One Identity Active Roles

Administration Guide

Copyright 2021 One Identity LLC.

ALL RIGHTS RESERVED.

This guide contains proprietary information protected by copyright. The software described in this guide is furnished under a software license or nondisclosure agreement. This software may be used or copied only in accordance with the terms of the applicable agreement. No part of this guide may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording for any purpose other than the purchaser's personal use without the written permission of One Identity LLC .

The information in this document is provided in connection with One Identity products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of One Identity LLC products. EXCEPT AS SET FORTH IN THE TERMS AND CONDITIONS AS SPECIFIED IN THE LICENSE AGREEMENT FOR THIS PRODUCT, ONE IDENTITY ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ONE IDENTITY BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ONE IDENTITY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. One Identity makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. One Identity does not make any commitment to update the information contained in this document.

If you have any questions regarding your potential use of this material, contact:

One Identity LLC.
Attn: LEGAL Dept
4 Polaris Way
Aliso Viejo, CA 92656

Refer to our Web site (http://www.OneIdentity.com) for regional and international office information.

Patents

One Identity is proud of our advanced technology. Patents and pending patents may apply to this product. For the most current information about applicable patents for this product, please visit our website at http://www.OneIdentity.com/legal/patents.aspx.

Trademarks

One Identity and the One Identity logo are trademarks and registered trademarks of One Identity LLC. in the U.S.A. and other countries. For a complete list of One Identity trademarks, please visit our website at www.OneIdentity.com/legal. All other trademarks are the property of their respective owners.

Legend

WARNING: A WARNING icon highlights a potential risk of bodily injury or property damage, for which industry-standard safety precautions are advised. This icon is often associated with electrical hazards related to hardware.

CAUTION: A CAUTION icon indicates potential damage to hardware or loss of data if instructions are not followed.

Active Roles Administration Guide
Updated - November 2021
Version - 7.5

Contents

Introduction **34**

About Active Roles **35**

Active Roles Main features 35

Technical overview 36

Presentation components 38

Active Roles console (MMC Interface) 38

Web Interface 38

Custom Interfaces 38

Active Roles ADSI Provider 38

Reporting 39

Service components 39

Data processing component 39

Configuration database 40

Audit trail 40

Network data sources 40

Security and administration elements 41

Access Templates for role-based administration 42

Policy Objects to enforce corporate rules 42

Managed Units to provide administrative views 43

Active Directory security management 45

Management of native security 45

Customization using ADSI Provider and script policies 46

Custom applications and user interfaces 46

Custom script policies 47

Dynamic groups 47

Workflows 48

Operation in multi-forest environments 49

Examples of use 51

Distributing administration 51

Integrating with other systems 51

Managing a multi-forest Active Directory design 52

Simplifying Active Directory structure 53
Handling organizational changes 53
User Account Management 54

Getting Started 55
Starting the Active Roles console 55
Delegating control to users for accessing MMC interface 56
Getting and using help 56
User Interface overview 57
Console tree 57
Details pane 58
Advanced pane 58
Active Roles Security and Links 59
Active Roles Policy 59
Native Security 59
Member Of and Members 60
Customizable Web Interface 60
Key features 60
Different interfaces for different roles 61
Role-based management of computer resources 62
View mode 63
Controlled objects 63
Using Managed Units 63
Setting up filter 64
Steps for sorting and filtering lists in the details pane 64
Finding objects 65
Steps for searching for a user, contact, or group 66
Steps for searching for a computer 67
Steps for searching for an Organizational Unit 68
Steps for using advanced search options 68
Steps for building a custom search 69
LDAP syntax 69
Search filter format 70
Operators 70
Wildcards 70
Special characters 71

Getting policy-related information 71
Performing Batch operations 74
Performing bulk operation 75
Performing bulk users password reset operation 76
Active Roles service account minimum permissions 77
Access to the Administration Service computer 77
Service publication in Active Directory 77
All script modules are executed under the security context of the Active Roles Service Account 78
Connecting to the Microsoft SQL database 79
Synchronizing native permissions to Active Directory 79

Rule-based Administrative Views 80
About Managed Units 80
How Managed Units work 81
Administering Managed Units 81
Creating a Managed Unit 82
Steps for creating a Managed Unit 88
Steps for modifying Managed Unit properties 90
Steps for modifying permission settings on a Managed Unit 91
Steps for modifying policy settings on a Managed Unit 92
Displaying members of a Managed Unit 93
Steps for displaying members of a Managed Unit 94
Adding or removing members from a Managed Unit 95
Steps for adding membership rules to a Managed Unit 96
Steps for removing membership rules from a Managed Unit 98
Steps for including a member to a Managed Unit 99
Steps for excluding a member from a Managed Unit 99
Steps for adding group members to a Managed Unit 100
Steps for removing group members from a Managed Unit 100
Copying a Managed Unit 101
Steps for copying a Managed Unit 101
Exporting and importing a Managed Unit 102
Renaming a Managed Unit 102
Steps for renaming a managed Unit 102
Deleting a Managed Unit 102

Steps for deleting a Managed Unit 103
Scenario: Implementing role-based administration across multiple OUs 103
Step 1: Creating the Managed Unit 104
Step 2: Adding users to the Managed Unit 104
Step 3: Preparing the Access Template 104
Step 4: Applying the Access Template 104
Deployment considerations 105
Managed Unit membership rules 105
Delegation of Managed Units 106
Working with Federated Authentication 107
Configuring Federated Authentication settings 108

Role-based Administration 110
Access Templates as administrative roles 110
How Access Templates work 111
Security synchronization 111
Access Template management tasks 112
Using predefined Access Templates 112
Active Directory 113
Azure 113
AD LDS (ADAM) 114
Computer Resources 114
Configuration 114
Exchange 115
Starling 115
User Interfaces 115
User Self-management 116
Creating an Access Template 116
Add Permission Entries wizard 118
Full Control access 119
Object access 120
Object property access 120
Creation/Deletion of child objects permission 121
Steps for creating an Access Template 123
Applying Access Templates 125
Steps for applying an Access Template 129

Managing Access Template links 132
Steps for managing Access Template links 134
Synchronizing permissions to Active Directory 137
Steps for synchronizing permissions to Active Directory 138
Managing Active Directory permission entries 139
Adding, modifying, or removing permissions 139
Steps for adding permissions to an Access Template 141
Steps for modifying permissions in an Access Template 142
Steps for removing permissions from an Access Template 143
Nesting Access Templates 144
Steps for managing nested Access Templates 146
Copying an Access Template 146
Steps for copying an Access Template 147
Exporting and importing Access Templates 147
Renaming an Access Template 148
Steps for renaming an Access Template 148
Deleting an Access Template 149
Steps for deleting an Access Template 149
Examples of use 150
Scenario 1: Implementing a Help Desk 150
Step 1: Preparing a Help Desk Access Template 151
Step 2: Creating a Help Desk group 151
Step 3: Applying the Help Desk Access Template 151
Scenario 2: Implementing Self-administration 152
Step 1: Preparing a self-administration Access Template 153
Step 2: Applying the self-administration Access Template 153
Deployment considerations 154
Delegation of Organizational Unit administration 155
Delegation of group administration 156
Delegation in a functional vs. hosted environment 157
Delegation in a functional environment 157
Delegation in a hosted environment 158
Windows claims-based Access Rules 159
Understanding Access Rules 160
Conditional Access Template links 160

Prerequisites for using Access Rules 160
Managing Windows claims 162
Enabling claim support 162
Claim Type management overview 163
Steps for managing Claim Types 167
Populating claim source attributes 168
Managing and applying Access Rules 169
Conditional expression editor 170
Applying an Access Rule 171
Steps for managing and applying Access Rules 172
Deploying an Access Rule (demonstration steps) 176
Step 1. Prerequisites 176
Step 2. Enable claim support 176
Step 3. Create Claim Type 177
Step 4. Create Access Rule 178
Step 5. Apply Access Rule 178

Rule-based AutoProvisioning and Deprovisioning 180
About Policy Objects 180
Provisioning Policy Objects 183
Deprovisioning Policy Objects 186
How Policy Objects work 189
Policy Object management tasks 190
Creating a Policy Object 190
Steps for creating a Policy Object 192
Adding, modifying, or removing policies 192
Steps for adding policies to a Policy Object 194
Steps for modifying policies in a Policy Object 194
Steps for removing policies from a Policy Object 195
Applying Policy Objects 195
Adding Managed Units or containers to policy scope 196
Adding Policy Objects to policy list for directory object 198
Steps for applying a Policy Object 199
Managing policy scope 200
Steps for managing Policy Object links 203
Steps for excluding an object from policy scope 205

Copying a Policy Object 206
Steps for copying a Policy Object 206
Renaming a Policy Object 207
Steps for renaming a Policy Object 207
Exporting and importing Policy Objects 208
Deleting a Policy Object 208
Steps for deleting a Policy Object 208
Policy configuration tasks 208
Property Generation and Validation 209
How this policy works 210
How to configure a Property Generation and Validation policy 212
User Logon Name Generation 233
How this policy works 234
How to configure a User Logon Name Generation policy 235
Steps for configuring a User Logon Name Generation policy 238
Scenario 1: Using uniqueness number 239
Scenario 2: Using multiple rules 242
Group Membership AutoProvisioning 245
How this policy works 245
How to configure a Group Membership AutoProvisioning policy 246
Steps for configuring a Group Membership AutoProvisioning policy 250
Scenario: Adding users to a specified group 252
Email Alias Generation 253
How this policy works 254
How to configure an E-mail Alias Generation policy 255
Steps for configuring an E-mail Alias Generation policy 257
Scenario: Generating e-mail alias based on user names 258
Exchange Mailbox AutoProvisioning 260
How this policy works 260
How to configure an Exchange Mailbox AutoProvisioning policy 261
Steps for configuring an Exchange Mailbox AutoProvisioning policy 262
Scenario: Mailbox store load balancing 262
Default creation options for Exchange mailbox 263
AutoProvisioning for SaaS products 264
How this policy works 264

Create Provisioning policy for Starling Connect 265
OneDrive Provisioning 266
How this policy works 266
Creating provisioning policy for OneDrive 266
Home Folder AutoProvisioning 267
How this policy works 267
How to configure a Home Folder AutoProvisioning policy 268
Steps for configuring a Home Folder AutoProvisioning policy 274
Using the built-in policy for home folder provisioning 275
Configuring the Home Folder Location Restriction policy 276
Scenario: Creating and assigning home folders 277
Script Execution 279
How this policy works 280
How to configure Script Execution policy 280
Steps for configuring a Script Execution policy 284
Scenario: Restricting group scope 285
Office 365 and Azure Tenant Selection 287
How this policy works 287
Configuring an O365 and Azure Tenant Selection policy 288
Applying a new policy 293
User Account Deprovisioning 295
How this policy works 295
How to configure a User Account Deprovisioning policy 296
Steps for configuring a User Account Deprovisioning policy 302
Scenario 1: Disabling and renaming the user account upon deprovisioning 304
Scenario 2: Managed Unit for deprovisioned user accounts 306
Office 365 Licenses Retention 308
How this policy works 308
How to configure Office 365 License Retention policy 309
Steps for configuring an Office 365 License Retention policy 310
Report on deprovisioning results 311
Group Membership Removal 311
How this policy works 311
How to configure a Group Membership Removal policy 312
Steps for configuring a Group Membership Removal policy 314

Scenario: Removing deprovisioned users from all groups 315
Exchange Mailbox Deprovisioning 316
How this policy works 317
How to configure an Exchange Mailbox Deprovisioning policy 318
Steps for configuring an Exchange Mailbox Deprovisioning policy 320
Scenario: Hide mailbox and forward e-mail to manager 321
Home Folder Deprovisioning 323
How this policy works 323
How to configure a Home Folder Deprovisioning policy 324
Steps for configuring a Home Folder Deprovisioning policy 325
Scenario: Removing access to home folder 326
User Account Relocation 327
How this policy works 327
How to configure a User Account Relocation policy 328
Steps for configuring a User Account Relocation policy 329
Scenario: Organizational Unit for deprovisioned user accounts 329
User Account Permanent Deletion 330
How this policy works 330
How to configure a User Account Permanent Deletion policy 331
Steps for configuring a User Account Permanent Deletion policy 332
Scenario: Deleting deprovisioned user accounts 333
Group Object Deprovisioning 333
How this policy works 334
How to configure a Group Object Deprovisioning policy 334
Steps for configuring a Group Object Deprovisioning policy 338
Scenario 1: Disabling and renaming the group upon deprovisioning 340
Scenario 2: Managed Unit for deprovisioned groups 341
Group Object Relocation 343
How this policy works 343
How to configure a Group Object Relocation policy 344
Steps for configuring a Group Object Relocation policy 345
Scenario: Organizational Unit for deprovisioned groups 345
Group Object Permanent Deletion 346
How this policy works 346
How to configure a Group Object Permanent Deletion policy 347

Steps for configuring a Group Object Permanent Deletion policy 348
Scenario: Deleting deprovisioned groups 349
Notification Distribution 349
How this policy works 350
How to configure a Notification Distribution policy 350
Configuring e-mail settings 351
Steps for configuring a Notification Distribution policy 352
Scenario: Sending deprovisioning notification 353
Report Distribution 355
How this policy works 355
How to configure a Report Distribution policy 355
Steps for configuring a Report Distribution policy 357
Scenario: Sending deprovisioning report 357
Deployment considerations 358
Checking for policy compliance 362
Steps to check for policy compliance 364
Deprovisioning users or groups 365
Default deprovisioning options 365
Delegating the Deprovision task 367
Using the Deprovision command 368
Report on deprovisioning results 368
Report contents 368
Restoring deprovisioned users or groups 375
Policy options to undo user deprovisioning 377
Delegating the task to undo deprovisioning 378
Using the Undo Deprovisioning command 378
Report on results of undo deprovisioning 379
Report contents 380
Container Deletion Prevention policy 384
Protecting objects from accidental deletion 385
Picture management rules 386
Policy extensions 388
Design elements 388
Policy type deployment 388
Policy type usage 389

Policy Type objects ... 389
Creating and managing custom policy types ... 390
Creating a Policy Type object ... 391
Changing an existing Policy Type object ... 392
Using Policy Type containers ... 394
Exporting policy types ... 394
Importing policy types ... 395
Configuring a policy of a custom type ... 395
Deleting a Policy Type object ... 396

Workflows ... 397
Understanding workflow ... 397
Key features and definitions ... 398
Workflow ... 398
Workflow definition ... 398
Workflow start conditions ... 398
Workflow instance ... 398
Workflow activity ... 399
Workflow Designer ... 399
Workflow engine ... 399
E-mail Notifications ... 399
About workflow processes ... 399
Workflow processing overview ... 401
About start conditions ... 402
Workflow activities overview ... 403
Approval activity ... 403
Approvers and escalation ... 404
Request for information ... 405
Customization ... 405
Notification ... 407
Notification activity ... 408
Notification recipients ... 408
Notification message ... 409
Web Interface address ... 409
E-mail server ... 409
Script activity ... 409

Notification 410
Error handling 410
If-Else activity 410
If-Else branch conditions 411
Error handling 413
Stop/Break activity 413
Add Report Section activity 414
Search activity 414
Search scenario 415
Object type 415
Search scope 416
Search options 419
Search for inactive accounts 423
Search filter 423
Notification 427
Error handling 427
"Run as" options 428
Additional settings 428
Stop Search activity 428
CRUD activities 428
"Create" activity 429
"Update" activity 432
"Add to group" activity 433
"Remove from group" activity 434
"Move" activity 435
"Deprovision" activity 436
"Undo deprovision" activity 436
"Delete" activity 437
Activity target 437
Notification 439
Error handling 439
"Run as" options 439
Additional settings 440
Save Object Properties activity 441
Retrieving saved properties 442

Modify Requested Changes activity 443
Configuring a workflow 445
Creating a workflow definition 445
Configuring workflow start conditions 446
Operation conditions 446
Initiator conditions 447
Filtering conditions 448
Configuring workflow parameters 453
Adding activities to a workflow 456
Configuring an Approval activity 457
Configure approvers 458
Configure escalation 460
Configure request for additional information 461
Configure request for review 462
Customize the header of the approval task page 462
Customize approval action buttons 463
Configuring a Notification activity 464
Events, recipients and messages 465
Active Roles Web Interface 466
E-mail server settings 467
Configuring a Script activity 467
Configuring an If-Else activity 469
Steps to configure error handling 470
Configuring conditions for an If-Else branch 470
Configuring a Stop/Break activity 474
Configuring an Add Report Section activity 474
Configuring a Search activity 476
Configure scope and filter 477
Configure notification 487
Configure error handling 487
Configure "run as" options 488
Configure additional settings 488
Configuring CRUD activities 489
"Create" activity 490
"Update" activity 493

"Add to group" activity 495
"Remove from group" activity 497
"Move" activity 498
"Deprovision" activity 500
"Undo deprovision" activity 501
"Delete" activity 502
Configuring notification 503
Configuring error handling 504
Configuring "run-as" options 504
Configuring additional settings 505
Configuring a Save Object Properties activity 506
Configuring a Modify Requested Changes activity 507
Enabling or disabling an activity 510
Enabling or disabling a workflow 510
Using the initialization script 511
Example: Approval workflow 512
Definition of terms 513
Approval 513
Approval rule (Approval activity) 513
Approval task 513
Approver 513
Initiator (requestor) 514
Notification 514
Operation 514
Operation target object 514
How it works 514
Action: Approve 515
Action: Reject 517
Multiple approvers 518
Multiple tasks 518
Creating and configuring an approval workflow 520
Creating a workflow definition 521
Specifying workflow start conditions 521
Specifying approvers 522
Configuring notification 522

Email based approval 523
Integration with Microsoft Outlook 524
Software and configuration requirements 524
Integration with non-Outlook e-mail clients 525
Software and configuration requirements 525
E-mail transport via Exchange Web Services 526
Configuration settings 526
Steps to configure the use of Exchange Web Services 527
Automation workflow 528
Automation workflow options and start conditions 529
Run the workflow on a schedule 529
Allow the workflow to be run on demand 530
"Run as" options 530
Additional settings 531
Parameters 531
Initialization script 532
Using automation workflow 533
Creating an automation workflow definition 533
Configuring start conditions for an automation workflow 534
Adding activities to an automation workflow 535
Running an automation workflow on demand 535
Viewing run history of an automation workflow 536
Terminating a running automation workflow 537
Disabling an automation workflow from running 537
Re-enabling an automation workflow to run 538
Delegating automation workflow tasks 538
Sample Azure Hybrid Migration 542
Managing Remote Mailbox 543
Office 365 automation workflow 544
Creating Office 365 shared mailbox 546
Enabling Azure Roles 546
Activity extensions 547
Design elements 547
Activity type deployment 547
Activity type usage 548

Policy Type objects ... 548
Creating and managing custom activity types ... 549
Creating a Policy Type object ... 550
Changing an existing Policy Type object ... 551
Using Policy Type containers ... 553
Exporting activity types ... 554
Importing activity types ... 554
Configuring an activity of a custom type ... 555
Deleting a Policy Type object ... 555

Temporal Group Memberships ... 557
Understanding temporal group memberships ... 557
Using temporal group memberships ... 559
Adding temporal members ... 560
Viewing temporal members ... 561
Rescheduling temporal group memberships ... 561
Removing temporal members ... 562

Group Family ... 564
Understanding Group Family ... 564
Design overview ... 565
How it works ... 566
Cross-domain Group Family ... 567
Group Family policy options ... 568
Creating a Group Family ... 569
Start the New Group Family wizard ... 569
Name the Group Family ... 569
Grouping Options ... 570
Location of managed objects ... 571
Selection of managed objects ... 572
Group-by properties ... 573
About multi-valued group-by properties ... 574
Capture existing groups manually ... 575
Group naming rule ... 576
Entry type: Group-by Property ... 578
Separate rule for each naming property ... 579

Group type and scope 579
Location of groups 580
Exchange-related settings 581
Group Family scheduling 583
Steps for creating a Group Family 583
Administering Group Family 586
Controlled groups 587
General tab 587
Controlled Groups tab 588
Group creation-related rules 590
Groupings tab 590
Schedule tab 591
Action Summary tab 591
Action summary log 592
Steps for administering a Group Family 592
Scenario: Departmental Group Family 594

