doesn't return the comments inserted into the audit log with the Write-AdminAuditLog cmdlet. As

you'll recall, the comments store the administrator-specified information that they want to write into the audit log, so these are data that you will want to be able to access. The data are held in the cmdlet parameters property of the audit entry, but if you include this property in the output set, all you will see is the string value "com…

the Search-AdminAuditLog cmdlet into an array and then looking at the appropriate record in the array. For example, these commands create an array and then examine the cmdlet parameters data in the first record in the array:

```
$AuditArray = Search-AdminAuditLog –StartDate '11/1/2010 00:00'
–EndDate '11/1/2010 23:59'
$AuditArray[1].cmdletparameters
```

Capturing audit data for administrator actions does not replace the need for operational discipline in the careful recording of changes made to server and organization configuration, including the following:

- Hotfixes and roll-up updates tested and applied, including any updates for add-on products
- Service packs for Windows and Exchange tested and applied
- Major network updates (for example, the introduction of a new DNS server)
- Installation of new Windows and Exchange servers
- Installation of any software on an Exchange server
- Changes to transport configuration such as the addition of a new connector or a change to transport settings

In this context, audit reports are helpful to record administrator actions, but they are not a complete solution.

Auditing mailbox access

Users gain access to mailboxes in different ways:

- Users log on and use their mailboxes as normal.
- An administrator can delegate full access to a mailbox to another user. See Chapter 6 for more information on this topic.
- Administrators can grant themselves access to a user's mailbox and then log on to it.

Normally administrators do not concern themselves about what happens inside user mailboxes. However, there are mailboxes that contain sensitive information that might need to be protected against attempts to conceal or remove items that are required by the company, typically to justify actions that the company or its employees took in a particular situation such as discussions with other companies relating to a merger, sale, or acquisition. Internally, sensitive information is often captured in discovery search mailboxes that must be monitored to detect any attempt to interfere with the data.

Best Practice: Protecting information by restricting access

Best practice in protecting information is to restrict access to people who can justify their access. In other words, you wouldn't open up your CEO's mailbox to all, but the CEO might delegate access to her executive assistant. In the same way, if a human resources (HR) investigation uses a specific discovery mailbox to capture information relating to potential harassment of an employee, you only grant Full Access to the discovery mailbox to the members of the HR department who absolutely need that access. Another best practice is to always remove access from users as soon as they no longer can justify the access.

Mailbox auditing is a feature introduced in Exchange 2010 SP1 that backs up best practice by allowing administrators to configure mailboxes so that details of specified actions are captured by Exchange. Audit entries are captured in the Audit subfolder of the Recoverable Items folder (the dumpster) and can be interrogated with the Search-MailboxAuditLog cmdlet. Mailbox actions are divided into three categories:

- **The mailbox owner** It is not normal to audit user actions, as they typically have full control over their mailbox contents. In addition, because mailbox owners use their mailboxes on a consistent and ongoing basis, the volume of audit entries is highest when auditing is enabled for the mailbox owner. For these reasons, when you compare the mailbox audit configuration for owners against the other categories, you'll see that the list of audit actions for owners is blank.
- **Delegates** Other users who have been assigned the SendAs, SendOnBehalf, or FullAccess permission can access some or all of a mailbox and take actions to affect its contents.
- **Administrative operations** These are operations such as mailbox moves, mailbox imports from PST, and mailbox discovery searches that are performed by administrators and affect mailbox contents in some way, if only to open folders.

Table 15-9 lists the various actions that Exchange can audit for a mailbox. Those marked with an asterisk (*) are part of the default set of actions that are marked for auditing when you enable auditing for a mailbox. For example, if you enable auditing for a mailbox, then Exchange will record details of all instances when a delegate sends a message using the *SendAs* permission. On the other hand, instances when delegates access the mailbox and send a message using *SendOnBehalf* permission are not captured unless you specifically mark this action for auditing. The decision to include one action over another in the default set of logged actions is probably explained by the fact that a message sent using the *SendAs* permission represents a higher degree of impersonation than one sent using the *SendOnBehalf* permission. We will discuss how to configure actions for auditing in a little while.

Table 15-9 Actions that can be audited for a mailbox

Action	Description	Admin	Delegate	Owner
BulkSync	Synchronization of a mailbox by an Outlook client configured in cached Exchange mode.	Yes*	Yes*	Yes
Copy	A message is copied to another folder in the mailbox or personal archive.	Yes	Yes	Yes
FolderBind	A mailbox folder is accessed (opened) by a client.	Yes*	Yes	Yes
HardDelete	A message is deleted permanently from the database (removed from the Recoverable Items folder).	Yes*	Yes*	Yes
MessageBind	A message is opened or viewed in the preview pane.	Yes	Yes	Yes
Move	A message is moved to another folder.	Yes*	Yes	Yes
MoveToDeletedItems	A message is deleted and moved into the Deleted Items folder.	Yes*	Yes	Yes
SendAs	A message is sent from the mailbox using the SendAs permission.	Yes*	Yes*	Yes
SendOnBehalf	A message is sent from the mailbox using the SendOnBehalf permission.	Yes*	Yes	Yes
SoftDelete	A message is deleted from the Deleted Items folder (and moved into the Recoverable Items folder).	Yes*	Yes*	Yes
Update	The properties of an item are updated.	Yes*	Yes*	Yes

Enabling mailboxes for auditing

The first step in the process is to enable the mailboxes that you want to audit by running the Set-Mailbox cmdlet. You cannot configure mailbox auditing with EMC or ECP. In this example, we enable auditing for the default discovery search mailbox.

```
Set-Mailbox –Identity 'Discovery Search Mailbox' –AuditEnabled $True
```

We can then check that the audit setting is in place with the Get-Mailbox cmdlet:

```
Get-Mailbox –Identity 'Discovery Search Mailbox' | Format-List Name, Aud*
```

```
Name             : DiscoverySearchMailbox {D919BA05-46A6-415f-80AD-7E09334BB852}
AuditEnabled     : True
AuditLogAgeLimit : 90.00:00:00
AuditAdmin       : {Update, Move, MoveToDeletedItems, SoftDelete, HardDelete, FolderBind,
SendAs, SendOnBehalf}
AuditDelegate    : {Update, SoftDelete, HardDelete, SendAs}
AuditOwner       : {MoveToDeletedItems, SoftDelete, HardDelete}
```

You can see that the act of enabling auditing for the mailbox has also assigned the default set of actions to be audited for the different categories of users who log onto the mailbox. You can also see that a property called *AuditLogAgeLimit* is present. This controls how long Exchange retains audit entries in the mailbox and the default value is 90 days. Once audit entries expire, they are removed from the mailbox by the MFA the next time it processes the mailbox.

You can set the value of *AuditLogAgeLimit* to anything up to 24,855 days. This amounts to just over 68 years, which should be sufficient for even the most retentive administrators. Oddly, for whatever reason, the actual coded maximum is 24,855 days, 3 hours, 14 minutes, and 7 seconds, which produces an audit age limit like this:

```
AuditLogAgeLimit : 24855.03:14:07
```

You can clear out all existing audit entries by setting the *AuditLogAgeLimit* property to 00:00:00. If you do this, Exchange will prompt you to confirm that all of the entries should be deleted and will proceed if you confirm that this is what you want to do.

You can decide to include or exclude audit actions for administrators, delegates, or owners by writing out the required actions into the *AuditAdmin*, *AuditDelegate*, and *AuditOwner* properties. For example, the owner of the default discovery search mailbox folder will never log onto it, so we can set the audit settings for the owner to "null". On the other hand, we might want to tweak the settings applied when administrators access the mailbox. Here's how we can make the change:

```
Set-Mailbox –Identity 'Discovery Search Mailbox' –AuditOwner $Null –AuditAdminUpdate,
Move, MoveToDeletedItems, SoftDelete, HardDelete,SendAs, SendOnBehalf –AuditEnabled
$True
```

To reverse the process and turn off auditing for a mailbox, we set the *AuditEnabled* flag to $False as follows:

```
Set-Mailbox –Identity 'Discovery Search Mailbox' –AuditEnabled $False
```

Accessing mailbox audit data

Mailbox audit information is written into the Audit subfolder of the Recoverable Items folder. However, this folder is invisible to any client so you cannot simply log onto the mailbox and browse through the audit entries. Instead, you have to submit a search with EMS and have Exchange retrieve and display the found entries to you. Searches are performed in two ways:

- The Search-MailboxAuditLog cmdlet performs a synchronous search for one or more mailboxes and returns the results on screen.
- The New-MailboxAuditLogSearch cmdlet can search across one or more mailboxes asynchronously in the background and return the results via email.

First, let's perform a simple search for audit entries for a single mailbox. In this example, we are looking for entries for a particular day, so we pass a start and end date. We then select a number of fields to be output for each entry that is found. Specifying the *–ShowDetails* parameter instructs Exchange to output details for each audit entry that it locates and passing "Delegate" to the *–LogonType* parameter restricts output to entries performed by a user who has delegate access to the mailbox. If you search many mailboxes for entries from an extended period, it is more than likely that Exchange might return thousands of entries. In this situation, you can use the *–ResultSize* parameter to specify how many entries you want to be returned. By default Exchange will output 1,000 entries.

```
Search-MailboxAuditLog –Identity 'Ruth, Andy' –ShowDetails –StartDate '5/12/2010
00:01'
–EndDate '5/18/2010 23:59' –LogonType Delegate –ResultSize 100 |
Format-Table Operation,
OperationResult, LogonUserDisplayName, ItemSubject, LastAcccessed
```

```
Operation      OperationResult  LogonUserDisplayName  ItemSubject          LastAccessed
-----------    ---------------  -------------------   ----------           -------------
SendOnBehalf   Succeeded        Executive Assistant   Travel Requests      11/05/2010 15:52:31
SendAs         Succeeded        Andrews, Lisa         Note from Peter      11/05/2010 15:54:41
SoftDelete     Succeeded        Smith, John                                11/05/2010 15:58:46
SendOnBehalf   Succeeded        Executive Assistant   Business directives  11/05/2010 16:05:09
```

This output is what you'd expect from a mailbox that has granted access to different users to perform actions on their behalf, which we see in the *SendOnBehalf* and *SendAs* entries.

Full access is obviously available to user John Smith because this user has been able to delete an item in the mailbox.

If you change the value passed to the *–LogonType* parameter to "Admin" you will see any operations performed against the mailbox as a result of administrative activity. For example, if a mailbox search is performed, you will probably see entries like this:

```
FolderBind       Succeeded      Administrator            11/05/2010 18:24:13
```

Of course, if you find something of interest, there is a lot more detail in an audit entry that can reveal additional information. For example, entries for a mailbox move will show detail like that shown here, whereas the *ClientInfoString* property for a mailbox search will contain "*Client=Management; Action=E-Discovery (mailbox search)*", so it's relatively easy to determine what administrative process accessed the mailbox.

```
FolderPathName           : \MailboxReplicationServiceSyncStates
ClientInfoString         : Client=MSExchangeRPC
ClientIPAddress          : 2002:c0a5:4134::c0a5:4134
ClientMachineName        : EXSERVER2
ClientProcessName        : MSExchangeMailboxReplication.exe
ClientVersion            : 14.1.160.2
InternalLogonType        : DelegatedAdmin
```

The New-MailboxAuditLogSearch cmdlet is designed to operate behind the scenes to fetch audit entries for perhaps many mailboxes on servers across the organization and respond with an email with an XML attachment that contains the search results. The XML data are complete but need to be poured through a formatter to make sense of them, or at least, to make sense for those of us who are not fluent in interpreting raw XML. The command to create a typical background mailbox audit log search looks like this:

```
New-MailboxAuditLogSearch –Name 'Unauthorized Delegate Access review' –LogonTypes
Delegate
–Mailboxes 'CEO Assistant', 'CEO', 'Senior VP-Finance' –StartDate '1/1/2010'
–EndDate '12/31/2010'
–StatusMailRecipients'ComplianceAuditMailbox@contoso.com'
```

Figure 15-36 shows how EMS acknowledges the submission of a new mailbox audit log search. The command that is run creates a search through the audit entries for delegate access that are stored in the three specified mailboxes between the start and end date. If you don't specify any mailboxes, Exchange will return audit data for every mailbox on an Exchange 2010 server in the organization that has been enabled for auditing.

When the search is complete, Exchange records the fact in event 4003 in the Application Event Log and sends an email containing the results to the email address or addresses

specified in the *–StatusMailRecipients* parameter. Figure 15-37 shows an example of the type of email delivered to these recipients. The text of the message contains the search criteria and the attached XML file contains the actual results. The recipients for mailbox audit reports must be mail-enabled objects known to the organization. Normally, they will be mailboxes or groups, but you can arrange for the email reports to go to external recipients such as your auditors, providing that you create a mail-enabled contact that contains their address.

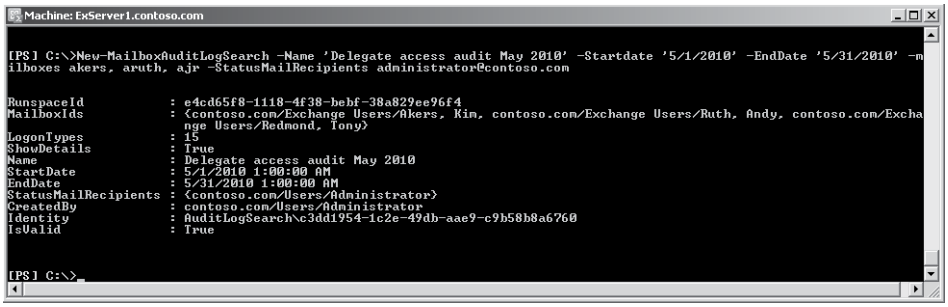


Figure 15-36 Launching a mailbox audit log search with New-MailboxAuditLogSearch.

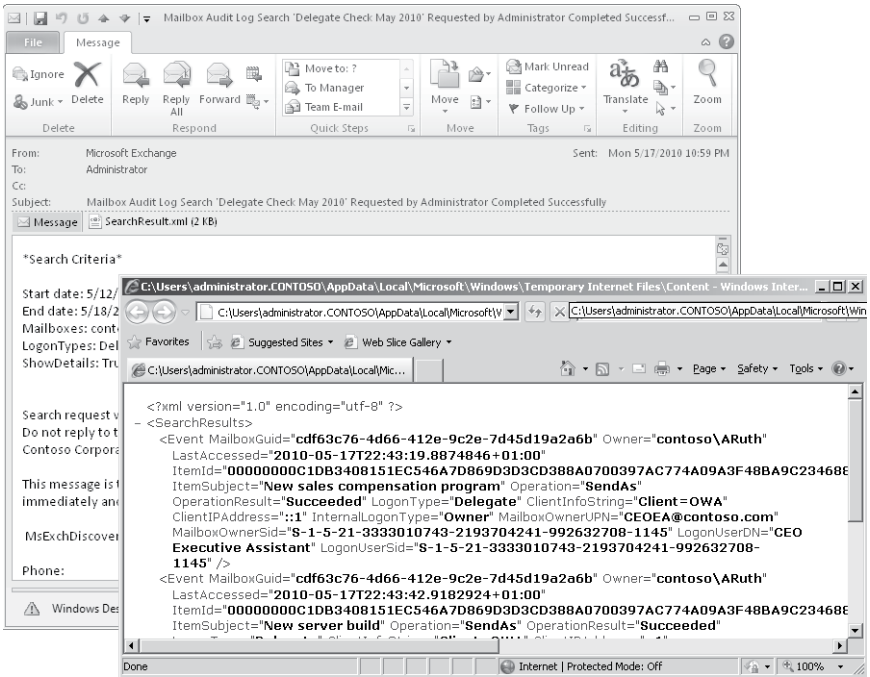


Figure 15-37 Viewing the message with details of a mailbox audit log search and the XML file containing the results.

No one could pretend that mailbox auditing is complete in terms of functionality or presentation. Forcing all interaction through EMS is acceptable if the output was easier to extract and interpret, but that's not the case. Eventually, Microsoft might do the work in a future version or service pack to integrate mailbox auditing into ECP and make this interesting and worthwhile functionality more accessible to administrators.

Message classifications

Microsoft introduced message classifications in Exchange 2007 in an attempt to allow users to apply business-specific labels to messages. The labels are stored as properties of the messages and can be acted on by transport rules. Users can also react to the classifications when they see them on messages through Outlook or Outlook Web App. For example, if a message is labeled "Super Critical," a user might be less likely to delete it and more likely to quickly respond to the request that the message contains.

The problems with message classifications are threefold. First, you rely on senders to apply classifications and receivers to respond appropriately to the classifications when messages arrive. Second, there's no automated method to distribute and publish message classification data to clients such as the approach used by Outlook protection rules, so you have to make an arrangement to distribute the necessary classification information to Outlook clients. By comparison, Outlook Web App picks up classifications direct from the server. Third, only Outlook 2007 and Outlook 2010 clients and Outlook Web App support classifications. Any other client simply ignores their presence.

Even with the acknowledged limitations, there are customers who find message classifications a very useful and worthwhile facility. The characteristics of these customers usually include the following:

- A requirement to classify messages in terms of their sensitivity, content, or intended audience that is well understood by the user population.
- A well-structured classification scheme for information that makes sense to users and the business. Ideally, email shares the same classification scheme as applied to documents and other information. Multiple classification schemes invariably cause confusion and result in a lesser degree of compliance by users.
- The ability to deploy updates to Outlook clients in a reliable manner.
- A population of Outlook 2007 and Outlook 2010 clients.

Typical classifications include labels such as Personal, Secret, Business Critical, Audit Retention, and so on. The diversity of businesses is reflected in the diversity of classifications that are created and applied to material. However, within an individual company, it is a good idea to keep the number of classifications to a minimum to avoid user confusion and to encourage compliance.

INSIDE OUT

Message classifications have value but are of limited use

It's important to understand that message classifications are purely indications of the importance of content in one way or another. Apart from being recognized as an actionable condition by transport rules, there is no other way to use a message classification to force Exchange to do anything. In short, message classifications are an inert and passive component of the overall MRM environment.

Creating a message classification

You can't create a new message classification using EMC or ECP. All manipulation of message classifications is performed through EMS. Before creating any message classifications, it's worthwhile to sit down and chart out the various classifications that you think are needed by the business. Too many classifications are likely to confuse users if they don't know which classification to apply in a given circumstance, whereas too few might not achieve the granularity of classification required by the business. Each classification therefore should be justified by business logic and its application should be easily explainable to a user. In other words, the use and purpose of every message classification must be blindingly obvious if it is to succeed.

Let's assume that we have done the necessary due diligence and have decided to create a new message classification called "Top Secret." We can do this with the New-MessageClassification cmdlet:

```
New-MessageClassification –Name 'Top Secret' –DisplayName 'Top Secret: Eyes Only'
–SenderDescription 'This message contains Top Secret information that must not be
shared outside the company' –RecipientDescription 'This message contains Top Secret
information that must not be shared outside the company: Do not forward!'
```

Table 15-10 explains the parameters that you can use with the New-MessageClassification cmdlet and their meaning.

Table 15-10 Parameters used with the New-MessageClassification cmdlet

Parameter	Description
Name	The internal name of the message classification that identifies the object to Exchange.
DisplayName	The name of the message classification that is visible to Outlook 2007 and Outlook 2010 and Outlook Web App clients when they select a classification to apply to a message. You can create a name of up to 64 characters.
SenderDescription	Text that describes the purpose of the message classification displayed by clients after a user adds the classification to a message.
RecipientDescription	Text displayed to recipients to help them understand what they should or should not do with a message bearing a particular classification. If this parameter is not set, clients use the value of the *SenderDescription* instead.
Locale	A parameter indicating that the message classification is for a particular language. If omitted, the message classification is created with the default locale, meaning that it is used by mailboxes of any language where a localized classification is not available.
RetainClassificationEnabled	Specifies whether the message classification persists with the message if it is forwarded or replied to. The default value is $True.
UserDisplayEnabled	Controls whether users see the *RecipientDescription* or *DisplayName* information for a message classification on a message that they receive. The default value is $True, meaning that recipients see this information. You can suppress it by setting the parameter to $False. In this case, Exchange carries the classification but doesn't reveal it to recipients. However, the classification can still be operated on by a transport rule.
DisplayPrecedence	Outlook and Outlook Web App only allow users to select a single message classification for a message but a transport rule can apply another classification. This parameter controls the order in which the classifications are displayed to users. The default is medium, but you can set other values: Highest, Higher, High, MediumHigh, Medium, MediumLow, Low, Lower, and Lowest.

Our new classification is now registered in Active Directory and is replicated throughout the forest. Users will be able to use the new classification the next time they connect to Exchange. We can retrieve the details of the classification with the Get-MessageClassification cmdlet. For example:

```
Get-MessageClassification –Identity 'Top Secret'
```

```
ClassificationID            : 7387906a-20fe-49e9-bedf-6a637f104b93
DisplayName                 : Top Secret: Eyes Only
DisplayPrecedence           : Medium
Identity                    : Default\Top Secret
```

```
IsDefault                   : True
Locale                      :
RecipientDescription        : This message contains Top Secret information that must not be
shared outside the company: Do Not forward!
RetainClassificationEnabled : True
SenderDescription           : This message contains Top Secret information that must not be
shared outside the company
UserDisplayEnabled          : True
Version                     : 0
```

Localized message classifications

The message classification that we just created will work for all languages, but let's assume that we have to support a group of French users. We can create a separate message classification for French that includes translated or localized text that Exchange will provide to clients connected to mailboxes that have French listed as a supported language. When we create the French version of the "Top Secret" message classification, we identify it as a localized version by including the locale parameter. For example:

```
New-MessageClassification –Name 'Top Secret' –DisplayName 'Top Secret: Eyes Only'
–SenderDescription "Ce message contientl'informationextrêmementsecrète qui ne doit
pas êtrepartagée en dehors de la compagnie"
–RecipientDescription"Ce message contient l'information extrêmement secrète qui ne
doit pas être partagée en dehors de la compagnie : N'expédiez pas !" –Locale 'fr-FR'
```

If you use the Get-MessageClassification cmdlet to list the message classifications known to Exchange after creating the localized version, you won't see the localized version listed because the cmdlet only returns classifications for the default language. To retrieve details of a localized classification, you have to pass its identifier. For example, in this instance we need:

```
Get-MessageClassification –Identity 'fr-FR\Top Secret'
```

Client access to message classifications

Because Outlook Web App reads in message classification data each time it creates a new session, assuming that Active Directory replication is working smoothly, new or updated message classifications are available to Outlook Web App clients the next time that they connect to a server. The situation is less satisfactory for Outlook because you have to instruct Outlook to read in classification data from a location with registry keys. Because of the difficulty of distribution of updates to all clients in a reliable and robust manner, it is bad news for administrators any time that you have to customize Outlook through registry keys.

Access to message classifications from Outlook Web App is very straightforward. When you create a new message, you can click the Restriction icon (envelope overlaid with a red Access Denied sign) to see the set of classifications available within the organization. The top screen shown in Figure 15-38 shows four available classifications. The No Restriction classification is the default value that Exchange applies to message if the user doesn't select an explicit classification. The Do Not Forward classification is in fact the default AD RMS template. We discuss AD RMS in detail very soon.

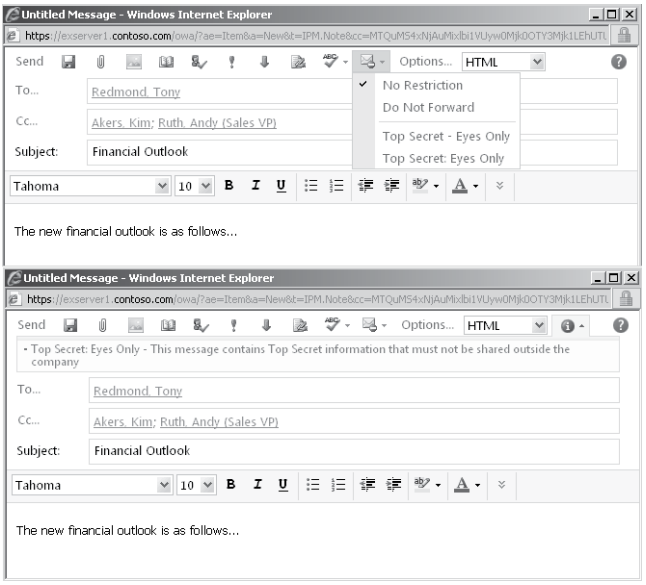


Figure 15-38 Using message classifications with Outlook Web App.

The important point that comes through here is that if you use AD RMS and message classifications, you have to make sure that you use a naming convention that clearly identifies message classifications and templates to users. The problem here is that we have an AD RMS template called "Top Secret – Eyes Only" and a message classification called "Top Secret: Eyes Only". The poor users won't know what to choose without clear direction and some help through the naming convention. Let's assume that this is a very clever user, so he knows that the "Top Secret: Eyes Only" classification is the one he wants. When you select a classification, Exchange displays the text defined in the classification's *SenderDescription* property. The bottom screen in Figure 15-38 shows how the text for the message classification is displayed. Much the same happens when the message is delivered except that the text is taken from the *RecipientDescription* property (Figure 15-39).