Dynamic Groups 596
Understanding dynamic groups 596
Cross-domain membership 597
Dynamic groups policy options 597
Managing dynamic groups 598
Converting a basic group to a dynamic group 598
Displaying the members of a dynamic group 601
Adding a membership rule to a dynamic group 601
Removing a membership rule from a dynamic group 603
Converting a dynamic group to a basic group 604
Modifying, renaming, or deleting a dynamic group 604
Scenario: Automatically moving users between groups 604
Step 1: Creating the groups 605
Step 2: Configuring the membership rules 605

Active Roles Reporting 606
Introduction 606
Collector to prepare data for reports 607
Starting the Active Roles Collector wizard 608

Collecting data from the network 608
Steps for collecting data from the network 610
Processing gathered events 612
Steps for processing gathered events 613
Importing events from an earlier database version 614
Deploying reports to the Report Server 614
Working with reports 615
Configuring the data source 615
Generating and viewing a report 616
Contents of the Active Roles Report Pack 617
Active Directory Assessment/Domains/ 617
Active Directory Assessment/Users/Account Information/ 617
Active Directory Assessment/Users/Exchange/ 618
Active Directory Assessment/Users/Obsolete Accounts/ 618
Active Directory Assessment/Users/Miscellaneous Information/ 619
Active Directory Assessment/Groups/ 619
Active Directory Assessment/Group Membership/ 619
Active Directory Assessment/Organizational Units/ 620
Active Directory Assessment/Other Directory Objects/ 620
Active Directory Assessment/Potential Issues/ 621
Active Roles Tracking Log/Active Directory Management/ 621
Active Roles Tracking Log/Dashboard/ 621
Active Roles Tracking Log/Active Roles Events/ 622
Active Roles Tracking Log/Active Roles Configuration Changes/ 622
Active Roles Tracking Log/Active Roles Workflow/ 622
Administrative Roles/ 623
Managed Units/ 624
Policy Objects/ 625
Policy Compliance/ 626

Management History 627
Understanding Management History 627
Considerations and best practices 628
Management History configuration 630
Change-tracking policy 630
Change Tracking log configuration 631

Replication of Management History data 632
Replication is not yet configured 633
Replication is already configured 633
Re-configuring replication of Management History data 634
Centralized Management History storage 635
Importing data to the new Management History database 635
Viewing change history 637
Workflow activity report sections 638
“Approval” activity 639
“Script” activity 641
“Stop/Break” activity 641
“Add Report Section” activity 641
“Create” activity 641
“Update” activity 641
“Add to group” activity 642
“Remove from group” activity 642
“Move” activity 643
“Deprovision” activity 643
“Undo deprovision” activity 643
“Delete” activity 644
Policy report items 644
Report section: Executing the 'User Logon Name Generation' policy 645
Report section: Executing the 'E-mail Alias Generation' policy 645
Report section: Executing the 'Exchange Mailbox AutoProvisioning' policy 645
Report section: Executing the 'Group Membership AutoProvisioning' policy 646
Report section: Executing the 'Home Folder AutoProvisioning' policy 646
Report section: Executing the 'Property Generation and Validation' policy 647
Report section: Executing policy script 'name' 648
Active Roles internal policy report items 648
Report section: Creating user mailbox 648
Report section: Creating linked mailbox 649
Report section: Creating equipment mailbox 649
Report section: Creating room mailbox 650
Report section: Creating shared mailbox 650
Report section: Moving mailbox 651

Report section: Deleting mailbox 651
Report section: Removing Exchange attributes 651
Report section: Enabling mailbox for Unified Messaging 651
Report section: Disabling Unified Messaging for mailbox 652
Report section: Resetting Unified Messaging PIN 652
Report section: Establishing e-mail address for group 652
Report section: Creating query-based distribution group 653
Report section: Establishing e-mail address for user 653
Report section: Establishing e-mail address for contact 653
Report section: Deleting e-mail address for group 654
Report section: Deleting e-mail address for user 654
Report section: Deleting e-mail address for contact 654
Report section: Converting user mailbox to linked mailbox 655
Report section: Converting linked mailbox to user mailbox 655
Examining user activity 655

Entitlement Profile 657
Understanding entitlement profile 657
About entitlement profile specifiers 658
Entitlement type 658
Entitlement rules 659
Resource display 660
About entitlement profile build process 660
Entitlement profile configuration 662
Creating entitlement profile specifiers 662
Changing entitlement profile specifiers 665
Pre-defined specifiers 667
Viewing entitlement profile 672
Authorizing access to entitlement profile 675

Recycle Bin 677
Understanding Recycle Bin 677
Finding and listing deleted objects 678
Searching the Deleted Objects container 678
Searching for objects deleted from a certain OU or MU 679
Restoring a deleted object 679

Delegating operations on deleted objects 681
Applying policy or workflow rules 682

AD LDS Data Management 684
Registering an AD LDS instance 684
Managing AD LDS objects 686
Adding an AD LDS user to the directory 687
Adding an AD LDS group to the directory 687
Adding or removing members from an AD LDS group 688
Disabling or enabling an AD LDS user account 688
Setting or modifying the password of an AD LDS user 689
Adding an organizational unit to the directory 690
Adding an AD LDS proxy object (user proxy) 690
Configuring Active Roles for AD LDS 691
Configuring Managed Units to include AD LDS objects 692
Viewing or setting permissions on AD LDS objects 693
Viewing or setting policies on AD LDS objects 694

One Identity Starling Join and Configuration through Active Roles 696
Configure Join to Starling 696
Prerequisites to configure One Identity Starling 697
Configuring Active Roles to join One Identity Starling 698
Disconnecting One Identity Starling from Active Roles 699

One Identity Starling Two-factor Authentication for Active Roles 700
Starling Two-Factor Authentication User Access template 700
ARS 2FA Users group 700
Pre-requisites to use One Identity Starling 2FA 701
Allowing two-factor authentication for Active Roles users 701
Steps to create ARS 2FA Users group manually 701
Registering to One Identity Starling 2FA 702
Logging in to Web interface through 2FA authentication 702
Logging in to MMC interface through 2FA authentication 703
Disallowing two-factor authentication for Active Roles users 703
Disabling or Enabling Starling 2FA Users from Configuration Center 703

Managing One Identity Starling Connect 704
Viewing Starling Connect settings in Active Roles Configuration Center 704

Create Provisioning policy for Starling Connect 705
Provision a new SaaS user using the Web interface 706
Provision an existing Active Roles user for SaaS products 707
Update the SaaS product user properties 708
Delete the SaaS product user 708
Deprovision an existing Active Roles user for SaaS products 708
Notifications for Starling operations 709
Configuring notification settings 710
SCIM attribute mapping with Active Directory 711

Azure AD, Office 365, and Exchange Online Management 714

Configuring Active Roles to Manage Hybrid AD Objects 716
Configuring Active Roles to manage Azure AD using the GUI 716
Configuring a new Azure tenant and consenting Active Roles as an Azure application 717
Importing an Azure tenant and consenting Active Roles as an Azure application 723
Viewing or modifying the Azure AD tenant type 727
Enabling OneDrive in an Azure tenant 729
Removing an Azure AD tenant 736
View Azure Health for Azure AD tenants and applications 738
View Azure Licenses Report 739
View Office 365 Roles Report 739
Azure Tenant Association 739
Configuring Active Roles to manage Hybrid AD using Management Shell 740
Adding an Azure AD tenant 740
Add an Azure AD Application 744
Active Roles Configuration steps to manage Hybrid AD objects 747
Configuring the Azure - Default Rules to Generate Properties policy 747
Active Roles Configuration to synchronize existing Azure AD objects to Active Roles 748
Configuring Sync Workflow to back-synchronize Azure AD Objects to Active Roles automatically using the Active Roles Synchronization Service Console 749
Configuring Sync Workflow to back-synchronize Azure AD Objects to Active Roles manually 751
Changes to Azure O365 Policies in Active Roles after 7.4.1 754
Managing Hybrid AD Users 755
Azure AD user management tasks using Web interface 755

Create a new Azure AD user 755
View or update the Azure AD user properties 757
Modify the Azure AD user Manager 757
Disable or re-enable an Azure AD user 758
Deprovision or undo deprovision of a Azure AD user 759
Add or remove a Azure AD user from a group 759
View the Change History and User Activity for an Azure AD user 760
Delete an Azure AD user 761
Hybrid User Management tasks using web interface 762
Create a new Hybrid user using web interface 762
Migrate an Exchange on-premise user to a Hybrid user 763
View or modify the Exchange Online properties of an Office 365 User 764
Azure AD user management tasks using Management Shell interface 770
Create a new Azure AD user 771
Update the Azure AD user properties 771
View the Azure AD user properties 771
Delete an Azure AD user 772
Office 365 license management for hybrid environment users 772
Assign Office 365 licenses to new hybrid users 772
Assign Office 365 licenses to existing hybrid users 773
Modify or remove Office 365 licenses assigned to hybrid users 774
Update Office 365 licenses display names 774
Unified provisioning policy for Azure O365 Tenant Selection, Office 365 License Selection, and Office 365 Roles Selection, and OneDrive provisioning 775
How this policy works 775
Configuring an O365 and Azure Tenant Selection policy 775
Applying a new policy 780
Office 365 roles management for hybrid environment users 782
Assign Office 365 roles to existing hybrid users 782
Modify Office 365 roles assigned to hybrid users 783
Managing Office 365 Contacts 784
Office 365 contact management tasks using Web interface 784
Create a new Office 365 contact 784
Modify the Office 365 Contact Properties 785
View the Change History for an Office 365 contact 785

Delete an Office 365 contact 786
Managing Hybrid AD Groups 786
Azure AD group management tasks using the Web interface 786
Create an Azure AD group 787
View or modify Azure AD group properties 788
Add or remove members to an Azure AD group 788
View the Change History for an Azure AD Group 789
Delete an Azure AD group 789
Azure AD Group management tasks using Management Shell interface 790
Create a new Azure AD Group 790
Update the Azure AD Group properties 791
Delete an Azure AD group 791
Add a member to Azure AD Group 791
Remove a member from Azure AD Group 791
Managing Office 365 Groups 791
Configuring O365 Groups with the Web Interface 792
Creating an O365 Group with the Web Interface 792
Modifying an O365 Group with the Web Interface 794
Adding or removing owners from an O365 Group with the Web Interface 796
Adding or removing members from an O365 Group with the Web Interface 797
Viewing the members of a dynamic O365 Group with the Web Interface 799
Viewing the change history of an O365 Group in the Web Interface 799
Deleting an O365 Group with the Web Interface 801
Office 365 Group management tasks using Management Shell interface 801
Create a new Office 365 Group 801
Update the Office 365 Group properties 802
Delete an Office 365 group 802
Adding members to an Office 365 Group with the Management Shell 802
Get a member from Office 365 Group 802
Get group from Office 365 Group 802
Removing members from an Office 365 Group with the Management Shell 803
Scheduling an O365 group synchronization task 803
Managing Azure Security Groups 804
Creating an Azure Security Group with the Web Interface 804
Modifying an Azure Security Group with the Web Interface 806

Adding or removing owners from an Azure Security Group with the Web Interface 808
Adding or removing members from an Azure Security Group with the Web Interface 809
Viewing the members of a dynamic Azure Security Group with the Web Interface 810
Viewing the change history of an Azure Security Group in the Web Interface 811
Deleting an Azure Security Group with the Web Interface 812
Managing cloud-only Azure users 813
Viewing cloud-only Azure user 813
Creating a new cloud-only Azure user 814
Viewing or modifying cloud-only Azure user properties 814
Configuring Microsoft OneDrive for cloud-only Azure users 815
Disabling cloud-only Azure user 815
Viewing and modifying Exchange Online properties 816
Resetting password for a cloud-only Azure user 817
Renaming Azure user 817
Viewing Azure membership 818
Viewing change history 818
Deleting an Azure user account 818
Managing cloud-only Azure guest users 819
Inviting an Azure guest user 820
Viewing Azure guest users 824
Disabling or Enabling an Azure guest user 824
Revoking the session of an Azure guest user 825
Resending the invitation to an Azure guest user 826
Renaming an Azure guest user 827
Viewing and updating the properties of an Azure guest user 828
Configuring the Identity settings of an Azure guest user 830
Configuring the Settings of an Azure guest user 832
Configuring the Job Info settings of an Azure guest user 833
Configuring the Contact Info settings of an Azure guest user 835
Configuring the Licenses settings of an Azure guest user 837
Configuring the O365 Admin Roles settings of an Azure guest user 838
Viewing or updating the Exchange Online properties of an Azure guest user 839
Configuring the Mail Flow Settings of an Azure guest user 840
Configuring the Delegation settings of an Azure guest user 841

Configuring Email Address settings for Azure guest users 842
Configuring Mailbox Features for Azure guest users 845
Configuring Mailbox Settings for Azure guest users 847
Deleting an Azure guest user 848
Configuring the O365 Group membership of an Azure guest user 849
Viewing the change history of an Azure guest user 851
Managing cloud-only Azure contacts 852
View cloud only Azure contacts 852
Create new cloud only Azure contacts 853
View or modify Azure contacts properties 853
Renaming Azure cloud contacts 854
Viewing and modifying Exchange Online properties 854
Viewing change history 855
Deleting an Azure contact 856
Changes to Active Roles policies for cloud-only Azure objects 856
Managing room mailboxes 857
Creating a new room mailbox 857
Viewing or modifying a room mailbox 859
Deleting a room mailbox 861

Managing Active Roles 863
Connecting to the Administration Service 864
Delegating control to users for accessing MMC interface 865
Steps for connecting to the Administration Service 866
Adding and removing managed domains 867
Steps for adding or removing a managed domain 868
Using unmanaged domains 869
Configuring an unmanaged domain 870
Evaluating product usage 871
Viewing product usage statistics 871
Delegating access to the managed object statistics 872
Scheduled task to count managed objects 873
Managed scope to control product usage 873
Voluntary thresholds for the managed object count 874
Installation label 875
Creating and using virtual attributes 876

Scenario: Implementing a Birthday attribute 877
Examining client sessions 880
Monitoring performance 881
Customizing the console 881
"Other Properties" page in object creation wizard 882
"Other Properties" tab in the Properties dialog box 883
Customizing object display names 885
Using Configuration Center 885
Configuration Center design elements 886
Configuring a local or remote Active Roles instance 887
Running Configuration Center 888
Pre-requisites to run the Configuration Center 888
Tasks you can perform in Configuration Center 889
Initial configuration tasks 890
Administration Service management tasks 891
Web Interface management tasks 897
Starling join configuration task 902
MMC interface access management 902
Logging management tasks 903
Solution Intelligence 904
Configuring gMSA as an Active Roles Service account 905
Changing the Active Roles Admin account 907
Enabling or disabling diagnostic logs 908
Active Roles Log Viewer 910
Using Log Viewer 910

SQL Server Replication 912
Replication terminology 912
Replication 913
Publisher 913
Subscribers 913
Distributor 913
Replication group 913
Standalone database server 914
Articles and publications 914
SQL Server Agent 914

Replication Agents 914
Snapshot Agent 914
Merge Agent 914
Understanding the Replication model 915
Replication group management 916
Promote 916
Add 916
Delete 916
Demote 916
Data synchronization and conflict resolution 917
SQL Server-related permissions 917
Configuring SQL Server 917
Configuring replication 918
The replication group 918
Creating a replication group 919
Adding members to a replication group 919
Steps for adding members to a replication group 921
Removing members from a replication group 922
Steps for removing members from a replication group 923
Monitoring replication 923
Using AlwaysOn Availability Groups 924
Availability Group setup in Active Roles 925
Configuring the database connection to use the listener 925
Using database mirroring 927
Role switching 927
Database Mirroring setup in Active Roles 928
Best practices 929
Viewing replication settings 930
Replication Agent schedule 931
Monitoring replication 932
Viewing database connection settings 933
Modifying database connection settings 933
Changing the service account 934
Changing the SQL Server Agent logon account 934
Modifying Replication Agent credentials 935

Windows authentication 935
Replication Agent connection to Subscriber 936
SQL Server authentication 937
Replication Agent connection to Publisher 937
Replication Agent connection to Subscriber 937
Moving the Publisher role 938
Recovering replication if Publisher is not available 940
Troubleshooting Replication failures 941
Replication Agent malfunction 941
Symptoms 941
Solution 941
Replication Agent authentication problems 942
Symptoms 942
Solution 942
SQL Server identification problems 943
Symptoms 943
Solution 943

Appendix A: Using regular expressions 946
Examples of regular expressions 948
Order of precedence 949

Appendix B: Administrative Template 950
Active Roles snap-in settings 950
Administration Service auto-connect settings 952
'Allowed Servers for Auto-connect' setting 953
'Disallowed Servers for Auto-connect' setting 954
'Additional Servers for Auto-connect' setting 954
Loading the Administrative Template 955

Appendix C: Communication ports 956
Access to the managed environment 956
Access to DNS servers 956
Access to domain controllers 956
Access to Exchange servers 957
Computer resource management 957
Computer restart 957

Home folder provisioning and deprovisioning 958
Access to SMTP server for e-mail integration 958
Access to AD LDS instances 958
Access to SMTP server for e-mail integration 958
Access to Active Roles Administration Service 958
Access to Web Interface 959

Appendix D: Active Roles and supported Azure environments 960
Non-federated 960
Synchronized identity 961
Federated 961
Azure Object Management supported in various Azure environments 962
Azure Object Management in Non-Federated environment 963
Azure Object Management in Federated and Synchronized Identity environments 964

Appendix E: Enabling Federated Authentication 968
Configuring the domain service account for Federated Authentication 968
Updating Local Policies 968
Creating SPN entries for the domain service account 969
Enabling delegation for Federated Authentication 970
Redistributable STS 971
Installing Redistributable STS 972
Configuring Redistributable STS in RStsApiAdmin 972
Configuring Redistributable STS in Configuration Center 973
Examples of configuring identity providers 974

Appendix F: Active Roles integration with other One Identity and Quest products 975

Appendix G: Active Roles integration with Duo MFA 979
Configuring Duo for Active Roles 979
Integrating Active Roles with Duo 979
Validating Duo Multi-Factor Authentication 981

Appendix H: Active Roles integration with Okta MFA 982
Configuring Okta for Active Roles 982
Integrating Active Roles with Okta 982
Validating Okta Multi-Factor Authentication 983

About us **985**
Contacting us 985
Technical support resources 985

1

Introduction

The *Active Roles Administration Guide* is designed for individuals who are responsible for creating and maintaining Active Roles' administrative structure. This document provides conceptual information about the product, and includes instructions for deploying a secure, distributed administrative structure that combines administrative policy enforcement, role-based delegation of administration, and flexible administrative views.

This guide also provides information for performing administrative tasks using the Active Roles web interface for Azure Active Directory and Office 365. The document includes instructions to help delegated administrators and help-desk operators perform day-today Azure AD administrative activities.

Active Roles facilitates administrators to configure and monitor Active Roles replication using Microsoft SQL Server tools. This guide details the SQL Server agents used during replication, accounts and logins used to access SQL Server, and strategies for monitoring and troubleshooting replication.

The *Active Roles Administration Guide* is supplemented with the *Active Roles User Guide* that provides information about the Active Roles console user interface, and includes instructions to help delegated administrators perform day-to-day administrative activities using the Active Roles console.

2

About Active Roles

Active Roles (formerly known as ActiveRoles®), delivers a reliable, policy-based administration and provisioning solution, allowing enterprises to fully benefit from Active Directory and Microsoft Exchange deployment.

One of the most valuable features of the product is the ability to automate provisioning tasks on directory objects in compliance with corporate administrative policies in corporate Active Directory and Exchange environments.

Active Roles provides consistent enforcement of corporate policies, a role-based administrative model, and flexible, rule-based administrative views, creating a reliable and secure environment for distributed administration and account provisioning.

NOTE: For information on the Active Roles features see the latest *Active Roles What's New Guide*.

Active Roles Main features

Before proceeding with the upgrade ensure to perform a database backup.

Active Roles (formerly known as ActiveRoles®) provides out-of-the-box user and group account management, strictly enforced administrator-based role security, day-to-day identity administration and built-in auditing and reporting for Active Directory and Azure Active Directory (AD) environments. The following features and capabilities make Active Roles a practical solution for secure management of objects in Active Directory and Active Directory-joined systems:

- **Secure access** Acts as a virtual firewall around Active Directory, enabling you to control access through delegation using a least privilege model. Based on defined administrative policies and associated permissions generates and strictly enforces access rules, eliminating the errors and inconsistencies common with native approaches to AD management. Plus, robust and personalized approval procedures establish an IT process and oversight consistent with business requirements, with responsibility chains that complement the automated management of directory data.

- **Automate object creation** Automates a wide variety of tasks, including:
  - Creating user, groups, and contacts in Active Directory and Azure AD
  - Creating mailboxes on Exchange Server and assigning licenses in Office 365
  - Managing on-premise Exchange and Exchange Online properties

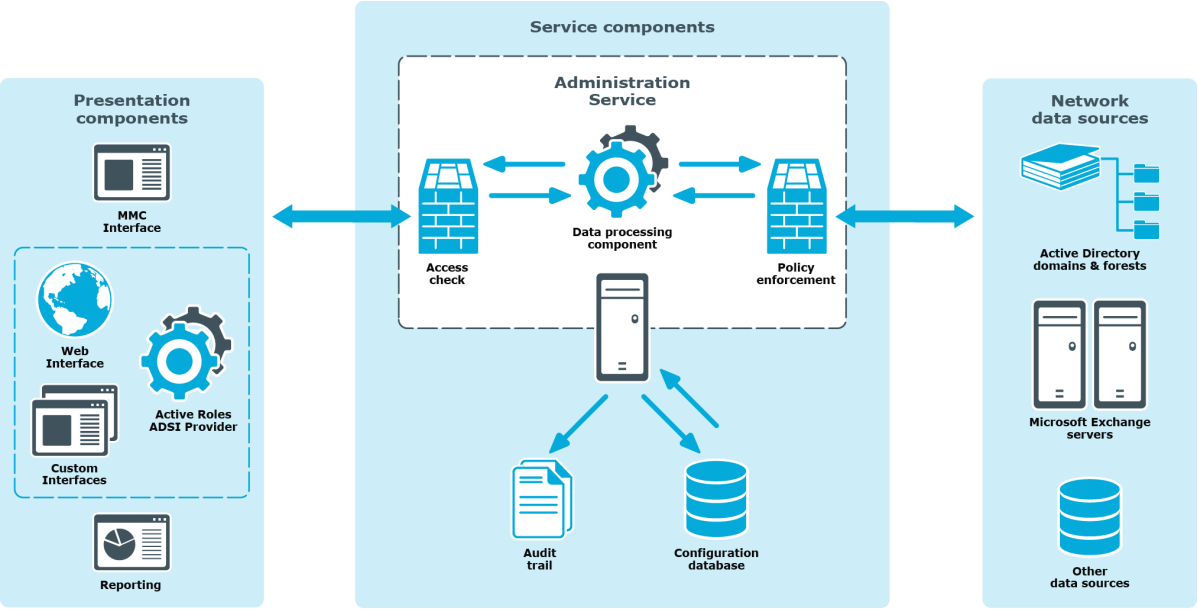
  Active Roles also automates the process of reassigning and removing user access rights in AD and AD-joined systems (including user and group de-provisioning) to ensure an efficient and secure administrative process over the user and group lifetimes. When a user's access needs to be changed or removed, updates are made automatically in Active Directory, Azure AD, Exchange, Exchange Online, SharePoint, Skype for Business, and Windows, as well as any AD-joined systems such as Unix, Linux, and Mac OS X.
- **Day-to-day directory management** Simplifies management of:
  - Exchange recipients, including mailbox assignment, creation, movement, deletion, permissions, and distribution list management
  - Groups
  - Computers, including shares, printers, local users and groups
  - Active Directory, Azure AD, Exchange Online and AD LDS

  Active Roles also includes intuitive interfaces for improving day-to-day administration and help desk operations via both an MMC snap-in and a Web interface.
- **Manage users, groups, and contacts in a hosted environment** Provides Synchronization Service to operate in hosted environments where accounts from client AD domains are synchronized with host domains. Active Roles enables user, group, and contact management from the client domain to the hosted domain, while also synchronizing attributes and passwords.
- **Consolidate management points through integration** Complements your existing technology and identity and access management strategy. Simplifies and consolidates management points by ensuring easy integration with many One Identity products and Quest products, including One Identity Manager, Authentication Services, Defender, ChangeAuditor, and GPO Admin. Active Roles also automates and extends the capabilities of PowerShell, ADSI, SPML and customizable Web interfaces.

Technical overview

Active Roles divides the workload of directory administration and provisioning into three functional layers—presentation components, service components, and network data sources.

Figure 1: Active Roles Components



The presentation components include client interfaces for the Windows platform and the Web, which allow regular users to perform a precisely defined set of administrative activities. The reporting solution facilitates automated generation of reports on management activities.

The service components constitute a secure layer between administrators and managed data sources. This layer ensures consistent policy enforcement, provides advanced automation capabilities, and enables the integration of business processes for administration of Active Directory, Microsoft Exchange, and other corporate data sources.

The Administration Database stores information about all permission and policy settings, and other data related to the Active Roles configuration.

On a very high level, the Active Roles components work together as follows to manipulate directory data:

1. An administrator uses the MMC interface or Web interface to access Active Roles.
2. The administrator submits an operation request, such as a query or data change to the Administration Service.
3. On receipt of the operation request, the Administration Service checks whether the administrator has sufficient permissions to perform the requested operation (access check).
4. The Administration Service ensures that the requested operation does not violate the corporate policies (policy enforcement).
5. The Administration Service performs all actions required by the corporate policies, before committing the request (policy enforcement).
6. The Administration Service issues operating system function calls to perform the requested operation on network data sources.

7. The Administration Service performs all related actions required by the corporate policies, after the request is processed by the operating system (policy enforcement).
8. The Administration Service generates an audit trail that includes records about all operations performed or attempted with Active Roles. Directory-change tracking reports are based on the audit trail.

Let us examine the three component layers.

Presentation components

The presentation components include user interfaces to serve a variety of needs. The user interfaces accept commands, display communication, and give results in a clear, concise fashion.

Active Roles console (MMC Interface)

The Active Roles console, also referred to as the MMC Interface, is a comprehensive administrative tool for managing Active Directory and Microsoft Exchange. It enables you to specify administrative roles and delegate control, define administrative policies and automation scripts, easily find directory objects, and perform administrative tasks.

Web Interface

Via the Web interface, intranet users with sufficient administrative rights can connect to Active Roles to perform basic administrative tasks, such as modifying user data or adding users to groups. The Web interface provides departmental and help-desk personnel with the administrative capabilities they need.

Custom Interfaces

In addition to the MMC and Web interfaces, Active Roles enables the development of custom interfaces that use the Active Roles ADSI Provider to access the features of Active Roles. Administrators familiar with scripting and programming can create custom interfaces to meet specific needs of the network administration.

Active Roles ADSI Provider

The Active Roles ADSI Provider operates as part of Presentation Components to enable custom user interfaces and applications to access Active Directory services through Active

Roles. The Active Roles ADSI Provider translates clients' requests into DCOM calls and interacts with the Administration Service.

The Active Roles ADSI Provider allows custom scripts and applications, such as Web-based applications, to communicate with Active Directory, while taking full advantage of the security, workflow integration and reporting benefits of Active Roles. For example, using the Active Roles ADSI Provider, Web-based pages can be created such that user property modifications made by help-desk operators are restricted by the corporate rules enforced by Active Roles.

Reporting

Active Roles offers comprehensive reporting to monitor administrative actions, corporate policy compliance, and the state of directory objects. The Active Roles reporting solution includes Data Collector and Report Pack.

Report Pack provides report definitions for creating reports based on the data gathered by Data Collector. Active Roles comes with an extensive suite of report definitions that cover all administrative actions available in this product.

Report Pack is deployed on Microsoft SQL Server Reporting Services (SSRS). You can use the tools included with SSRS to view, save, print, publish, and schedule Active Roles reports.

Data Collector is used to gather data required for reporting. The Data Collector Wizard allows you to configure and schedule data collection jobs.

Once configured, Data Collector retrieves data from various sources, accessing them via the Active Roles Administration Service, and stores the data in a SQL Server database. Data Collector also provides a means for managing the gathered data, including the ability to export or delete obsolete data.

Service components

At the core of Active Roles lies the Administration Service. It features advanced delegation capabilities and ensures the reliable enforcement of administrative policies that keep data current and accurate. The Administration Service acts as a bridge between the presentation components and network data sources. In large networks, multiple Administration Services can be deployed to improve performance and ensure fault tolerance.

Data processing component

The data processing component accepts administrative requests and validates them by checking permissions and rules stored in the Administration Database. This component manages the network data sources, retrieving or changing the appropriate network object data based on administrative requests and policy definitions.

The data processing component operates as a secure service. It logs on with domain user accounts having sufficient privileges to access the domains registered with Active Roles (managed domains). The access to the managed domains is limited by the access rights of those user accounts.

Configuration database

The Administration Service uses the configuration database to store configuration data. The configuration data includes definitions of objects specific to Active Roles, assignments of administrative roles and policies, and procedures used to enforce policies. The configuration database is only used to store Active Roles configuration data. It does not store copies of the objects that reside in the managed data sources, nor is it used as an object data cache.

Active Roles uses Microsoft SQL Server to host the configuration database. The replication capabilities of SQL Server facilitate implementation of multiple equivalent configuration databases used by different Administration Services.

Active Roles now supports database configuration on on-premises databases and Azure SQL databases. Azure SQL database variants, such as, Azure SQL database, Azure SQL Managed instance, and Azure SQL Elastic pool can be configured in Active Roles.

NOTE: Active Roles supports database configuration over an encrypted SQL Server configuration. For more information see KB article

https://support.oneidentity.com/kb/262157/is-sql-server-encryption-supported-

Audit trail

The data processing component provides a complete audit trail by creating records in the event log on the computer running the Administration Service. The log shows all actions performed and by whom, including actions that were not permitted. The log entries display the success or failure of each action, as well as which attributes were changed.

Network data sources

Through the Administration Service, Active Roles accesses and controls the object data stored in the following data sources:

- **Active Directory domains & forests** Provides the directory object information in Active Directory domains.
- **Microsoft Exchange servers** Provides information about mailboxes maintained by Microsoft Exchange.
- **Azure AD** Provides information about users in Azure Active Directory.
- **Microsoft Office 365** Provides information about users in Office 365.

- **Exchange Online** Provides information about users in Exchange Online.
- **Other data sources** Provides information about objects that exist outside of Active Directory. This includes information from corporate databases, such as human resources databases, and information about computer resources, such as services, printers, and network file shares.

Active Roles is designed to help with the use and management of these data sources. Directory administrators can define and enforce business rules and policies to ensure that the data in the managed data sources remains current and accurate.

With Active Roles, you can utilize the information stores from a wide variety of data sources in your network, such as human resource data or inventories. You can use scripting to integrate these important data sources. This reduces the duplication of work, reduces data pollution, and allows for the validation of information that is often stored in more than one database.

Active Roles makes it possible for a custom script to receive control upon a request to perform an administrative operation, such as object creation, modification, or deletion. Custom scripts can be invoked through Policy Objects, which Active Roles uses to enforce corporate rules. For example, you could implement a Policy Object containing a custom script that will receive control whenever Active Roles is requested to create a user object in a certain OU.

The Policy Object could be configured so that Active Roles continues with the user creation only after a certain piece of the script (the pre-create event handler) has successfully executed. In this way, the script prohibits the creation of user objects whose properties violate corporate rules. It prevents the population of object properties with values taken from external data sources, and generates default property values in accordance with the corporate rules.

The Policy Object may also be configured to pass control to another piece of the script (the post-create event handler) immediately after a user object is successfully created. This enables the script to trigger additional actions, required by corporate rules, after the object has been created. For example, it can update external data stores, provision the user with access to resources, and notify that the user object has been created.

Security and administration elements

Active Roles offers three key security and administration elements, which are stored as objects in the Administration Database:

- Access Templates
- Policy Objects
- Managed Units

These elements enable any user or group in Active Directory to be given limited and effectively controlled administrative privileges.

Users and groups that are given administrative permissions in Active Roles are referred to as *Trustees*. Trustees can be assigned to Managed Units or directory objects and containers.

Trustees do not need special administrative rights within Active Directory. To give Trustees access to Active Directory, Active Roles implements proxy mechanisms that use Access Templates to specify the level of access. When Trustees exercise their access permissions, these mechanisms use Policy Objects to trigger additional actions, such as running integration scripts and validating input data.

When designating a user or group as a Trustee, you must specify the Access Templates that control what the Trustee can do. Permissions granted to a group are extended to all members of that group. To reduce administration time, administrative control should be delegated to groups, rather than to individual users.

To implement policy constraints and automation, you must configure and apply Policy Objects that invoke built-in or custom procedures upon administrative requests. Policy procedures may include running custom scripts to synchronize Active Directory data with other data sources, performing a data validity checkup, and initiating additional administrative operations.

Access Templates for role-based administration

An *Access Template* is a collection of permissions that define what actions can be performed by an administrative role. Active Roles applies Access Templates to directory objects, containers, and administrative views (Managed Units) in relation to groups and users designated as Trustees.

Active Roles offers an extensive suite of preconfigured Access Templates that represent typical administrative roles, enabling the correct level of administrative authority to be delegated quickly and consistently. Access Templates significantly simplify the delegation and administration of management rights, speed up the deployment of the delegation model, and reduce management costs. The preconfigured Access Templates are discussed in the Active Roles Access Templates Available out of the Box document.

Access Templates enable centralized administrators to define administrative roles with various levels of authority, speeding up the deployment of access control and streamlining change tracking of permission settings across the enterprise.

It is also possible to create custom Access Templates based on business requirements. Custom Access Templates can be modified at any time. When an Access Template is modified, the permission settings on all objects where that Access Template is applied change accordingly.

Policy Objects to enforce corporate rules

A *Policy Object* is a collection of administrative policy definitions that specify corporate rules to be enforced. Access Templates define who can make changes to a piece of data, and Policy Objects control what changes can be made to the data. Active Roles enforces corporate rules by linking Policy Objects to:

- Administrative views (Managed Units)
- Active Directory containers

- Individual (leaf) directory objects

Policy Objects define the behavior of the system when directory objects are created, modified, moved, or deleted. Policies are enforced regardless of a Trustee's permissions.

A Policy Object includes stored policy procedures and specifications of events that activate each procedure. Based on policy requirements, a policy procedure could:

- Validate specific property values
- Allow or deny entire operations
- Trigger additional actions

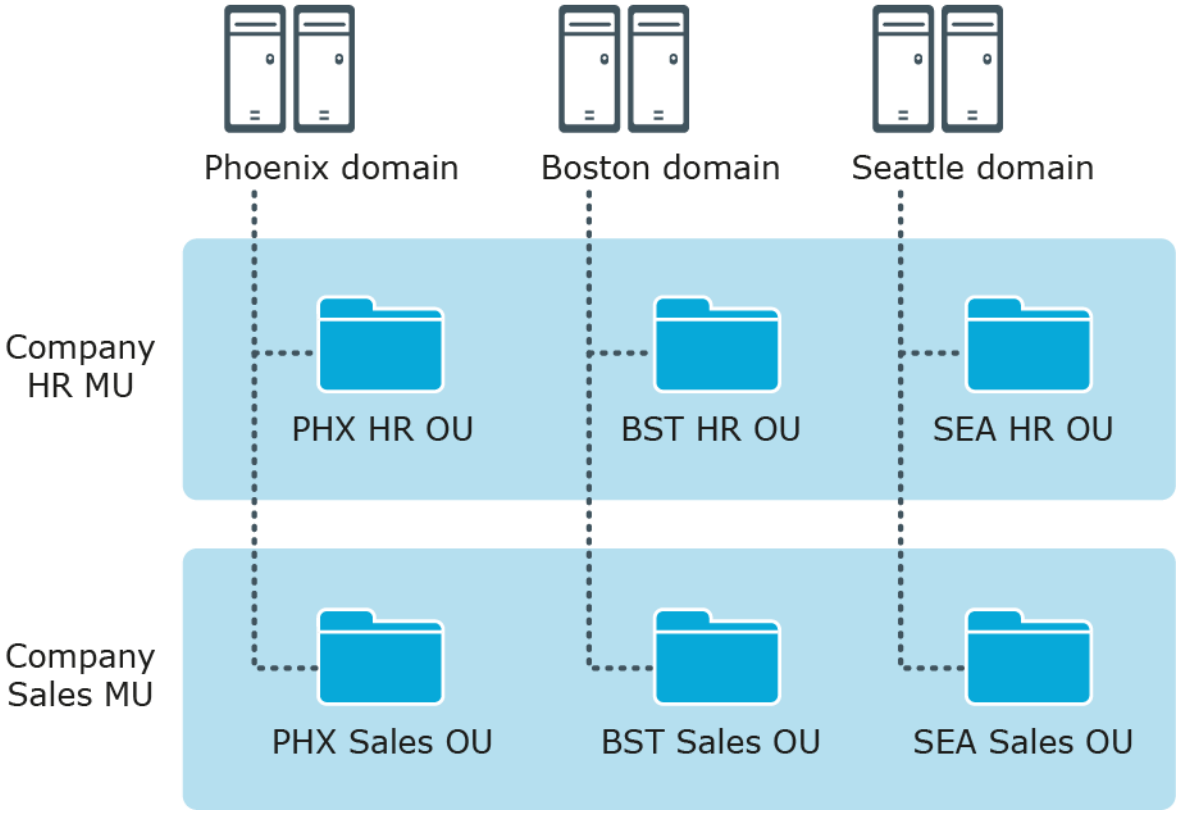
A Policy Object associates specific events with its policy procedures, which can be built-in procedures or custom scripts. This provides an easy way to implement sophisticated validation criteria, synchronize different data sources, and combine a number of administrative tasks into a single batch.

Managed Units to provide administrative views

A *Managed Unit* is a collection of objects collectively managed with Active Roles, created for the distribution of administrative responsibilities, enforcement of business rules and corporate standards, and management of complex network environments. Using Managed Units, the management framework can be separated from the Active Directory design. Directory objects can easily be grouped into administrative views, regardless of their location in Active Directory.

For example, the Active Directory design might be based on geographic location, with domains named after cities or regions and organizational units named after corporate departments or groups. However, Managed Units could be designed to manage specific departments or groups that are divided across multiple geographic locations.

Figure 2: Managed Units



In this example, each AD domain has a Human Resources (HR) OU and a Sales OU. The Active Roles design has an HR MU and a Sales MU. The HR MU enables administrators to configure the policies and security restrictions needed for all HR users regardless of their location, while the Sales MU enables the same for all Sales users.

Managed Units are defined with the use of membership rules—criteria used by Active Roles to evaluate whether or not an object belongs to a given Managed Unit. This enables Managed Units to dynamically change as the network environment changes. For example, you can define a Managed Unit by specifying rules that include all objects whose properties match specific conditions. The specified rules will force the new or modified objects to be members of the correct Managed Unit.

Managed Units extend the functionality of organizational units (OUs), providing convenient scope to delegate administration and enforce corporate rules. A Managed Unit has the following characteristics:

- Represents a collection of objects (one object can belong to more than one Managed Unit)
- Supports rule-based specifications for its members (a Managed Unit only holds objects that satisfy the membership rules specified for the Managed Unit)
- Can hold directory objects that reside in different organizational units, domains, forests, and other Managed Units

Active Roles ensures that permission and policy settings specified for a Managed Unit are inherited by all objects that belong to that Managed Unit. When a directory container belongs to a Managed Unit, all child objects in that container inherit the permission and

policy settings defined at the Managed Unit level. This inheritance continues down the directory tree within all container objects that are members of the Managed Unit.