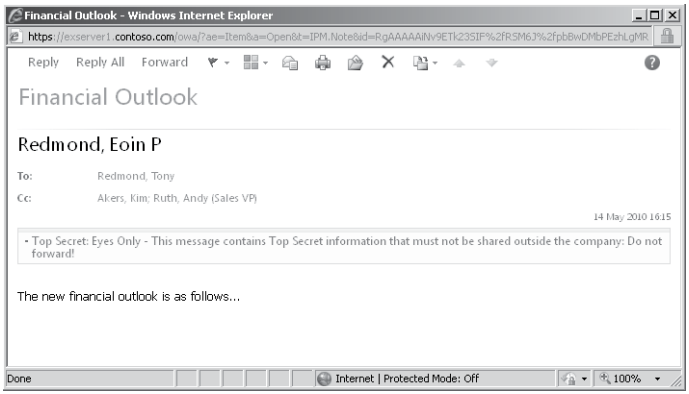


Figure 15-39 Recipient text displayed by Outlook Web App for a message classification.

Making message classifications available to Outlook 2007 and Outlook 2010 clients is a multistage process. You have to do the following:

1. Export the message classifications from Active Directory to an XML file. Exchange provides the Export-OutlookClassification.ps1 script for this purpose. Run the script to output the XML file as follows:

   ```
   Export-OutlookClassification.ps1 > c:\Temp\Classifications.XML
   ```

2. The XML file contains information about the message classifications that are currently known to Exchange and establishes the set of classifications that are available to Outlook. It must be regenerated each time you make changes to classifications or add or delete classifications. The XML content for a classification looks like this:

   ```
   <Classification>
   <Name>Top Secret: Eyes Only</Name>
   <Description>This message contains Top Secret information that must not be
   shared outside the company</Description>
   <Guid>7387906a-20fe-49e9-bedf-6a637f104b93</Guid>
   <AutoClassifyReplies />
   </Classification>
   ```

3. Each Outlook client must be separately enabled to use message classifications. This is done by updating the system registry to instruct Outlook to enable classifications and where to find the XML file that describes the available classifications. A suitable registry update file for Outlook 2010 is shown here. Change “14.0” to “12.0” in the first line to create an update file for Outlook 2007 clients. The three values instruct Outlook where to find the XML file, enable message classifications, and assert that the classifications placed on messages are trusted by the Exchange organization. Exchange 2003 supports a different form of message classification so you should

set this registry value to 0 (zero) for users who still have their mailboxes on an Exchange 2003 server.

```
[HKEY_CURRENT_USER\Software\Microsoft\Office\14.0\Common\Policy]
"AdminClassificationPath"="\\ExServer1\Outlook\Classifications.xml"
"EnableClassifications"=dword:00000001
"TrustClassifications"=dword:00000001
```

4. The next time Outlook starts it discovers that classifications are enabled and attempts to read in the XML file. If the XML file is unavailable for some reason (for example, the server that hosts the share is offline or Outlook is working offline and can't contact the server), Outlook won't be able to add classifications to messages. If you want classifications to be available all the time, you can either distribute the XML file to every PC so that it is available locally or you can put it in a network share and use the Windows offline files feature to synchronize the file so that it is in the local files cache and therefore available even when offline.

All of the steps just described are reasonably straightforward to accomplish on a single computer. Things get messy when you have to ensure that every computer in a large organization has access to the XML file and receives updates after they are made available in a reasonably guaranteed manner. Once this is done, probably through a mixture of group policies and offline file synchronization, you are ready to use message classifications with Outlook clients.

> **Note**
>
> **If an Outlook client uses an outdated classifications file, it will not be able to add new classifications to messages, but it will be able to see new classifications if they have been added by a client that has access to an updated XML file.**

See Chapter 16, "Rules and Journals," for an example of how to use the Top Secret message classification in a transport rule that stops messages marked with this classification from going outside the organization.

Protecting content

Message classifications are useful to a point. They don't impose any restrictions on users and apart from providing some advice as to how important information contained in a message is, users can blissfully ignore their invocation to deal with the information in any particular way. Another approach is necessary if you want to impose restrictions on users

as to what they can and cannot do with content. This isn't a new requirement. For almost 20 years, companies have attempted to protect sensitive information that users transmit in email. The first attempts were based on message encryption and required the sender and recipient to share common (public) keys. The first versions of Exchange-based message encryption used the Windows Public Key Infrastructure (PKI). Some companies experienced success with message encryption but others found that PKI-based systems require a lot of administrative knowledge and intervention (at times) and users often didn't use encryption when they should. In addition, software and hardware upgrades for server and clients had to be carefully managed to ensure that keys were preserved. The result was often disappointing in terms of the degree of protection achieved across the organization.

Software vendors began to explore different approaches to content protection toward the end of the 1990s. These solutions typically required users to explicitly protect a message by invoking a client add-in and relied on communication with specially designated servers. It was difficult to achieve true protection across a range of Web, mobile, and desktop and laptop clients working in offline and online modes. Costs were high because of the need to buy, deploy, and manage additional client and server software and achieving interoperability with partners was tricky. Needless to say, mergers and acquisitions posed even more challenges.

Apart from the ability to send S/MIME protected messages, Exchange 2010 supports two more sophisticated mechanisms to protect content:

- **Information Rights Management (IRM)** IRM relies on templates that describe different sensitivity levels for content and the actions that users can take when they receive protected contents. IRM depends on AD RMS and is now well-integrated throughout Exchange 2010 in terms of client support (Outlook Web App, Outlook, Windows Mobile, and BlackBerry), its support for federation, and the ability to apply IRM through transport rules. Exchange is able to decrypt IRM protected content to inspect it during transport rules processing and discovery searches and can journal protected content in a satisfactory manner. In short, IRM is the approach most suitable for enterprises that are willing to dedicate sufficient resources to achieving a sophisticated and comprehensive level of protection.

- **Outlook protection rules** An Exchange 2010 administrator can create rules that are pushed out to Outlook 2010 clients, which then use the rules to automatically apply AD RMS templates to protect content as users create and send messages. For example, a rule might say that any message sent to a specific distribution group must be protected. Outlook protection rules are in their first iteration and have a major dependency in that they cannot be used without first deploying Outlook 2010 clients.

Planning for the deployment of protected content within an enterprise is not just a matter of selecting and deploying technology. User education is often the biggest obstacle

to overcome, especially in large, multinational, and highly distributed companies. It's also not a matter that the Exchange administrators can decide on alone. The Active Directory management team has to be involved if you want to use AD RMS and the corporate security team and legal group need to be involved to ensure that content protection is aligned with other security initiatives and complies with any legal or regulatory regime to which the company is exposed. The topic is therefore complex at many levels and the best idea is to research the information available on TechNet and other sources before you progress too far along toward a decision to deploy.

Active Directory Rights Management Services

Active Directory Rights Management Service (AD RMS) is a new version of Windows Rights Managment Service (RMS) that allows companies to protect sensitive information by applying policies that dictate how that information can be accessed and shared by users. In an email context, AD RMS can be used with Exchange 2010 to stop users from sending or forwarding messages to unauthorized recipients or otherwise sharing contents by printing, cutting and pasting into other messages, and so on. RMS had a somewhat checkered history because it is very Windows-centric and didn't meet the needs of companies that had heterogeneous environments. AD RMS runs on a Microsoft Windows 2008 server and leverages Active Directory extensively, so it still poses some issues for companies that don't depend on Microsoft for large sections of their IT infrastructure. However, the latest implementation does offer some new features in conjunction with Exchange 2010 that make it more interesting to a messaging administrator. One major advance is the fact that Outlook Web App now supports protected messages for non-Microsoft browsers running on non-Windows platforms, so it is now possible to deploy rights management in a heterogeneous multiplatform infrastructure and expect that users will be able to read protected information. (In addition, Microsoft Outlook for Mac OS X supports AD RMS, as do Windows Mobile 6.1 and later.)

Every message that flows through Exchange passes through the transport service. It is therefore sensible to use this choke point as the place to impose restrictions on email content. Transport protection encryption is an integration that allows you to use AD RMS with Exchange transport rules to apply AD RMS templates to messages as they are processed by the transport service. For example, you could apply the standard Do Not Forward AD RMS template to outgoing messages sent by your senior management team to members of the board of directors to stamp the messages as highly confidential and to ensure that they cannot be forwarded. A key part of making this feature work is that the Exchange 2010 transport system can read protected messages so that journaling and transport rules can be applied. In response to user demands, Exchange 2010 Unified Messaging also supports protection for voice mail messages by marking private messages with the Do Not Forward AD RMS template before they are submitted to a hub transport service.

AD RMS is supported by Outlook 2010, Outlook Web App, Windows Mobile, and some non–Windows Mobile devices. Windows Mobile clients depend on the AD RMS prelicensing agent that was first shipped in Exchange 2007 SP1. The agent runs on hub transport servers and proactively requests licenses to allow Outlook 2007 (and later) and Windows Mobile 6.0 (and later) clients to read protected content without having to first submit credentials. The idea is to make protected content less onerous to deploy and use.

CAUTION

It's worth emphasizing that technology is one part of the solution for information leakage. Many projects have announced their intention to deploy technology to protect information, only to run into the two rocks of user discontent and heterogeneous systems. If users don't buy into the idea that information must be protected and accept the technology that is deployed for this purpose, they will attempt to get around the strictures and information will continue to leak. And if the chosen technology cannot accommodate all the platforms used within an organization, including all varieties of mobile devices, the same lack of success is firmly on the project's horizon.

Installing Active Directory Rights Management

There's plenty of information available in TechNet to help guide you through the detail of installing and configuring Active Directory Rights Management Service so we do not provide a detailed step-by-step guide here. Essentially, the following are the headline steps.

1. Decide on the server that will host the AD RMS cluster (the server that will control the licenses that underpin rights protection within the organization). The server can run Windows 2008 SP2 (a hotfix is required for this version of the operating system) or Windows 2008 R2. It is not best practice to install AD RMS on the same server as Exchange 2010.

2. Install the Active Directory Rights Management Services role on the server (Figure 15-40). The installation wizard leads you through many screens and you might want to perform an installation on a test server before you attempt to introduce AD RMS into the production environment.

3. Perform the postinstallation steps to configure Exchange 2010 to use IRM.

4. Validate that everything is running properly and that users can protect and access content using IRM.

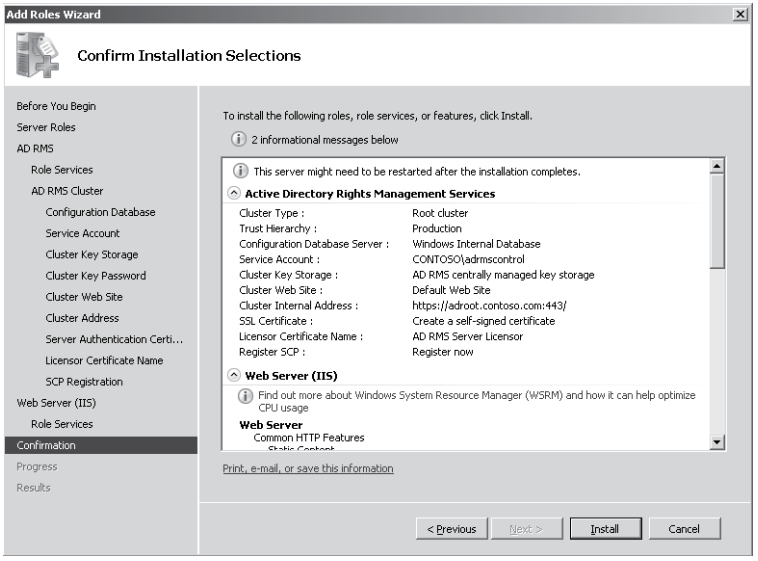


Figure 15-40 Installing Active Directory Rights Management Services on a Windows 2008 R2 server.

After a successful installation of Active Directory Rights Management Service, there are still some steps that must be performed before AD RMS is ready to support Exchange:

- The certificate specified for use by the installation (or created by the installation) must be installed on Exchange mailbox and hub transport servers. You can export the certificate from the server that supports the AD RMS cluster that will be used by Exchange and import it into the Exchange mailbox and hub transport servers. The certificate is required to provide Secure Sockets Layer (SSL) communications between Exchange and the AD RMS server.

- The Exchange Servers security group must be granted read and write permission to the AD RMS cluster certification pipeline. This is a file called \inetpub\wwwroot_wmcs\certification\ServerCertification.asmx located on the AD RMS cluster. Exchange hub transport servers need to be able to access this file to be able to use the AD RMS prelicensing agent to request licenses to access protected content proactively on behalf of clients.

- The account for the federated delivery mailbox must be added to the super users group for the AD RMS cluster. This step allows Exchange to decrypt protected messages as they pass through the transport system (to apply rules based on message content), to journal protected messages, to incorporate protected items into Exchange content indexes, and to allow Outlook Web App users to use IRM.

- IRM features must be enabled for messages sent to internal recipients (within the same Exchange organization) to allow them to access AD RMS templates. To perform this step, run the Set-IRMConfiguration cmdlet as follows:

  ```
  Set-IRMConfiguration –InternaLicensingEnabled $True
  ```

  You can check the IRM configuration with the Get-IRMConfiguration cmdlet.

After taking these steps, you can verify that the IRM configuration is valid by running the Test-IRMConfiguration cmdlet. In this example we use two email addresses to verify that prelicensing and journal encryption works:

```
Test-IRMConfiguration –Recipient TRedmond@contoso.com –Sender epr@contoso.com
```

```
Results : Checking Exchange Server ...
              - PASS: Exchange Server is running in Enterprise.
          Loading IRM configuration ...
              - PASS: IRM configuration loaded successfully.
          Retrieving RMS Certification Uri ...
              - PASS: RMS Certification Uri: https://adroot.contoso.com/_wmcs/certification.
          Verifying RMS version for https://adroot.contoso.com/_wmcs/certification ...
              - PASS: RMS Version verified successfully.
          Retrieving RMS Publishing Uri ...
              - PASS: RMS Publishing Uri: https://adroot.contoso.com/_wmcs/licensing.
          Acquiring Rights Account Certificate (RAC) and Client Licensor Certificate (CLC)
...
              - PASS: RAC and CLC acquired.
          Acquiring RMS Templates ...
              - PASS: RMS Templates acquired.
          Retrieving RMS Licensing Uri ...
              - PASS: RMS Licensing Uri: https://adroot.contoso.com/_wmcs/licensing.
          Verifying RMS version for https://adroot.contoso.com/_wmcs/licensing ...
              - PASS: RMS Version verified successfully.
          Creating Publishing License ...
              - PASS: Publishing License created.
          Acquiring Prelicense for 'tredmond@contoso.com' from RMS Licensing Uri
(https://adroot.contoso.com/_wmcs/licensing)
          ...
              - PASS: Prelicense acquired.
          Acquiring Use License from RMS Licensing Uri
(https://adroot.contoso.com/_wmcs/licensing)
              - PASS: Use License acquired.

          OVERALL RESULT: PASS
```

After you are sure that the basic AD RMS configuration is working, you can consider the finer details of your deployment. By default, AD RMS installs a template called Do Not Forward that users can apply to messages to mark them as confidential. A template defines the rights that users have over items to which the template is applied and you might want

to create some additional templates to meet business needs within the company. For example, Figure 15-41 shows the properties of a template called Top Secret – Do Not Share as viewed through the AD RMS console. This template is purposely restrictive because we don't want to allow users to do much except view its contents and reply to the messages that are marked with the template. Behind the scenes the AD RMS server distributes new and updated templates to clients so that they are available for use.

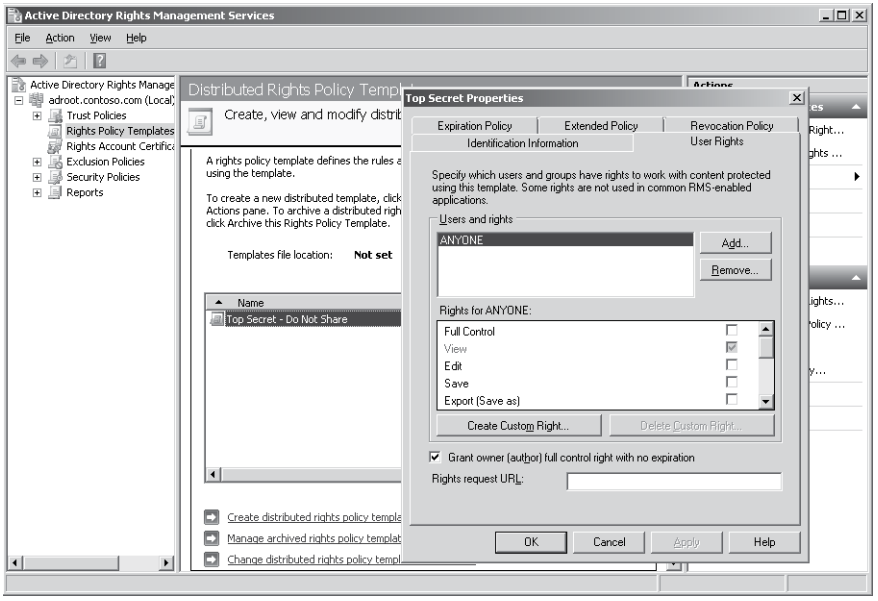


Figure 15-41 Viewing the properties of the "Top Secret" AD RMS template.

You can discover the set of templates that are currently defined with the Get-RMSTemplate cmdlet. For example:

```
Get-RMSTemplate
```

```
Name                          Description
--------                      ---------------
Top Secret – Do Not Share     The Top Secret template is applied to our most secret documents
Do Not Forward                Recipients can read this message, but they can't forward, print, or
```

Using AD RMS to protect content

Figure 15-42 shows the Top Secret template being applied to a new message with Outlook Web App. The same drop-down menu is used for message classifications, which we discussed earlier in this chapter, so if you define some message classifications, you'll see them

listed here along with AD RMS templates. Once again, this reinforces the necessity for a good naming convention to help users do the right thing.

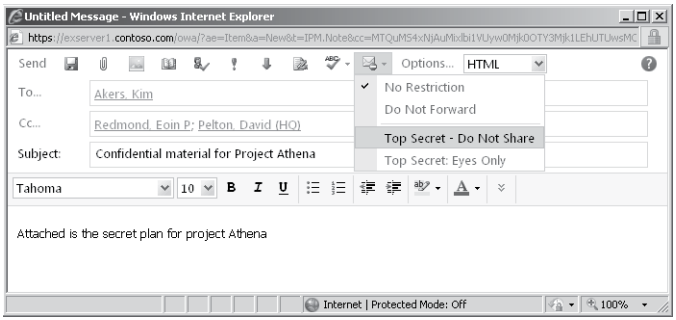


Figure 15-42 Marking a message with the Top Secret AD RMS template.

Once a template is applied, it remains with the message no matter where it passes through the messaging system. The transport system is able to decrypt protected content. Decryption occurs first in the transport pipeline to ensure that subsequent agents can process the content to apply transport and journal rules. Protected messages can be discovered by mailbox searches so that any relevant protected items are included in the content captured in a discovery mailbox.

The first time that a client accesses protected content during a session, it contacts Exchange and the AD RMS server to fetch the credentials that are necessary to access protected items. The understandable need to retrieve credentials can slow down access to protected content, especially when the client is separated from the AD RMS server by an extended network connection. Outlook Web App is a lot less obvious than Outlook is when it comes to configuring itself for rights management, as the user will probably be unaware of the work going on behind the scenes. As you can see from Figure 15-43, Outlook 2010 provides the user with a lot more insight about the configuration process.

The user interface of clients also adjusts to take account of the rights defined in the template. As you can see in Figure 15-44, Outlook Web App has disabled the Forward and Print options because these rights are not allowed for items marked with the Top Secret template. In addition, if you reply to a message the client will not be able to include the text of the original message, as it is protected, so the original message will be added as an attachment to the reply.

You can also see evidence that the transport system has processed the message because the corporate disclaimer has been appended. We will develop the topic further when we discuss transport rules in Chapter 16 and see how the transport system can automatically apply AD RMS templates to protect confidential information in messages as they pass from user to user.

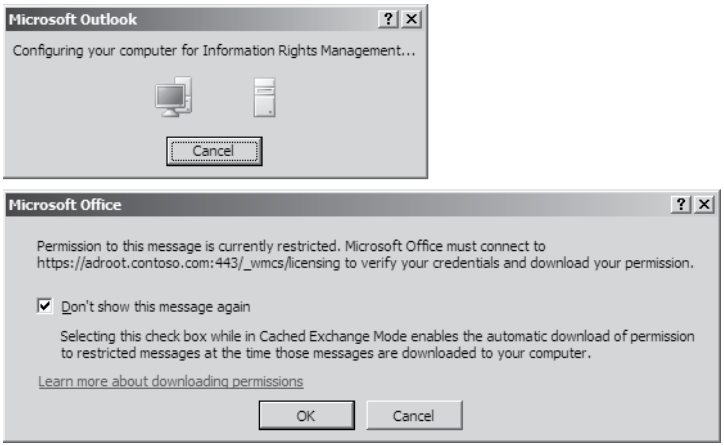


Figure 15-43 Outlook 2010 is configured for rights management.

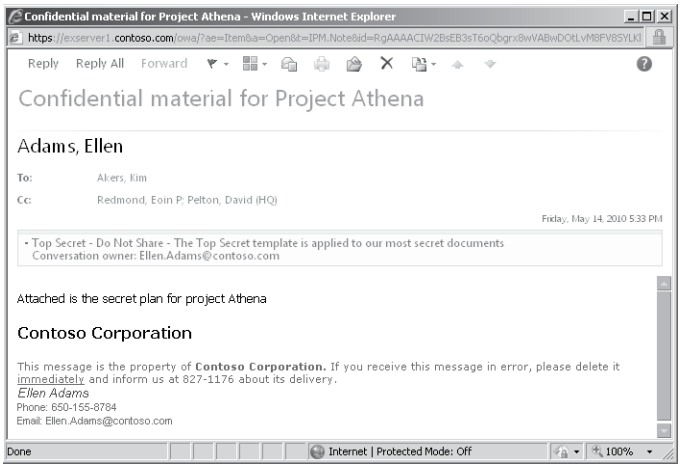


Figure 15-44 Viewing a message marked with the Top Secret AD RMS template.

Figure 15-45 shows how Outlook 2010 displays a protected message (all versions from Outlook 2003 SP2 are able to access protected content). The View Permissions option is accessed by clicking the highlighted bar that shows details of the template that protects the item. When selected, Outlook displays the permissions defined in the template. As you can see from the list, the template applied to this item does not allow users to do much except view or reply to the item. They cannot print, save, export, or even copy the item (and the restriction on copying goes so far as to prohibit any attempt to save a view of the content with a screen capture tool).

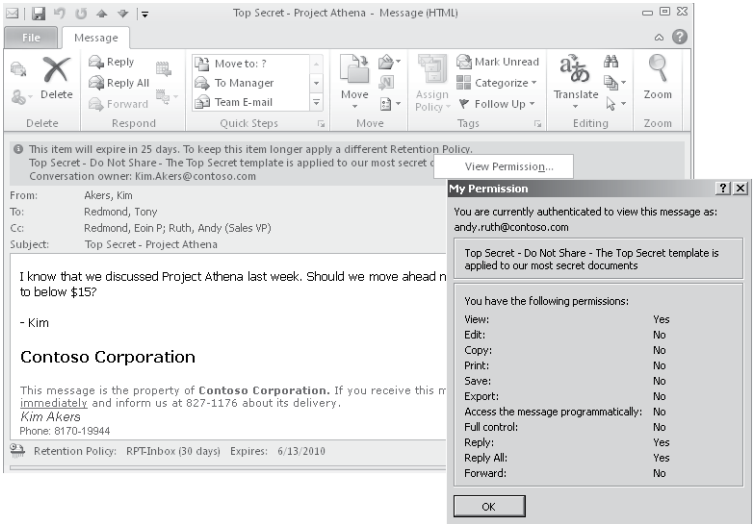


Figure 15-45 Viewing permissions on a protected message with Outlook 2010.

Unless they can access the necessary credentials, external recipients won't be able to access protected content if a user forwards a protected message outside the organization. This is the biggest value of applying protection to messages because it stops users from being able to share confidential or other sensitive information either by accident (forwarding the wrong message or sending it to the wrong recipient) or deliberately. Information leakage is deemed by some companies to be a form of corporate sabotage and carries severe consequences for the responsible employee.

Protecting information on Windows Mobile clients

The ability to protect content also extends to Windows Mobile clients as long as they run Windows Mobile 6.1 or later. As with many advanced features for mobile devices, the newer the operating system, the easier it is to use advanced features like rights management.

The ability to support rights management for Exchange is accomplished by having ActiveSync prefetch the credentials necessary to view protected content as it downloads items to the mobile device. All of the necessary decryption and encryption is performed by the Client Access Server (CAS) during synchronization after the CAS recognizes that the client is capable of handling protected content. Part of the synchronization process downloads the list of available rights templates to the device for local storage and access. A protected item that arrives on the mobile device is decrypted and the processing to display the template and respect the rights that it describes becomes the responsibility of the mobile device.

Outgoing messages are processed by the CAS to ensure that template restrictions are applied before new items are transmitted to recipients. However, this is an implementation that might not be supported by all ActiveSync clients, so it's something that you need to verify as you test mobile devices to include in your deployment.

Rights management enhancements in Exchange 2010 SP1

A number of changes are made in Exchange 2010 SP1 to improve the effectiveness of rights protection. Protected attachments can now be read using the Outlook Web App WebReady document viewer. Also, Microsoft now supports IRM between on-premise servers and Exchange running in its Office 365 hosted service. This is accomplished through the Trusted Publishing Domain (TPD) feature that allows the RMS cluster running in the on-premise deployment to trust and use the licenses issued by the Office 365 servers and vice versa. Much the same approach can be taken to deploy federated IRM between two organizations.

After a rough start, Microsoft has improved the implementation and capabilities of its rights management solution to a point where it is very usable, providing that you deploy the latest clients. However, the success or failure of any solution that aims to protect confidential material is not dependent on technology alone; it is much more important to achieve strong user buy-in and appreciation of the need to protect the material. Unless this aspect of the project is achieved, it will eventually fail because users will find other ways to share information with each other, maybe by pasting it into Facebook!

Outlook Protection Rules

If you use Outlook 2010 clients and have AD RMS deployed, you can use a complementary function called Outlook Protection Rules that allows Exchange administrators to create new rules and distribute them to Outlook users to have protection automatically applied at the client rather than waiting for messages to be transmitted to Exchange. Although it adds another layer of security for messages in corporate email systems, this feature is also intended for use in scenarios where companies use Exchange in hosted services and want to ensure that content is protected against snooping or other unauthorized access by the managers of the hosted service. Outlook Protection Rules require you to install an add-on to Outlook 2010 before they work. After that, the administrator creates and distributes new rules using the set of cmdlets described in Table 15-11. You need to be an organization administrator to be able to use these cmdlets.

Table 15-11 **Outlook Protection Rule cmdlets**

Cmdlet	Use
New-OutlookProtectionRule	Create a new Outlook Protection Rule
Enable-OutlookProtectionRule	Enable an Outlook Protection Rule and make it available to clients
Get-OutlookProtectionRule	Return information about a selected Outlook Protection Rule or all of the rules within the organization
Set-OutlookProtectionRule	Set properties of an existing Outlook Protection Rule
Remove-OutlookProtectionRule	Remove an Outlook Protection Rule from the organization
Disable-OutlookProtectionRule	Disable an Outlook Protection Rule

Essentially, an Outlook Protection Rule establishes the conditions for when to apply the rule, states the scope of recipients for whom the rule applies, and tells Outlook what AD RMS template it should apply when the conditions and scope are satisfied. Outlook monitors new messages as they are created and will load the necessary templates when it first detects that it might need them during a session. The scope can be internal recipients, all recipients, or specific recipients.

To take a practical example, let's assume that we have been told that we need to protect any message sent to the CEO's staff. This EMS command creates a new rule that applies the default Do Not Forward AD RMS template to any message sent to the SMTP address specified in the rule (in this case, the address is for the distribution group used to map all of the CEO staff):

```
New-OutlookProtectionRule -Name "CEO Staff Communications" -SentTo
CEOStaff@contoso.com -ApplyRightsProtectionTemplate "Do Not Forward" Priority 1
```

After an Outlook protection rule is defined, it is distributed to clients using Exchange Web Services. If you make a change to a rule, it will take about an hour to redistribute the rule as they are cached for better performance. You can force a rule to be distributed by recycling the Microsoft Internet Information Services (IIS) process. In addition, you have to restart Outlook to make new rules available to the client.

For the rule to work properly, Outlook must have access to the AD RMS template. The default Do Not Forward template is automatically made available to clients when they contact the AD RMS server for the first time. If you create new templates, they will have to be distributed so that the client can access the XML that describes the template. To check that a template is available to Outlook, create a new message and click the Options tab to view the list of templates available through the Permission list. If the template doesn't appear here it means that it has not yet been distributed to the client computer.

For advice about how best to distribute templates, see the AD RMS Template Deployment Step-by-Step guide at *http://go.microsoft.com/fwlink/?LinkID=153712*.

Rules help compliance, too

Exchange supports transport and journal rules. These could be considered to be part of the compliance landscape, but because they are capable of solving so many more problems for a hard-pressed Exchange administrator they deserve their own chapter. Let's go and talk about rules, and we will see how some of the topics presented in this chapter reoccur in that domain.

Index

Symbols

*** (asterisk), 152**
! (exclamation mark), 768
– (hyphen), 105, 113
. operator, 104, 119, 122
"- -" address space, 915
*** address space, 915**
$_ variable, 104, 119, 122
" (quotation marks), 104, 124

A

A records, 675, 678, 942
ABNF (Augmented Backus-Naur Form), 1118
accepted domains
  about, 886–887
  creating, 888
  updating, 889
Account Picker dialog box, 310
ACID test, 409
ACLL (attempt copy last logs), 439–440, 495
ACLs (access control lists). *See also* RBAC
  about, 147
  cmdlet support, 80
  Organization Management role and, 158
  viewing group members, 323
Activate Database Copy Wizard, 488, 496
activation blocks, 499–500, 509
Active Directory (AD)
  access to data, 1104
  ADSIEdit support, 22
  caching configuration data, 28
  custom attributes, 35–36
  DAG considerations, 473
  data storage locations, 21
  deployment preparations, 12, 15, 19, 23–24, 31–33
  ECP support, 215
  edge servers and, 909
  EMC support, 190
  EMS dependencies, 86
  evaluating attributes, 1099–1100
  Exchange installation process, 20–21
  installation preparations, 41
  mailbox naming conventions, 257
  MailTips and, 763
  overriding site link costs, 808–809
  planning for global catalogs, 29–31
  PowerShell support, 126–128
  remote management tools, 40
  role of ADAccess, 25–28
  routing and, 806–808
  schema customizations, 34–35
  schema upgrades, 21, 31–32
  split permissions model, 175–177
  USGs supported, 71–72, 157
Active Directory Application Mode (ADAM), 54
Active Directory Federation Services role, 205
Active Directory Lightweight Directory Services. *See* AD LDS
Active Directory Management Gateway Service, 126
Active Directory Module for PowerShell, 127
Active Directory objects, 185–186
Active Directory Rights Management Services. *See* AD RMS
Active Directory schemas. *See* schemas
Active Directory Topology Diagrammer, 808
Active Directory Topology service, 61, 805
Active Directory Users and Computers
  backward pointers, 108
  bulk updates and, 118
  changing group properties, 244
  EMC and, 186
  recipient properties, 185
Active Manager
  about, 426, 432–435
  ACLL and, 439–440
  activation blocks, 499
  automatic database transitions, 435–436
  best copy selection, 437–438
  DACP bit, 511
  mounting mailbox databases, 379
ActiveSync
  about, 555, 618–620
  affinity and, 696
  Blackberry mobile devices, 636–637
  blocking mobile devices, 626–632
  CAS support, 652, 658
  certificate management and, 208
  client support, 556
  client throttling and, 638
  creating policies, 171
  debugging, 635
  disabling access, 573
  ECP support, 602
  EMS restrictions, 187

generating reports, 622–623
mobile devices and, 555
reporting synchronized devices, 623–626
setting policies, 620–622
testing mobile connectivity, 636
wiping lost devices, 632–634
AD. *See* Active Directory (AD)
AD LDS (Active Directory Lightweight Directory Services)
edge servers and, 54, 909
ESE support, 922
mail-enabled recipients, 918
safelist aggregation and, 961
viewing data synchronized into, 916
AD RMS (Active Directory Rights Management Services)
content protection and, 1072–1073
discovery searches, 1036, 1041
enhancements, 5
installing, 1073–1076
IRM support, 1071
message classifications and, 1068
OWA support, 578, 611
transport rules and, 1107–1109
AD RMS Super Users group, 1042
ADAccess, 25–28
ADAM (Active Directory Application Mode), 54
ADAM_MSExchange service, 62
Add Mailbox Database Copy Wizard, 477
Add New Database Copy Wizard, 477
Add-ADPermission cmdlet, 306, 930
Add-AttachmentFilterEntry cmdlet, 953
Add-DatabaseAvailabilityGroupServer cmdlet, 463
Add-DistributionGroupMember cmdlet, 93, 250
Add-IPBlockListEntry cmdlet, 932
Add-IPBlockListProvider cmdlet, 933
Add-MailboxDatabaseCopy cmdlet, 477, 482, 486, 501
Add-MailboxPermission cmdlet, 274, 308, 349
Address Book service
about, 61, 654
affinity and, 696
CAS support, 652
directory connections, 700
address lists, 602
Address Rewrite agent, 955–957
address rewriting, 955–957
ADDriver (Exchange Authorization Library), 83
Add-RoleGroupMember cmdlet, 160
Add-WindowsFeature cmdlet, 43–44
Admin Audit Log agent, 1050, 1052
AdminAuditLogs folder, 1052
Administrator account
auditing mailbox access, 1057
configuring IMAP4 clients, 615
determining role groups, 167
discovery mailboxes, 302
mail-enabled, 236
Organization Management role and, 179
administrator options (ECP), 227–236
$ADminSessionADSettings global variable, 128
ADN (Application Delivery Network), 693–694
ADSI utility, 118
ADSIEdit
about, 22
cautions using, 23
displaying group membership, 313
safelist collection attributes, 962
testing system messages, 877
troubleshooting support, 35, 53
viewing ThumbnailPhoto attribute, 111
Advanced Query Syntax (AQS) format, 1038
affinity in load balancing, 696–697
agent logs, 957–960
aliases, mailbox, 261
All Rooms address list, 344
Allow Backups to Overwrite flag, 543
analysis framework, 1158–1159
–and operator, 113
Anonymous account, 615
AnonymousUsers permission group, 832
anti-spam agents
about, 813, 923
address rewriting, 955–957
agent logs, 957–960
attachment filtering, 953–955
backscattering, 935–936
best practices and, 924
connection filtering, 925, 931–934
content filtering, 926, 946–953
header firewalls, 824–825
installing, 924–925
order of processing, 925–926
recipient filtering, 926, 939–940
safelist aggregation, 961–963
scripts reporting activity, 959
sender filtering, 925, 934–935
sender ID filtering, 926, 936, 940–945
sender reputation, 936–938
tarpitting and, 940
x-headers, 926–929
Anti-spam update service, 61
antivirus products, 907, 964–965
APIPA (Automatic Private IP Addressing), 460
APIs (application programming interfaces)
about, 647–648
CAS installation priority, 656
EWS considerations, 648–650
Apple iPhones, 630
Application and Services Logs, 514
Application Delivery Network (ADN), 693–694
Application Event Log
back pressure conditions, 857

backups and, 534–535
database limit message, 361
diagnostic information, 253
ecTooManyMountedDatabases error, 498
lingering move requests, 746
named properties, 403
page-level corruption, 450
problem mailboxes, 392
protection against excessive growth, 396
tracking background maintenance, 390
Application Host configuration file, 87–88
application programming interfaces (APIs)
about, 647–648
CAS installation priority, 656
EWS considerations, 648–650
ApplicationImpersonation role, 164
AQS (Advanced Query Syntax) format, 1038
arbitration mailboxes, 299, 339, 504
archives. *See* personal archives
Assign Services to Certificate Wizard, 211
assignment policies. *See* role assignment policies
assignments. *See* role assignments
asterisk (*), 152
asynchronous mailbox moves, 711–712
Attach Files dialog box, 596
Attachment Filter agent, 953–955
attachment filtering, 953–955
attachments
compression and, 371
filtering, 953–955
handling, 595
processing, 608
scanning, 1105–1107
attempt copy last logs (ACLL), 439–440, 495
attribute synchronization, 728
audio codecs, 642
audit mailboxes, 1052
Audit subfolder, 1028
auditing
administrator actions, 973, 1049–1057
best practices, 1058
disabling, 1050
mailboxes, 256, 312, 1059–1061
Augmented Backus-Naur Form (ABNF), 1118
authentication
IMAP4 clients, 612, 615
KerbAuth module, 87
MaxConcurrentAPI setting and, 702
Outlook clients, 691
POP3 clients, 612
protected groups and, 320
receive connectors, 840
send connectors, 849
wiping lost devices and, 634
authoritative domains, 887
authorization, RBAC and, 316
AutoArchive feature (Outlook), 976
AutoConfiguration option (Outlook), 668–670
AutoDatabaseMountDial setting (mailbox database), 495–499
Autodiscover service
about, 653, 663
affinity and, 697
AutoConfiguration option, 668–670
CAS support, 652, 663
example demonstrating, 649
logging actions, 670–673
MailTips and, 765
Remote Connectivity Analyzer and, 673
Service Connection Point, 663–666
Set-OutlookProvider cmdlet and, 656
settings overview, 666–667
site scope, 668
SRV record pointers, 675–676
static configuration, 673–674
WCF and, 475
Automatic Picture Download setting (Outlook), 971
Automatic Private IP Addressing (APIPA), 460
auto-reply (email messages), 274, 290, 889
autosignatures, 1095

B

B2B mode (setup program), 47
Background Intelligent Transfer Service (BITS), 783
background maintenance
about, 370–371
content maintenance, 388–390
identifying tasks, 570
index-aging table and, 388
scheduling, 387–388
backscattering, 935–936
Backup Exec, 532
backups and restores
Allow Backups to Overwrite flag, 543
complete server backups, 552
database management, 360, 394
database size and, 360
deleted item recovery, 539
deployment preparations, 14
dial-tone database recovery operation, 540
Last Good Backup date, 536
questioning need for, 528–530
recovery databases, 538–551
testing, 537
Windows Server Backup, 530–538
bad item limit, 727, 745
basic authentication, 612
BATV (Bounce Address Tag Verification), 936
BCS (best copy selection), 437–438

BES (BlackBerry Enterprise Server)
ActiveSync support, 636–637
client throttling and, 639
conflict resolution, 567
connection considerations, 556
ExPerfWiz limitations, 1165
permissions, 381
best copy selection (BCS), 437–438
beta versions, testing, 6
BITS (Background Intelligent Transfer Service), 783
BlackBerry mobile devices. *See also* BES (BlackBerry Enterprise Server)
ActiveSync support, 636–637
conflict resolution, 567
connection considerations, 556
ExPerfWiz limitations, 1165
IRM support, 1071
junk mail filter, 969
permissions, 381
block replication, 446–447
Blocked Senders list, 971
bootstrap replication account, 913
bounce action, 1112
Bounce Address Tag Verification (BATV), 936
BPOS (Business Productivity Online Suite)
about, 8
ECP support, 214
internal use, 816
Online service, 977
personal archives and, 981
remote PowerShell support, 80
braces, 113
build numbers, 571
bulk updates, 118–122, 198
Business Productivity Online Suite. *See* BPOS (Business Productivity Online Suite)

C

CA (certification authority), 208, 210
cached Exchange mode, 557, 560
caching
configuration data, 28
group expansion results, 825–826
GUID, 825–826
mailbox permissions, 309
transport rules, 1110–1111
Calendar Assistant, 276, 570, 793
Calendar Repair Assistant (CRA), 794–795, 798–799
calendars
accessing, 349
anonymous access, 592
creating retention policy tags for, 992
mailbox settings, 276
OWA support, 602
restricted, 593
sharing, 588–590
sharing with Internet users, 590–594
callback verification, 935
CALs (Client Access Licenses)
ActiveSync support, 620
archiving solutions and, 976
deployment preparation, 7, 12, 19
determining number needed, 19, 203, 256
determining requirements for, 201
discovery searches and, 1034
FPE support, 909, 964
journaling support, 1116
Outlook and, 557
troubleshooting, 262
Capacity Calculator (Microsoft), 364
capacity planning
assumptions, 364
maximum database size and, 359–363
CAS (Client Access Server)
about, 652–656
additional information, 651
antivirus products and, 964
architectural overview, 657
assigning static ports, 698
Autodiscover service, 663–676
backup considerations, 552
certificate management and, 208, 688–690
client throttling and, 639
connecting to, 89
cross-site connections and, 513–514, 679–680
decommissioning servers and, 52
deployment preparations, 17
directory information access, 662
ECP delivery reports and, 227
edge servers and, 54
enhancements, 2
file transfer and, 784
firewall considerations, 661
IMAP4 support, 612
installation priority, 655–656
installation requirements, 46, 269
interoperability, 703–704
language settings, 271–272
linking to mailbox databases, 659–660
load balancing considerations, 692–702
MAPI endpoints and, 655
Outlook Anywhere and, 652–654, 691–692
Outlook client support, 661
perimeter networks, 684–685
personal archives and, 981
polling interval, 560
POP3 support, 612
preparing for transition, 703–704
server recovery and, 552
Test-ECPConnectivity cmdlet and, 143

Test-MRSHealth cmdlet and, 144
Test-OWAConnectivity cmdlet and, 143
upgrading servers, 509
CAS arrays
about, 676–677
creating, 678–679
cross-site connections and, 679–680
deployment considerations, 677
load balancing and, 681–682, 693–695
upgrading servers, 682–684
Cascading Style Sheets (CSS), 597
case sensitivity
comparison operators, 113
date/time formats, 272
CCR (cluster continuous replication)
about, 425
database considerations, 360, 451
deployment preparations, 14
system shutdowns and, 510
Test-ReplicationHealth cmdlet and, 141
transport dumpsters and, 894
cd command, 124
CDOEX API, 648
CDOEXM utility, 118
CEIP (Customer Experience Improvement Program), 58–60
CEIPEnable registry value, 60
certificate management
ActiveSync, 208
CAS, 208, 688–690
edge servers, 208
EMC, 208–212
EWS, 208
hub transport, 208
IMAP4, 208
mailboxes, 208
OWA, 208
POP3, 208
security and, 208–209
SMTP, 208
SSL, 208
TLS, 208
Unified Messaging, 208
certification authority (CA), 208, 210
CheckDatabaseRedundancy.ps1 script, 521–522
checkpoint files
about, 375, 417
mounting databases and, 546
restoring, 542
checksum validation
background maintenance and, 384–386
transaction logs, 417
circular logging
about, 419–421, 517
best practices, 475
DAG and, 381, 475–476
enabling, 381–382
removing database copies and, 502
restored databases and, 544
RetainDeletedItemsUntilBackup property, 394
transaction logs and, 448
transport queues, 859
Clean-MailboxDatabase cmdlet, 287
Clear-ActiveSyncDevice cmdlet, 632
Client Access Licenses. ***See*** **CALs (Client Access Licenses)**
Client Access Server. ***See*** **CAS (Client Access Server)**
client throttling, 637–641
clients. ***See also*** **CAS (Client Access Server)**
accessing message classifications, 1067–1070
blocking mailbox connections, 570–573
blocking mailbox server access, 573–574
CAS support, 661
deployment preparations, 13
EWS considerations, 648–650
listing connections, 569–570
MailTips interaction, 765
message hygiene, 965–971
types supported, 555–556
Unified Messaging, 641–647
client-side processing
ActiveSync mailbox policies, 621
configuring IMAP4 clients, 615–617
filter support, 114–117
inbox rules, 221
cluster continuous replication (CCR)
about, 425
database considerations, 360, 451
deployment preparations, 14
system shutdowns and, 510
Test-ReplicationHealth cmdlet and, 141
transport dumpsters and, 894
cluster network object (CNO), 465
clustering. ***See*** **Windows Failover Clustering**
cmdlets
about, 75, 78
accessing, 78, 84
auditing considerations, 1051
CEIP data gathering, 59
edge servers and, 910
EMS support, 93–96
Exchange 2010 SP1, 165
Exchange support, 78
–Filter parameter, 114–117
help available for, 94–96
–Identity parameter, 106–108
–IgnoreDefaultScope parameter, 128
moving mailboxes, 710
piping output, 96, 104, 109–111, 963
–ResultSize parameter, 113–114
roles and, 151
script recommendations, 94

syntax changes, 86
testing, 139–146
–Verbose parameter, 117, 136, 139
writing restrictions, 153
CNAME records, 675
CNO (cluster network object), 465
Collaboration Data Objects, 648
CollectOverMetrics.ps1 script, 522
CollectReplicationMetrics.ps1 script, 523
command line
ISE versus, 122–126
setup program version, 47
comparison operators, 113
compliance
about, 973–974
ActiveSync status report, 623
auditing administrator actions, 1049–1057
auditing mailbox access, 1057–1064
deployment preparations, 14
discovery searches, 1033–1049
dumpster and, 973, 975, 1025–1033
litigation hold and, 973, 1022–1024, 1031, 1033
message classifications, 1064–1070
messaging records management and, 989–1018
Outlook protection rule cmdlets, 1080–1081
personal archives, 976–987
protecting content, 1070–1080
retention hold and, 973, 1021
compression
DAG considerations, 470
database size and, 371
LZX, 775
mail attachments and, 371
transaction logs, 445–446
conditional statements, 105, 235
configuration data/files
Application Host, 87–88
backup considerations, 552
caching, 28
cautions editing, 824
edge servers, 823, 923
editing with Notepad, 596
EMC settings, 192, 201
handling attachments, 595
limits on user mailboxes, 822–823
move reports, 740
MRS settings, 708
OWA considerations, 583–584, 595
RCA service, 685
restoring, 57
routing tables, 826–829
transport system, 816–830
conflict resolution
Outlook and, 567–568
public folder replication, 390
Conflicts folder, 567–568
Connect Mailbox Wizard, 288
ConnectFunctions.ps1 script, 86
Connection Filter agent, 931–934, 947, 957
connection filtering, 925, 931–934
connectivity logging
controlling, 881–882
interpreting, 883–884
Connect-Mailbox cmdlet, 289
connectors
auditing configuration changes, 1057
delivery, 893–894
EDK and, 804
foreign, 893–894
legacy gateway, 829
linked, 853–854
receive, 811, 831–834, 855, 911
send, 828, 841–852, 855
consistency check, 534
consistent databases, 411
contacts
distribution groups and, 313
importing photos, 111
mail-enabled, 200, 341–342, 1121
naming conventions, 200
OWA support, 602
Recipient Management role and, 227
Content Filter agent
agent logs, 957
content filtering, 946–953
FPE and, 909
junk mail filter and, 971
managing with EMS, 951–952
safelist aggregation and, 961
Content Filter Properties dialog box, 947
content filtering
about, 926, 946–947
managing Content Filter agent, 951–952
SCL thresholds, 948–950
SCL values in transport rules, 953
content index catalogs, 1035
content protection
about, 1070–1071
AD RMS and, 1072–1080
Windows Mobile and, 1079
contiguity, maintaining, 370–371
continuous replication, 426–428, 462, 524
Conversation Action Settings folder, 565
conversation views
about, 563–567
ActiveSync support, 618
OWA support, 577
ConvertTo-HTML cmdlet, 131
ConvertTo-MessageLatency.ps1 script, 1151
ConvertTo-SecureString cmdlet, 263

corporate disclaimers, 1091–1097
cost-benefit analysis, 637
CRA (Calendar Repair Assistant), 794–795, 798–799
Credential service, 914
crimson events, 514
cross-forest implementation, 23, 728
cross-site connections, 513–514, 679–680
CSS (Cascading Style Sheets), 597
CSV files
ActiveSync reports, 623, 625
agent logs, 957
bulk updates, 118–119, 278
connectivity logs, 883
outputting, 130, 322
protocol logging, 687, 884–886
CSVDE utility, 118
Customer Experience Improvement Program (CEIP), 58–60
customizations, programming and
Active Directory schemas, 34–35
deployment preparations, 14–16