Active Directory security management

The Active Roles MMC Interface makes it easy to examine and manage permission entries in Active Directory, by showing the access available to each user, along with the scope of their access. A centralized view of all permission entries for any given object helps with the analysis and administration of permissions in Active Directory. For each permission entry, the view displays a number of entry properties, including the permission description, origin, and security principal. From the main window, additional properties can be displayed and the native security editor can be accessed.

The centralized display of native security allows the administrator to quickly view permissions assigned to objects in Active Directory, and to determine whether the permission is inherited. The list of permission entries can be sorted by security principal name to determine who has access to the selected object. If a permission entry is inherited, Active Roles identifies the object from which the permission originates, so that the administrator can easily find and edit the permission entry for that object.

The Active Roles MMC Interface provides the capability to view the permissions for an object by simply clicking the object to display the permission entries in a centralized view. This makes it easier for the administrator to verify the permissions on security-sensitive objects, and to identify possible security problems.

Management of native security

Active Roles Access Templates can be used to specify permissions in Active Directory. Designed to support the role-based grouping of permissions, Access Templates provide an efficient mechanism for setting and maintaining access control, simplifying and enhancing the management of permissions in Active Directory.

To provide this capability, Active Roles gives the administrator the option to keep Active Directory native security updated with selected permissions specified using Access Templates. This option, referred to as Permissions Propagation, is intended to provision users and applications with native permissions to Active Directory. The normal operation of Active Roles does not rely on this option.

For Active Roles permission entries with the Permissions Propagation option set, Active Roles generates Active Directory native permission entries in accordance with the Active Roles permissions. Once set, the option ensures that every time Active Roles permission assignments or templates change, the associated native permission entries change accordingly.

Customization using ADSI Provider and script policies

Active Roles offers the facility to customize its off-the-shelf functionality using scripts and applications that interact with the Administration Service. It allows a high degree of customer modification to meet specific business and organizational needs. This gives customers greater flexibility when using the product, and enables them to build solutions that can easily be integrated with existing systems and data.

The following list shows some of the ways in which the product can be customized:

- Using the Active Roles ADSI Provider, the existing proprietary applications or custom Web-based interfaces could communicate with Active Roles to perform administration and provisioning tasks on user accounts and groups.
- Using policy scripts, custom corporate rules could be enforced to regulate data format and administrative workflows.
- Using policy scripts, the data stored in an HR database or ERP system could be incorporated into the administration and provision of users.

Active Roles makes it possible for user-developed scripts and applications to manipulate directory objects through the Administration Service (*persistent objects*), and to take control of objects that are in the process of being created, modified, or deleted with Active Roles (*in-process objects*).

Having programmatic access to persistent and in-process objects makes it easy for developers to customize Active Roles in these two areas:

- Creating custom applications and user interfaces
- Enforcing corporate administrative policies by running custom scripts (*script policies*)

Custom applications and user interfaces

A custom application or user interface can be created to manipulate directory objects in Active Roles. Active Roles offers the ADSI Provider to communicate with the Administration Service using standard COM interfaces that conform to the Microsoft ADSI 2.5 specification.

Custom applications are executables that provide data to the Administration Service or retrieve and process data from the Administration Service. For example, an organization with a separate Human Resources database could develop and deploy a custom application that extracts personal information from the database, and then passes it to the Administration Service in order to facilitate user account provisioning.

Custom user interfaces are usually Web-based interfaces that distribute certain tasks to users. Custom user interfaces can also be used to streamline the workflow of network administrators and help-desk operators. For example, Web-based pages could be created so that help-desk operators only see the fields related to user properties that they can view and modify, according to the corporate standards.

Both custom applications and user interfaces rely on the Active Roles ADSI Provider to access the functionality of Active Roles.

Custom script policies

Active Roles provides the ability to implement administrative policies by running user-developed scripts. This makes it possible to:

- **Facilitate the provisioning of user accounts** Populate user properties through external database integration and automate multi-step provisioning tasks.
- **Maintain the integrity of directory content** Prevent inconsistency of Active Directory data by enforcing update-sequence and data-format policies across the enterprise.
- **Enforce business rules** Maintain security design and capture administration expertise by integrating business rules into the administrative workflow.

Once configured, the custom script-based policies are enforced without user interaction. Active Roles automatically handles the execution of policy scripts that supplement particular administrative operations and trigger additional administrative actions. For example, policy scripts can be used to:

- Perform a sophisticated validity check on input data
- Synchronously change information in multiple data sources, such as the Active Directory store, Microsoft Exchange server, and HR or ERP-system database
- Ensure that delegated administrators follow a prescribed administrative workflow
- Link multiple administrative tasks into one operator transaction

Dynamic groups

Active Roles helps streamline group maintenance by defining group membership dynamically, with rule-based membership criteria. Dynamic group membership eliminates the need to manually update membership lists for security and distribution groups.

To automate the maintenance of group membership lists, Active Roles provides:

- Rule-based mechanism that automatically adds and removes objects to groups whenever object attributes change in Active Directory
- Flexible membership criteria that enable both query-based and static population of groups

The membership criteria fall into these categories:

- **Include Explicitly** Ensures that specified objects are included in the membership list regardless of any changes made to the objects.

- **Include by Query** Populates the membership list with objects that have certain properties. When an object is created, or when its properties are changed, Active Roles adds or removes it from the membership list depending on whether the object's properties match the search criteria.
- **Include Group Members** Populates the membership list with members of specified selected groups. When an object is added or removed from the selected groups, Active Roles adds or removes that object from the membership list.
- **Exclude Explicitly** Ensures that specified objects are not in the membership list regardless of any changes made to the objects.
- **Exclude by Query** Ensures that objects with certain properties are not in the membership list. Active Roles automatically removes objects from the membership list depending on whether the objects' properties match the search criteria.
- **Exclude Group Members** Ensures that members of specified groups are not in the membership list. When an object is added to any one of the selected groups, Active Roles automatically removes that object from the membership list.

These membership criteria are also applicable to Managed Units.

Workflows

Active Roles provides a rich workflow system for directory data management automation and integration. Based on Microsoft's Windows Workflow Foundation technology, this workflow system enables IT to define, automate and enforce management rules quickly and easily. Workflows extend the capabilities of Active Roles by delivering a framework that enables combining versatile management rules such as provisioning and de-provisioning of identity information in the directory, enforcement of policy rules on changes to identity data, routing data changes for approval, e-mail notifications of particular events and conditions, as well as the ability to implement custom actions using script technologies such as Microsoft Windows PowerShell or VBScript.

Suppose you need to provision user accounts based on data from external systems. The data is retrieved and then conveyed to the directory by using feed services that work in conjunction with Active Roles. A workflow can be created to coordinate the operations in account provisioning. For example, different rules can be applied for creating or updating accounts held in different containers.

Workflows may also include approval rules that require certain changes to be authorized by designated persons (approvers). When designing an approval workflow, the administrator specifies which kind of operation causes the workflow to start, and adds approval rules to the workflow. The approval rules determine who is authorized to approve the operation, the required sequence of approvals, and who needs to be notified of approval tasks or decisions.

By delivering e-mail notifications, workflows extend the reach of management process automation throughout the enterprise. Notification activities in a workflow let people be notified via e-mail about events, conditions or tasks awaiting their attention. For example, approval rules can notify of change requests pending approval, or separate notification rules can be applied to inform about data changes in the directory. Notification messages

include all necessary supporting information, and provide hyperlinks enabling message recipients to take actions using a standard Web browser.

The logic of an automated management process can be implemented by using administrative policies in Active Roles. Yet creating and maintaining complex, multi-step processes in that way can be challenging. Workflows provide a different approach, enabling IT administrators to define a management process graphically. This can be faster than building the process by applying individual policies, and it also makes the process easier to understand, explain and change.

Operation in multi-forest environments

Active Directory organizes network elements into a hierarchical structure based on the concept of containers, with the top-level container being referred to as a forest. Today, many real-world Active Directory implementations consist of several forests. Common reasons for multi-forest deployments are the isolation of the administrative authority, organizational structure issues (e.g., autonomous business units and decentralized IT departments), business policy, or legal and regulatory requirements.

This section provides information on the features and benefits of Active Roles as applied to environments where multiple Active Directory forests have been deployed.

With Active Roles, you can create a scalable, secure, and manageable infrastructure that simplifies user and resource management in a multi-forest environment. Benefits of deploying Active Roles in such environments include:

- Centralized management of directory data in domains that belong to different forests
- Administrative views spanning forest boundaries
- The ability to delegate administrative control of directory data where appropriate, without regard to forest boundaries
- Policy-based control and automation of directory data management across forest boundaries

By registering Active Directory domains with Active Roles, you form a collection of managed domains that represents an Active Roles security and administrative boundary in Active Directory. The collection need not be restricted to domains from a single forest. Rather, you can register domains from any forest in your environment, configuring the Active Roles Administration Service to use the appropriate administrative credentials on a per-domain basis.

To centralize management of directory data across the managed domains, Active Roles retrieves and consolidates the Active Directory schema definitions from all forests to which those domains belong. The consolidated schema description is stored in the Active Roles configuration database, and contains information about the object classes and the attributes of the object classes that can be stored in the managed domains. By using the consolidated schema, Active Roles extends the scope of its administrative operations to cover the entire collection of managed domains regardless of forest boundaries.

Active Roles allows administrators to organize directory objects (such as users, groups, computers, and so on) into a relational structure made up of rule-based administrative

views (referred to as Managed Units), each of which includes only the objects that meet certain membership criteria defined by the administrator. This structure can be designed independently from the logical model of Active Directory, which is based on the concept of containers and thus implies rigid boundaries between containers, be it forests, domains or organizational units. Administrators can configure Managed Units so that each Unit represents the appropriate collection of directory objects that reside in the same Active Directory container or in different containers, with different forests not being the exception.

To facilitate the management of directory data, Active Roles provides for administrative delegation at the Managed Unit level as well as at the level of individual containers in Active Directory. Through delegation, authority over directory objects held in a given Unit or container can be transferred to certain users or groups. Delegation of control over Managed Units provides the ability to distribute administration of directory data among individuals trusted to perform management of specific groups and types of objects, without taking into account the location of the objects in the Active Directory structure. Thus, Active Roles makes it easy to delegate control of directory data from one forest to users or groups located in the same forest or in a different forest.

Active Roles also allows policy-based control and automation of directory data management to be implemented at the Managed Unit level. By applying policy and automation rules to Managed Units, administrators can ensure consistent control of the well-defined collections of directory objects located in different organizational units, domains, or forests. In addition, policy and automation rules can be consistently applied to different containers, whether in the same forest or in different forests, which provides the platform for complex automation scenarios that involve cross-forest operations. An example could be provisioning users from one forest with resources in another forest.

When adding objects to a group, Active Roles allows you to select objects from different managed domains, including those that belong to different forests. This operation requires a trust relationship between the domain that holds the group and the domain that holds the object you want to add to the group. Otherwise, Active Directory denies the operation and, therefore, Active Roles does not allow you to select the object. Note that Active Directory automatically establishes trust relationships between domains within one forest. As for domains in different forests, administrators must explicitly establish trust relationships as needed.

The rule-based mechanisms that Active Roles provides for auto-populating groups can also be freely used in multi-forest environments. You can configure rules to have Active Roles populate groups with objects that reside in different domains, whether in the same forest or in different forests. However, the capabilities of Active Roles to automatically manage group membership lists are also restricted by the Active Directory constraints that only allow a group to include objects from the domain that holds the group or from the domains trusted by that domain. In other words, unless a trust relationship is established between the domain that holds the group and the domain that holds a given object, the object cannot be added to the group, neither manually nor automatically by Active Roles.

Examples of use

Active Roles can be configured to provide a wide range of directory management solutions, allowing organizations to create more secure, productive, and manageable Active Directory and Microsoft Exchange environments. This section highlights how Active Roles helps to address the challenges faced by enterprises today.

Distributing administration

Suppose a large company wants to introduce distributed administration, but wants to avoid the large costs involved in training their Help Desk and business units to correctly use complex administrative tools. In this situation, there is the need for an easy-to-use tool, to control what actions the Help Desk and business units can perform, and to enforce company policies and procedures.

Solution

Active Roles allows organizations to create Managed Units and to designate Trustees over those Managed Units. Trustees only see the objects to which they have access. They are given only the rights they need for the objects within these Managed Units, down to individual properties. Unlike native Active Directory organizational units, Managed Units provide virtual boundaries that span across domains and forests, offering more flexible delegation capabilities.

Delegating limited control over Managed Units efficiently eliminates the need for high-level administrative user ID's, allowing organizations to securely distribute administrative authority to local management. To improve network security and make distributed administration safe, Active Roles defines and enforces customizable administrative polices.

Active Roles allows organizations to safely implement administration for business units. If a company has a number of different business units, each of equal importance and each located in a separate office, a single network administrator could support all of the sites. Active Roles allows the company to create a single Managed Unit, giving an administrator control over users and resources that span multiple domains.

Integrating with other systems

Suppose a company wants to integrate its HR system, administration, and physical security to provide a workflow that reduces repetitive data. Normally, the HR team creates a user profile, the IT team also creates a user profile in Windows and Exchange, and the security team activates an access card for the new employee. The three teams do not synchronize with each another and instead duplicate their work. This results in increased administration costs and introduces security issues. For example, some individuals may no longer work for the company but may still have valid user ID’s and access cards. In this scenario, there

is a need to integrate the company's HR system and other systems, and to automate the execution of user provisioning tasks.

Solution

With Active Roles, a suitable property set can be established to include data from network data sources other than Active Directory. For instance, a property set might be configured to retrieve a user's personal information from an HR database. When the user account is created, this data could then be passed to Active Directory and Microsoft Exchange. If these property values change, an update could be made to both Active Directory and to the HR system.

Active Roles also provides the ability to set up administrative policies that reduce the amount of input required to carry out a task. For example, when a user moves to a different location, Active Roles could automatically update the user's profile in the HR system, based only on the change to the user's site code or department in Active Directory. Additionally, when a user joins or leaves the company, their access card could automatically be enabled or disabled.

Managing a multi-forest Active Directory design

Suppose a host company has client customers who need to place domain controllers on their premises. In Active Directory, every domain controller holds a writable copy of the schema and configuration of the entire forest. Anyone with administrative or backup/restore rights on any domain controller, or physical access to any domain controller, could potentially disrupt the entire forest. For instance, they could attempt to circumvent Windows security, or they could edit the Active Directory database, and the changes would be propagated to all domains in the forest. To avoid such an incident, the company needs to create a separate forest for each client who requires domain controllers on their premises. Otherwise, the actions of one malicious user could affect directory service delivery for other clients in the same forest.

Having multiple forests increases the complexity of the Active Directory structure. This in turn leads to increased administration, as each forest needs separate directory service administration. In this case, there is a need for an administrative system that enables the cross-forest management of Active Directory.

Solution

Active Roles provides a unified management structure that can extend across multiple Active Directory forests. The Active Roles user interface provides a single interface for the management of Active Directory domains that belong to different forests. It offers administrative views (Managed Units) that can hold objects from multiple forests, thereby enabling the unified application of corporate rules and roles across forest boundaries.

With its ability to safely delegate administration in multi-forest environments, Active Roles provides the necessary level of control for the host company's customers, while enabling the company to implement role-based security, and restrict the customers' administrative actions based on corporate policies.

For security reasons, it may be unacceptable to have an administrative tool with the same level of rights as a domain administrator. This is because administrative access to an entire domain in a forest may be used to gain administrative access to the whole forest, via the elevation of privileges attack. Active Roles can operate in a multi-forest environment within a precisely defined scope of access to domains, with no special requirement to have administrative access to entire domains or security-sensitive containers. This addresses the need for a product that provides advanced administrative capabilities, while effectively preventing the elevation of privileges.

Simplifying Active Directory structure

Suppose a company wants to design an Active Directory structure based on physical location. As a rule, the administration/IT department, business units, and Exchange team would each prefer to have a different structure. As a result, they agree to a compromise that doesn't fully satisfy their requirements. Clearly, there is a need to simplify the Active Directory structural requirements.

Solution

In Active Roles, Managed Units allow organizations to achieve acceptable security boundaries without setting up extra domains or organizational units. This significantly simplifies the Active Directory structure and reduces security risks.

By using Managed Units for delegation purposes, Active Roles creates a rule-based overlay of Active Directory for administration. This simplifies the process of choosing an Active Directory structure. Different administrative tasks often require different OU structures. For instance, an OU structure designed purely for the delegation of administration differs from an OU structure shaped purely for Group Policy. It becomes much easier to design an Active Directory structure by using Managed Units to handle delegation issues.

Handling organizational changes

Consider a company in the process of re-organization. Multiple departments are changing names, merging, or separating from one another. Such reorganization involves an increase in administrative, security, and business liabilities, as well as the high cost of manually updating data. This situation demands a means to automatically update and move the data.

Solution

Active Roles provides the ability to define administrative policies that make organizational changes easier to handle. By using Managed Units, rule-based overlays of the actual data in Active Directory can be set up for both the current and planned organizational structures. Administrative policies can be specified so that when data moves from one Managed Unit to another, policy definitions will automatically be applied, based on the change. This will update properties, such as the user’s manager, department, group memberships, and OU memberships.

As another example, consider a user who changes departments. Depending on the department to which the user moves, Active Roles could automatically move the user's data, change the user's group memberships, and specify to whom the user reports.

User Account Management

Suppose a company provides services based on Active Directory and Microsoft Exchange. The company relies on the Active Directory infrastructure as a basis for their service offerings.

Configuration of Active Directory involves setting security and partitioning the directory, so that any user has proper access to directory resources. It is paramount to have a framework that facilitates the creation of new user accounts and the assignment of appropriate access rights. There is a need for a robust system that maintains user creation and management with minimal administrative effort.

Solution

Active Roles offers a reliable solution to simplify and safely distribute user account management. It addresses the need to create and manage a large number of user accounts, and to ensure that each user can only access their own resources. By implementing an administrative model based on business rules, Active Roles allows domain-level administrators to easily establish and maintain very tight security, while facilitating the provisioning of new users with the appropriate access to IT resources

Active Roles has the ability to safely delegate routine user-management tasks to designated persons. By incorporating policy enforcement and role-based security, Active Roles allows the organization to restrict the administrative actions according to the corporate policies defined by the high-level administrators. In addition, it allows the administrators to change the policies, ensuring that new policy settings are automatically propagated and enforced without additional development.

Active Roles makes it simpler for the organization to delegate authority to administrative and support groups, while enhancing the overall security. The Web Interface can serve as an administrative tool that allows the assistant administrators to manage users, groups, and mailboxes. Active Roles ensures that all actions performed by a Web Interface user are in compliance with the corporate security policies.

3

Getting Started

- Starting the Active Roles console
- User Interface overview
- View mode
- Using Managed Units
- Setting up filter
- Finding objects
- Getting policy-related information

Starting the Active Roles console

The Active Roles console, also referred to as MMC Interface, is a comprehensive administrative tool for managing Active Directory and Microsoft Exchange. With the Active Roles console, you can easily find directory objects and perform administrative tasks.

To start the Active Roles console
Depending upon the version of your Windows operating system, click **Active Roles 7.5 Console on the Apps page or select All Programs** | **One Identity Active Roles 7.5** | **Active Roles 7.5 Console from the Start menu.**

NOTE:

- Normally, the Active Roles console automatically chooses the Administration Service and establishes a connection. If the console cannot connect to the Administration Service or you want to manually select the Administration Service, see "Connecting to the Administration Service" in the *Active Roles Administration Guide*.
- For more information on extending the Active Roles provisioning and account administration capabilities to your cloud applications, click **Learn More** in the **Action|About Active Roles|What's New** tab from the console window.

Delegating control to users for accessing MMC interface

By default, on installing Active Roles, all users are allowed to log in to the MMC interface. To manage the MMC interface access for a user, you must configure the options using **Configuration Center** | **MMC Interface Access**| **Manage settings**. Selecting this option restricts all non Active Roles Administrators from using the console. All delegated users are affected, however, it does not apply to Active Roles Administrators.