D

DAC (Datacenter Activation Coordination)
about, 510–511
cross-site connections, 513–514
planning for resilience, 511–513
DAG (Database Availability Group)
about, 428–430, 455–462
activating mailbox database copies, 488–492
activation blocks, 499–500
adding database copies, 477–478
adding database copies with EMS, 482–484
AllowCrossSiteRPCClientAccess property, 680
antivirus products and, 964
applying updates to servers, 492–493
backup considerations, 528, 537
building, 462–468
circular logging and, 381, 475–476
database considerations, 378
dealing with failed servers, 493–494
decommissioning servers and, 51
deployment preparations, 14
design considerations, 515–520
Exchange support, 18
handling initial seeding errors, 479
handling storage hangs, 507–508
high availability feature, 12
investigating problems, 468–469
lagged databases, 484–488
managing properties, 469–471
MAPI network, 471–472
message submission changes, 455
monitoring database copies, 480–481
moving database locations, 383, 500–502
moving databases, 495–499
moving mailboxes, 736
naming conventions, 200, 456
network considerations, 471–475
NetworkCompression property, 470
NetworkEncryption property, 470
operating system requirements, 432, 463, 516
path sharing, 454
recovery considerations, 57
removing database copies, 502–506
removing servers, 506–507
renaming databases, 453
reseeding database copies, 481–482
scripts for managing, 520–523
testing replication within, 141
transaction log replication, 440–451
troubleshooting, 284
upgrading servers, 508–509
viewing properties, 456
Windows clustering, 431–432
DAMs (deferred action messages), 221
DAP (Datacenter Activation Protocol), 511
data compression. *See* compression
Data Protection Manager (DPM), 485, 532, 537
data replication. *See* replication
Database Availability Group. *See* DAG (Database Availability Group)
Database Availability Groups role, 172
database copies
activating, 488–492, 496, 499
active server load and, 538
adding to DAGs, 477–478
adding with EMS, 482–484
backups and, 527–530, 533, 537–538
lagged copies as disaster recovery, 488
monitoring, 480–481
public folder databases, 533
removing, 502–506
reseeding, 481–482
switching over to, 538
database management
about, 374–375
background maintenance, 370–371
backups/permanent removal, 394
content maintenance tasks, 388–390
controlling named properties, 401–403
corrupt item detection/isolation, 391–394
creating mailbox databases, 377–381
ISINTEG considerations, 398–401
protection against excessive growth, 396–397
protection against high latency, 395
scheduling background maintenance, 387–388
Store Driver fault isolation, 397–398
tracking background maintenance, 390–391
updating mailbox databases after installation, 381–383
database mobility, 427

database portability, 427
database scopes, 174–175
Database Size Limit registry value, 362
databases. *See also* mailbox databases
accommodating future growth, 283
allocating mailboxes to, 266–269
Allow Backups to Overwrite flag, 543
assigning GUIDs, 454
best practices, 378
circular logging, 475–476
consistent and inconsistent, 411
decommissioning servers and, 51
defragmenting, 404–407
dismounting versus failovers, 436
EMC restrictions, 187
encrypting, 401
estimating size, 283
filtering, 175, 375
IsExcludedFromProvisioning flag, 266
IsSuspendedFromProvisioning flag, 266
lagged, 484–488
logical corruption, 485
mailbox schemas, 372–374
maintaining contiguity, 370–371
maximum size, 359–363, 516
mounting, 439, 496, 498
moving locations, 383, 454, 500–502
moving mailboxes between, 155, 709–746
naming conventions, 200
page-level corruption, 450
problems moving, 495–499
reassigning from servers, 493
rebuilding, 407
recovery, 538–551
renaming, 382, 452
replicated, 453
reporting size via email, 134–135
restore considerations, 360, 531, 538–551
seeding, 451, 479
Standard Edition limitations, 361–362
unique names, 451–455
usage statistics, 407–409
white page within, 283
Datacenter Activation Coordination. *See* DAC (Datacenter Activation Coordination)
Datacenter Activation Protocol (DAP), 511
date/time formats, 272
debugging ActiveSync, 635
debugging scripts, 106
deduplication feature, 1043–1045
Default Archive and Retention Policy, 985–987, 994
Default Archive Policy, 985
Default Offline Address Book, 380
default policy tags (DPT)
about, 991
in sample retention policy, 996
naming, 997
retention policies and, 1021
Default Role Assignment Policy
about, 168
ECP considerations, 245
MyDistributionGroups role and, 245, 247
RBAC considerations, 150
updating contact details, 152
user roles listed, 168–169
deferred action messages (DAMs), 221
defragmentation
about, 404–406
database size and, 361
ESEUTIL and, 406–407
delayed fan-out technique, 810
Delegated Setup role, 158
Delegated Setup USG, 71
delegates
auditing mailbox access, 1058
personal archives and, 978
delete action, 1112
Deleted Items folder
as default folder, 365
conversation views and, 564
creating, 269
creating retention tags, 1000–1001
dumpster and, 1025
Ignore feature and, 565
notification messages, 585
personal archives and, 986
Deletions folder, 1027, 1029, 1031, 1033
delivery connectors, 893–894
delivery reports
administrator searches, 228–232
dynamic distribution groups, 233
ECP support, 224–227
protected groups and, 320
searching, 234–235
troubleshooting, 229
delivery status notifications (DSNs), 871–875, 935, 1101
Deployment Assistant, 18
deployments
actions list, 11–12
Active Directory support, 12, 15, 19, 23–24, 31–33
CAS arrays, 677
Deployment Assistant support, 18
multiforest, 23
preparing Exchange 2003, 17
preparing Exchange 2007, 12, 15–17
test plans, 12–13
testing for operational processes, 14
testing for programming and customization, 14–16
Details Templates Editor, 1130–1134
DHCP (Dynamic Host Configuration Protocol), 456, 461

diagnostic information
Outlook performance and, 671
setting levels, 251–253
dial-tone database recovery operation, 540
dirty pages, 413–414, 417
Disable MAPI Clients registry key, 573
Disable-AddressListPaging cmdlet, 299
Disable-JournalRule cmdlet, 1125
Disable-Mailbox cmdlet, 93, 285–286, 987
Disable-OutlookProtectionRule cmdlet, 1081
disaster recovery, lagged copies as, 488
disaster recovery mode (setup program), 47
discard notifications, 896, 898
disclaimers, corporate, 1091–1097
Discover-ExchangeServer command (EMC), 89
discovery actions. *See* legal discovery
discovery mailboxes
about, 299–302
accessing search results, 1040–1042
best practices, 1058
creating, 73, 301–302
default quota, 301
permissions, 302
recommended location, 302
usage considerations, 73
Discovery Management role group
discovery mailboxes and, 300
discovery searches and, 1033, 1044, 1047
Exchange 2010 SP1, 160
linking to tasks, 151
managing mailbox permissions, 307
role assignment information, 167
search support, 302
security principal purpose, 71
discovery searches
about, 1033–1035
accessing results, 1040–1042
annotating, 1046
concurrent, 1049
deduplication feature, 1043–1045
dumpster and, 1027, 1029, 1043
EMS support, 1047–1049
exiting employees and, 1034
for multimailboxes, 1037–1040
litigation hold and, 1022
logging, 1045
personal archives and, 976
retention tags and, 1003
single item recovery and, 1029
unsearchable items, 1035–1037
DiscoverySearchMailbox account, 73
Dismount-Database cmdlet, 412, 476, 551
distinguished name, 333, 453, 617
distribution groups. *See also* dynamic distribution groups
about, 312–313
adding mailboxes, 264
assigning quotas, 281
assigning roles, 164
backward pointers, 313
creating, 242–243, 247–249
custom MailTips, 770
defining default location, 238–242
group expansion, 318–319
group owners, 316–318
hidden, 215, 238
identifying membership, 244–247
importing photos, 111
mail flow settings, 304
maintaining, 249–251
managing with ECP, 237–251
moderated recipients, 334
multimailbox searches, 1037
naming conventions, 200, 238–242, 259
processing large, 825
protected groups, 319–320
Recipient Management role and, 227
resource forests and, 24
room lists, 238, 314–316
self-maintaining groups, 321–322
SMTP addresses, 234
tracking group usage, 324
version numbers, 312
viewing members, 322–323
DIT files, 34
$DN variable, 333
DNS (Domain Name System)
DAG and, 429
NonExistentDomain message, 865
receive connectors, 837
send connectors, 849
SRV records, 675
DNS zone, 675
domain controllers
ADAccess flag, 27–28
determining availability, 26
EMC settings, 192
Exchange restrictions, 33, 463
load balancing and, 702
MaxConcurrentAPI setting, 702
Outlook considerations, 237
partial attribute set, 111
support restrictions, 26
Domain Name System (DNS), 429, 675
domains
accepted, 886–889
authoritative, 887
relay, 887
remote, 889–891
DPM (Data Protection Manager), 485, 532, 537

DPT (default policy tags)
about, 991
in sample retention policy, 996
naming, 997
retention policies and, 1021
drizzle-mode synchronization, 569, 687, 714
DSNs (delivery status notifications), 871–875, 935, 1101
DSProxy service, 654, 662
dumpster
about, 498, 894, 973
defined, 1025
discovery searches and, 1027, 1029, 1043
enhancements, 1027–1028
folders and, 1028
identifying contents, 1031
litigation hold and, 1033
managing parameters, 1032–1033
moving, 712
process overview, 1025–1027
recovering deleted items, 1025
retention period and, 1026
retention states, 1033
single item recovery, 1029–1030, 1033
sizing considerations, 528
soft delete and, 975
dynamic distribution groups
about, 324–325
advantages, 313
conditional properties, 331
creating, 198, 326–329
creating with custom filters, 329–334
delivery reports, 233
determining membership, 264
ECP considerations, 228
enabling moderation, 337
mail flow settings, 304
OPATH filters, 238, 325–326
populating, 313
viewing group members, 323
Dynamic Host Configuration Protocol (DHCP), 456, 461

E

ECP (Exchange Control Panel)
about, 182, 213–214
accessing, 214
accessing personal tags, 1011
administrator options, 227–236
affinity and, 696
application overview, 215–216
architectural overview, 76
basic user options, 216–227
CAS support, 652
delivery reports, 224–227
discovery searches, 1033
displaying objects, 216
enhancements, 2
Exchange 2010 SP1 updates, 215
inbox rules, 220–223
managing groups, 237–251
managing mailbox settings, 236
Mobile Phones option, 632
Outlook limitations, 559
RBAC support, 179, 182, 214, 217
role assignment policy and, 152
role groups and, 157
self-maintaining groups and, 321
setting litigation hold, 1024
setting server diagnostics, 251–253
Test-ECPConnectivity cmdlet and, 143
transport and journal rules, 235, 1087
without Exchange mailbox, 235–236
EDB database file, 358, 479
Edge Rules agent, 957
edge servers. *See also* anti-spam agents
about, 806, 908–910
Address Rewrite agent, 955–957
Attachment Filter agent, 953–955
CAS installation priority, 655
certificate management and, 208
configuration file, 823
defined, 54
FPE considerations, 965
installing, 54
message hygiene and, 908–909
permissions, 833
receive connectors and, 811
remote PowerShell and, 79
shadow redundancy and, 896
synchronization and, 909, 911–922
throttling, 824
TLS support, 930
transport pipeline, 892–893
transport rules, 1088–1089
Edge Subscriptions role, 172
Edgesync service, 61
EdgeTransport.exe config file, 823, 827, 859, 1119
EDK (Exchange Development Kit), 804, 893
EHLO commands, 936
email address policies
about, 290–292
creating, 293–296
creating with custom filters, 297
email policy priority, 292
mask variables, 295
setting priorities, 297–298
virtual list view, 299
Email Address Policies role, 172
email signatures, 603
email systems. *See also* Outlook (Microsoft)
ActiveSync support and, 619

attachment processing, 608
autoreply messages, 274, 290
conversation views, 559, 563–567
database size and, 361
delayed fan-out technique, 810
discovery actions, 974
email address policies, 290–299
EMC support, 185
foreign connectors, 893
inbox rules, 220–223
language packs, 54
legacy policies, 291
remote domains and, 889
reporting data via, 134–135
testing mailflow, 145–146
throttling and, 854–857
voice mail preview, 643
EMC (Exchange Management Console)
about, 181
accepted domains and, 889
accessing data, 190–193
Active Directory objects, 185–186
architectural overview, 76
auto-generated PowerShell commands, 195–196
backward pointers, 108
CEIP opt-in, 59
certificate management, 208–212
changing columns, 194
connection filtering, 932
console overview, 183–184
custom attributes, 35–36
DAG considerations, 456
database management, 375
determining CALs needed, 19
Discover-ExchangeServer command, 89
displaying group membership, 313
email address policies, 290
enhancements/new features, 184–185
Export mailbox option, 164
installing edge servers, 54
language considerations, 47
Last Good Backup date, 536
linked connectors and, 854
mailbox databases, 267
mail-enabled accounts and, 89
mail-enabled contacts and, 200
managing mailbox permissions, 308
managing multiple organizations, 204–205
managing objects, 187–188
Maximum Number Of Objects option, 193
moving mailboxes, 710, 719–721, 730
naming conventions, 199–201
object versions, 69
organizational health data, 201–204
personal archives and, 979
reconnecting mailboxes, 287
refresh option, 191
remote PowerShell support, 79, 185
renaming databases, 452
reporting licenses, 69–70
role groups and, 157
scheduling background maintenance, 387
sharing policies, 205–207
sorting objects, 257
starting, 188–190
troubleshooting, 29, 189
updating archive mailbox name, 985
version numbers, 67
EMEA OU, 173–174
EMS (Exchange Management Shell)
accepted domains and, 889
adding database copies, 482–484
adding recipient photos, 111–112
agent logs, 958
architectural overview, 76
bulk mailbox creation, 278–279
bulk updates, 118–120
client/server filters, 114–117
cmdlets supported, 93–96
command editing, 96–99
controlling background maintenance, 386
creating OWA policies, 604
creating reports in HTML, 131
creating retention tags, 998
DAG considerations, 456, 460
discovery search support, 1047–1049
email address policies, 290
execution policies, 137–138
exporting mailbox data, 758–760
finding disconnected mailboxes, 132
generating transcripts, 117
handling code changes, 120–122
handling information returned by, 99–100
help support, 94
identities, 106–108
importing mailbox data, 753
installing edge servers, 54
looking for large folders, 129–130
mailbox import/export requests, 747
mail-enabled accounts and, 89
major dependencies, 86–91
managing Content Filter agent, 951–952
mounting recovery databases, 544
moving mailboxes, 710, 723–725
OPATH filters, 113
outputting CSV files, 130
personal archives and, 981
piping output, 96, 104, 109–111
RBAC support, 87, 152
remote PowerShell support, 79, 82

renaming databases, 453
reporting data via email, 134–135
role groups and, 157
selective output, 100–102
sending messages, 132–133
setting language values, 136
setting mailbox quotas, 281
testing cmdlets, 139–146
troubleshooting, 35
using variables, 103–105
–Verbose cmdlet parameter, 136
viewing group members, 322
Enable-AddressListPaging cmdlet, 299
Enable-AntiSpamUpdates cmdlet, 960
Enable-ExchangeCertificate cmdlet, 690
Enable-JournalRule cmdlet, 1125
Enable-Mailbox cmdlet, 264, 981, 983
Enable-OutlookAnywhere cmdlet, 692
Enable-OutlookProtectionRule cmdlet, 1081
Enable-TransportAgent cmdlet, 956
encryption
DAG properties, 469–470
database, 401
journaling and, 1120
Kerberos-based, 79
MAPI considerations, 661
Outlook considerations, 559
S/MIME and, 219
wiping lost devices and, 634
endpoints
CAS, 682
DAG, 470
MAPI, 431, 653–655
NSPI, 662
Enron investigation, 975
Entourage
Autodiscover service, 663
CAS support, 652
client throttling and, 638
deployment preparations, 13
envelope journaling, 1116
–eq operator, 113
equipment mailboxes, 265, 344, 355
ESE (Extensible Storage Engine)
about, 358–359
AD LDS support, 922
backup support, 530, 535
handling storage hangs, 508
I/O enhancements, 366
lost flush detection, 485
monitoring servers, 433
performance enhancements, 360
transaction logs, 409–423
transaction process, 413–417
transferring logs, 440
transport queues and, 859
ESE scanning, 384
ESEUTIL utility
background maintenance, 383
database defragmentation and, 404, 406–407
ISINTEG utility and, 398–401
rebuilding databases and, 407
transaction logs and, 412, 444
validating recovered databases, 543–544
ESMTP MAIL Service, 840
ESMTP verbs, 884, 897, 901, 906
ethical firewalls, 1101
EWS (Exchange Web Services)
affinity and, 696
CAS support, 652–653
certificate management, 208
client considerations, 648–650
client throttling and, 638
deployment preparations, 13, 15
IPv6 and, 475
search catalog, 497
XSO layer, 813
$ExBin variable, 103
ExBPA (Exchange Best Practice Analyzer)
about, 11, 1129
checking system settings, 139
DAG considerations, 473
organizational health data, 201
Performance Troubleshooter Assistant and, 1162
RBAC reports, 178–179
setup program support, 48
Exchange 2000, 40
Exchange 2003
decommissioning routing groups, 900
deployment preparations, 17
EDK-based development, 893
handling link status updates, 900–901
linking to Exchange 2010, 898–901
migrating from, 4–5, 8–9
Exchange 2007
cmdlet support, 78
compliance features, 976
deployment preparations, 12, 15–17, 34
EDK-based development, 893
migrating from, 4–5, 8–9
reporting licenses, 69
upgrading workstations, 11
Exchange 2010
development priorities, 2–3
editions supported, 18–19
EDK-based development, 893
linking Exchange 2003 to, 898–901
services supported, 60–62
version support, 10–11
Exchange 2010 Coexistence Edition, 18

Exchange 2010 Enterprise Edition, 18
Exchange 2010 Gateway Edition, 18
Exchange 2010 SP1
client connections, 570
cmdlets in, 165
cross-site connections, 680
DAG OS requirements, 432
defining group locations, 239
ECP updates, 215
EMC considerations, 204
improvements contained, 65–66
mailbox aliases, 261
OWA enhancements, 575–578
RBAC enhancements, 170–179
removing mailboxes, 286
role assignments in, 160
sharing policies, 206
transport system, 901–906
Exchange 2010 Standard Edition, 18, 361–362
Exchange 2010 Store. *See* Store (Exchange 2010)
Exchange Admins group, 165
Exchange All Hosted Organizations USG, 71
Exchange Authorization Library (ADDriver), 83
Exchange Best Practice Analyzer. *See* ExBPA (Exchange Best Practice Analyzer)
Exchange Certificate Wizard, 210
Exchange Control Panel. *See* ECP (Exchange Control Panel)
Exchange development group, 1170
Exchange Development Kit (EDK), 804, 893
Exchange Install Domain Servers group, 72
Exchange LegacyInterOp USG, 71
Exchange Load Generator 2010 (LoadGen), 1165–1167
Exchange Management Console. *See* EMC (Exchange Management Console)
Exchange Management Shell. *See* EMS (Exchange Management Shell)
Exchange Management tools, 503
Exchange Organization Administrators security group, 32
Exchange Performance Wizard, 1162–1165
Exchange Personal Information property set, 21
Exchange Queue Viewer, 862, 867–868
Exchange Routing Objects, 647
Exchange Search, 46, 1035
Exchange Server Authentication, 849
Exchange Servers role group
Exchange Install Domain Servers group and, 72
managing mailbox permissions, 307
security principal purpose, 71
Exchange Trace Analyzer (ExTRA), 50
Exchange Trusted Subsystem role group
FSW considerations, 458
managing mailbox permissions, 307
scenario example, 74
security principal purpose, 71
split permission model, 175–177
Exchange Users OU, 601
Exchange Web Services. *See* EWS (Exchange Web Services)
Exchange Windows Permissions role group, 71, 175–177
ExchangeInstallPath variable, 90
Exchange.ps1 script, 124, 128
ExchangeSetup.log file, 49–50
ExchangeUsers permission group, 832
exclamation mark (!), 768
execution policies (EMS), 137–138
Exim MTA, 936
$ExInstall variable, 103
ExLogAnalyzer tool, 1159
ExMon utility, 637
ExOLEDB API, 648
ExPerfWiz, 1162–1165
Export-ActiveSyncLog cmdlet, 623
Export-CSV cmdlet, 322
ExportEdgeConfig.ps1 script, 923
exporting data
about, 747–748, 758
EMS support, 758–760
from edge servers, 923
limiting PST access, 760–762
RBAC support, 749–750
Export-Mailbox cmdlet, 77, 164, 747, 749
Export-OutlookClassification.ps1 script, 1069
Export-TransportRuleCollection cmdlet, 923, 1112
$ExScripts variable, 103
Extensible Storage Engine. *See* ESE (Extensible Storage Engine)
ExTRA (Exchange Trace Analyzer), 50

F

failover. *See* Windows Failover Clustering
false positives, 961
$False variable, 103
Favorites folder, 584
fax integration, 647
FederatedEmail account, 73
FIFO (first in, first out) order, 713, 905
File Distribution service, 61, 764, 783
File Share Witness. *See* FSW (File Share Witness)
filter pack (Microsoft), 46
filters/filtering
attachment filtering, 953–955
client/server support, 114–117
connection filtering, 925, 931–934
content filtering, 926, 946–953
custom, 329–334
database, 175, 375
email address policies, 290, 297
for personal archives, 983
junk e-mail filter, 946
LDAP searches and, 618
OPATH, 113, 238, 325–326

OWA support, 602
recipient filtering, 926, 939–940
sender filtering, 925, 934–935
sender ID filtering, 926, 936, 940–945
testing effectiveness, 333
URL, 965
firewalls
CAS considerations, 661
edge servers and, 910, 912
ethical, 1101
moving mailboxes and, 729
remote management and, 460
x-header, 824–825
first in, first out (FIFO) order, 713, 905
Flaherty, Paul, 491
Folder List shortcut, 568
folders. *See also* managed folders; public folders
dumpster and, 1028
I/O considerations, 365
language and, 269–272
large, 129–130
multiple tags for same, 1005
OWA processing, 577
security considerations, 224
setting quotas, 1032
special, 269
ForEach cmdlet, 120
Forefront Identity Manager 2010, 15
Forefront Protection for Exchange 2010 (FPE), 909, 936, 960, 964
Forefront TMG, 665, 691, 694, 965
foreign connectors, 893–894
forests
AD deployment scenarios, 23–24
deployment preparations, 12
edge servers and, 54
setting object scope, 127–128
Format-List cmdlet, 1036
Forms-based Authentication service, 61
FPE (Forefront Protection for Exchange 2010), 909, 936, 960, 964
FQDN (fully qualified domain name)
CAS arrays, 676, 678
hostnames, 617
receive connectors, 837
remote servers and, 873
send connectors, 849
server considerations, 26, 204
FSW (File Share Witness)
changing location, 471
DACP bit, 511
DAG considerations, 457–459, 469
maintaining quorum, 461, 467
PAM and, 431
Full Access permission, 302, 306–311
fully qualified domain name. *See* FQDN (fully qualified domain name)

G

GAC (Global Assembly Cache), 649
GAL (Global Address List)
adding recipient photos, 111–112
group owners and, 316
hidden groups and, 215, 238, 322
hidden mailboxes and, 300
OAB support, 772
selecting groups, 654
sorting objects, 257
user-created groups and, 249
Web browser limitations, 581
gap coalescing technique, 367
gateway routing table (GWART), 900
Get-AcceptedDomain cmdlet, 889, 916
Get-ActiveSyncDeviceAccessRule cmdlet, 629
Get-ActiveSyncDeviceStatistics cmdlet, 623–625, 631
Get-ActiveSyncOrganizationSettings cmdlet, 627
Get-ActiveSyncVirtualDirectory cmdlet, 188
Get-AdminAuditLogConfig cmdlet, 1051
Get-ADSite cmdlet, 808
Get-ADSiteLink cmdlet, 808
Get-AgentLog cmdlet, 958
Get-AntispamFilteringReport.ps1 script, 960
Get-AntispamSCLHistogram.ps1 script, 960
Get-AntispamTopBlockedSenderDomains.ps1 script, 960
Get-AntispamTopBlockedSenderIPs.ps1 script, 960
Get-AntispamTopBlockedSenders.ps1 script, 960
Get-AntispamTopRBLProviders.ps1 script, 960
Get-AntispamTopRecipients.ps1 script, 960
Get-AttachmentFilterEntry cmdlet, 953
Get-AttachmentFilterListConfig cmdlet, 954
Get-CalendarProcessing cmdlet, 273, 350
Get-CASMailbox cmdlet, 116, 572, 625
Get-ClientAccessArray cmdlet, 679
Get-ClientAccessServer cmdlet, 126, 666–667
Get-Command cmdlet, 95
Get-Content cmdlet, 111, 120, 126, 913
Get-Credential cmdlet, 140, 728
Get-DatabaseAvailabilityGroup cmdlet, 433, 461, 471
Get-DatabaseAvailabilityGroupNetwork cmdlet, 472
Get-DeliveryAgentConnector cmdlet, 894
Get-DistributionGroup cmdlet, 115–116, 247
Get-DistributionGroupMember cmdlet, 322, 334
Get-DynamicDistributionGroup cmdlet, 116, 330
Get-ECPVirtualDirectory cmdlet, 216
Get-EmailAddressPolicy cmdlet, 290
Get-EventLogLevel cmdlet, 253
Get-ExBlog cmdlet, 94
Get-ExchangeAdministrator cmdlet, 159
Get-ExchangeServer cmdlet
about, 93, 1157

code example, 126
EMC settings, 192
–Identity parameter, 106
selecting storage locations, 377
version information, 67
Get-ExecutionPolicy cmdlet, 137
Get-FailedContentIndexDocuments cmdlet, 1036
Get-Group cmdlet, 108
Get-Help cmdlet, 93–95
Get-IMAPSettings cmdlet, 613
Get-InboxRule cmdlet, 224
Get-IPAllowListConfig cmdlet, 931
Get-IPBlockListConfig cmdlet, 931
Get-JournalRule cmdlet, 1124
Get-Mailbox cmdlet
about, 93
applying retention policies, 1008
–Arbitration parameter, 299
–BatchName parameter, 725
checking server restrictions, 572
determining assigned retention policies, 1010
displaying retention policy information, 1015, 1017
EMC support, 191
EMS support, 76
enabling all mailboxes, 1032
enabling for auditing, 1060
filter support, 115–116
looking for large folders, 130
managed folders, 982
object version example, 68
personal archives and, 983
piping output, 963
RBAC considerations, 84
–RecipientTypeDetails parameter, 301
recovery databases and, 546
removing database copies, 505
restoring mailbox data, 547
retrieving list of mailboxes, 1053
retrieving mailbox groups, 1020
role group example, 83
setting user limits, 822
testing filter effectiveness, 333
Get-MailboxAutoReplyConfiguration cmdlet, 273–274
Get-MailboxCalendarConfiguration cmdlet, 273, 275, 351
Get-MailboxCalendarFolder cmdlet, 593
Get-MailboxCalendarSettings cmdlet, 351
Get-MailboxDatabase cmdlet
about, 94
adding database copies, 483
assigning OABs, 380
checking current settings, 659
discovering GUID, 362
finding distinguished name, 333
lagged databases, 487
moving database locations, 500
removing database copies, 504, 506
retrieving current server, 301
scheduling background maintenance, 387
standard journaling and, 1122
–Status parameter, 405, 443, 536
Get-MailboxDatabaseCopyStatus cmdlet
applying updates to DAG servers, 493
lagged databases, 487
moving database locations, 501–502
performance considerations, 452
replication activity status, 443, 480
valid database status codes, 481
Get-MailboxExportRequest cmdlet, 750, 759
Get-MailboxExportRequestStatistics cmdlet, 750, 759–760
Get-MailboxFolder cmdlet, 224, 273, 276
Get-MailboxFolderStatistics cmdlet, 85, 109, 130, 1031
Get-MailboxImportRequest cmdlet, 750, 754
Get-MailboxImportRequestStatistics cmdlet, 750, 755
Get-MailboxJunkEmailConfiguration cmdlet, 274, 277
Get-MailboxMessageConfiguration cmdlt, 274, 277
Get-MailboxRegionalConfiguration cmdlet, 272–273
Get-MailboxRestoreRequestStatistics cmdlet, 550
Get-MailboxSearch cmdlet, 1047, 1049
Get-MailboxServer cmdlet
CRA support, 795–796
disconnected mailboxes, 132
listing client connections, 569
message tracking, 1137
script initialization, 126
Get-MailboxSpellingConfiguration cmdlet, 274, 277
Get-MailboxStatistics cmdlet
about, 94, 108, 284
accessing log data, 740
–Archive parameter, 984
EMC support, 191
generating data, 284
identifying mailbox name, 289
–IncludeMoveHistory parameter, 741
–IncludeMoveReport parameter, 741
piping output, 109, 284
quarantined mailboxes, 392
RBAC support, 85
recovery databases and, 546
removing database copies, 505
RPC support, 131
–Server parameter, 284
soft deleted mailboxes, 716
split in item storage, 739
Get-MailContact cmdlet, 115
Get-ManagementRole cmdlet, 151, 217
Get-ManagementRoleAssignment cmdlet
about, 167, 217
piping roles, 153
troubleshooting role groups, 163
viewing role assignments, 160–161, 167

Get-ManagementRoleEntry cmdlet, 151–152
Get-Message cmdlet, 866
Get-MessageClassification cmdlet, 1066–1067
Get-MessageTrackingLog cmdlet, 1142–1145, 1151, 1156–1157
Get-MoveRequest cmdlet
about, 723
–BatchName parameter, 725
examining current moves, 735
releasing suspended requests, 733
–ResultSize UnLimited parameter, 719
Get-MoveRequestStatistics cmdlet
about, 724
–IncludeReport parameter, 740
move request details, 713, 720, 733, 739
orphaned requests, 735
TotalMailboxSize property, 739
viewing StalledDueToHA status, 443
Get-OABVirtualDirectory cmdlet, 188, 783
Get-OfflineAddressBook cmdlet, 778, 789
Get-OrganizationalUnit cmdlet, 185, 198
Get-OrganizationConfig cmdlet
about, 203
accessing group properties, 241
IsAddressListPagingEnabled attribute, 299
updating group naming policy, 241
Get-OutlookProtectionRule cmdlet, 1081
Get-OWAVirtualDirectory cmdlet, 188
Get-POPSettings cmdlet, 613
Get-PublicFolder cmdlet, 389
Get-PublicFolderDatabase cmdlet, 387, 536
Get-Queue cmdlet, 862
Get-ReceiveConnector cmdlet
enabling logging, 885
granting permissions, 840
tarpitting support, 940
viewing default connectors, 831
viewing details, 838, 916
Get-Recipient cmdlet
EMC startup and, 188
–Filter parameter, 116
OAB support, 772
personal archives and, 983
VLV support, 299
Get-RecipientFilterConfig cmdlet, 939
Get-RetentionPolicy cmdlet, 994, 1006
Get-RetentionPolicyTag cmdlet
checking properties, 1001
displaying retention policy information, 1017
–IncludeSystemTags parameter, 994
reviewing retention policy tags, 1003
Get-RMSTemplate cmdlet, 1076
Get-RoleGroup cmdlet, 166
Get-RoutingGroupConnector, 899
Get-RPCClientAccess cmdlet, 661
Get-SendConnector cmdlet, 824, 850, 915
Get-SenderFilterConfig cmdlet, 935
Get-SenderIDConfig cmdlet, 944
Get-SenderReputationConfig cmdlet, 938
Get-StoreUsageStatistics cmdlet, 407–409
Get-SystemMessage cmdlet, 876
Get-ThrottlingPolicy cmdlet, 638–639
Get-TransportAgent cmdlet, 1114
Get-TransportConfig cmdlet, 94, 816, 1116
Get-TransportPipeline cmdlet, 891, 924
Get-TransportRule cmdlet, 1086, 1090
Get-TransportRuleAction cmdlet, 1112–1113
Get-TransportServer cmdlet
configuration settings, 827
connectivity logging, 882
message tracking, 1137
protocol logging and, 886
script initialization example, 126
send connectors, 841
validating edge synchronization, 916
Get-UMServer cmdlet, 126
Get-User cmdlet
about, 108
filter support, 115–116
remote PowerShell considerations, 87
Set-Mailbox cmdlet and, 118
viewing updated properties, 196
Get-UserPrincipalNameSuffix cmdlet, 184
Gigabit Ethernet, 468, 472
Global Address List. ***See*** **GAL (Global Address List)**
Global Assembly Cache (GAC), 649
global catalogs
ADAccess flag, 27
determining availability, 26
planning for, 29–31
publishing photos, 111
globally unique identifier. ***See*** **GUID (globally unique identifier)**
Gmail.com, 942
gold build, 668
Goldman, Dave, 791
Google Calendar, 590
Google Desktop for Windows, 1165
Google.com, 942, 945
GPO (Group Policy Object), 661
Group Metrics service, 763
Group Policy Object (GPO), 661
Group-Object cmdlet, 94
groups. ***See*** **distribution groups**
GSM 06.10 audio codec, 642
GSM G.711 audio codec, 642
–gt operator, 113
GuessSmart process, 668
Guest account, 615
GUID (globally unique identifier)

assigning, 454
caching, 825–826
discovering, 362
mailbox identity and, 107–108, 289
personal archives and, 984
Guillet, Jeff, 630
GWART (gateway routing table), 900

H

HAB (hierarchical address book), 791–793
hashing, safelist collections and, 962
HeaderFilterAgent transport agent, 403
Help Desk role group
ActiveSync policies, 171–172
linking to tasks, 150
non-mail-enabled accounts, 236
permissions, 78
Herding Cats theme, 576
hierarchical address book (HAB), 791–793
Hierarchical Storage Management (HSM), 813
high availability feature. *See* DAG (Database Availability Group)
high latency. *See* latency
HighAvailability channel, 514
$home variable, 103
hosted environments, outsourcing to, 709
hostnames
FQDN considerations, 617
minimizing number of, 690
hot fixes
applying, 49
auditing, 1057
Hotmail email service, 668
HP Information Archive, 975
HSM (Hierarchical Storage Management), 813
HTML format
creating reports in, 131
data compression, 371
MailTips support, 771–772
transport rule disclaimers, 1093
HTTP protocol
ActiveSync status report, 623
Outlook client connections, 699
RPC support, 559, 561–562, 570
HTTPS protocol, 591
hub transport servers
about, 801–802
accepted domains, 886–889
architectural overview, 802–814
back pressure, 857–858
backup considerations, 552
CAS installation priority, 655–656
certificate management and, 208
configuration settings, 816–830
customizable system messages, 871–880
delivery connectors, 893–894
dynamic distribution groups and, 325
ECP considerations, 235
email address policies, 290
Exchange 2010 SP1, 901–906
foreign connectors, 893–894
group expansion and, 319
installing anti-spam agents, 924–925
linked connectors, 853–854
linking Exchange 2003/2010, 898–901
logging, 880–886
mailbox databases and, 269
mailbox servers and, 858
message hygiene and, 907
message submission and, 455
message tracking and, 1140
moderated requests, 338
receive connectors, 831–834
remote domains, 889–891
role of, 811–813
send connectors, 841–852
shadow redundancy, 894–898
site considerations, 809
Store Driver and, 397
throttling, 854–857
transport pipeline, 891–893
transport queues, 859–871
transport rules, 1088–1089
upgrading, 1085
Hygiene Management role group, 71, 160
hyphen (-), 105, 113

I

I Love You virus, 907
IBM, 942
iCalendar protocol, 590, 893
ICS (Incremental Change Synchronization), 714
ICS (Internet Calendar Sharing), 594
identities, 106–108
Identity Lifecycle Management (ILM), 15
–Identity parameter (cmdlets), 106–108
IF/ELSEIF conditional statements, 105
IFilter (Adobe), 46, 1035, 1106
Ignoredlogs directory, 445
IIS (Internet Information Services)
deployment preparations, 15
EMS dependencies, 87
installation requirements, 44
MailTips considerations, 771
Outlook protection rules, 1081
RBAC support, 83
IIS Manager, 87
–ilike operator, 113
ILM (Identity Lifecycle Management), 15
IM (Instant Messaging), 604

IMAIL component, 402
IMAP4 (Internet Messaging Access Protocol 4)
about, 61
affinity and, 697
CAS support, 652–653, 658
certificate management, 208
client support, 555, 610–617
client throttling and, 638, 640
configuring client access, 615–617
configuring servers, 612–615
deployment preparations, 13
disabling access, 573
dumpster and, 1025
testing connectivity, 144
IMF (Intelligent Message Agent), 946
Import-CSV cmdlet, 278
ImportEdgeConfig.ps1 script, 923
Import-ExchangeCertificate cmdlet, 690
importing data
about, 747–748
EMS support, 753
from edge servers, 923
into personal archives, 758
multiple concurrent imports, 753
planning considerations, 750–757
RBAC support, 749–750
retrieving job information, 754–757
importing photos, 111–112
Import-Mailbox cmdlet
about, 749
PST files and, 273
replacement cmdlet for, 77, 164, 747
Import-Module ServerManager cmdlet, 44
Import-RecipientDataProperty cmdlet, 111
Import-TransportRuleCollection cmdlet, 923, 1112
Inbox folder
creating retention tags, 1000–1001
languages and, 269
opening from another mailbox, 310
storing text messages, 619
views and, 365
inconsistent databases, 411
Incremental Change Synchronization (ICS), 714
incremental resynchronization, 449–450
index-aging table, 388
Information Rights Management (IRM), 1041, 1071, 1073
Information Store
about, 61, 358
backups and, 535, 538
caching permissions, 309
handling storage hangs, 508
MailTips and, 763
monitoring servers, 433
multithreading support, 391
transaction logs and, 411, 440, 444
inheritance
mailbox settings, 274, 280
roles and, 156
initialization scripts, 82, 86, 92, 125–126
–inotlike operator, 113
Install-AntiSpamAgents.ps1 script, 924
installation
Customer Experience Improvement Program and, 58–60
delegating setup activity, 57–58
edge servers, 54
language packs, 47, 54–55
Microsoft filter pack, 46
prerequisite system components, 42–46
process overview, 20–21, 39–40
progress watermark, 49
recovering failed servers, 55–57
running setup program, 46–49
setup program log files, 49–50
time requirements, 48
uninstalling Exchange, 51–52
installation mode (setup program), 47
Instant Messaging (IM), 604
Integrated Scripting Environment (ISE), 122–126
Integrated Windows Authentication, 612, 615
Intelligent Message Agent (IMF), 946
Internet Calendar Sharing (ICS), 594
Internet Information Services. *See* IIS
Internet Messaging Access Protocol 4. *See* IMAP4
Internet Security and Acceleration (ISA), 473
Internet service provider (ISP), 669
I/O operations
about, 364–365, 517
Exchange 2007 characteristics, 368
Exchange 2010 characteristics, 366–368
gap coalescing technique, 367
maintaining contiguity, 370–371
Store schema and, 365
transaction logs, 418
IP addresses
as source of spam, 932
DAG considerations, 456, 460, 470, 474
receive connectors, 837
SRL list and, 937
IP Allow list, 925, 932
IP Block list, 925, 932
iPhones, 630
IPsec, 440, 849
IPv4, 460, 475
IPv6, 460, 475
IRM (Information Rights Management), 1041, 1071, 1073
ISA (Internet Security and Acceleration), 473
ISE (Integrated Scripting Environment), 122–126
ISINTEG utility, 383, 398–401
ISP (Internet service provider), 669

J

Jet database engine, 358
journal reports, 1116–1120
journal rules
  about, 1122–1123
  creating, 1123–1125
journaling
  about, 975, 1114–1115
  alternate recipients, 1120
  assessing load, 1125
  controlling named properties, 403
  ECP considerations, 235, 602
  envelope, 1116
  interorganization, 1127
  items placed in managed folders, 1018
  message-only, 1116
  messages, 339
  options supported, 1116
  process overview, 1115
  securing mailboxes, 1126
  standard, 1121–1122
junk e-mail filter, 946, 966–971
Junk E-Mail folder
  about, 968
  creating retention tags, 1000
  PCL considerations, 928
  safelist collections and, 961, 963
Junk E-Mail Options Assistant, 793, 798, 961, 963

K

KerbAuth module, 87
Kerberos-based encryption, 79

L

lagged databases, 484–488
language support
  content filtering and, 947
  EMC, 185
  HABs and, 793
  installing language packs, 47, 54–55
  mailboxes, 264, 269–272
  MailTips, 771
  message classifications, 1067
  retention policies, 1016
  setting language values, 136
  switching languages, 271
  voice mail preview, 643–644
Last Good Backup date, 536
latency
  DAG considerations, 473
  mail notifications and, 562
  protecting against, 395
  Test-ECPConnectivity cmdlet and, 143
  tracking message, 1151–1153
LCR (local continuous replication), 14, 425, 655
LDAP (Lightweight Directory Access Protocol)
  ADAccess flag, 27
  configuring IMAP4 client access, 616, 618
  edge synchronization, 914
  email address policies, 291, 297
  managing groups, 237
  moving mailboxes, 729
  OPATH and, 326
  split connections, 662
LDF files, 34, 41
LDIFDE utility, 118
LDP utility, 918
least-cost routing, 807–808
legacy systems
  duplicate mailbox database names, 455
  email policies, 291
  foreign connectors, 893
  moving mailboxes, 713
legal discovery
  about, 974–976
  for email systems, 974
  PST limitations, 978
Lehman Brothers, Inc., 975
Lightweight Directory Access Protocol. *See* LDAP
–like operator, 113
linked connectors, 853–854, 907
linked mailboxes, 260
linked value replication (LVR), 313
Linux environment, 924
litigation hold
  defined, 973
  dumpster and, 1033
  keeping items indefinitely, 1031
  putting mailbox on, 1022–1024
  user considerations, 1023
LLR (lost log resilience), 449
load balancing
  affinity in, 696–697
  CAS arrays and, 681–682, 693–695
  CAS servers and, 692–695
  domain controllers and, 702
  Exchange 2010 SP1, 901–902
  hardware-based solutions, 694
  moving mailboxes, 709
  software-based solutions, 694
  SSL offloading, 702
  Web services and, 701
Load Generator (LoadGen), 1165–1167
local continuous replication (LCR), 14, 425, 655
Local Failures folder, 568
log files
  agent logs, 957–960
  Application Event Log, 253
  Autodiscover support, 670–673

capturing crimson events, 514
connectivity logging, 881–884
DAG maintenance operations, 468
discovery searches, 1045
EMS command logs, 197–199
interpreting entries, 1142–1150
message tracking, 229, 1139–1142
moving mailboxes, 740–742
protection against excessive growth, 396–397
protocol logging, 685–687, 884–886
setup program, 49–50
transaction logs, 409–423
transport system, 880–886
Log Record Checksum (LRCK), 418
log replication. *See* transaction log replication
Log Search service, 231
log sets, 410–412
logical unit number (LUN), 531
Logoff.aspx file, 599
Logon.aspx file, 599
LogParser utility, 1158
lossy failure, 716
lost devices, wiping, 632–634
lost flush, 485
lost log resilience (LLR), 449
Lotus Domino, 647
Lotus Notes email system, 293, 893
LRCK (Log Record Checksum), 418
–lt operator, 113
LUN (logical unit number), 531
LVR (linked value replication), 313
LZ77 algorithm, 470
LZX compression, 775

M

Mac platform
AD RMS support, 1072
Autodiscover service, 663
CAS support, 652
client throttling and, 638
deployment preparations, 13
OWA considerations, 556
Mail Flow Troubleshooter, 1130
MAIL FROM information, 934, 937, 940, 955
mail notifications
about, 561–562
automatically deleting, 585
OWA support, 603
mail queue database, 859–860, 909
Mail Recipient Creation role, 154
Mail Submission service
about, 61, 805
back pressure and, 858
notification events, 802
Mailbox Assistant service, 61, 570, 794, 878
mailbox database master, 433
mailbox databases
activating copies, 488–492
allocating, 266–269, 362–363
AutoDatabaseMountDial setting, 495–499
background maintenance, 370–371
circular logging and, 420
creating, 377–381
duplicate names, 455
ESE scanning, 384
identifying problem mailboxes, 391
important files, 375
layout considerations, 372–374
linking CAS, 659–660
mounting, 379, 439
moving to new location, 382
permissions, 381
protection against excessive growth, 396–397
RpcClientAccessServer property, 659–660, 662, 679–680
transaction logs, 409–423, 714
updating after installation, 381–383
mailbox delivery prioritization, 901, 904–906
Mailbox Import Export role/role group, 164, 167, 172
mailbox permissions
Full Access, 306–311
Send As, 303–306
SendOnBehalf, 304–306, 309–310, 1058
mailbox properties
IssueWarningQuota, 279
Mail Flow Setting tab, 303–304
manipulating settings, 273–277
ProhibitSendQuota, 279
ProhibitSendReceiveQuota, 279
QuotaNotificationSchedule, 280
viewing, 192
Mailbox Properties dialog box, 196
mailbox provisioning agents, 266–269
Mailbox Replication Service. *See* MRS (Mailbox Replication Service)
Mailbox Search role, 164
mailbox servers
about, 805
AutoDatabaseMountDial setting, 439
block replication, 446–447
blocking access, 573–574
CAS installation priority, 655
deployment preparations, 17
EMC restrictions, 188
firewall considerations, 661
hub transport and, 858
listing client connections, 569–570
managing DAG memberships, 463
message tracking and, 1138, 1140
NTFS deadlocks, 11
personal archives and, 977

tuning, 654
mailbox signature, 726
MailboxDatabaseFailureItems channel, 514
mailboxes. *See also* distribution groups; moving mailboxes; personal archives
accuracy of space available, 406
ActiveSync policies, 621
aliases, 261
applying retention tags to, 1007–1009
arbitration, 299, 339, 504
archiving solutions, 265, 283, 973
attachment processing, 608
attributes and names, 257–258
auditing, 256, 312, 1052, 1057–1064
autoreply settings, 274, 290
blocking client connections, 570–573
bulk creation, 277–279
certificate management and, 208
completing setup, 264–265
concurrently open, 558
contents of, 284
counting, 203
creating, 259–284
custom MailTips, 770
customizing retention policies for, 1010
decommissioning servers and, 51
deployment preparations, 14
differentiation for types, 256
disabling, 285–289
disconnected, 132
discovery, 73, 299–302, 1040–1042
dumpster and, 1029
ECP considerations, 235–236
edge servers and, 54
email address policies, 290–299
enhancements, 2
equipment, 265, 344, 355
exporting data, 747–762
identifying problem, 391
importing data, 747–762
importing photos, 111
inbox rules, 220–223
integrating creation with policy, 260
language support, 264, 269–272
linked, 260
litigation hold, 973
mail-enabled contacts, 341–342
maintenance guidelines, 255–256
management role assignment policies, 168
managing settings, 236, 273–277
moderated recipients, 334–341
multiforest deployments and, 24
naming conventions, 200, 257–259
OWA policies, 600–604
passwords and, 261, 263
putting on litigation hold, 1022–1024
putting on retention hold, 1021
quarantined, 392–393
quota considerations, 279–284, 301, 369, 594–595
Recipient Management role and, 227
reconnecting, 286–289
recovery database and, 539
remote PowerShell support, 81
removing, 285–289
removing photo data, 112
removing retention policies, 1009, 1017
restoring, 531, 547–551
scanning in managed folders, 982
SCL thresholds and, 950
search, 504
securing, 1126
setting language values, 136
setting permissions, 303–311
testing mailflow, 145–146
troubleshooting, 261–263, 268, 667
user limits, 822–823
validating names, 310
views and, 365
mail-enabled accounts
Administrator account, 236
distribution groups and, 313
security considerations, 235
support considerations, 89
mail-enabled contacts
about, 341–342
mail flow settings, 304
mail users and, 342–343
moderated requests and, 338
naming conventions, 200
protecting, 341
standard journaling, 1121
MailTips
about, 762–765
CAS support, 653, 693
client interaction, 765
configuring, 766–768
custom, 770–771
IMAP4 limitations, 611
mail users, 343
mailbox naming conventions, 259
mail-enable contacts, 341
moderated mailboxes, 340
multilingual custom, 771
OAB support, 790
Outlook limitations, 559
performance considerations, 765
POP3 limitations, 611
protected groups, 320
restrictions, 771
self-maintaining groups, 321

setting language values, 136
types of, 767
user experience, 768–769
major releases, 571
Majority Node Set clusters, 461
Manage Database Availability Group Wizard, 463, 507
Managed Folder Assistant. *See* MFA (Managed Folder Assistant)
managed folders
about, 973, 975
backward compatibility, 990
journaling about items placed in, 1018
MRM and, 1018
personal archives and, 981
scanning mailboxes in, 982
upgrading from, 1018
management role assignment, 149
management role assignment policies, 149–150, 168–169
management role entries, 149
management role groups, 149, 165
management role scope, 149
management roles, 148, 168, 189
MAPI (Messaging Application Programming Interface)
antivirus products, 907
API considerations, 648
Autodiscover process and, 513
background, 555
BES support, 1165
CAS support, 652, 658
client connections, 570, 572
client throttling and, 638
compression and, 470
controlling named properties, 401–403
DAG considerations, 471
determining endpoint, 431
disabling encryption, 661
ElcMoveDate property, 1018
mailbox database schema, 373
moving mailboxes, 729
preserving mailbox signatures, 726
relocating endpoints, 653–655
RPC Client Access Layer and, 573, 638
Test-MAPIConnectivity cmdlet, 141
MaxNumExchange registry key, 558
Measure-Command cmdlet, 92
meetings
attendee lists, 592
CRA handling discrepancies, 795
forwarding requests, 585–586
Melissa virus, 907
MESO (Microsoft Exchange System Objects) object, 33
message classifications
about, 1064–1065
client access to, 1067–1070
creating, 1065–1066
localized, 1067
transport rules and, 1108–1109
usage considerations, 1065
message hygiene
address rewriting, 955–957
agent logs, 957–960
anti-spam agents and, 923–929
antivirus products, 907, 964–965
attachment filtering, 953–955
backscattering, 935–936
client defense, 965–971
connection filtering, 925, 931–934
content filtering, 926, 946–953
defined, 907
edge servers and, 908–922
header firewalls, 824–825
options available, 907–908
recipient filtering, 926, 939–940
safelist aggregation, 961–963
sender filtering, 925, 934–935
sender ID filtering, 926, 936, 940–945
sender reputation, 936–938
tarpitting and, 940
Message Labs, 907
message moderation, 1104
message tracking
about, 1135–1138
interpreting log entries, 1142–1150
log files generated on servers, 1139–1142
mailbox servers and, 1138
measuring message latency, 1151–1153
options for analyzing logs, 1158–1159
Message Tracking role, 151, 228
Message Tracking tool, 1130
message transfer agents (MTAs), 935
message-only journaling, 1116
messages. *See also* message hygiene
ActiveSync support, 618
autoreply, 274, 290
best practices, 835
common causes for failure, 874–875
content filtering considerations, 948
conversation views, 564
corrupt, 391–394
customizing, 878–880
delivery reports, 320
DSN, 871–875, 935
EMC restrictions, 188
foreign versus delivery connectors, 894
hub transport roles, 811–813
journaling, 339
mail notifications, 561
mailbox delivery prioritization, 901, 904–906
MailTips support, 767
moderated requests, 337–340