To be able to log in to the MMC interface, the user must be delegated with the **User Interfaces** access rights on the **User Interfaces** container under **Server Configuration**. User Interfaces Access templates that provide the access rights are available as part of the Active Roles built-in Access templates in the **User Interfaces** container.

To delegate the control to users in the User Interfaces container you must apply the User Interface Access Template

1. In the console tree, expand **Active Roles | Configuration | Server Configuration**.
2. Under **Server Configuration**, locate the **User Interfaces** container, right-click it, and click **Delegate Control**.
3. On the **Users or Groups** page, click **Add**, and then select the users or groups to which you want to delegate the control. Click **Next**.
4. On the **Access Templates** page, expand the **Active Directory** | **User Interfaces** folder, and select the check box next to **User Interface Management-MMC Full control**.
5. Click **Next** and follow the instructions in the wizard, accepting the default settings.
6. After you complete these steps, the users and groups you selected in Step 3 are authorized to log in to the MMC interface.
7. Click **OK** to close the **Active Roles Security** dialog box.

Getting and using help

Active Roles Help explains concepts and includes instructions for performing tasks with the product.

You can use the following guidelines to get assistance while you work:

- To access Active Roles Help, click **Help** on the **Action** menu or **Help Topics** on the **Help** menu.
- To view description of a dialog box, click the **Help** button in the dialog box or press F1.

- To view a brief description of a menu command or a toolbar button, point to the command or button. The description is displayed in the status bar at the bottom of the window.

You can print a single Help topic or all Help topics under a selected heading.

To print a single Help topic

1. On the menu bar, click **Help** and then click **Help Topics**.
2. In the left pane of the Help viewer, expand the heading that contains the topic you want to print, and then click the topic.
3. On the Help viewer toolbar, click **Options**, click **Print**, and then click **OK**.

To print all Help topics under a heading

1. On the menu bar, click **Help**, and then click **Help Topics**.
2. In the left pane of the Help viewer, click the heading that contains the topics you want to print.
3. On the Help viewer toolbar, click **Options**, and then click **Print**.
4. In the **Print Topics** dialog box, click **Print the selected heading and all subtopics**, and then click **OK**.

User Interface overview

The Active Roles console window is divided into two panes. The left pane contains the console tree, showing the items that are available in the console. The right pane, known as the details pane, displays information about items you select in the console tree. You can perform most management tasks from this pane using commands on the **Action** menu.

Additional information is displayed in the lower sub-pane of the details pane when you check the **Advanced Details Pane** command on the **View** menu. You can perform management tasks from the lower sub-pane using commands on the **Action** menu.

Console tree

The left pane of the Active Roles console contains the console tree.

The console tree root is labeled **Active Roles**. The name of the Administration Service is shown in square brackets. If you have Advanced view mode selected for Active Roles console display (**View | Mode**), the following folders are shown under the console tree root:

- **Configuration** Contains all Active Roles proprietary objects held in containers with appropriate names.

- **Active Directory** Contains a list of domains registered with Active Roles. In this folder, you can browse domains for directory objects (users, group, computers), and perform management tasks on those objects.
- **AD LDS (ADAM)** Contains a list of AD LDS directory partitions registered with Active Roles. In this folder, you can browse partitions for directory objects (users, group, containers), and perform management tasks on those objects.
- **Applications** Contains a list of applications integrated with Active Roles, such as Reporting, and allows for quick access to those applications.

The console display mode determines which folders are displayed in the console tree. For more information, see View mode later in this document.

Details pane

When you select an item in the console tree, the details pane changes accordingly. To perform administrative tasks, click items in the details pane and use commands on the **Action** menu. The **Action** menu commands also appear on the shortcut menu that you can access by right-clicking items in the console tree or details pane.

By default, the objects listed in the details pane are sorted in ascending order by object name. You can change the sorting order by clicking a column heading. You can add and remove columns in the details pane using the **Choose Columns** command on the **View** menu.

In the Active Roles console you can apply filters to the details pane in order to search for directory objects. To configure a filter, select a domain and then click **Filter Options** on the **View** menu. It is also possible to find an object in the details pane by typing a few characters. This will select the first item in the sorted column that matches what you typed.

Advanced pane

The advanced pane appears at the bottom of the details pane if you check **Advanced Details Pane** on the **View** menu. You can use the advanced pane to administer an object selected in the console tree or details pane: right-click an existing entry in the list to administer it, or right-click a blank area of the advanced pane to add a new entry.

The advanced pane is composed of a number of tabbed pages. The selected object determines which tabs are displayed. All possible tabs in the advanced pane and their descriptions are as follows:

- **Active Roles Security** Lists Active Roles Access Templates applied to the selected object.
- **Links** Lists the objects to which the selected Access Template is applied.
- **Active Roles Policy** Lists Active Roles Policy Objects applied to the selected object.

- **Native Security** Lists Active Directory permission entries specified for the selected object.
- **Member Of** Lists groups to which the selected object belongs.
- **Members** Lists members of the selected group.

NOTE: The console displays the **Active Roles Security**, **Active Roles Policy**, and **Native Security** tabs for a selected object only if your user account has the **Read Control** right to the selected object.

Depending on the tab you have selected in the advanced pane, the toolbar displays the following buttons to help you work with the entries on the tab.

Active Roles Security and Links

Table 1: Active Roles Security and Links

	Apply additional Access Templates to the selected object.
	Display Access Templates that affect the selected object owing to inheritance.
	Synchronize from Active Roles security to Active Directory security.

Active Roles Policy

Table 2: Active Roles Policy

	Apply additional Policy Objects to the selected object.
	Display Policy Objects that affect the selected object owing to inheritance.

Native Security

Table 3: Native Security

	Display permission entries that are inherited from parent objects.
	Display default permission entries specified by the AD schema.

Member Of and Members

Table 4: Member Of and Members

	Add the selected object to groups.
	Set the group as the primary group for the selected object.

Customizable Web Interface

The Active Roles Web Interface is a customizable Web-based application that facilitates administration, while taking full advantage of Active Roles' security, workflow integration, and reporting benefits. To help distribute administrative tasks, the Web Interface allows you to configure multiple Web sites with individual sets of user interface elements. Each Web site can be customized to meet specific business and organizational needs.

Key features

Key features of the Web Interface include the following.

Role-based suite of interfaces

Customized interfaces (Web Interface sites) can be installed and configured for administrators, help desk operators, and end users. Administrators use an interface that supports a wide range of tasks, whereas help desk operators use a tailored, dedicated interface to expedite the resolution of trouble tickets. Network end users have access to an interface for self-administration. Multiple interfaces with different configurations can be deployed so that there is no need to re-configure the Web Interface for particular roles.

Dynamic configuration based on roles

The Web Interface dynamically adapts to the specific roles assigned to the users. A user can see only the commands, directory objects, and object properties to which the user's role provides administrative access. Objects and commands beyond the scope of the user are removed from the Web Interface, streamlining the execution of administrative tasks.

Point-and-click customization

It is straightforward to configure the user interface. Administrators can set up a suitable set of user interface elements without writing a single line of code. Administrators can add and remove commands or entire menus, assign tasks and forms to commands, modify forms used to perform tasks, and create new commands, tasks, and forms. All configuration settings are saved in a persistent storage so that the Web Interface users are always presented with the properly configured interfaces that suite their roles.

Instant application of administrative policies

User input is efficiently supplemented and restricted based on administrative policies defined in Active Roles. The Web Interface displays property values generated in accordance with the policies, and prohibits the input of data that violates them. User input is checked against the policies before committing the operation request, and if a violation is detected, the user can immediately correct the input.

Fully-featured management solution

The Web Interface supports all administrative tasks on Active Directory objects such as users, groups, and computers, and on computer resources such as services, printers, network file shares, and local users and groups. With its advanced customization capabilities, the Web Interface serves as a complete administrative tool, providing suitable interfaces for any administrative role.

User Profile Editor

Provided they have the necessary Active Roles permissions, end users can view or change their personal data. Due to the reliable enforcement of business rules based directory entry, the Web Interface makes these tasks safe and secure. With User Profile Editor, Active Roles enables IT to manage, but not necessarily participate, in these time-consuming tasks, resulting in decreased help desk calls and IS administration time.

Support for multiple languages

The Web Interface allows users to select their preferred language. Changing the language affects all menus, commands, and forms associated with the Web Interface, as well as tool tips and help.

Different interfaces for different roles

The Web Interface allows multiple Web sites to be installed with individual, customizable configurations. The following configuration templates are available out-of-the box:

- **Site for Administrators** Supports a broad range of tasks, including the management of all directory objects and computer resources.
- **Site for Help Desk** Handles typical tasks performed by Help Desk operators, such as enabling or disabling accounts, resetting passwords, and modifying certain properties of users and groups.
- **Site for Self-Administration** Provides User Profile Editor, allowing end users to manage personal or emergency data through a simple-to-use Web interface.

Each Web site configuration template provides an individual set of commands installed by default. The Web site can be customized by adding or removing commands, and by modifying Web pages (forms) associated with commands.

Although the Web Interface dynamically adapts to roles assigned to users, the ability to tailor separate Web sites to individual roles gives increased flexibility to the customer. It helps streamline the workflow of directory administrators and help-desk personnel. Static configuration of interface elements ensures that Web Interface users have access to the specific commands and pages needed to perform their duties.

Role-based management of computer resources

Active Roles provides the ability to delegate administration of computer resources, such as services and printers. Delegated administrators can use the Active Roles Web Interface to manage computer resources with a single, consolidated tool. Active Roles, along with the Web Interface, enables the delegation of administrative tasks on the following computer resources:

- **Services** Start or stop a service, view or modify properties of a service.
- **Network File Shares** Create a file share, view or modify properties of a file share, stop sharing a folder.
- **Logical Printers** Pause, resume or cancel printing, list documents being printed, view or modify properties of a printer.
- **Documents being printed (print jobs)** Pause, resume, cancel or restart printing of a document, view or modify properties of a document being printed.
- **Local groups** Create or delete a group, add or remove members from a group, rename a group, view or modify properties of a group.
- **Local users** Create or delete a local user account, set a password for a local user account, rename a local user account, view or modify properties of a local user account.
- **Devices** View or modify properties of a logical device, start or stop a logical device.

Active Roles provides a comprehensive set of Access Templates that are available out of the box for delegating computer management tasks. By applying Access Templates of the "Computer Resources" category to a computer account, the rights of delegated administrators can be specified on the corresponding computer's resources.

Delegated administrators should use the Web Interface rather than the Active Roles console (MMC Interface) to manage computer resources. Although the console provides certain tools for computer resources management, the console user needs the native administrator rights on the computer in order to use those tools. The rights specified through "Computer Resources" Access Templates have no effect in the tools provided by the console for computer resources management.

View mode

In the Active Roles console you can choose view mode—Basic, Advanced, or Raw. Changing view mode makes it possible to filter out advanced objects and containers from the display.

Basic mode displays Active Directory objects and Managed Units, and filters out objects and containers related to the Active Roles configuration. Basic mode should normally be used by delegated administrators and help-desk operators.

Advanced mode displays all objects and containers except those reserved for Active Roles internal use. Advanced mode is designed for administrators who are responsible for configuring the system and managing Active Roles proprietary objects.

Raw mode displays all objects and containers defined in the Active Roles namespace. This mode is primarily designed for troubleshooting.

With Raw mode, the console displays all data it receives from the Administration Service. With Basic or Advanced mode, some data is filtered out. For example, the **Configuration** folder is not shown in the console tree with Basic mode. Another example is the **Configuration Container** folder used to display the Active Directory configuration naming context, which is displayed with Raw mode only. In addition, there are some commands and property pages that are only displayed when the console is in Raw mode.

In short, when you choose Raw mode, the snap-in displays everything it is able to display. Otherwise, some items are hidden. Note that changing view mode does not modify any items. Rather, this only shows or hides particular items from the display.

To change view mode, click **Mode** on the **View** menu. In the **View Mode** dialog box, click **Basic Mode**, **Advanced Mode**, or **Raw Mode**.

Controlled objects

The Active Roles console provides for visual indication of the objects to which Access Templates or Policy Objects are linked. The console marks those objects by adding an arrow icon at the lower-left corner of the icon that represents the object in the console tree or details pane. As a result, the icon looks similar to the following image: .

To enable this feature, click **Mark Controlled Objects** on the **View** menu, and select check boxes to specify the category of object to be marked.

Using Managed Units

Active Roles offers these key security and administration elements:

- **Trustees** Users or groups that have permissions to administer users, groups, computers, or other directory objects.
- **Permissions and Roles** Permissions are grouped in Access Templates (roles) to define how a Trustee can manage directory objects.
- **Managed Units** Collections of directory objects delegated to Trustees for administration.

The directory administrator defines which users or groups are designated as Trustees, which roles and permissions are assigned to Trustees, and what objects are included in Managed Units.

Managed Units are used to determine the directory objects that a Trustee can administer. As a Trustee, you can administer Managed Units for which you have assigned permissions. Managed Units containing objects you are authorized to administer are displayed under **Managed Units** in the console tree.

When you select a Managed Unit in the console tree, the details pane displays a list of objects included in that Managed Unit. To administer objects, select them from the list and use the commands on the **Action** menu.

If a Managed Unit includes a container, such as an Organizational Unit, the container is displayed under the Managed Unit in the console tree. When you select a container in the console tree, the details pane lists all child objects and sub-containers held in that container.

Setting up filter

The Active Roles console makes it possible to apply a filter to display only the objects that match the filtering criteria. To apply a filter, select an Active Directory object or container and click the **Filter** button on the toolbar: . This displays the **Filter Options** dialog box where you can set up a filter. After you set up a filter, the filtering criteria immediately take effect on all lists of Active Directory objects in the Active Roles console.

Steps for sorting and filtering lists in the details pane

To sort objects in the details pane

1. Click a column heading to sort by the contents of that column.
2. Click the column heading again to switch between ascending and descending sort order.

To add or remove columns in the details pane

1. On the **View** menu, click **Choose Columns** or **Add/Remove Columns**.
2. Do the following, and then click **OK**:
   - To add a column, in **Available columns**, click the column you want to display, and then click **Add**.
   - To remove a column, in **Displayed columns**, click the column you want to hide, and then click **Remove**.
   - To re-order columns, click a column name in **Displayed columns**, and then click **Move Up** or **Move Down** to change the position of the column.

NOTE: In the advanced details pane, you can add or remove columns from a list in the upper sub-pane or in the lower sub-pane: click the list in the sub-pane you want to modify, and then follow the steps above.

Filter options help you search for particular objects in the details pane. You can view all objects or only objects of selected type, configure the number of items that can be displayed for each folder, or create custom filters using object attributes and LDAP queries.

To select view filter options

1. On the **View** menu, click **Filter Options**.
2. Do one of the following, and then click **OK**:
   - To view all objects, click **Show all types of objects**. With this option, the filter is turned off.
   - To view objects of certain types, click **Show only the following types of objects**, and select check boxes next to the types of objects you want to view.
   - To view objects that match custom filtering criteria, click **Create custom filter**. Then, **Customize** and configure your filtering criteria by using the instructions outlined in Steps for building a custom search.
3. Optionally, in **Maximum number of items displayed per folder**, modify the maximum number of objects that can be displayed in the console. The default maximum number of objects displayed in the console is 2,000 objects.

Finding objects

In the Active Roles console you can search for objects of different types using the **Find** window. To access the **Find** window, right-click a container and click **Find**.

From the **In** list, you can select the container or Managed Unit you want to search. The list includes the container that you selected before activating the **Find** window. To add containers to the list, click **Browse**. From the **Find** list, you can select the type of the objects you want to find.

When you select an object type, the **Find** window changes accordingly. For example, **Users, Contacts, and Groups** searches for users, contacts, or groups using criteria such

as user name, a note describing a contact, or the name of a group. In the **Find** list, Active Roles splits the **Users, Contacts, and Groups** category into three, providing the option for a more streamlined search.

By selecting **Custom Search** from the **Find** list, you can build custom search queries using advanced search options:

Using the **Find** window, you can search for any directory objects, such as users, groups, computers, Organizational Units, printers or shared folders. It is also possible to search for Active Roles configuration objects such as Access Templates, Managed Units, and Policy Objects. When you search for Access Templates, Policy Objects or Managed Units and select an appropriate object type from the **Find** list, the relevant container appears in the **In** list.

Once the search has completed, the objects matching the search criteria (search results) are listed at the bottom of the **Find** window. You can quickly find an object in the search results list by typing a few characters. This will select the first name that matches what you typed.

Once you have found the object, you can manage it by right-clicking the entry in the search results list, and then clicking commands on the shortcut menu.

Steps for searching for a user, contact, or group

To search for a user, contact, or group

1. On the **Action** menu, click **Find** to display the **Find** window.
2. In the **Find** box, click one of the following:
   - **Users, Contacts, and Groups**, to find users, groups, and contacts that match your search criteria.
   - **Users**, to find only users that match your search criteria.
   - **Groups**, to find only groups that match your search criteria.
   - **Contacts**, to find only contacts that match your search criteria.
3. In the **In** box, select the domain, container or Managed Unit you want to search, or click **Browse** to locate a domain, container or Managed Unit.
4. Type in a name, a description, or both:
   - In the **Name** box, type the name of the object you want to find.
   - In the **Description** box, type the description of the object you want to find.

   You can search using partial search criteria. For example, **B** in the **Name** box will return all objects whose name begins with the letter **B**, such as Backup Operators.
5. Click **Find Now** to start your search.

NOTE:

- You can use the **Advanced** tab for more powerful search options. For details, see Steps for using advanced search options.
- The found users, groups, or contacts are displayed at the bottom of the **Find** window.
- You can manage found users, groups, or contacts directly from the list in the **Find** window: right-click a list item, and then use commands on the shortcut menu to perform management tasks.

Steps for searching for a computer

To search for a computer

1. On the **Action** menu, click **Find** to display the **Find** window.
2. In the **Find** box, click **Computers**.
3. In the **In** box, select the domain, container or Managed Unit you want to search, or click **Browse** to locate a domain, container or Managed Unit.
4. In the **Name** box, type the name of the computer you want to find.

   You can search using partial search criteria. For example, **B** in the **Name** box will return all computers whose name begins with the letter **B**.
5. Optionally, in the **Role** box, click one of the following:
   - **Domain Controller**, to find only domain controllers.
   - **Workstations and Servers**, to find only workstations and servers (not domain controllers).
6. Click **Find Now** to start your search.

NOTE:

- You can use the **Advanced** tab for more powerful search options. For details, see Steps for using advanced search options.
- The found computers are displayed at the bottom of the Find window
- You can manage found computer objects directly from the list in the **Find** window: right-click a list item, and then use commands on the shortcut menu to perform management tasks.

Steps for searching for an Organizational Unit

To search for an Organizational Unit

1. On the **Action** menu, click **Find** to display the **Find** window.
2. In the **Find** box, click **Organizational Units**.
3. In the **In** box, select the domain, container or Managed Unit you want to search, or click **Browse** to locate a domain, container or Managed Unit.
4. In the **Name** box, type the name (or a part of the name) of the Organizational Unit you want to find.
5. Click **Find Now** to start your search.

NOTE:

- You can use the **Advanced** tab for more powerful search options. For details, see Steps for using advanced search options.
- The found Organizational Units are displayed at the bottom of the **Find** window.
- You can manage found Organizational Units directly from the list in the **Find** window: right-click a list item, and then use commands on the shortcut menu to perform management tasks.

Steps for using advanced search options

To use advanced search options

1. On the **Action** menu, click **Find** to display the **Find** window.
2. In the **Find** box, click the kind of object for which you want to search.
3. Click the **Advanced** tab.
4. Click the **Field** button, and select the object property you want to query.
5. In **Condition**, click the condition for your search, and then, in **Value**, type a property value, in order to find the objects that have the object property matching the condition-value pair you have specified.
6. Click **Add** to add this search condition to your search.
7. Repeat steps 4 through 6 until you have added all of the desired search conditions.
8. Click one of the following:
    - If you want to find the objects that meet all of the conditions specified, click **AND**.

- If you want to find the objects that meet any of the conditions specified, click **OR**.