moving mailboxes and, 727
moving to delivery queues, 861–862
nested moderation, 336
OOF notices, 763
pickup directory and, 869–870
problem queues and, 865
quota, 878–880
remote domains and, 889
retrieving deleted, 389
sending from EMS, 132–133
setting language values, 136
shadow redundancy and, 895–896, 898
Store Driver fault isolation, 397
submission changes, 455
submission queue, 861
system, 871–880
testing mailflow, 145–146
tracking logs, 229
transaction logs, 409
troubleshooting deleted, 132
messaging administrators, 801–802
Messaging Application Programming Interface. *See* MAPI
messaging records management. *See* MRM (messaging records management)
metadata, 299, 340, 763
MFA (Managed Folder Assistant)
about, 793, 989
automatically applying retention policies, 1014
implementing retention policies, 1018–1021
processing time, 1020
responsibilities, 997
retention policies and, 985, 1008
setting schedules, 798
timeslots and processing, 1019
work cycle definitions, 797
MFCMAPI utility, 221
Microsoft Access, 284, 1159
Microsoft Capacity Calculator, 364
Microsoft Corporation, 907, 942, 965
Microsoft Excel, 284, 957
Microsoft Exchange ADAM service, 62
Microsoft Exchange Credential service, 62
Microsoft Exchange Replication Service. *See* Replication Service
Microsoft Exchange Security Groups OU, 71–72, 157, 160
Microsoft Exchange Server 2010. *See* Exchange 2010
Microsoft Exchange System Objects (MESO) object, 33
Microsoft filter pack, 46
Microsoft Mail, 563
Microsoft Management Console. *See* MMC (Microsoft Management Console)
Microsoft .NET Framework
edge servers and, 54
installation requirements, 40, 42
.NET TCP Port Sharing service, 46
Microsoft Outlook. *See* Outlook
Microsoft Search (MSFTESQL-Exchange), 62
Microsoft System Center
Data Protection Manager, 485, 532, 537
Operations Manager, 139, 843, 1147, 1151, 1159
Microsoft Update, 960
Microsoft Virtual PC, 636
migration from older platforms, 4–5, 8–9. *See also* upgrades
MIME (Multipurpose Internet Mail Extensions), 870, 954, 1118
Mimosa Systems NearPoint, 975
minor releases, 571
MMC (Microsoft Management Console)
ADSIEdit, 22
Queue Viewer, 862, 867–868
mobile devices. *See also* Windows Mobile
ActiveSync considerations, 555, 619, 621
audio codecs, 642
blocking based on type, 626–630
blocking on per-user basis, 631–632
deployment preparations, 13
keyboard limitations, 619
OWA screen design, 577
quarantine for, 629
search catalog, 497
synchronizing, 624
testing connectivity, 636
wiping lost devices, 632–634
moderated recipients
about, 334–336
dynamic distribution groups, 337
mail-enabled contacts, 341
moderated mailboxes, 340–341
moderated requests, 337–340
monitoring
database copies, 480–481
submission queue, 901, 903–904
Monitoring service, 61
Mount-Database cmdlet, 379–380, 476, 545
mounting
databases, 439, 496, 498
mailbox databases, 379
recovery databases, 544–546
Move Database Path Wizard, 382
Move Mailbox Wizard, 710, 717–718
Move Mailboxes role, 155, 160
Move Offline Address Book Wizard, 782
Move-ActiveMailboxDatabase cmdlet
about, 496–497
activating database copies, 496, 499
reassigning databases, 493
–SkipClientExperienceChecks parameter, 497, 513
Move-DatabasePath cmdlet, 383, 501–502
Move-Mailbox cmdlet, 710, 724–725
MoveMailbox.ps1 script, 724

Move-OfflineAddressBook cmdlet, 782
Move-TransportDatabase.ps1 script, 860
moving mailboxes
about, 709–711
accessing log data, 740–742
asynchronously, 711–712
between Exchange versions, 727–729
checking request status, 731
clearing move requests, 722
concurrency considerations, 732
EMC process, 710, 719–721, 730
EMS support, 710, 723–725
handling errors, 744–746
high availability and, 736–738
legacy mailboxes, 713
lingering requests, 746
message arrival and, 727
Move Mailbox Wizard, 710, 717–718
Move Mailboxes role, 155, 160
MRS process, 713–715
orphaned requests, 735
outputting reports to files, 741
performance considerations, 736
planning considerations, 732–736
preserving mailbox signature, 726
preventing data loss, 716–717
provisioning requirements, 743
reporting, 738–742
RTM considerations, 730
troubleshooting, 731
with personal archives, 729–731
MP3 codec, 642
MRM (messaging records management)
about, 989
managed folders and, 1018
retention policies, 990, 995–997, 1004–1017
retention tags, 990–994, 997–1004
system tags, 994
upgrading from managed folders, 1018
voice mail and, 1002
MRS (Mailbox Replication Service). *See also* moving mailboxes
about, 61, 707
configuration file, 708
ICS support, 714
log replication/replay, 442
MaxSendSize setting, 751
moving mailboxes, 710
processing overview, 713–715
restoring mailbox data, 549
Test-MRSHealth cmdlet and, 144
MRSProxy service, 729
MSExchangeAB service. *See* Address Book service
MSExchangeADTopology service, 61
MSExchangeAntiSpamUpdate service, 61
MSExchangeEdgeCredential service, 62
MSExchangeEdgeSync service, 61
MSExchangeFBA service, 61
MSExchangeFDS service, 61, 764, 783
MSExchangeIMAP4 service. *See* IMAP4
MSExchangeIS service. *See* Information Store
MSExchangeMailboxAssistants service, 61, 570
MSExchangeMailboxReplication service. *See* MRS (Mailbox Replication Service)
MSExchangeMailSubmission service. *See* Mail Submission service
MSExchangeMonitoring service, 61
MSExchangePop3 service, 61
MSExchangeProtectedServiceHost service, 61
MSExchangeRepl service. *See* Replication Service
MSExchangeRPC service. *See* RPC Client Access service
MSExchangeSA service, 25, 62
MSExchangeSearch service, 62
MSExchangeServiceHost service, 62
MSExchangeThrottling service, 62
MSExchangeTransport service, 62, 824
MSExchangeTransportLogSearch service, 62
Ms-Exch-Schema-Version-Pt object, 33
MSFTESQL-Exchange service, 62
MTA service, 901
MTAs (message transfer agents), 935
multiforest deployments, 23
Multipurpose Internet Mail Extensions (MIME), 870, 954, 1118
MX records, 846, 919, 942
MyBaseOptions role, 160, 169
MyContactInformation role, 160, 169, 218
MyDistributionGroupMembership role, 169, 245
MyDistributionGroups role, 169, 245, 247, 249
MyProfileInformation role, 169
MyRetentionPolicies policy, 1012
MyTextMessaging role, 169
MyVoiceMail role, 169

N

Name Service Provider interface (NSPI), 662, 699
named properties, controlling, 401–403
namespaces, 655, 886
naming conventions
DAG, 200, 456
databases, 200
distribution groups, 200, 238–242, 259
EMC, 199–201
mailboxes, 200, 257–259
mail-enabled contacts, 200
messaging connectors, 201
public folders, 200
retention tags, 997–998
role assignments, 161
servers, 39

variables, 105
NDRs (nondelivery reports)
back pressure and, 858
CEIP and, 58
customizing, 875–878
DSN messages, 872
example, 402
sender ID, 944
transaction logs and, 410
–ne operator, 113
NetLogon service, 28
NetStat program, 700
network interface cards (NICs), 468, 471–472
Network Load Balancing (NLB), 678, 681–682
networks
DAG, 471–475
perimeter, 684–685
Never Delete managed folder, 1018
New Accepted Domain Wizard, 888
New Address List Wizard, 786–787
New Database Availability Group Wizard, 456
New E-Mail Address Policy Wizard, 293–294
New Exchange Certificate Wizard, 212
New Federation Trust Wizard, 205
New Journal Rule Wizard, 1123
New Mail User Wizard, 342
New Mailbox Database Wizard, 377
New Mailbox Wizard
about, 259–260
applying retention tags, 1007
bulk mailbox creation and, 277
manipulating settings, 261, 263–265
troubleshooting, 29, 261–263
New Offline Address Book Wizard, 788
New Outlook Web App Mailbox Policy Wizard, 601
New Receive Connector Wizard, 835
New Retention Policy Tag Wizard, 992, 998
New Sharing Policy Wizard, 206
New-AcceptedDomain cmdlet, 888, 912
New-ActiveSyncDeviceAccessRule cmdlet, 627
New-ActiveSyncMailboxPolicy cmdlet, 171
New-AddressList cmdlet, 787
New-AddressRewriteEntry cmdlet, 956
New-AutodiscoverVirtualDirectory cmdlet, 702
New-ClientAccessArray cmdlet, 678
New-DatabaseAvailabilityGroup cmdlet, 456, 462, 473
New-DatabaseAvailabilityGroupNetwork cmdlet, 474
New-DistributionGroup cmdlet, 250, 315, 1051
New-DynamicDistributionGroup cmdlet, 329
New-ECPVirtualDirectory cmdlet, 216
New-EdgeSubscription cmdlet, 912–913, 915
New-EmailAddressPolicy cmdlet, 295–297
New-ExchangeCertificate cmdlet, 690
New-InboxRule cmdlet, 223
New-JournalRule cmdlet, 1123–1124
New-Mailbox cmdlet
–Archive parameter, 981
auditing example, 1051
creating extensions for, 135
database scoping, 175
–Discovery parameter, 302
enforcing settings, 263
–Equipment parameter, 265
remote PowerShell, 83
–Room parameter, 265
–TemplateInstance parameter, 273
New-MailboxAuditLogSearch cmdlet, 1061–1062
New-MailboxDatabase cmdlet
automatic provisioning, 267
output example, 136, 379
–Recovery parameter, 544
New-MailboxExportRequest cmdlet
about, 750, 758
as replacement cmdlet, 77, 164
–ExcludeFolders parameter, 759
–IncludeFolders parameter, 759
–IsArchive parameter, 759
search annotation, 1046
–SourceRootFolder parameter, 759
–TargetRootFolder parameter, 759
New-MailboxFolder cmdlet, 273, 276
New-MailboxImportRequest cmdlet
about, 749, 753
as replacement cmdlet, 77, 164
–IsArchive parameter, 758
New-MailboxRepairRequest cmdlet, 399
New-MailboxRestoreRequest cmdlet, 547, 549–551
New-MailboxSearch cmdlet, 1047–1049
New-MailContact cmdlet, 341
New-MailMessage cmdlet, 132
New-MailUser cmdlet, 343
New-ManagementRole cmdlet, 165, 218
New-ManagementRoleAssignment cmdlet
about, 749
assignment policies, 169
ECP user options, 217–218
unscoped roles, 165
New-ManagementScope cmdlet, 156
New-MessageClassification cmdlet, 1065–1066
New-MoveRequest cmdlet
about, 94, 723–724, 733
database scoping and, 175
–DoNotPreserveMappingSignature parameter, 727
moving between Exchange versions, 727
moving discovery mailboxes, 302
MRS support, 719
provisioning requirements, 743
–Suspend parameter, 734
–TargetDatabase parameter, 455
New-OfflineAddressBook cmdlet, 789

New-OrganizationRelationship cmdlet, 205
New-OutlookProtectionRule cmdlet, 1081
New-OWAMailboxPolicy cmdlet, 604
New-PSSession cmdlet, 82
New-PublicFolderDatabaseRepairRequest cmdlet, 399
New-ReceiveConnector cmdlet, 912
New-RetentionPolicy cmdlet, 1006
New-RetentionPolicyTag cmdlet
creating personal tags, 1002
creating retention policy tags, 1000
default tags, 1003
–Type parameter, 990
upgrading from managed folders, 1018
New-RoleAssignmentPolicy cmdlet, 169
New-RoleGroup cmdlet, 159, 163
New-RoutingGroupConnector cmdlet, 899
New-SendConnector cmdlet
creating send connectors, 850, 854
defining address spaces, 847
edge servers and, 912
New-SystemMessage cmdlet, 875, 878
New-ThrottlingPolicy cmdlet, 641
New-TransportRule cmdlet, 1086, 1101, 1103, 1106
New-WebServicesVirtualDirectory cmdlet, 702
NICs (network interface cards), 468, 471–472
NLB (Network Load Balancing), 678, 681–682
node majority model, 466
noncircular logging, 421–422
nondelivery reports. *See* NDRs (nondelivery reports)
NonExistentDomain message, 865
–not operator, 113
Notepad editor, 596, 824
Notes application, 603
–notlike operator, 113
Novell GroupWise email system, 293, 893
NSPI (Name Service Provider interface), 662, 699
NT Authoritynonymous Logon account, 930
NTBACKUP utility, 530
$Null variable, 103, 1009

O

OAB (Offline Address Book)
about, 772
affinity and, 697
assigning to databases, 380–381
customized, 785–790
decommissioning servers and, 52
downloading, 772–776
email address policies, 298
forcing data downloads, 112
generating, 776–781
generating segmented, 786
group metrics and, 764
LZX compression, 775
MailTips support, 790
moving generation server, 782
updating files, 781
Web-based distribution, 783–784
OABInteg tool, 791
object versions, 68–69
objects
ECP display constraints, 216
managing, 187–188
setting scope, 127–128
sorting, 257
OCS (Office Communications Server), 610
Office Communications Server (OCS), 610
Offline Address Book. *See* OAB (Offline Address Book)
OLD. *See* online defragmentation (OLD)
OnConnect transport event, 931
one-way trust, 24
online defragmentation (OLD)
about, 384
database size and, 360
online maintenance and, 384
tracking, 384
Online service (BPOS), 977
OnMailCommand event, 934
OOF (Out of Office) notices, 763, 767, 889
OPATH filters, 113, 238, 325–326
operating systems
DAG requirements, 432, 463, 516
language packs, 55
upgrading workstations, 11
Web browser support, 579–580
opportunistic TLS, 830
–or operator, 113
Organization Management role/role group
accessing cmdlets, 83
ACL support, 158
Administrator account and, 179
delegating roles, 158, 164
ECP support, 214
Exchange 2010 SP1, 160
linking to tasks, 150
managing DAG properties, 469
non-mail-enabled accounts, 236
permissions and, 78, 150
role assignment information, 167
security principal purpose, 72
unscoped roles, 165
organizational health data, 201–204
Organizational Health Gathering Wizard, 201
OST files, 361, 687
Out of Office (OOF) notices, 763, 767, 889
Out-HTML cmdlet, 131
Outlook (Microsoft)
AD RMS support, 1073
AutoArchive feature, 976
AutoConfiguration option, 668–670

Autodiscover service, 663
Automatic Picture Download setting, 971
blocking access to mailbox servers, 573–574
blocking client connections, 570–573
CAS support, 652, 656, 658, 661
Clean Up Conversation option, 566
configuring for cached mode, 562
conflict resolution, 567–568
connection considerations, 555
conversation views, 559, 563–567
delivery reports, 225
depicted, 558
deployment preparations, 13
development background, 557–558
diagnostic information, 671
enhancements, 557
functionality limitations, 559–560
gathering information, 666
group maintenance, 237
Ignore feature, 565
inbox rules, 222
IRM support, 1071
junk email filter, 946, 966–971
listing client connections, 569–570
mail notifications, 561–562
NSPI endpoints, 662
Outlook Anywhere, 562–563, 665–666
OWA and, 574–575, 584
personal archive support, 979
polling mechanism, 559–560
room list groups, 315
room mailboxes, 344
safelist aggregation, 961
synchronizing, 774
TCP connections, 698
upgrade considerations, 8
viewing group members, 322–323
Outlook Anywhere
accessing mailboxes, 667
accessing SCPs, 665–666
affinity and, 696
CAS support, 652–654, 691–692
forcing faster connections, 562–563
Remote Connectivity Analyzer and, 673
split connections, 662
Outlook protection rules, 1071, 1080–1081
Outlook Rules Wizard, 222
outsourcing
message hygiene, 907
to hosted environments, 709
OWA (Outlook Web App)
about, 574–575
AD RMS support, 1073
affinity and, 696
attachment processing, 608
CAS support, 652–653, 659
certificate management and, 208
client support, 556, 569–570
client throttling and, 638
configuration files, 583–584, 595
conflict resolution, 567
conversation views, 566, 577
customizations, 597–600
delivery reports, 225
deployment preparations, 13, 16
deprecated functionality, 578–579
disabling access, 573
ECP support, 214
enhancements, 2
Exchange 2010 SP1, 575–578
Favorites folder and, 584
feature segmentation, 604–610
forwarding meeting requests, 585–586
IRM support, 1071
junk email filter, 946
language settings, 269, 271–272
long signatures, 587
mailbox policies, 600–604, 609–610
mailbox quotas, 594–595
OCS support, 610
opening another user's mailbox, 310–311
Outlook and, 574–575, 584
personal archive support, 979
preserving mailbox signatures, 726
room mailboxes, 344
safelist aggregation, 961
search catalog, 497
sharing calendars, 588–590
sharing calendars with Internet users, 590–594
S/MIME support, 219, 309
Test-OWAConnectivity cmdlet and, 143
themes, 576, 597–600
viewing group members, 322
Web parts, 578, 586
OWA Light, 582
OWAURLs API, 648

P

PAM (Primary Active Manager) role, 431, 433–435, 489
parameters, positional, 106
partial attribute set (PAS), 111–112
PAS (partial attribute set), 111–112
Pasetting.edb database, 937
passwords
mailboxes and, 261, 263
OWA support, 577, 602–603
wiping lost devices and, 634
PBX (private branch exchange), 641
PCL (phishing confidence level), 926, 971
PDC (primary domain controller), 27

performance
affinity and, 696
database size and, 361
EMC and, 185
ESE and, 360
Get-MailboxDatabaseCopyStatus cmdlet, 452
logging diagnostics, 671
mail notifications and, 562
MailTips feature, 765
mounted databases, 498
moving mailboxes, 736
OST/PST files, 361
OWA folder processing, 577
polling interval and, 560
remote PowerShell, 92
voice mail preview and, 643
performance counters, 1159–1162
Performance Monitor
about, 1130, 1159–1162
background maintenance, 384, 386
database defragmentation, 404
low space conditions, 423
tracking online defragmentation, 384
Performance Troubleshooter Assistant, 1130, 1162–1165
perimeter networks, 684–685
Permission Group Properties dialog box, 833
permission groups, 832, 836, 839
permissions
auditing mailbox access, 1058
DAG considerations, 456
delegated, 192
deployment preparations, 14, 21, 32
discovery mailboxes, 302
EMC support, 185
Full Access, 302, 306–311
header firewalls, 824
installation considerations, 47
mailbox databases, 381
personal archives and, 979
RBAC support, 80, 84, 90, 161
room mailboxes, 349
SendAs, 303–306, 1058
SendOnBehalf, 304–306, 309–310, 1058
setting for mailboxes, 303–311
split permissions model, 175–177
personal archives
about, 976–979
checking space usage, 984
database size and, 360, 363
default policy, 985–987
deployment preparations, 14
disabling, 987
enabling, 979–985
enabling properties, 983–984
filtering for, 983
for mailboxes, 265, 283, 973
importing data into, 758
limitations, 978
moving mailboxes into, 729–731
Outlook limitations, 559
target market, 976
transfer limitations, 979
troubleshooting, 988
updating mailbox names, 985
usage considerations, 987
personal tags
about, 991
accessing, 1011
creating, 1002
naming, 997
sample retention policy, 996
phishing confidence level (PCL), 926, 971
phone dialing plans, 34
photos
importing, 111–112
removing data from mailboxes, 112
pickup directory, 869–870
Ping utility, 912
piping output
$_ variable, 104
about, 109–111
cmdlets and, 96, 284, 963
for roles, 153
remote PowerShell and, 120–122
PKI (Public Key Infrastructure), 1070
planning considerations
capacity planning, 359–364
for resilience, 511–513
global catalogs, 29–31
importing PST data, 750–757
moving mailboxes, 732–736
poison queues, 864
polling mechanism, 559–560
POP3 (Post Office Protocol 3)
affinity and, 697
CAS support, 652–653, 658
certificate management, 208
client support, 555, 610–611
client throttling and, 638, 640
deployment preparations, 13
disabling access, 573
dumpster and, 1025
testing connectivity, 144
POP3 service, 61
ports
edge server support, 910
static, 698
positional parameters, 106
Post Office Protocol 3. *See* POP3 (Post Office Protocol 3)
Postfix MTA, 936

Postini, 907
postmarks, 972
PowerShell. *See also* remote PowerShell
about, 75
Active Directory support, 126–128
architectural overview, 76
auto-generated commands, 195–196
automating processes, 650
automating scripts, 510
client throttling and, 638, 640
debugging scripts, 106
deployment preparations, 15–16
edge servers and, 54
EMS dependencies, 86
Exchange support, 76–78
installation requirements, 40, 42, 44
ISE support, 122–126
memory limits, 115
quotation marks and, 104
table of commands, 44
transport rules, 1086
–Verbose cmdlet parameter, 136
version number information, 67
viewing service dependencies, 63
PowerShell Community Extensions project, 132
PowerShell scripts. *See* scripts
Prepare-MoveRequest.ps1 script, 744
Primary Active Manager (PAM) role, 431, 433–435, 489
primary domain controller (PDC), 27
private branch exchange (PBX), 641
private keys, 690
problem queues, 865–867
Process Tracking Log Tool, 1159
$Profile variable, 125
profiles, 124–125
programming and customizations
Active Directory schemas, 34–35
deployment preparations, 14–16
ProhibitSendQuota property, 118
Promodag Reports, 1159
Properties dialog box, 937, 1019, 1121
protected groups, 319–320
Protected Service Host service, 61
protocol logging, 685–687, 884–886
provisioning agents, mailbox, 266–269
PST files
about, 273
exporting mailbox data, 77, 747–748, 758–760
importing mailbox data, 77, 747–748, 750–757
information reliability and, 752
limiting user access, 760–762
management problems, 977
performance problems, 361
personal archives and, 977
searches and, 1029
public folder databases
background maintenance, 388
backup and restore considerations, 537, 539
circular logging and, 381
clusters and, 524
database copies and, 533
database management, 399, 401
ESE support, 358
recovery databases and, 539
repair requests, 399
replication support, 483
setting age limits, 389
transaction logs, 411
updating after installation, 381
Public Folder management console, 1130
Public Folder Management role group, 72, 151
public folders
customizing quota messages, 878–880
distribution considerations, 313, 784
mailbox import/export requests, 748
naming conventions, 200
OWA support, 603
replicating, 390
setting age limits, 389
transaction logs and, 411
Public Key Infrastructure (PKI), 1070
public keys, 1070
Purges folder, 1029, 1031
$pwd variable, 103

Q

quarantine
for mailboxes, 392–393
for mobile devices, 629
Quarantine threshold, 949
queries
discovering mailboxes, 264
dynamic distribution groups, 324–326
OPATH syntax, 113, 325
RecipientContainer property, 326
Queue Viewer, 862, 867–868, 1130
queues
delivery, 861–862
poison, 864
problem, 865–867
status indicators, 863
submission, 861, 901, 903–904
unreachable, 864
viewing, 862–864
QuickConfig command (WinRM), 89
quorum model, 466
quotas for folders, 1032
quotas for mailboxes
archiving solutions and, 978
customizing messages, 878–880

default for discovery mailboxes, 301
exceeding, 594–595
move requests and, 725
setting, 279–284
supporting larger, 369
quotation marks (“), 104, 124

R

RBAC (role-based access control). *See also* entries beginning with “role”
about, 147, 1130
architectural overview, 76
authorization requests, 316
cmdlet writing restrictions, 153
defined, 76
ECP and, 179, 182, 214, 217
EMS dependencies, 87, 152
EWS and, 649
Exchange 2010 SP1 enhancements, 170–179
execution policies and, 137
group owners and, 316
mailbox example, 81
mailbox import/export requests, 749–750
mail-enabled accounts, 89
major elements, 148–149
PowerShell and, 650
remote PowerShell and, 79, 85
troubleshooting, 90
validation rules, 179
RBL (real-time block list), 925, 932
RCA. *See* RPC Client Access (RCA) layer
RCA (Remote Connectivity Analyzer), 673, 692, 1129–1130, 1167–1169
RCA Connectivity log, 687
RCA service. *See* RPC Client Access service
RCPT TO information, 939, 955
read-only domain controllers (RODCs), 26, 31
real-time block list (RBL), 925, 932
receive connectors
about, 811, 831–834
creating, 835–840
edge servers and, 911
MaxAcknowledgementDelay property, 898
throttling, 855
Recipient Block list, 926, 939–940
Recipient Filter agent, 939–940, 947, 957
recipient filtering, 926, 939–940
Recipient Management role/role group
ECP administrator options, 227
ECP user options, 217
linking to tasks, 150
permissions, 78
security principal purpose, 72
Recipient Policies role, 172
Recipient Update Service (RUS), 290–291
Records Management role group, 72, 151
Recoverable Items folder, 1003, 1028, 1043
recovering failed servers, 55–57
recovery databases
about, 538–539
cleanup, 551
dial-tone database recovery operation, 540
dismounting, 551
mounting, 544–546
performing restores, 540–542
removing, 551
restoring mailbox data, 547–551
validating, 543–544
recovery mode (setup program), 56
Recovery Storage Group (RSG), 428, 539
Recovery Wizard, 540–542
RedistributeActiveDatabases.ps1 script, 491, 521
Refresh
In Progress message, 185, 188
registry. *See* system registry
Reject threshold, 949
relay domains, 887
release to manufacturing version. *See* RTM (release to manufacturing) version
Remote Connectivity Analyzer (RCA), 673, 692, 1129–1130, 1167–1169
Remote Desktop Connection, 204
remote domains, 889–891
remote management
backups and restores, 531
creating firewall rule, 460
creating remote mailboxes, 266
installation requirements, 40
moving mailbox databases, 382
tarpitting and, 940
wiping lost devices, 632–634
remote PowerShell
about, 79–80
adding recipient photos, 111–112
advantages, 91–93
affinity and, 697
architectural overview, 76, 81
bulk updates, 118–120
client/server filters, 114–117
command editing, 96–99
connecting to, 84–86
disadvantages, 93
EMC and, 79, 185, 188
execution cautions, 86
generating transcripts, 117
handling code changes, 120–122
handling information returned by EMS, 99–100
help support, 94
identities, 106–108
ISE support, 122–126

KerbAuth module, 87
mailbox example, 81
memory limits, 115
OPATH filters, 113
piping output, 96, 104, 109–111
selective output, 100–102
startup considerations, 92
using variables, 103–105
Remote Procedure Call. *See* RPC (Remote Procedure Call)
Remote Server Administrative Tools Active Directory Management Tools (RSAT-ADDS), 54
RemoteExchange.ps1 script, 86
Remove-AcceptedDomain cmdlet, 889
Remove-ADPermission cmdlet, 306
Remove-AttachmentFilterEntry cmdlet, 954
Remove-ContentFilterPhrase cmdlet, 952
Remove-DatabaseAvailabilityGroupNetwork cmdlet, 475
Remove-DatabaseAvailabilityGroupServer cmdlet, 463, 507
Remove-ECPVirtualDirectory cmdlet, 216
Remove-IPBlockListEntry cmdlet, 932
Remove-JournalRule cmdlet, 1125
Remove-Mailbox cmdlet
about, 285
controlling access to, 154
Exchange 2010 SP1, 286
reconnecting mailboxes and, 286, 288
–StoreMailboxIdentity parameter, 289
Remove-MailboxDatabase cmdlet, 551
Remove-MailboxDatabaseCopy cmdlet, 501, 506
Remove-MailboxExportRequest cmdlet, 750
Remove-MailboxImportRequest cmdlet, 750
Remove-MailboxPermission cmdlet, 277
Remove-MailboxSearch cmdlet, 1047
Remove-MailContact cmdlet, 154
Remove-ManagementRoleAssignment cmdlet, 169
Remove-Message cmdlet, 867
Remove-MoveRequest cmdlet
about, 720, 724, 731, 736
lingering move requests, 746
troubleshooting, 731
Remove-OutlookProtectionRule cmdlet, 1081
Remove-RetentionPolicyTag cmdlet, 1002, 1010
Remove-RoleGroupMember cmdlet, 160
Remove-StoreMailbox cmdlet, 286, 716
Remove-TransportRule cmdlet, 1086
renaming databases, 382, 452
RepAdmin utility, 21, 32
repair jobs, canceling, 400
replay directory, 871
replication. *See also* transaction log replication
block, 446–447
bootstrap replication account, 913
continuous, 426–428, 462
database considerations, 453
deployment preparations, 14
disabling, 32
multicopy, 528
PAS considerations, 112
public folder databases, 483
public folders, 390
shadow redundancy and, 895
testing with DAGs, 141
validating, 21
Replication Service
about, 61, 436
ACLL support, 439
Active Manager support, 433
backups and, 538
HighAvailability channel and, 514
IgnoredLogs directory, 445
transaction logs and, 444
reports. *See also* delivery reports
ActiveSync, 622–626
audit, 1052–1057
HTML, 131
journaling, 1116–1120
moving mailboxes, 738–742
outputting to files, 741
RBAC, 178–179
via email, 134–135
request for comments (RFC), 941
Reread Logon Quotas Interval registry entry, 309
Res1.log file, 422
Res2.log file, 422
reserved logs, 375, 422–423
Resource Booking Attendant/Assistant
about, 793
processing requests per policy, 352–355
providing policy direction, 347–351
room lists and, 315
resource domains, 24
resource forest, 24
resource mailboxes. *See also* equipment mailboxes; room mailboxes
about, 260, 343–345
creating, 265
defining custom properties, 345–346
personal archives and, 982
processing meeting requests, 352–354
providing policy direction, 347–351
restore process. *See* backups and restores
Restore-Mailbox cmdlet, 546–549
Resume-MailboxDatabaseCopy cmdlet, 502
Resume-MailboxExportRequest cmdlet, 750
Resume-MailboxImportRequest cmdlet, 750
Resume-Message cmdlet, 867
Resume-MoveRequest cmdlet
about, 720, 724, 733
handling errors, 744
Resume-Queue cmdlet, 867

retention hold, 973, 1021
retention period
deleted items, 539
dumpster and, 1026
IMAP4 limitations, 611
Outlook limitations, 559
POP3 limitations, 611
recommendations, 1026
setting, 287
retention policies
about, 973
accessing personal tags, 1012
creating, 1004–1007
customizing for specific mailboxes, 1010
default, 985–987
default policy tags and, 1021
designing, 995–997
exception to changing, 1010
implementing with MFA, 1018–1021
language support, 1016
limiting tags in, 1007
MFA automatically applying, 1014
modifying, 1009–1010
MRM support, 990
personal archives and, 981
PST limitations, 978
removing from mailboxes, 1009, 1017
retention hold and, 973
sample, 995–996
user interactions with, 1012–1017
retention policy tags (RPT)
about, 991
creating, 999
creating for calendars, 992
naming conventions, 997
sample policy, 996
retention tags
applying to mailboxes, 1007–1009
creating, 998–1002
limiting in policies, 1007
MRM support, 990
naming, 997–998
troubleshooting, 1002, 1005
types of, 990–994
upgrading from managed folders, 1018
RFC (request for comments), 941
RFC 2821, 1117
Richard, Pat, 46
RODCs (read-only domain controllers), 26, 31
role assignment policies
about, 168–169
default, 150, 152, 168–169
limiting access, 152–153
management, 149–150
role assignments
about, 152, 160–161
controlling, 162
default, 158
determining, 166–168
Exchange 2010 SP1, 160
naming convention, 161
RBAC triangle, 148
role entries, 149, 152
role groups. *See also* specific role groups
about, 156–159
assigning scopes, 156, 162–163
assignment information, 166–168
common problems, 90–91
creating, 113, 159–160
default, 158
defined, 83
ECP support, 170–174
management, 149
RBAC triangle, 148
troubleshooting, 163
USGs and, 71, 157
role scopes
assigning to role groups, 156, 162–163
defined, 155
implicit write, 155
management, 149
RBAC triangle, 148
role-based access control. *See* RBAC (role-based access control)
roles
built-in, 151
cmdlets and, 151
creating for specific tasks, 154–155
delegating, 158, 164
ECP considerations, 179
inheritance considerations, 156
management, 148, 189
RBAC triangle, 148
special, 164
unscoped, 165–166
roll-up updates (RU)
about, 63
applying, 49
applying to servers in CAS arrays, 682–684
auditing, 1057
system registry information, 68
testing, 64
room lists, 238, 314–316
room mailboxes
about, 343–344
creating, 265
defined, 259
personal archives and, 982
populating room lists, 315

properties affecting meeting requests, 352–355
providing policy direction, 347–351
Resource Information tab, 347–348
Resource Policy tab, 347–348
Root folder, 1031
root forest, 24
round robin, 697
routing
Active Directory and, 806–808
least-cost, 807–808
version-based, 813–814
routing groups, decommissioning, 900
Routing Log Viewer, 829, 1130
routing tables, 826–829, 900
RPC (Remote Procedure Call)
about, 562
Exchange 2010 support, 11
firewall rules and, 460
Get-MailboxStatistics cmdlet and, 131
Outlook considerations, 559
transport system and, 805
RPC Client Access (RCA) layer
about, 429, 657–659
CAS support, 653
considerations moving databases, 498
MAPI connections, 573, 638
RPC Client Access service
about, 657
affinity and, 696
cross-site connections, 679–680
dynamic mapping exception, 698
logging considerations, 685–687
MAPI support, 573
rerouting client connections, 498
RPC over HTTP, 559, 561–562, 570
RPC over HTTPS, 656, 691, 696
RPC over TCP, 562
RPT (retention policy tags)
about, 991
creating, 999
creating for calendars, 992
naming conventions, 997
sample policy, 996
RSAT-ADDS (Remote Server Administrative Tools Active Directory Management Tools), 54
RSG (Recovery Storage Group), 443, 539
RTF files, 371
RTM (release to manufacturing) version
client throttling and, 641
components, 77
example, 65
moving mailboxes and, 730
personal archives and, 985
servers in CAS arrays, 684
RU (roll-up updates)
about, 63
applying, 49
auditing, 1057
servers in CAS arrays, 682–684
system registry information, 68
testing, 64
RUS (Recipient Update Service), 290–291

S

SACL (system access control list), 28
Safe Senders list, 971
safelist aggregation, 961–963
SAM (Standby Active Manager) role, 433–435
Sarbanes-Oxley Act, 973
SATA (Serial ATA) storage, 528
Scales, Glen, 96, 649
scanning attachments, 1105–1107
SCC (single copy cluster), 14
Scheduling Assistant, 793
scheduling mailbox maintenance, 387–388
Schema Admins group, 41
schemas
customizing, 34–35
enhancements, 365
extending, 41
I/O considerations, 364
mailbox database layouts, 372–374
ThumbnailPhoto attribute and, 111
troubleshooting, 35
upgrading, 21, 31–32
SCL (spam confidence level)
content filtering and, 926
in transport rules, 953
junk mail filter and, 971
Sender ID agent and, 944
x-headers and, 946
SCL thresholds
about, 948–949
per-mailbox, 950
setting, 950
SCOM alerts, 396
scope
database, 174–175
management role, 149
role, 148–149, 155–156, 162–163
send connectors, 852
site, 668
SCP (Service Connection Point), 663–666
SCR (standby continuous replication)
about, 425
database considerations, 360, 451
deployment preparations, 14
moving databases and, 655
Test-ReplicationHealth cmdlet and, 141

scripts
activating mailbox database copies, 491
automating, 510
bulk mailbox creation, 277–279
calling, 123–124
cmdlet recommendations, 94
connecting remote PowerShell, 86
DAG management, 520–523
debugging, 106
distinguished names and, 453
edge server configuration data, 923
EMS execution policies and, 138
ExchangeSetupLogs folder, 50
for moving databases, 724
initialization, 82, 86, 92, 125–126
installing anti-spam agents, 924
location considerations, 166
mailbox naming conventions and, 258
maintaining groups, 251
measuring message latency, 1151
message classifications, 1069
moving mail queue database, 860
moving mailboxes, 744
protection against excessive growth, 396–397
protection against high latency, 395
reporting anti-spam activity, 959
transport rules, 1086
unscoped roles, 165
search catalog, ignoring, 497
search engines, restricted calendars, 593
Search Indexer (MSExchangeSearch), 62
search mailboxes, 504
Search-AdminAuditLog cmdlet, 1054–1057
searches
delivery reports, 234–235
discovery mailboxes and, 299
Discovery Management role group support, 302
LDAP considerations, 618
PSTs and, 1029
wildcard characters, 94
Search-Mailbox cmdlet, 1034
Search-MailboxAuditLog cmdlet, 1058, 1061
Search-MessageTrackingReport cmdlet, 231, 234–235
Secure Multipurpose Internet Mail Extensions. *See* S/MIME
Secure Sockets Layer. *See* SSL (Secure Sockets Layer)
security
CEIP data and, 60
certificate management and, 208–209
confidential information, 230
content protection, 1070–1081
credentials in plain text, 133
EMS execution policies, 137–138
Kerberos-based encryption, 79
mailbox maintenance and, 256
message hygiene and, 908
multiforest deployments, 24
password in plain text, 263
personal accounts and, 235
PST files and, 977
RBAC considerations, 161
TLS considerations, 830
security groups
about, 71–73, 312
creating, 243–244
ECP support, 215, 243–244
mail users and, 342
protected groups and, 320
split permissions model, 175–177
security support provider (SSP), 470
seeding databases
about, 451
handling initial errors, 479
reseeding copies, 481–482
self-maintaining groups, 321–322
send connectors
about, 828, 841–844
creating, 845–851
selecting, 851–852
throttling, 855
SendAs permission, 303–306, 1058
Sender Filter agent, 934–935, 957
sender filtering, 925, 934–935
Sender Filtering agent, 937
Sender ID agent, 940–945
sender ID filtering, 926, 936, 940–945
Sender ID Wizard, 942, 945
Sender Policy Framework (SPF), 929, 940
Sender Policy Framework project, 941
Sender Reputation agent, 936–938
Sender Reputation Level (SRL), 936–938
SendMail MTA, 936
Sendmail system, 836
Send-MailMessage cmdlet, 132
SendOnBehalf permission, 304–306, 309–310, 1058
Send-SMTPMail cmdlet, 132
Sent Items folder
creating retention tags, 1000
languages and, 269
storing text messages, 619
views and, 365
Serial ATA (SATA) storage, 528
Server Extension for Windows Backup (WsbExchange) service, 62
Server Failures folder, 568
Server Management role group, 72, 150
Server Manager, 45, 463
Server Message Block (SMB), 440
ServerManagerCmd command, 43–44
server-side processing
ActiveSync mailbox policies, 621

configuring IMAP4 servers, 612–615
filter support, 114–117
inbox rules, 221
Service Connection Point (SCP), 663–666
Service Host service, 62
service level agreements (SLAs), 1151
service packs. *See also* Exchange 2010 SP1
about, 63, 65
applying to servers in CAS arrays, 682–684
auditing, 1057
Exchange 2010 SP1, 65–66
installation requirements, 40, 47
Set-ActiveSyncOrganizationSettings cmdlet, 626
Set-AdminAuditLogConfig cmdlet, 1050–1052
Set-ADSite cmdlet, 809
Set-ADSiteLink cmdlet, 808–809
Set-Alias cmdlet, 124
Set-AttachmentFilterListConfig cmdlet, 955
Set-AutodiscoverVirtualDirectory cmdlet, 702
Set-CalendarProcessing cmdlet
about, 273
additional information, 276
automatically deleting notifications, 585
Calendar Assistant and, 276
updating properties, 351
Set-CASMailbox cmdlet
allowing accessing, 631–632
applying OWA policies, 601
parameters supported, 570
piping input into, 120
restricting access, 572–574
setting properties, 76
Set-ClientAccessServer cmdlet, 667–668
Set-Contact cmdlet, 771
Set-DatabaseAvailabilityGroup cmdlet
adding IP addresses, 460
changing FSW location, 471
DatacenterActivationMode property, 511
editing DAG properties, 458, 470
Set-DatabaseAvailabilityGroupNetwork cmdlet, 474
Set-DistributionGroup cmdlet
–ArbitrationMailbox parameter, 340
–AutoApproveNestedDLEnabled parameter, 339
–ByPassNestedModerationEnabled parameter, 335
changing group properties, 244, 246
custom MailTips, 770
EMC support, 186
–MailTipTranslations parameter, 771
–ManagedBy parameter, 318
marking groups, 315
moderated requests, 338
protected groups, 320
Set-DynamicDistributionGroup cmdlet, 326, 334, 337
Set-ECPVirtualDirectory cmdlet, 216
Set-EdgeSyncServiceConfig cmdlet, 921
Set-EmailAddressPolicy cmdlet, 295, 298, 888
Set-EventLogLevel cmdlet, 390, 782
Set-Exchange2010Prereqs.ps1 script, 45
Set-ExchangeServer cmdlet, 192, 378, 591
Set-ExecutionPolicy cmdlet, 137
Set-Group cmdlet, 791–793
Set-IMAPSettings cmdlet, 613–615
Set-InboxRule cmdlet, 223
Set-IPAllowListConfig cmdlet, 931
Set-IPBlockListConfig cmdlet, 931
Set-IPBlockListEntry cmdlet, 932
Set-IPBlockListProvidersConfig cmdlet, 931
Set-IRMConfiguration cmdlet, 1042, 1075
Set-JournalRule cmdlet, 1125
Set-Mailbox cmdlet
about, 93
AntiSpamBypassEnabled property, 951
applying retention tags, 1007
CalendarRepairEnabled property, 797
custom MailTips, 770
EMC support, 186
EMS support, 76
enabling mailboxes for auditing, 1059
enabling single item recovery, 1032
Get-User cmdlet and, 118
implementing MailTips, 340
listing roles supporting, 152
–MailTipTranslations parameter, 771
management role assignment policies, 169
manipulating settings, 263, 281–282
OAB support, 789
personal archives and, 984
preferred language setting, 270
putting mailbox on litigation hold, 1022
putting mailbox on retention hold, 1021
RBAC considerations, 84
removing picture data, 112
removing policies from mailboxes, 982
resource mailboxes, 346
–RetentionURL parameter, 1008
role group example, 83
SCL thresholds, 950
security mailboxes, 1126
setting storage quota, 1053
setting user limits, 822
Set-MailboxAutoReplyConfiguration cmdlet, 273–274
Set-MailboxCalendarConfiguration cmdlet, 273, 275, 351
Set-MailboxCalendarFolder cmdlet, 594
Set-MailboxCalendarSettings cmdlet, 351, 586
Set-MailboxDatabase cmdlet
activating mailbox database copies, 490
BackgroundDatabaseMaintenance property, 386
blocking provisioning temporarily, 267
controlling permanent removal, 394
DataMoveReplicationConstraint property, 750–752

default deleted items retention period, 1032
enabling circular logging, 420, 476
OAB support, 789
renaming databases, 382, 453
scheduling background maintenance, 388
setting retention period, 287
standard journaling and, 1122
updating properties, 660
Set-MailboxDatabaseCopy cmdlet, 175, 483, 486
Set-MailboxExportRequest cmdlet, 750
Set-MailboxImportRequest cmdlet, 749
Set-MailboxJunkEmailConfiguration cmdlet, 274, 277
Set-MailboxMessageConfiguration cmdlt, 274, 277
Set-MailboxRegionalConfiguration cmdlet, 272–274
Set-MailboxSearch cmdlet, 1047
Set-MailboxServer cmdlet
activation blocks, 499
AutoDatabaseMountDial property, 495
CRA support, 796
DatabaseCopyAutoActivationPolicy property, 499
delivery reports and, 230
disabling MAPI encryption, 661
group metrics and, 764–765
limiting mounted databases, 498
message tracking, 1137, 1139
setting work cycles, 798
SubmissionServerOverrideList property, 802
Set-MailboxSpellingConfiguration cmdlet, 274, 277
Set-MailContact cmdlet, 341
Set-MailPublicFolder cmdlet, 771
Set-MailUser cmdlet, 343
Set-MoveRequest cmdlet, 745
Set-OABVirtualDirectory cmdlet, 669, 783
Set-OfflineAddressBook cmdlet, 780, 783, 790
Set-OrganizationConfig cmdlet, 201, 766, 950
Set-OutlookProtectionRule cmdlet, 1081
Set-OutlookProvider cmdlet, 563, 656, 666
Set-OWAMailboxPolicy cmdlet, 604, 607, 610
Set-OWAVirtualDirectory cmdlet, 592
Set-POPSettings cmdlet, 613–614
Set-PSDebug cmdlet, 106
Set-PublicFolder cmdlet, 389
Set-PublicFolderDatabase cmdlet, 388
Set-ReceiveConnector cmdlet
about, 93
administrative comments, 840
authenticated connections, 833
configuring settings, 820, 822
protocol logging and, 885
throttling and, 855–856
Set-RecipientFilterConfig cmdlet, 940
Set-RemoteDomain cmdlet, 890
Set-ResourceConfig cmdlet, 345–346
Set-RetentionPolicy cmdlet, 1009
Set-RetentionPolicyTag cmdlet, 1002, 1010, 1017
Set-RoleAssignmentPolicy cmdlet, 169
Set-RPCClientAccess cmdlet, 573–574, 661
Set-SendConnector cmdlet
configuring settings, 821, 851
protocol logging and, 886
throttling and, 855, 857
updating source servers, 848
Set-SenderFilterConfig cmdlet, 935
Set-SenderIDConfig cmdlet, 945
Set-SenderReputationConfig cmdlet, 938
Set-Service cmdlet, 612
Set-SharingPolicy cmdlet, 592
Set-SystemMessage cmdlet, 876
Set-TransportAgent cmdlet, 953, 1114
Set-TransportConfig cmdlet
configuring settings, 816, 818–820, 823, 844
–InternalSMTPServers parameter, 925
–JournalingReportNdrTo parameter, 1120
shadow redundancy and, 897
Set-TransportRule cmdlet, 1086, 1103
Set-TransportServer cmdlet
configuring settings, 807
connectivity logging, 881–882
disabling data collection, 230
DSN messages, 872
message tracking, 1139
pickup directory location, 869
protocol logging and, 886
routing tables, 827
throttling, 855
setup program
architectural overview, 76
command-line version, 47
–DBFilePath parameter, 452
delegating setup activity, 57–58
depicted, 42
installation modes supported, 47
installing edge servers, 54
Language option, 55
log files, 49–50
–LogFolderPath parameter, 452
–mdbname parameter, 452
PrepareAD option, 31, 34, 41–42
PrepareAllDomains option, 32
PrepareDomain option, 32
PrepareLegacyExchangePermissions option, 35
PrepareSchema option, 31–32, 34
PS option, 31
recoverserver option, 56–57
remote PowerShell and, 79
repair constraints, 53
running, 46–49
split permissions model, 176
troubleshooting, 49, 53
uninstalling Exchange, 47, 51–52