9. Click **Find Now** to start your search. The found objects are displayed at the bottom of the window.

Steps for building a custom search

To build a custom search

1. On the **Action** menu, click **Find** to display the **Find** window.
2. In the **Find** box, click **Custom Search**.
3. In the **In** box, select the domain, container or Managed Unit you want to search, or click **Browse** to locate a domain, container or Managed Unit.
4. Do one of the following:
   - On the **Custom Search** tab, follow Steps 4-9 of the procedure outlined in Steps for using advanced search options.
   - On the **Advanced** tab, specify a search filter using LDAP syntax.
5. Click **Find Now** to start your search.

LDAP syntax

Search filters enable you to define search criteria and provide more efficient and effective searches. The search filters are represented by Unicode strings.

The Active Roles console supports the standard LDAP search filters as defined in RFC2254.

The following table lists some examples of standard LDAP search filters.

Table 5: LDAP search filters

Search filter	Description
`(objectClass=*)`	All objects
`(&(objectCategory=person)(objectClass=user)(!cn=andy))`	All user objects but "andy"
`(sn=sm*)`	All objects with a surname that starts with "sm"
`(&(objectCategory=person)(objectClass=contact)(|(sn=Smith)(sn=Johnson)))`	All contacts with a surname equal to "Smith" or "Johnson"

Search filter format

Search filters use one of the following formats:

```
<filter>=(<attribute><operator><value>)
```

or

```
(<operator><filter1><filter2>)
```

In this example, *<attribute>* stands for the LDAP display name of the attribute by which you want to search.

Operators

The following table lists some frequently used search filter operators.

Table 6: Search filter operators

Logical Operator	Description
=	Equal to
~=	Approximately equal to
<=	Lexicographically less than or equal to
>=	Lexicographically greater than or equal to
&	AND
|	OR
!	NOT

Wildcards

You can also add wildcards and conditions to a search filter. The following examples show substrings that can be used to search the directory.

Get all entries:

```
(objectClass=*)
```

Get entries containing “bob” somewhere in the common name:

```
(cn=*bob*)
```

Get entries with a common name greater than or equal to “bob”:

```
(cn>='bob')
```

Get all users with an e-mail attribute:

```
(&(objectClass=user)(mail=*))
```

Get all user entries with an e-mail attribute and a surname equal to “smith”:

```
(&(sn=smith)(objectClass=user)(mail=*))
```

Get all user entries with a common name that starts with “andy”, “steve”, or “margaret”:

```
(&(objectClass=user) | (cn=andy*)(cn=steve)(cn=margaret))
```

Get all entries without an e-mail attribute:

```
(!(mail=*))
```

Special characters

If any of the following special characters must appear in the search filter as literals, they must be replaced by the listed escape sequence.

Table 7: Special characters in Search filter

ASCII Character	Escape Sequence Substitute
*	\2a
(	\28
)	\29
\	\5c
NUL	\00

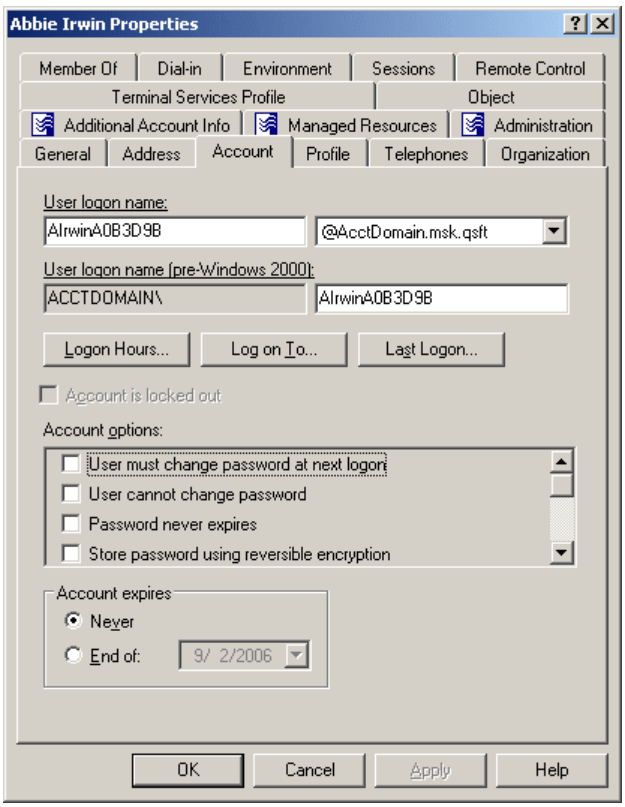
In addition, arbitrary binary data may be represented using the escape sequence syntax by encoding each byte of binary data with the backslash (\) followed by two hexadecimal digits. For example, the four-byte value 0x00000004 is encoded as \00\00\00\04 in a filter string.

Getting policy-related information

In object creation wizards and properties dialog boxes, some property labels may be displayed as hyperlinks. This indicates that Active Roles enforces policy restrictions on the property.

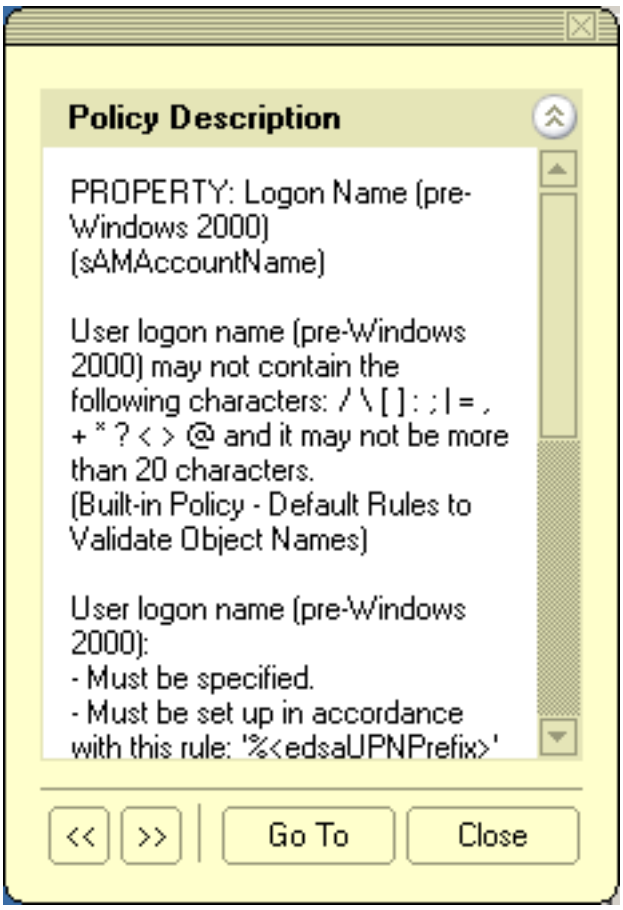
In the following figure, the **User logon name** and **User logon name (pre-Windows 2000)** labels are underlined, which means that these properties are under the control of a certain policy defined with Active Roles.

Figure 3: Policy related information



To examine the policy in detail, you can click the label. For example, if you click **User logon name (pre-Windows 2000)**, the Active Roles console presents you with a window similar to the following figure.

Figure 4: Policy description



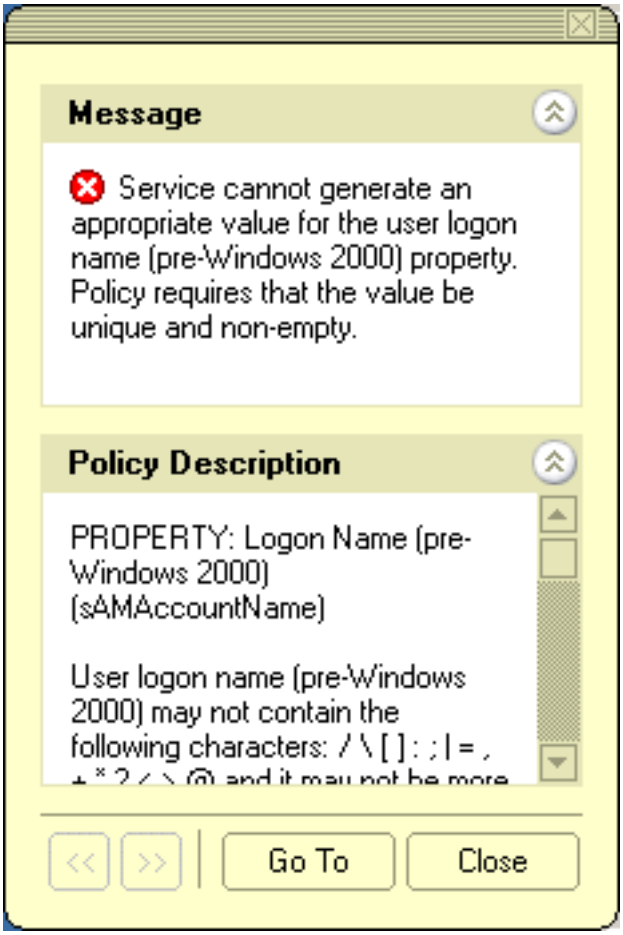
The window may display the following information:

- **Policy Description** Provides a brief description of the policy.
- **Message** Details the problem if the supplied property value violates the policy.

You can click arrows in the lower-left corner to display description of other policies enforced on the given property.

The **Message** section is displayed whenever the specified property value violates the policy. The following figure illustrates the situation where a value has not been supplied for a mandatory property.

Figure 5: Policy violation message



When you click **Go To** in this window, the console moves the pointer to the field that needs to be corrected. You can type or select an appropriate value to correct your input.

Performing Batch operations

In the Web Interface, you can select multiple objects (such as users, groups and computers), and then apply a certain command to your selection of objects. This allows you to perform a batch operation on all the selected objects at a time instead of executing the command on each object separately. The Web Interface supports the following batch operations:

- **Delete** Allows you to delete multiple objects at a time.
- **Deprovision** Allows you to deprovision multiple users or groups at a time.
- **Move** Allows you to move a batch of objects to a different organizational unit or container.

- **Add to groups** Allows you to add a batch of objects to one or more groups of your choice.
- **Update object attributes** Allows you to perform bulk attributes operations for multiple users at a time.
- **Reset Password** Allows you to reset the password for multiple users at a time.

Batch operations are available in the list of objects on the following Web Interface pages:

- **Search** This page lists the search results when you perform a search.
- **View Contents** This page displays the objects held in a given organizational unit, Managed Unit, or container.

To perform a batch operation, select the check box next to the name of each of the desired objects in the list, and then click a command in the top area of the Command pane. This executes the command on each object within your selection.

NOTE: Active Roles administrators can customize Web Interface by adding and removing commands, and modifying pages associated with commands. For more information, see "Customizing the Web Interface" in the *Active Roles Web Interface Administration Guide*.

Performing bulk operation

Active Roles Web interface enables you to perform bulk attributes operation for multiple users at a time.

To perform bulk attribute operation

1. On the Navigation bar, click **Directory Management**.
2. On the **Views** tab in the Browse pane, click the required container.
3. From the list of objects displayed for the selected container, select the required users for which you need to perform bulk attributes operation.

   The batch operations that can be performed on users are displayed in the Command pane.
4. In the Command pane, click **Update object attributes**.

   The **Update object attributes** window is displayed, which lists the user attributes that can be selected for bulk operation.
5. From the Attribute List tab, select the required attribute on which you want to perform the bulk operation, and click the + symbol.
6. On the **Update object attributes** dialog box that is displayed, in the **New Value** field, enter a value for the attribute, and click **OK**.

   The selected attribute with the updated value is displayed in the Select attribute table.

7. Repeat step 5 and step 6 to select and update more attributes, and then click **Next**.

   The **Preview tab | Operation Summary** section displays the summary of the selected attributes with the new values to be updated after the bulk operation is performed. To export the details, click **Export as CSV**.

8. Click **Finish**, to complete the bulk operation on the selected attributes for the multiple users.

NOTE:

- The bulk operation does not complete and an error is displayed if no attributes are selected or if no changes are made to the values of the attributes selected for the bulk operation.
- The bulk operation cannot be performed beyond 1500 users. However, you can configure the limit to increase the number of users. For more information on configuring the limit, see https://support.oneidentity.com/active-roles/kb/200735/not-able-to-query-or-update-groups-with-more-than-1500-members.

Performing bulk users password reset operation

Active Roles Web interface enables you to reset the password for multiple users at a time.

To perform bulk users password reset operation

1. On the Navigation bar, click **Directory Management**.
2. On the **Views** tab in the Browse pane, click the required container.
3. From the list of objects displayed for the selected container, select the required users for which you need to perform password reset operation.

   The batch operations that can be performed on users are displayed in the Command pane.

4. In the Command pane, click **Reset Password**.

   The Reset Password window is displayed.

5. On the General tab dialog box, click **Generate** to generate a new password for the selected users.
6. Under **Account** options, select the check box corresponding to the required rule to be applied for change of password, and then click **Save**.

   The password reset gets completed and the changes can be viewed on the selected user's Change History tab.

Active Roles service account minimum permissions

As Active Roles performs operations on objects on behalf of delegated users, the Active Roles service account which is used to manage the Active Directory domain requires adequate permissions. One Identity recommends to manage the domain using an account which is a member of the Domain Admins role group. If this configuration is not used, then guidance and documentation provided by One Identity may not be relevant.

Specify separate accounts for service and for managing the domain to separate the tasks performed by the service account from domain management. In this configuration scenario, the service account can be configured to run with the minimum permissions specified below, but the proxy account should be a member of the Domain Admins role group to stay within the One Identity Active Roles support model.

The service account credential has five main roles:

- Access to the Administration Service computer
- Service publication in Active Directory
- All script modules are executed under the security context of the Active Roles Service Account
- Connecting to the Microsoft SQL database
- Synchronizing native permissions to Active Directory, if Active Roles is configured

Access to the Administration Service computer

The service account must be a member of the Administrators group on the computer running the Administration Service.

Service publication in Active Directory

The Administration Service attempts to publish itself in the Active Directory. This enables Active Roles clients to automatically discover the Administration Service. This functionality is non-critical and if permissions are not granted, this will not prevent the service from functioning as expected, instead Active Roles clients won't automatically discover the Active Roles Administration Service. They will still be able to connect if the service name or IP address is available. Service publication requires that the service account have the following permissions on the Aelita sub-container of the System container in the domain of the computer running the Administration Service:

- Create Container Objects
- Create **serviceConnectionPoint** Objects
- Delete the **serviceConnectionPoint** objects in the System container
- Write permission for the keywords attribute of the **serviceConnectionPoint** objects in the System container

Along with the mentioned permissions, the service account (or the override account, if specified), must have these permissions on the Aelita sub-container of the System container in every managed domain. If an account has the domain administrator rights, then it has the required permissions by default. Otherwise, provide the permissions to the account by using the ADSI Edit console. The following instructions apply to the ADSI Edit console that ships with Windows Server 2012, Windows Server 2012 R2, Windows Server 2016, or Windows Server 2019.

To grant permissions for Administration Service publication in Active Directory

1. Open the ADSI Edit console and connect to the Domain naming context.
2. In the console tree, expand the System container, right-click the Aelita subcontainer, and then click Properties. If the Aelita container does not exist, create it: right-click System, point to New, click Object, and then, in the Create Object wizard, select the Container class and specify Aelita for the cn value.
3. On the Security tab in the Properties dialog box, click Advanced.
4. On the Permissions tab in the Advanced Security Settings dialog box, click Add.
5. On the Permission Entry page, configure the permission entry:
    - Click the Select a principal link, and select the desired account.
    - Verify that the Type box indicates Allow.
    - Verify that the Applies onto box indicates This object and all descendant objects.
    - In the Permissions area, select the Create container objects and Create serviceConnectionPoint objects check boxes.
    - Click OK
6. Click OK to close the Advanced Security Settings dialog box, and then click OK to close the Properties dialog box.

All script modules are executed under the security context of the Active Roles Service Account

The permissions needed by custom scripts will vary according to the needs of the scripts, and ideally should be reviewed on a case-by-case basis as a Best Practise security model.

Connecting to the Microsoft SQL database

In some configurations, assigning these permissions to the service account are optional, as a SQL Authentication credential may also be specified and the necessary permissions then be assigned to that SQL Authentication credential. For more information on the necessary SQL Server permissions, see *SQL Server Permissions* topic in the *Active Roles Quick Start Guide*.

Synchronizing native permissions to Active Directory

The service account must have the Read Permissions and Modify Permissions rights on the Active Directory objects and containers where it is desired to use the Active Roles security synchronization feature.

4

Rule-based Administrative Views

- About Managed Units
- Administering Managed Units
- Scenario: Implementing role-based administration across multiple OUs
- Deployment considerations

About Managed Units

Enterprises usually design their OU-based network structure on geographical or departmental boundaries, restricting the ability to delegate administration outside these boundaries. However, they can face situations that require objects to be grouped together in ways that differ to the OU structure.

Active Directory offers a comprehensive delegation model. However, since the scope of delegation is defined using Organizational Units, distributed administration in Active Directory is constrained by the OU structure.

In Active Directory, without changing the directory structure, it is impossible to re-group objects so that the new "groups" support inheritance for their members when delegating control or enforcing policy. As a solution to this inflexible, OU-based structure, Active Roles provides the facility to configure administrative views that meet any directory management needs. The administrative views (Managed Units) allow distributed administration to be independent of the OU hierarchy.

Thus, Active Roles provides Managed Units (MUs)—securable, flexible, rules-based administrative views. MUs represent dynamic virtual collections of objects of different types. MUs may include any directory objects, regardless of their location in the network. This allows objects to be grouped into administrative views that are independent of the OU-based structure.

Managed Units allow organizations to implement OU structures on a geographical basis, but distribute administration on a functional basis. For example, all users in a particular department, regardless of their location in different OUs, could be grouped into a single Managed Unit for the purposes of delegating access control and enforcing administrative policy. The members of that Managed Unit would remain in their geographically defined OUs, leaving the OU structure unaffected.

Managed Units make it possible to organize an enterprise in any particular way, without changing the underlying domain and OU structure. Managed Units can include directory objects from different domains, trees and forests, as well as from other Managed Units. In addition, different Managed Units can have common members. These features of Managed Units create an environment that is both secure and easy to manage.

How Managed Units work

Membership rules determine whether an object is a member of a certain MU. For example, you might specify a membership rule that states: all users from OU **A** whose full names start with **B** belong to this MU. The membership rule is then implemented as a query that searches OU **A** for users with full names starting with **B**. Active Roles stores the query as a part of the MU properties, and executes it whenever a list of MU members is created or refreshed.

Active Roles allows permission and policy settings to be specified at the level of Managed Units. Inheritance of permission and policy settings from the Managed Unit level works seamlessly across the Active Directory environment.

As the environment changes, the memberships of objects held in Managed Units also change automatically to adapt to the new environment, therefore object permission and policy settings change as well. Managed Units dynamically adapt to changes in the enterprise, simplifying the maintenance of permission and policy settings on directory objects.

Each Managed Unit provides a convenient scope for delegated administration. Delegated administrators no longer have to browse the hierarchy of OUs to search for managed objects. With Active Roles, administrative control of each MU can be delegated to specific individuals and groups, just as control of OUs can be delegated. Using Managed Units, all objects managed by a delegated administrator are located in one place.