Set-User cmdlet
enabling user accounts, 85
setting properties, 186, 196
updating mailbox properties, 264
Set-WebServicesVirtualDirectory cmdlet, 669, 702
sexual harassment scenario, 2, 529
SHA-256 algorithm, 961
shadow redundancy, 894–898, 901, 906
Shadow Redundancy Manager (SRM), 898
ShadowRedundancy queue, 864
sharing policies, 205–207
Sharing Policy Assistant, 798
Short Message Service (SMS)
ActiveSync support, 619
CEIP data gathering, 58
foreign connectors, 893
mobile devices and, 619
Outlook limitations, 559
Silverlight development platform, 596
Simple Mail Transfer Protocol. *See* SMTP (Simple Mail Transfer Protocol)
Simple Object Access Protocol (SOAP), 647
single copy cluster (SCC), 14
Sirana AppAnalyzer, 1159
site scope, 668
SLAs (service level agreements), 1151
sliding window algorithm, 418
SmartHost queue, 864
SmartScreen technology, 946–947, 960, 966
SMB (Server Message Block), 440
S/MIME (Secure Multipurpose Internet Mail Extensions)
discovery searches, 1036
dragging and dropping attachments, 578
encryption and, 219
OWA support, 603
personal digital signatures and, 309
SMS (Short Message Service)
ActiveSync support, 619
CEIP data gathering, 58
foreign connectors, 893
mobile devices and, 619
Outlook limitations, 559
SMTP (Simple Mail Transfer Protocol)
accepted domains and, 889
certificate management, 208
client throttling and, 640
configuring IMAP4 clients, 615
discard notifications, 896
distribution groups, 234
edge servers and, 911
email address policies, 290
foreign connectors and, 893
GuessSmart process and, 669
journal reports and, 1117
linked connectors, 853–854
load balancing, 902
mailbox considerations, 195, 255, 289, 311
message considerations, 132, 146
receive connectors, 811, 831–834, 855
remote domains, 889
routing tables, 829
send connectors, 828, 841–852, 855
Sender ID agent, 940
shadow redundancy and, 898
transport system and, 806, 891
version-based routing, 813
SmtpRelay queue, 864
SOAP (Simple Object Access Protocol), 647
Software Quality Metrics (SQM), 60
sorting objects, 257
Sort-Object cmdlet, 94
spam. *See* message hygiene
spam confidence level. *See* SCL (spam confidence level)
spell checking, 277, 582, 603
SPF records, 929, 940
split brain syndrome, 511
SQL, ESE and, 358–359
SQM (Software Quality Metrics), 60
SRL (Sender Reputation Level), 936–938
SRM (Shadow Redundancy Manager), 898
SRV records, 675–676
SSL (Secure Sockets Layer)
CAS support, 655, 691
certificate management, 208
deployment preparations, 15
IMAP4 servers, 612
remote PowerShell and, 85
SSL offloading, 702
SSP (security support provider), 470
standard journaling, 1121–1122
Standby Active Manager (SAM) role, 433–435
standby continuous replication (SCR)
about, 425
database considerations, 360, 451
deployment preparations, 14
moving databases and, 655
Test-ReplicationHealth cmdlet and, 141
StartDAGServerMaintenance.ps1 script, 510, 522, 574
Start-EdgeSynchronization cmdlet, 914, 921
Start-MailboxSearch cmdlet, 1047
Start-ManagedFolderAssistant cmdlet, 1020
STARTTLS security, 616
static Autodiscover configuration, 673–674
static IP addresses, 456, 460
static ports, 698
STM (streaming file), 402
S/TNEF (format), 1118
StopDAGServerMaintenance.ps1 script, 510, 522
Stop-Transcript cmdlet, 117

storage considerations
Active Directory locations, 21
discovery mailboxes, 302
handling storage hangs, 507–508
I/O and, 364
SATA, 528
selecting locations, 377
troubleshooting, 444
voice mail preview, 646
storage groups, 428
Store (Exchange 2010)
API considerations, 648
database defragmentation, 404–407
database management, 374–403
database usage statistics, 407–409
dealing with I/O, 364–374
enhancements, 2, 357, 360, 366
index-aging table, 388
logical corruption, 485
mailbox database layouts, 372–374
maintaining contiguity, 370–371
major components, 358
MAPI connections and, 653
maximum database size, 359–363
moderated requests and, 338
nightly online maintenance tasks, 383
transaction logs, 409–423
Store Driver
about, 1145
client connections, 570
connection limits, 397–398
fault isolation, 397–398
Store dumpster, 1025
submission queues, 861, 901, 903–904
Support Diagnostics role, 164
Suspend-MailboxDatabaseCopy cmdlet, 487, 493, 501
Suspend-MailboxExportRequest cmdlet, 750
Suspend-MailboxImportRequest cmdlet, 750, 757
Suspend-Message cmdlet, 867
Suspend-MoveRequest cmdlet, 720, 724
switchovers, 433
Symantec Enterprise Vault, 975, 978
Sync Issues folder, 567–568
synchronization
ActiveSync support, 619, 623–626
ADAccess flag, 27
attribute, 728
computer clocks, 91
conflict resolution, 567–568
drizzle-mode, 569, 687, 714
edge servers, 909, 911–922
ICS, 714
incremental resynchronization, 449–450
ongoing, 919–922
Outlook considerations, 774
troubleshooting, 35
validating, 915–919
wiping lost devices and, 634
system access control list (SACL), 28
System Attendant service, 25, 62, 793–794
system registry
concurrent open mailboxes, 558
controlling PST access, 761–762
database size limit, 362
installation watermark, 49
OWA considerations, 584
quarantined mailboxes, 394
roll-up update information, 68
static Autodiscover configuration, 674
version number information, 67
system shutdowns, 510
system tags, 994–995
SystemMailbox account, 72

T

TAP (Technology Adoption Program), 6, 92
tarpitting, 940
TCP (Transmission Control Protocol), 440, 470, 562, 698
Technology Adoption Program (TAP), 6, 92
Test-AssistantHealth cmdlet, 794
Test-ECPConnectivity cmdlet, 143, 216
Test-EdgeSynchronization cmdlet, 919
Test-ExchangeSearch cmdlet, 142
testing
backup and restore process, 537
beta versions, 6
cmdlets, 139–146
customized text, 876
deployment preparations, 12–13
filter effectiveness, 333
for operational processes, 14
for programming and customization, 14–16
mobile connectivity, 636
roll-up updates, 64
voice mail preview, 645
Web browsers, 579
Test-IPBlockListProvider cmdlet, 933
Test-IRMConfiguration cmdlet, 1075
Test-MailFlow cmdlet, 139, 145–146
Test-MAPIConnectivity cmdlet, 141
Test-MRSHealth cmdlet, 144
Test-OutlookConnectivity cmdlet, 692
Test-OWAConnectivity cmdlet, 143
Test-POP3Connectivity cmdlet, 144
Test-PowerShellConnectivity cmdlet, 92
Test-ReplicationHealth cmdlet, 141, 501
Test-SenderID cmdlet, 945
Test-ServiceHealth cmdlet, 140
Test-SystemHealth cmdlet, 139–140
text messaging, 604, 619

Text Messaging Delivery Agent Connector, 894
text messaging routing agent, 894
Thawte (vendor), 208
themes
about, 597–600
user-selectable, 576, 603
Threat Management Gateway (TMG), 665, 691, 694, 965
throttling, transport system, 854–857
Throttling service, 62
ThumbnailPhoto attribute, 111
thumbnails, 781
time to live (TTL), 429
time/date formats, 272
timeslot expirations, 1019
timestamps, 413, 713
TLS (Transport Layer Security)
certificate management, 208
edge servers and, 930
IMAP4 servers, 612
opportunistic, 830
routing overview, 806
security considerations, 830
TMG (Threat Management Gateway), 665, 691, 694, 965
TNEF (Transport Neutral Encapsulation Format), 811
TNEF format, 1118
tombstone maintenance, 388–390
Top 'N' Words Assistant, 798
Topology Discovery process (ADAccess), 26–28
TPD (Trusted Publishing Domain), 1080
Tracking Log Explorer, 1130, 1153–1157
transaction log replication
about, 429
block replication, 446–447
compression, 445–446
incremental resynchronization, 449–450
seeding databases, 451
truncating, 448–449
transaction logs
about, 358, 375, 409
checksum validation, 417
circular logging and, 419–421, 448, 476
compressing, 445–446
creating, 378
DAG considerations, 440–445
I/O considerations, 418
latency and, 473
log sets, 410–412
lossy failure, 716
mailbox databases, 409–423, 714
moving location, 454
noncircular logging, 421–422
recovered databases and, 544
replication and replay, 440–451
reserved logs, 422–423
restoring, 542
size recommendations, 411
storage groups and, 428
transactions, buffers, commitment, 413–417
truncating, 536
transcription, voice mail preview, 642, 644–646
transcripts, generating, 117
Transmission Control Protocol (TCP), 440, 470
Transport Layer Security. ***See*** **TLS (Transport Layer Security)**
Transport Log Search service, 62
Transport Neutral Encapsulation Format (TNEF), 811
transport pipeline, 891–893, 924
transport queues
delivery queues, 861–862
pickup directory, 869–870
problem queues, 865–867
Queue Viewer, 862, 867–868
replay directory, 871
submission queues, 861
viewing queues, 862–864
Transport Rule agent, 953
transport rules
about, 1083–1084
AD RMS and, 1107–1109
available actions, 1112–1113
basic structure, 1088
blocking sending external mail, 1102–1105
caching, 1110–1111
creating corporate disclaimers, 1091–1097
custom transport agents, 1113
ECP considerations, 235, 1087
edge versus hub, 1088–1089
ethical firewalls, 1101
evaluating AD attributes, 1099–1100
examples, 1085–1086
message classifications and, 1108–1109
moderated workflow, 1097–1099
priorities for, 1114
scanning attachments, 1105–1107
SCL values in, 953
setting priorities, 1089–1091
transferring between versions, 1111
Transport service, 62, 824, 931, 1110
transport system
about, 801–802
accepted domains, 886–889
architectural overview, 802–814
auditing configuration changes, 1057
back pressure, 857–858
configuration settings, 816–830
customizable system messages, 871–880
delivery connectors, 893–894
Exchange 2010 SP1, 901–906
foreign connectors, 893–894
linked connectors, 853–854
linking Exchange 2003/2010, 898–901

logging, 880–886
receive connectors, 831–834
remote domains, 889–891
send connectors, 841–852
shadow redundancy, 894–898
throttling, 854–857
transport pipeline, 891–893
transport queues, 859–871
Troubleshoot-DatabaseLatency.ps1 script, 395
Troubleshoot-DatabaseSpace.ps1 script, 396–397
troubleshooting
accessing mailboxes, 667
CALs, 262
client throttling, 639
connectivity logging, 882
DAGs, 284
deleted messages, 132
delivery reports, 229
diagnostic levels and, 253
EMC, 29, 189
log file support, 49–50
mailboxes, 261–263, 268
moving mailboxes, 731
multiple tags for same folder, 1005
personal archives, 988
piping output, 110–111
protection against excessive growth, 396–397
quarantined mailboxes, 393
RBAC, 90
retention tags, 1002, 1005
retrieving deleted messages, 389
role groups, 163
setup program, 49, 53
Store Driver fault isolation, 397–398
synchronization issues, 35
WinRM, 90
WS-Management service, 89
$True variable, 103
truncating transaction logs, 448–449, 536
Trusted Publishing Domain (TPD), 1080
TTL (time to live), 429

U

UAC (User Account Control), 47
UAdndex.oab file, 774
UBrowse.oab file, 774
UDetails.oab file, 774
UDP (User Datagram Protocol), 559–561
UM Management role group, 72, 151
UM Reporting Assistant, 798
unattended installations, 47
UNC (universal naming convention), 457
Unified Messaging
about, 641–642
ActiveSync support, 619
AD schema support, 34
CAS installation priority, 655
certificate management and, 208
deployment preparations, 17
ECP support, 215, 603
EMC restrictions, 187
enhancements, 5
fax integration, 647
IPv6 and, 475
language pack support, 55
Outlook limitations, 559
voice mail preview, 642–646
Windows Desktop Experience feature, 42
Uniform Resource Identifier (URI), 667
Uniform Resource Locator (URL), 48, 311, 701, 965
uninstall mode (setup program), 47, 51–52
uninstalling Exchange, 51–52
universal naming convention (UNC), 457
Universal Principal Name (UPN), 120, 261
universal security groups. *See* USGs (universal security groups)
UNIX environment, 924
unreachable queues, 864
Unscoped role management role, 164
unscoped roles, 165–166
Update Database Copy Wizard, 481
Update-EmailAddressPolicy cmdlet, 297–298
Update-FileDistributionService cmdlet, 782
Update-MailboxDatabaseCopy cmdlet, 482, 497
Update-OfflineAddressBook cmdlet, 781, 789
Update-SafeList cmdlet, 963
UPdndex.oab file, 774
upgrades. *See also* roll-up updates; service packs
Active Directory schemas, 21, 31–32
antivirus products and, 965
auditing, 1057
best practices, 64
EMS considerations, 118–120
for workstations, 11
hot fixes, 49, 1057
hub transport servers, 1085
in-place, 8–9
migration from older platforms, 4–5, 8–9
motivation for, 3–9
Outlook builds, 571
servers in CAS arrays, 682–684
servers in DAGs, 508–509
UPN (Universal Principal Name), 120, 261
URdndex.oab file, 774
URI (Uniform Resource Identifier), 667
URL (Uniform Resource Locator), 48, 311, 701, 965
usage statistics/reports
for ActiveSync, 623
for databases, 407–409
for discovery mailboxes, 73

for distribution groups, 324
User Account Control (UAC), 47
user accounts
creating groups, 247–249
identifying groups, 244–247
linked mailboxes, 260
protocolsetting attribute, 85
recovery database and, 539
resource forests and, 24
User Datagram Protocol (UDP), 559–561
User Options role, 171
Users OU, 72, 92, 239
USGs (universal security groups)
assigning roles, 164
created in Active Directory, 71–72, 157
DAG considerations, 458
defined, 312
Exchange Install Domain Servers group, 72
mail-enabled, 249
role groups and, 71, 157
self-maintaining groups and, 321
unscoped roles, 165
UTmplts.oab file, 774

V

validation
checksum, 384–386, 417
data replication, 21
edge server synchronization, 915–919
mailbox names, 310
RBAC rules, 179
recovery databases, 543–544
variables
common/user-defined, 103–105
naming conventions, 105
Verisign (vendor), 138, 208
version numbers
about, 65–68
beta, 6
deployment preparations and, 17
Exchange 2010 support, 10–11
groups and, 312
object versions, 68–69
RTM, 65
version-based routing, 813–814
Versions folder, 1030–1031
ViewEntireForest property, 128
View-Only Management role, 189
View-Only Organization Management role group, 72, 150
View-only Organization role, 228
View-Only Recipients role, 171
views. *See also* conversation views
expiry dates controlling, 388
I/O considerations, 365
Outlook limitations, 559
virtual directories
calendar vdir, 591
language settings, 272
remote PowerShell and, 85
segmentation properties, 604–610
Test-OWAConnectivity cmdlet and, 143
virtual list view (VLV), 299
virtual private networks (VPNs), 559, 656
virtual servers, 48, 59
Virus Scanning API (VSAPI), 806, 907, 964
viruses. *See* message hygiene
Vista operating system, 11, 55
VLV (virtual list view), 299
voice mail
journaling messages, 1116
retention tags and, 1002
voice mail preview
about, 642–646
Outlook limitations, 559
VPNs (virtual private networks), 559, 656
VSAPI (Virus Scanning API), 806, 907, 964
VSS (Volume ShadowCopy Services)
about, 531–532
backup considerations, 529–530
database size and, 360
deployment preparations, 14
HighAvailability channel and, 514
restoring databases, 360
Windows Server Backup and, 530, 532–533

W

WAN Optimization Controllers (WOCs), 830
WCF (Windows Communications Framework), 475, 648
Web browsers
AD RMS support, 578
multiple sessions, 583
operating system support, 579–580
OWA Light support, 582
OWA support, 556, 579–582
spell checking capability, 582
testing considerations, 579
Web parts, 578, 586
Web services
load balancing and, 701
MailTips support, 763
WebDAV (Web Distributed Authoring and Versioning)
API considerations, 647–648
CAS installation priority, 656
deployment preparations, 15
upgrade considerations, 8
WebReady converter, 606
Where-Object cmdlet, 94, 108, 166
white space, 283, 396
wildcard characters
asterisk, 152

in searches, 94
in string compares, 113
Windows 7 operating system, 11
Windows Certificate Services, 209
Windows Communications Framework (WCF), 475, 648
Windows Desktop Experience feature, 42
Windows Desktop Search, 1035
Windows Failover Cluster Manager, 466–467
Windows Failover Clustering
about, 431–432
DAG considerations, 460, 463, 516
installing, 463
IPv6 and, 475
NIC considerations, 468
server limitations, 429
Windows Firewall, 661
Windows Load Balancing, 682
Windows Mobile
ActiveSync and, 618, 620
AD RMS support, 1072–1073
audio codecs, 642
Autodiscover service, 663
client considerations, 556
conflict resolution, 567
content protection, 1079
deployment preparations, 13
IRM support, 1071
junk mail filter, 969
mail notifications, 561
testing connectivity, 636
Windows Remote Management. *See* WinRM (Windows Remote Management)
Windows Server 2008
cmdlet considerations, 126
DAG OS requirements, 432
Exchange 2010 support, 10–11
installation prerequisites, 42–46
language packs, 55
Windows Server Backup
about, 530–531
consistency check, 534
database copies and, 537–538
process overview, 533–536
restore process, 540–542
server recovery and, 552
VSS and, 530, 532–533
Windows terminal services, 11
WinRM (Windows Remote Management)
EMS dependencies, 87
installation requirements, 40, 42
QuickConfig command, 89
remote PowerShell and, 80
troubleshooting, 90
WSMan module, 87–88
WMA audio codec, 642
WOCs (WAN Optimization Controllers), 830
Wong, Meng Weng, 941
work cycles, 797–799
working directory, 124
workstations, upgrading, 11
Write-AdminAuditLog cmdlet, 1053, 1057
write-ahead logging model
about, 409–410
circular logging, 419–421
I/O considerations, 418
log sets, 410–412
noncircular logging, 421–422
reserved logs, 422–423
transactions, buffers, commitment, 413–417
WSBExchange.exe. *See* Windows Server Backup
WSMan module, 87–88
WS-Management service, 80, 89

X

X.400 protocol, 816
X.509 certificates, 208
x-headers
adding by anti-spam agents, 926–929
application support, 403
content filtering and, 946
firewall support, 824–825
IMAIL support, 402
journaling, 1116
transport configuration settings, 822
X-Microsoft-Exchange-ProcessedByJournaling (x-header), 1116
XML format, 140, 635
XML templates, 1162–1165
X-MS-Exchange-Organization-Antispam-Report (x-header), 928, 946
X-MS-Exchange-Organization-OriginalSize (x-header), 822
Xobni, 1165
XPRESS, 470
XQDISCARD verb, 884, 897
XSHADOW ESMTP verb, 884, 897
XSO layer, 813

Y

Yahoo! Calendar, 590

Z

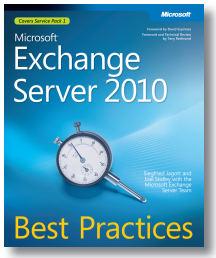
Zenprise Mobile Manager, 1165

About the Author

Before retiring in March 2010, **Tony Redmond** spent 26 years at Digital Equipment Corporation, Compaq, and HP, ending up as the Vice-President and Chief Technology Officer for HP Services. Throughout that time, he maintained a close interest in enterprise email and worked on products such as ALL-IN-1, MailWorks, and TeamLinks at Digital, and then formed a group to work on Microsoft Exchange Server 4.0 in 1996. Over the next 14 years, Tony stayed interested in Exchange and wrote extensively about the technology. This is his tenth book covering Exchange.

Although Tony now runs his own consulting company, his range of other interests limits the time he spends doing real work. You can track his activities on his blog at *http://thoughtsofanidlemind.wordpress.com/*.

Resources for Microsoft Exchange Server and Forefront

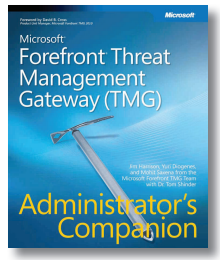


Microsoft® Exchange Server 2010 Best Practices

Siegfried Jagott and Joel Stidley with the Microsoft Exchange Server Team

ISBN 9780735627192

Apply real-world best practices, field-tested solutions, and candid advice for administering Exchange Server 2010 and SP1—and optimize your operational efficiency and results.

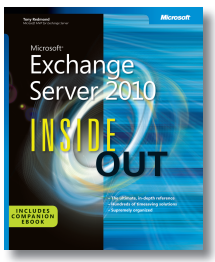


Microsoft Forefront® Threat Management Gateway (TMG) Administrator's Companion

Jim Harrison, Yuri Diogenes, and Mohit Saxena from the Microsoft Forefront TMG Team with Dr. Tom Shinder

ISBN 9780735626386

Help protect your business from Web-based threats with this essential administrator's reference to planning, deploying, and managing Forefront TMG—successor to Microsoft ISA Server.



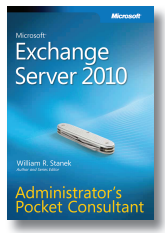
Microsoft Exchange Server 2010 Inside Out

Tony Redmond

ISBN 9780735640610

Pre-order now

This supremely organized reference packs all the details you need to deploy and manage your Exchange Server 2010–based system—from the inside out. Covers SP1.



Microsoft Exchange Server 2010 Administrator's Pocket Consultant

William R. Stanek

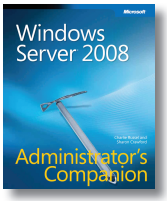
ISBN 9780735627123

Portable and precise, this pocket-sized guide delivers ready answers for the day-to-day administration of Exchange Server 2010.

microsoft.com/mspress

Microsoft® Press

Windows Server 2008—Resources for Administrators



Windows Server® 2008 Administrator's Companion

Charlie Russel and Sharon Crawford

ISBN 9780735625051

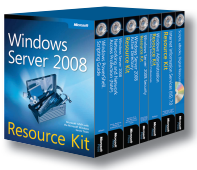
Your comprehensive, one-volume guide to deployment, administration, and support. Delve into core system capabilities and administration topics, including Active Directory®, security issues, disaster planning/recovery, interoperability, IIS 7.0, virtualization, clustering, and performance tuning.

Windows Server 2008 Administrator's Pocket Consultant, Second Edition

William R. Stanek

ISBN 9780735627116

Portable and precise—with the focused information you need for administering server roles, Active Directory, user/group accounts, rights and permissions, file-system management, TCP/IP, DHCP, DNS, printers, network performance, backup, and restoration.

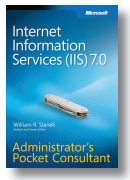


Windows Server 2008 Resource Kit

Microsoft MVPs with Microsoft Windows Server Team

ISBN 9780735623613

Six volumes! Your definitive resource for deployment and operations—from the experts who know the technology best. Get in-depth technical information on Active Directory, Windows PowerShell® scripting, advanced administration, networking and network access protection, security administration, IIS, and more—plus an essential toolkit of resources on CD.

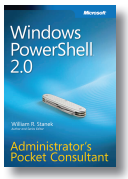


Internet Information Services (IIS) 7.0 Administrator's Pocket Consultant

William R. Stanek

ISBN 9780735623644

This pocket-sized guide delivers immediate answers for administering IIS 7.0. Topics include customizing installation; configuration and XML schema; application management; user access and security; Web sites, directories, and content; and performance, backup, and recovery.



Windows PowerShell 2.0 Administrator's Pocket Consultant

William R. Stanek

ISBN 9780735625952

The practical, portable guide to using *cmdlets* and scripts to automate everyday system administration—including configuring server roles, services, features, and security settings; managing TCP/IP networking; monitoring and tuning performance; and other essential tasks.

ALSO SEE

Windows PowerShell 2.0 Best Practices

ISBN 9780735626461

Windows® Administration Resource Kit: Productivity Solutions for IT Professionals

ISBN 9780735624313

Windows Server 2008 Hyper-V™ Resource Kit

ISBN 9780735625174

Windows Server 2008 Security Resource Kit

ISBN 9780735625044

Get Certified—Windows Server 2008

Ace your preparation for the skills measured by the Microsoft® certification exams—and on the job. With 2-in-1 *Self-Paced Training Kits*, you get an official exam-prep guide + practice tests. Work at your own pace through lessons and real-world case scenarios that cover the exam objectives. Then, assess your skills using practice tests with multiple testing modes—and get a customized learning plan based on your results.

EXAMS 70-640, 70-642, 70-646

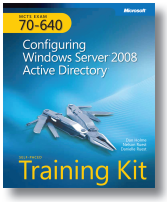
MCITP Self-Paced Training Kit: Windows Server® 2008 Server Administrator Core Requirements

ISBN 9780735625082

EXAMS 70-640, 70-642, 70-643, 70-647

MCITP Self-Paced Training Kit: Windows Server 2008 Enterprise Administrator Core Requirements

ISBN 9780735625723



EXAM 70-640

MCTS Self-Paced Training Kit: Configuring Windows Server® 2008 Active Directory®

Dan Holme, Nelson Ruest, and Danielle Ruest

ISBN 9780735625136

EXAM 70-647

MCITP Self-Paced Training Kit: Windows® Enterprise Administration

Orin Thomas, et al.

ISBN 9780735625099

EXAM 70-642

MCTS Self-Paced Training Kit: Configuring Windows Server 2008 Network Infrastructure

Tony Northrup, J.C. Mackin

ISBN 9780735625129

EXAM 70-643

MCTS Self-Paced Training Kit: Configuring Windows Server 2008 Applications Infrastructure

J.C. Mackin, Anil Desai

ISBN 9780735625112

EXAM 70-646

MCITP Self-Paced Training Kit: Windows Server Administration

Ian McLean, Orin Thomas

ISBN 9780735625105

ALSO SEE

Windows Server 2008, Administrator's Pocket Consultant, Second Edition

William R. Stanek

ISBN 9780735627116

Windows Server 2008 Administrator's Companion

Charlie Russel, Sharon Crawford

ISBN 9780735625051

Windows Server 2008 Resource Kit

Microsoft MVPs with Windows Server Team

ISBN 9780735623613

microsoft.com/mspress

Microsoft® Press