Administering Managed Units

This section guides you through the Active Roles console to administer Managed Units. The following topics are covered:

- Creating a Managed Unit
- Displaying members of a Managed Unit
- Adding or removing members from a Managed Unit
- Copying a Managed Unit
- Exporting and importing a Managed Unit
- Renaming a Managed Unit
- Deleting a Managed Unit

Creating a Managed Unit

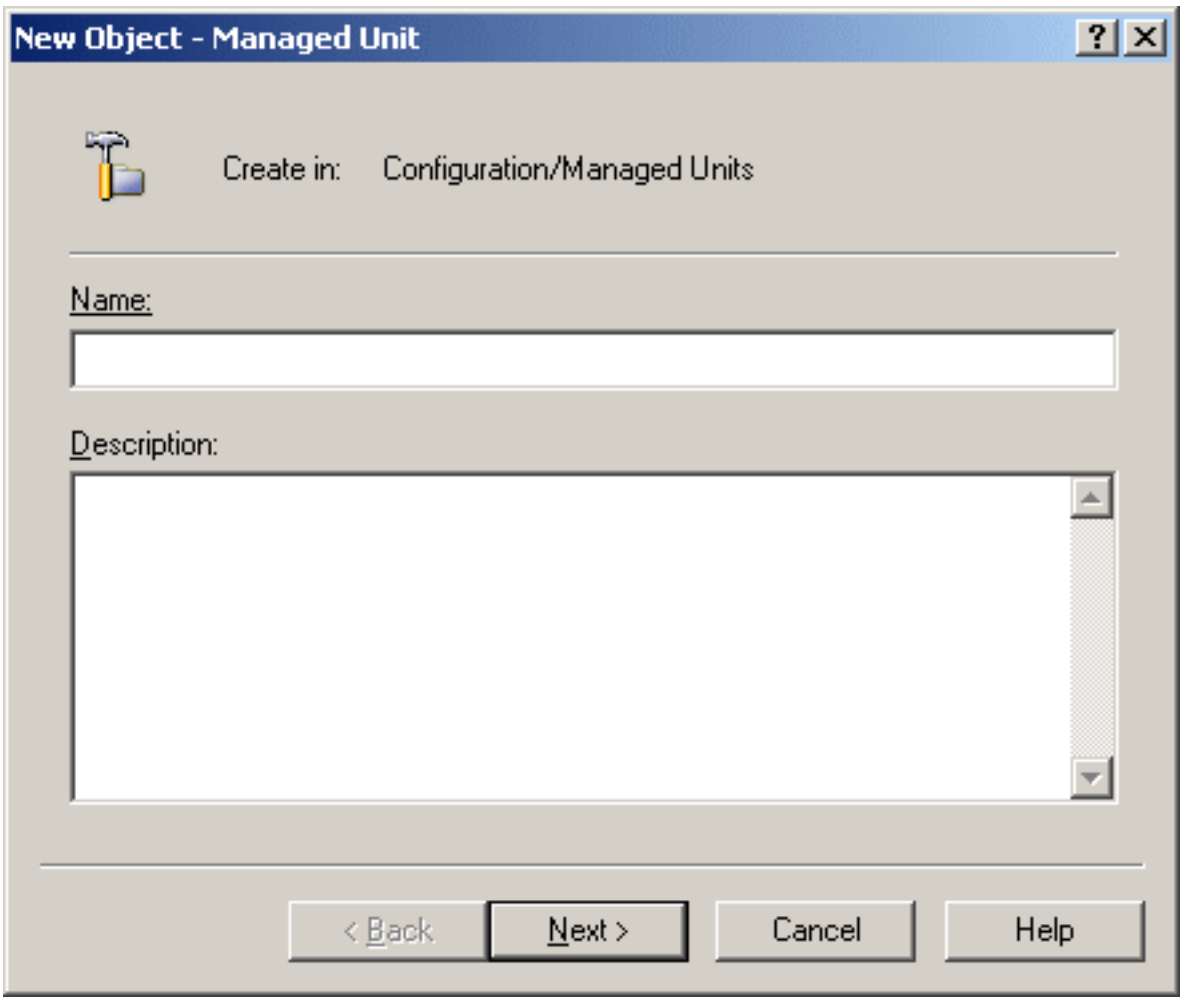
The Active Roles console provides the New Object – Managed Unit wizard to create Managed Units. You can start the wizard from the **Managed Units** container, located under **Configuration** in the console tree: right-click **Managed Units** in the console tree, and select **New | Managed Unit**.

If you need to manage a large number of Managed Units, it is advisable to create containers that hold only specified Managed Units for easy location: in the console tree, right-click **Managed Units** and select **New | Managed Unit Container**. Then, you can use the wizard to create a Managed Unit in that container: right-click the container and select **New | Managed Unit**.

> NOTE: Only users with administrative access to the Administration Service (members of the Active Roles Admin account) are permitted to create Managed Units. For more information about the Active Roles Admin account, refer to the *Active Roles Quick Start Guide*.

The first page of the wizard looks as shown in the following figure.

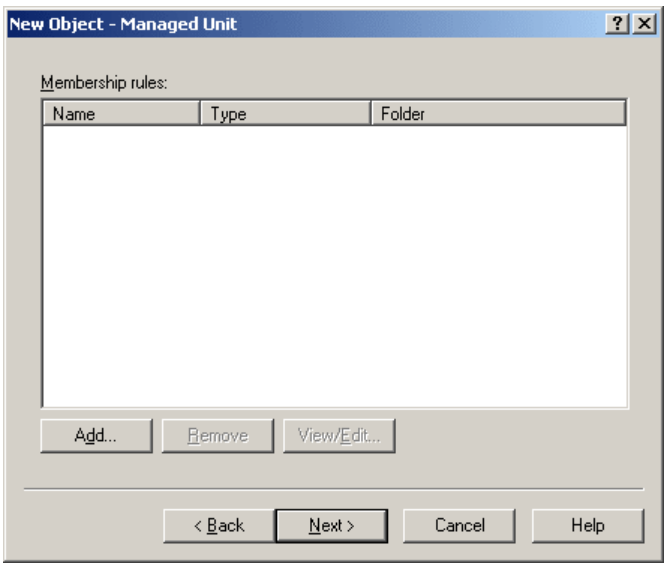
Figure 6: Managed unit - Name and Description



On this page, type in the name and description for the Managed Unit. The Active Roles console will display the name and description in the list of Managed Units in the details pane.

Click **Next**. The second page of the wizard looks as shown in the following figure.

Figure 7: Managed unit - include objects



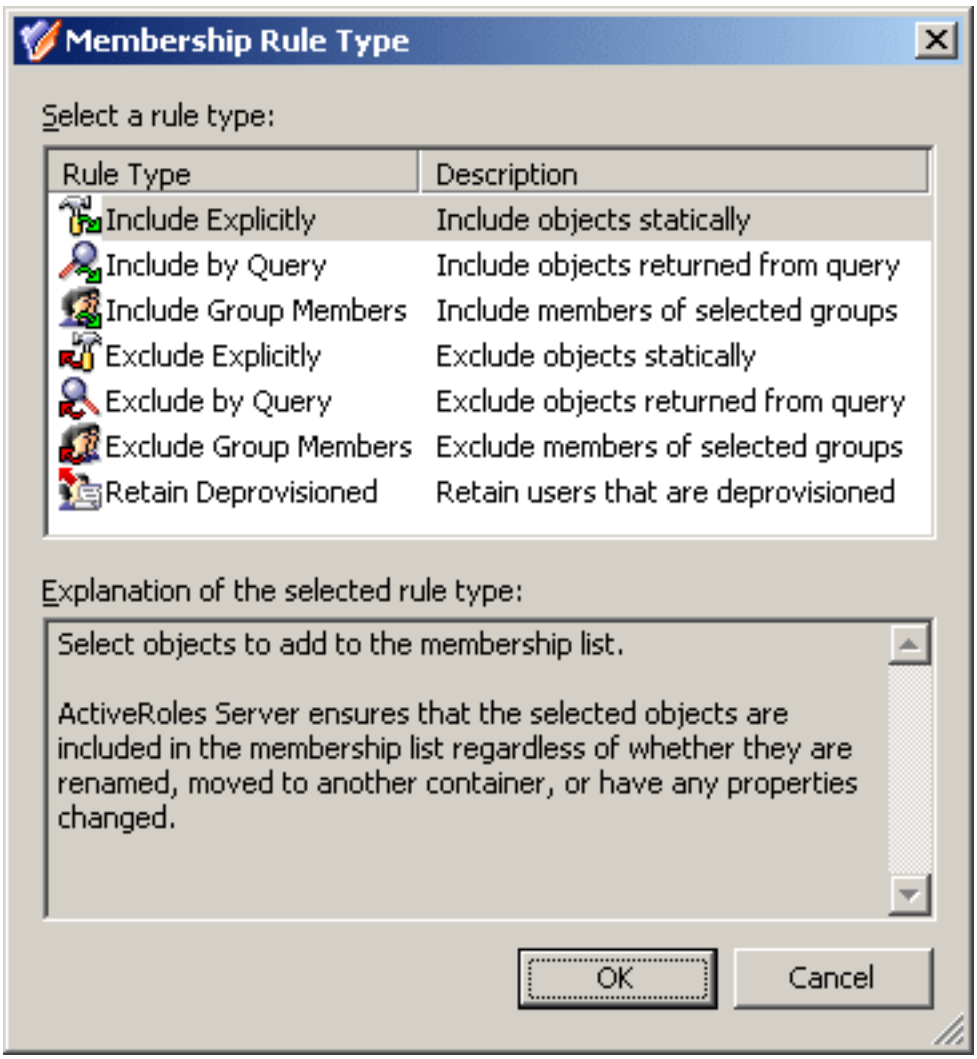
This page lets you specify which objects you want to be included in the Managed Unit.

Membership of a Managed Unit is determined by membership rules. Members of a Managed Unit are those objects that match criteria defined in the membership rules. A list of members is dynamically updateable: When you create a new object that satisfies the criteria in the membership rule, the object is included into the MU automatically. When an object no longer matches the criteria specified in the membership rule (for example, when the object is renamed or moved), it is automatically removed from the membership list.

A membership rule may take a form of search query, object static inclusion and exclusion rule, and group member's inclusion and exclusion rule.

To specify a membership rule, click **Add**. This displays the **Membership Rule Type** dialog box, shown in the following figure.

Figure 8: Managed Unit - membership rule type



In this dialog box, select a type of membership rule. In the lower box, you can read a description that explains which membership rules can be created using the selected type.

The **Include Explicitly** rule type allows you to select objects to be statically added to the Managed Unit. If you select a container, such as an OU, the entire sub-tree rooted in that container is included in the Managed Unit. Active Roles ensures that the selected objects are included in the Managed Unit regardless of whether they are renamed, moved to another container, or have any properties changed.

The **Exclude Explicitly** rule type allows you to select objects to be statically excluded from the Managed Unit. Active Roles ensures that the selected objects are excluded from the membership list regardless of whether they are renamed, moved, or have any properties changed. Because the **Exclude Explicitly** rule takes precedence over all other types of rule, the selected objects will be excluded from the Managed Unit even if another rule states that they should be included. Note that this rule type can be used to exclude only those objects that match one of the inclusion rules.

The **Include Group Members** rule type allows you to select the groups which members you want to include in the Managed Unit. Active Roles dynamically populates the membership list with the objects that belong to the selected groups. When an object is added or removed from the selected groups, Active Roles adds or removes that object from the membership list of the Managed Unit.

The **Exclude Group Members** rule type allows you to select groups whose members will be excluded from the Managed Unit. Active Roles ensures that the members of the selected groups are removed from the membership list of the Managed Unit. When an object is added to any one of the selected groups, Active Roles automatically removes that object from the membership list. Note that this rule type can be used to exclude only those objects that match one of the inclusion rules.

The **Include by Query** rule type allows you to define criteria the objects must match to be included in the Managed Unit. Active Roles dynamically populates the membership list with the objects that have certain properties. When an object is created, or when its properties are changed, Active Roles adds or removes it from the membership list depending on whether the objects' properties match the defined criteria.

The **Exclude by Query** rule type allows you to define criteria the objects must match to be excluded from the Managed Unit. Active Roles ensures that the objects with certain properties are excluded from the membership list. Active Roles automatically removes objects from the membership list depending on whether the objects' properties match the defined criteria. Note that this rule type can be used to exclude only those objects that match one of the inclusion rules.

The **Retain Deprovisioned** rule is intended to adjust the behavior of Managed Units towards deprovisioned objects, such as deprovisioned users or groups. Once an object is deprovisioned, the default behavior is to automatically remove that object from all Managed Units it was a member of. If there is a need to keep deprovisioned objects in certain Managed Units, you can satisfy this requirement by adding the **Retain Deprovisioned** rule to those Managed Units. This rule causes the Managed Unit to include both the regular and deprovisioned objects that meet the membership rules for that Managed Unit. Without this rule, the Managed Unit does not include any deprovisioned objects.

Note that the rules that exclude objects from a Managed Unit have an effect on only those objects that match one of the inclusion rules for that Managed Unit. For example, if a container object is explicitly included in a Managed Unit, all objects held in that container are also included in the Managed Unit and cannot be excluded by applying exclusion rules. An exclusion rule can only be used to exclude the entire container from the Managed Unit since the container is the only object that matches an inclusion rule. The objects that are held in the container do not match any inclusion rule, and therefore are not affected by exclusion rules.

In the **Membership Rule Type** dialog box, select a rule type, and click **OK**.

If you have selected the **Include Explicitly** or **Exclude Explicitly** rule type, the **Select Objects** dialog box is displayed. Select the objects you want to include or exclude from the Managed Unit, click **Add**, and then click **OK**.

If you have selected the **Include Group Members** or **Exclude Group Members** rule type, the **Select Objects** dialog box is displayed. The list of objects in that dialog box consists of groups. Select groups, click **Add**, and then click **OK**. All members of the selected groups will be included or excluded from the Managed Unit.

If you have selected the **Include by Query** or **Exclude by Query** rule type, the **Create Membership Rule** dialog box, similar to the **Find** dialog box, is displayed. In that dialog box, define the criteria that objects must match to be included or excluded from the Managed Unit.

After you have added one membership rule, you can add further membership rules for the same Managed Unit.

If you add several membership rules to the Managed Unit and some of them conflict with each other, then the conflict is resolved by a rule that defines the following order of precedence:

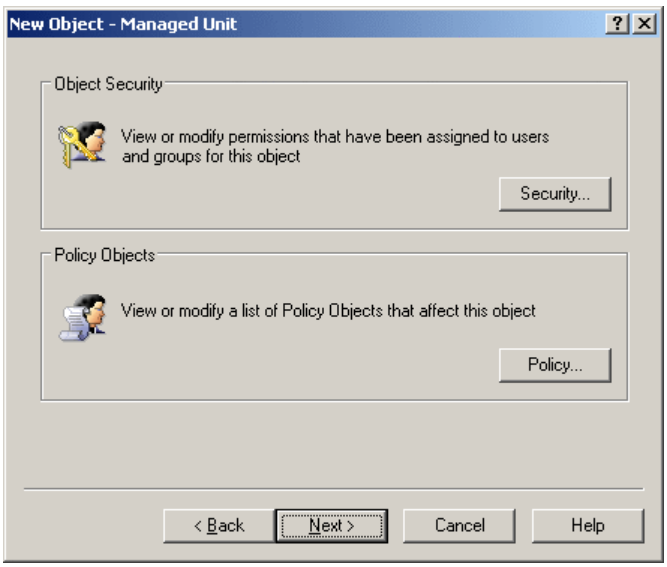
1. Exclude Explicitly
2. Include Explicitly
3. Exclude by Query
4. Exclude Group Members
5. Include by Query
6. Include Group Members

According to this, for example, the **Exclude Explicitly** rule takes precedence over all other types of rule. Therefore, the selected objects will be excluded from the Managed Unit even if another rule states that they should be included (for example, the objects that match the criteria defined in the **Include by Query** membership rule, or belong to a group selected in the **Include Group Members** rule).

> NOTE: An exclusion rule type can be used to exclude only those objects that match one of the inclusion rules. For example, if a given Organizational Unit is included in a Managed Unit by an inclusion rule, all child objects held in the Organizational Unit are also included in that Managed Unit. However, only the entire Organizational Unit rather than its individual child objects can be excluded from the Managed Unit.

Once you have added membership rules, click **Next**. This displays a page shown in the figure that follows.

Figure 9: Managed unit - Permission and Policy settings



You can use this page to specify the permission and policy settings for the Managed Unit. When finished, click **Next**, and then click **Finish**. For information about permission settings, see Applying Access Templates later in this document. For information about policy settings, see Applying Policy Objects later in this document.

Steps for creating a Managed Unit

To create a Managed Unit

1. In the console tree, under **Active Roles** | **Configuration** | **Managed Units**, locate and select the folder in which you want to add the Managed Unit.

   You can create a new folder as follows: Right-click Managed Units and select New | Managed Unit Container. Similarly, you can create a sub-folder in a folder: Right-click the folder and select New | Managed Unit Container

2. Right-click the folder, and select **New** | **Managed Unit** to start the New Object - Managed Unit wizard.
3. On the first page of the wizard, do the following, and then click **Next**:
   a. In the **Name** box, type a name for the Managed Unit.
   b. In the **Description** box, type any optional information about the Managed Unit.
4. On the second page of the wizard, click **Add**. This displays the **Membership Rule Type** dialog box.
5. Select the type of the membership rule to create, and then click **OK**:
   - To create a rule that statically adds members to the Managed Unit, click **Include Explicitly**.

- To create a rule that statically excludes members from the Managed Unit, click **Exclude Explicitly**.
- To create a rule that adds all members of a certain group to the Managed Unit, click **Include Group Members**.
- To create a rule that excludes all members of a certain group from the Managed Unit, click **Exclude Group Members**.
- To create a rule that populates the Managed Unit with the objects that match certain search criteria, click **Include by Query**.
- To create a rule that prevents the Managed Unit from including the objects that match certain search criteria, click **Exclude by Query**.
- To create a rule that prevents the deprovisioned objects, such as deprovisioned users or groups from being removed from the Managed Unit, click **Retain Deprovisioned**.

If you selected the Include by Query rule type or the Exclude by Query rule type in Step 5, the Create Membership Rule dialog box is displayed. Otherwise (except for the Retain Deprovisioned rule type), the Select Objects dialog box is displayed.

6. Complete the **Create Membership Rule** or **Select Objects** dialog box by following the instructions that are given later in this topic.
7. Repeat steps 4 through 6 until you have added all of the desired membership rules. Then, click **Next**.
8. On the next page of the wizard, do the following, and then click **Next**:
   - Click **Security** to specify permission settings on the Managed Unit.
   - Click **Policy** to specify policy settings on the Managed Unit.

   For information on how to specify security and policy settings, see Steps for modifying permission settings on a Managed Unit and Steps for modifying policy settings on a Managed Unit later in this document.
9. On the completion page of the wizard, click **Finish**.

To complete the Create Membership Rule dialog box

1. From the **Find** list, select the class of objects you want the membership rule to include or exclude from the Managed Unit. For example, when you select **Users**, the membership rule includes or excludes the users that match the conditions you specify.
2. From the **In** list, select the domain or folder that holds the objects you want the membership rule to include or exclude from the Managed Unit. For example, when you select an Organizational Unit, the membership rule includes or excludes only the objects that reside in that Organizational Unit.

   To add folders to the **In** list, click **Browse** and select folders in the **Browse for Container** dialog box.
3. Define the criteria of the membership rule. For example, to include or exclude the objects that have the letter T at the beginning of the name, type T in **Name**. You can

use an asterisk (*) to represent any string of characters.

4. Optionally, click **Preview Rule** to view a list of objects that match the criteria you have defined.
5. Click **Add Rule**.

To complete the Select Objects dialog box

1. In the **Look in** list, click the domain or folder that holds the objects you want to select. To add a folder to the list, click **Browse**.
2. Do one of the following, and then click **OK**:
3. In the list of objects, double-click the object you want to add.

   OR

4. In the lower box, type the entire name, or a part of the name, of the object you want to add. Then, click **Check Names**.

NOTE:

- You can also use the **Properties** command to add or remove membership rules from an existing Managed Unit: Right-click the Managed Unit, click **Properties**, and then click the **Membership Rules** tab in the **Properties** dialog box.
- For information on how to display a list of members of a Managed Unit, see Displaying members of a Managed Unit later in this document.
- The **Create Membership Rule** dialog box is similar to the **Find** dialog box you use to search for objects in the directory. Once you have specified your search criteria, the **Add Rule** function saves them as a membership rule. For more information on how to specify search criteria, see Finding objects earlier in this document.
- The **Find** list includes the **Custom Search** entry. Selecting that entry displays the **Custom Search** tab, enabling you to build custom membership rules using advanced options, as well as to build advanced membership rules using the Lightweight Directory Access Protocol (LDAP), which is the primary access protocol for Active Directory. For more information about using advanced search options, seeSteps for building a custom search and Steps for using advanced search options earlier in this document.

Steps for modifying Managed Unit properties

To modify properties of a Managed Unit

1. In the console tree, expand **Active Roles | Configuration | Managed Units**.
2. Under **Managed Units**, locate the Managed Unit you want to modify, right-click it, and click **Properties**.

3. Use the tabs in the **Properties** dialog box to view or modify properties of the Managed Unit.
4. When finished, click **OK**.

NOTE:

- The **Membership Rules** tab displays a list of membership rules for a given Managed Unit. You can add, remove, or modify membership rules as needed. For more information, see Adding or removing members from a Managed Unitlater in this document.
- On the **Administration** tab, you can use **Policy** to add and remove Policy Object links that determine which administrative policies are enforced on the Managed Unit. For more information, see Steps for modifying policy settings on a Managed Unit later in this document.
- On the **Administration** tab, you can use **Security** to add and remove Access Template links that define Trustees and their permissions for the Managed Unit. For more information, see Steps for modifying permission settings on a Managed Unit later in this document.

Steps for modifying permission settings on a Managed Unit

To modify permission settings on a Managed Unit

1. In the console tree, expand **Active Roles | Configuration | Managed Units**.
2. Under **Managed Units**, locate the Managed Unit you want to modify, right-click it, and click **Delegate Control**.
3. In the **Active Roles Security** dialog box, do the following:
   - To add permissions to the Managed Unit, click **Add** and follow the instructions in the Delegation of Control Wizard to create an Access Template link. For information on how to use the Delegation of Control wizard, see Applying Access Templates later in this document.
   - To remove permissions from the Managed Unit, select Access Template links from the list, and click **Remove**. Alternatively, you can revoke permissions by disabling Access Template links: Select one or more links, and then click **Disable**.
   - To view or modify properties of an Access Template link on the Managed Unit, select the link from the list and click **View/Edit**.
   - To modify an Access Template link so that the permissions defined by the link are also added to Active Directory, select the link from the list and click **Sync to AD**.
4. Click **OK** to close the **Active Roles Security** dialog box.

NOTE:

- The **Active Roles Security** dialog box displays a list of Access Template links, with each list item indicating a Trustee and the Access Template that is used to specify the Trustee's permissions.
- By default, the list of Access Template links displays all the links that determine the permission settings on the Managed Unit, regardless of whether a link was created on the Managed Unit itself or on a container that holds the Managed Unit. To change the display of the list, clear the **Show inherited** check box.
- An Access Template link can be removed from a Managed Unit if the link was created on that Managed Unit. Only the links that meet this condition are displayed when you clear the **Show inherited** check box, so you can remove them by clicking **Remove**.
- You can also use the advanced details pane to view, add, remove, or modify Access Template links on a Managed Unit: Select the Managed Unit, and then, on the **Active Roles Security** tab in the advanced details pane, right-click an Access Template link or a blank area, and use commands on the shortcut menu. For information about the advanced details pane, see Advanced pane earlier in this document.

Steps for modifying policy settings on a Managed Unit

To modify policy settings on a Managed Unit

1. In the console tree, expand **Active Roles | Configuration | Managed Units**.
2. Under **Managed Units**, locate the Managed Unit you want to modify, right-click it, and click **Enforce Policy**.
3. In the **Active Roles Policy** dialog box, do the following:
   - To add policies to the Managed Unit, click **Add** and select the Policy Object that defines the policies. You can select multiple Policy Objects at a time.
   - To remove policies from the Managed Unit, select the Policy Object that defines the policies, and click **Remove**. Alternatively, you can remove the effect of a Policy Object on the Managed Unit by selecting the **Blocked** check box next to the name of the Policy Object.
   - To modify policies, select the Policy Object that defines the policies, and click **View/Edit**.
4. Click **OK** to close the **Active Roles Policy** dialog box.

NOTE: The **Active Roles Policy** dialog box lists all the Policy Objects that define the policy settings on the Managed Unit, regardless of whether a Policy Object was added on the Managed Unit itself or on a container that holds the Managed Unit. You can view a list of Policy Objects that were added directly on the Managed Unit: Click **Advanced** and then clear the **Show inherited** check box.

Only the Policy Objects that were added directly on the Managed Unit can be removed. However, even if the **Remove** button is unavailable, you can select the **Blocked** check box. In this way, you remove the effect of the Policy Object on the Managed Unit. At any time, you can restore the effect of the Policy Object on the Managed Unit by clearing the **Blocked** check box.

You can also use the advanced details pane to add, remove, block, or modify Policy Objects that define the policy settings on a Managed Unit: Select the Managed Unit, and then, on the **Active Roles Policy** tab in the advanced details pane, right-click a Policy Object or a blank area, and use commands on the shortcut menu. For information about the advanced details pane, Advanced pane earlier in this document.

Displaying members of a Managed Unit

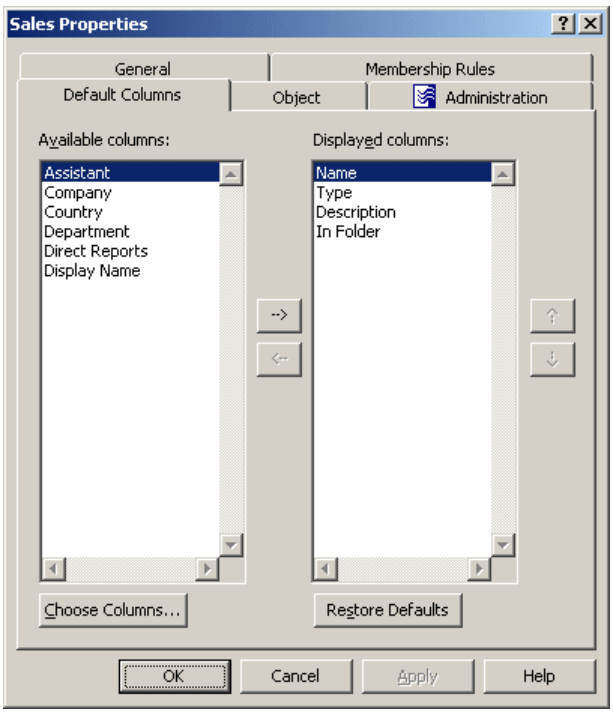
Members of a Managed Unit are objects that match the criteria specified in the membership rules for the Managed Unit.

To display the members of a Managed Unit, expand **Configuration/Managed Units** in the console tree, and then click a Managed Unit in the console tree. Members of the Managed Unit are displayed in the details pane.

For each Managed Unit, it is possible to preset an individual set of columns to display in the details pane. This allows you to customize administrative views on a per-Managed Unit basis.

To preset columns in the details pane for a given Managed Unit, right-click the Managed Unit, click **Properties**, and go to the **Default Columns** tab. The tab is similar to the following figure.

Figure 10: Managed Unit - Preset columns



You can add a column to display by double-clicking its name in the **Available columns** list. To add columns to the **Available Columns** list, click **Choose Columns**. In the **Choose Columns** dialog box, you can select columns and, if necessary, modify the names to be displayed in column headings.

Double-clicking a column name in **Available Columns** adds the name to the **Displayed Columns** list. Click **OK**. The new column is displayed in the details pane after refreshing the view. Right-click **Managed Units** in the console tree and click **Refresh**; then, select the Managed Unit in the console tree: the new column appears in the details pane.

Steps for displaying members of a Managed Unit

To display the members of a Managed Unit

1. In the console tree, expand **Active Roles | Configuration | Managed Units**.
2. Under **Managed Units**, locate and select the Managed Unit.

   The members of the Managed Unit are listed in the details pane.

To customize the list of Managed Unit members in the details pane

1. Right-click the Managed Unit, and click **Properties**.
2. In the **Properties** dialog box, click the **Default Columns** tab.
3. On the **Default Columns** tab, add or remove column names from the **Displayed**

Columns list.

4. Click **OK**.

NOTE:

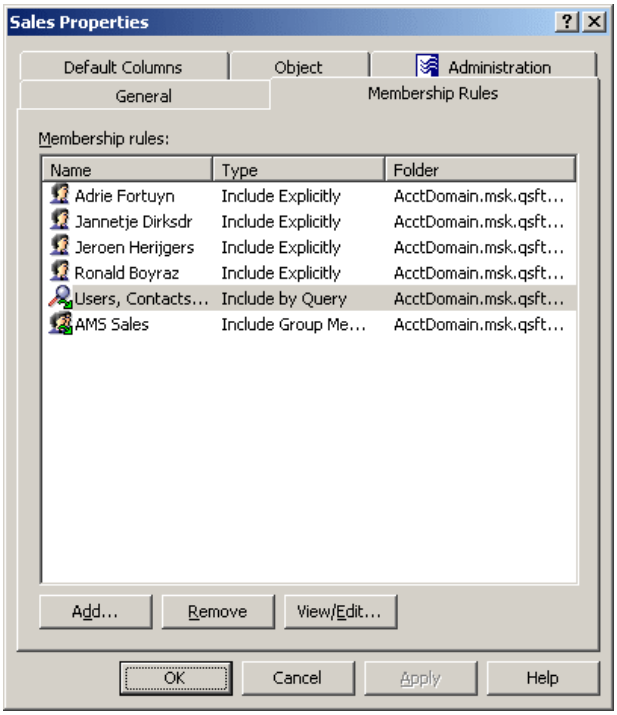
- For each Managed Unit, it is possible to configure an individual list of the default columns to display in the details pane, so you can perform the customization on a per-Managed Unit basis.
- You can populate the **Displayed columns** list by double-clicking column names in the **Available columns** list on the **Default Columns** tab. You can remove columns by double-clicking column names in the **Displayed columns** list.
- To add column items to the **Available Columns** list, click **Choose Columns**. In the **Choose Columns** dialog box, you can select columns and, if necessary, modify column names.
- For your changes to the **Displayed columns** list to take effect, the details pane needs to be refreshed: Right-click **Managed Units** in the console tree and click **Refresh**.

Adding or removing members from a Managed Unit

Members of a Managed Unit are defined by membership rules. Therefore, to add or remove members from a Managed Unit, you need to add, delete, or modify membership rules.

To add, delete or modify membership rules for a Managed Unit, display the **Properties** dialog box for that Managed Unit, and then click the **Membership Rules** tab. The tab is similar to the following figure.

Figure 11: Managed Unit - Adding or removing members



The **Membership Rules** tab displays a list of membership rules, with each entry indicating the name, type, and scope of the rule.

To add a membership rule, click **Add**. This displays the **Membership Rule Type** dialog box, discussed earlier in this chapter (see Creating a Managed Unit).

To modify a membership rule, select it from the **Membership rules** list, and click **View/Edit**. Only query-based rules can be modified in that way. If you select a rule of a different type, the **View/Edit** button is unavailable.

To delete a membership rule, select it from the **Membership rules** list, and click **Remove**.

As you add, modify or delete membership rules, the list of Managed Unit members automatically changes.

Steps for adding membership rules to a Managed Unit

To add a membership rule to a Managed Unit

1. In the console tree, expand **Active Roles | Configuration | Managed Units**.
2. Under **Managed Units**, locate the Managed Unit you want to modify, right-click it, and click **Properties**.

3. On the **Membership Rules** tab, click **Add**. This displays the **Membership Rule Type** dialog box.
4. Select the type of the membership rule you want to create. Do one of the following, and then click **OK**:
   - To create a rule that statically adds members to the Managed Unit, click **Include Explicitly**.
   - To create a rule that statically excludes members from the Managed Unit, click **Exclude Explicitly**.
   - To create a rule that adds all members of a certain group to the Managed Unit, click **Include Group Members**.
   - To create a rule that excludes all members of a certain group from the Managed Unit, click **Exclude Group Members**.
   - To create a rule that populates the Managed Unit with the objects that match certain search criteria, click **Include by Query**.
   - To create a rule that prevents the Managed Unit from including the objects that match certain search criteria, click **Exclude by Query**.
   - To create a rule that prevents the deprovisioned objects, such as deprovisioned users or groups, from being removed from the Managed Unit, click **Retain Deprovisioned**.

   If you select the Include by Query rule type or the Exclude by Query rule type in Step 5, the Create Membership Rule dialog box is displayed. Otherwise (except for the Retain Deprovisioned rule type), the Select Objects dialog box is displayed.
5. Complete the **Create Membership Rule** or **Select Objects** dialog box by following the instructions that are given later in this topic.
6. Click **OK** to close the **Properties** dialog box.

To complete the Create Membership Rule dialog box

1. From the **Find** list, select the class of objects you want the membership rule to include or exclude from the Managed Unit. For example, when you select **Users**, the membership rule includes or excludes the users that match the conditions you specify.
2. From the **In** list, select the domain or container that holds the objects you want the membership rule to include or exclude from the Managed Unit. To add folders to the **In** list, click **Browse**.
3. Define the criteria of the membership rule. For example, to include or exclude the objects that have the letter T at the beginning of the name, type T in **Name**. You can use an asterisk (*) to represent any string of characters.
4. Optionally, click **Preview Rule** to view a list of objects that match the criteria you have defined.
5. Click **Add Rule**.

To complete the Select Object dialog box

1. In the **Look in** list, click the domain or folder that holds the objects you want to select. To add a folder to the list, click **Browse**.
2. Do one of the following, and then click **OK**:
3. In the list of objects, double-click the object you want to add.
4. In the lower box, type the entire name, or a part of the name, of the object you want to add. Then, click **Check Names**.

NOTE:

- The only way to populate Managed Units is by adding membership rules. The members of a Managed Unit are the objects that match the criteria defined by the membership rules.
- To display members of a Managed Unit, click the Managed Unit in the console tree. The members of the Managed Unit are displayed in the details pane.
- The **Create Membership Rule** dialog box is similar to the **Find** dialog box you use to search for objects in the directory. Once you have specified your search criteria, Active Roles allows you to save them as a membership rule, forcing the membership list to include the objects that match the search criteria. For instructions on how to specify search criteria in the **Create Membership Rule** dialog box, see Finding objects earlier in this document.
- The **Find** list includes the **Custom Search** entry. Selecting that entry displays the **Custom Search** tab, enabling you to build custom membership rules using advanced options, as well as to build advanced membership rules using the Lightweight Directory Access Protocol (LDAP), which is the primary access protocol for Active Directory. For more information about using advanced search options, see Steps for building a custom search and Steps for using advanced search options earlier in this document.

Steps for removing membership rules from a Managed Unit

To remove a membership rule from a Managed Unit

1. In the console tree, expand **Active Roles | Configuration | Managed Units**.
2. Under **Managed Units**, locate the Managed Unit you want to modify, right-click it, and click **Properties**.
3. On the **Membership Rules** tab, select the membership rule you want to remove, and then click **Remove**.

Steps for including a member to a Managed Unit

To include a member to a Managed Unit

1. In the console tree, expand **Active Roles | Configuration | Managed Units**.
2. Under **Managed Units**, locate the Managed Unit you want to modify, right-click it, and click **Properties**.
3. On the **Membership Rules** tab, click **Add**. The **Membership Rule Type** dialog box appears.
4. In the **Membership Rule Type** dialog box, click **Include Explicitly**, and then click **OK**. The **Select Objects** dialog box appears.
5. Use the **Select Objects** dialog box to locate and select the object (or objects) you want to explicitly include in the Managed Unit.

   For general instructions on how to configure membership rules, see Steps for adding membership rules to a Managed Unit earlier in this document.
6. Click **OK** to close the **Properties** dialog box.

Steps for excluding a member from a Managed Unit

To exclude a member from a Managed Unit

1. In the console tree, expand **Active Roles | Configuration | Managed Units**.
2. Under **Managed Units**, locate the Managed Unit you want to modify, right-click it, and click **Properties**.
3. On the **Membership Rules** tab, click **Add**. The **Membership Rule Type** dialog box appears.
4. In the **Membership Rule Type** dialog box, click **Exclude Explicitly**, and then click **OK**. The **Select Objects** dialog box appears.
5. Use the **Select Objects** dialog box to locate and select the object (or objects) you want to explicitly exclude from the Managed Unit.

   For general instructions on how to configure membership rules, see Steps for adding membership rules to a Managed Unit earlier in this document.
6. Click **OK** to close the **Properties** dialog box.

Steps for adding group members to a Managed Unit

To add group members to a Managed Unit

1. In the console tree, expand **Active Roles | Configuration | Managed Units**.
2. Under **Managed Units**, locate the Managed Unit you want to modify, right-click it, and click **Properties**.
3. On the **Membership Rules** tab, click **Add**. The **Membership Rule Type** dialog box appears.
4. In the **Membership Rule Type** dialog box, click **Include Group Members**, and then click **OK**. The **Select Objects** dialog box appears.
5. Use the **Select Objects** dialog box to locate and select the group (or groups) whose members you want to be included in the Managed Unit.

   For general instructions on how to configure membership rules, see Steps for adding membership rules to a Managed Unit earlier in this document.
6. Click **OK** to close the **Properties** dialog box.

Steps for removing group members from a Managed Unit

To remove group members from a Managed Unit

1. In the console tree, expand **Active Roles | Configuration | Managed Units**.
2. Under **Managed Units**, locate the Managed Unit you want to modify, right-click it, and click **Properties**.
3. On the **Membership Rules** tab, click **Add**. The **Membership Rule Type** dialog box appears.
4. In the **Membership Rule Type** dialog box, click **Exclude Group Members**, and then click **OK**. The **Select Objects** dialog box appears.
5. Use the **Select Objects** dialog box to locate and select the group (or groups) whose members you want to be excluded from the Managed Unit.

   For general instructions on how to configure membership rules, see Steps for adding membership rules to a Managed Unit earlier in this document.
6. Click **OK** to close the **Properties** dialog box.

Copying a Managed Unit

With the Active Roles console, you can create copies of Managed Units. This feature helps you re-use existing Managed Units.

To create a copy of a Managed Unit, right-click the Managed Unit, and click **Copy**. This opens the Copy Object – Managed Unit wizard. You can complete the wizard by following the instructions in the Creating a Managed Unit section, earlier in this chapter.

Steps for copying a Managed Unit

To copy a Managed Unit

1. In the console tree, expand **Active Roles | Configuration | Managed Units**.
2. Under **Managed Units**, locate the Managed Unit you want to copy.
3. Right-click the Managed Unit, and then click **Copy**. The Copy Object - Managed Unit wizard starts.
4. On the first page of the wizard, do the following, and then click **Next**:
   a. In the **Name** box, type a name for the Managed Unit.
   b. In the **Description** box, type any optional information about the Managed Unit.
5. On the second page of the wizard, you can add, remove, and modify the membership rules that were copied from the original Managed Unit. Do the following:
   - To add a membership rule to the new Managed Unit, click **Add**.
   - To remove a membership rule from the new Managed Unit, select the membership rule from the list, and click **Remove**.
   - To modify a membership rule for the new Managed Unit, select the membership rule from the list, and click **View/Edit**.

   For instructions on how to configure a membership rule, see Steps for adding membership rules to a Managed Unit earlier in this document.
6. Click **Next**.
7. On the next page of the wizard, do the following:
   - Click **Security** to specify permission settings on the Managed Unit.
   - Click **Policy** to specify policy settings on the Managed Unit.

   For instructions on how to specify security and policy settings, see Steps for modifying permission settings on a Managed Unit and Steps for modifying policy settings on a Managed Unit earlier in this document.
8. Click **Next**, and then lick **Finish**.

NOTE: The membership rules, permission settings, and policy settings are copied from the original Managed Unit and can be modified in the Copy Object - Managed Unit wizard.

Exporting and importing a Managed Unit

With the Active Roles console, you can export Managed Units to an XML file and then import them from that file to populate another instance of Active Roles. The export and import operations provide a way to move Managed Units from a test environment to a production environment.

> NOTE: When you export and then import a Managed Unit, only membership rules are transferred along with other properties of the Managed Unit. The permission and policy settings of the Managed Unit are not exported. You need to reconfigure them manually after you import the Managed Unit.

To export Managed Units, select them, right-click the selection, and select **All Tasks | Export**. In the **Export Objects** dialog box, specify the file where you want to save the data, and click **Save**.

To import Managed Units, right-click the container where you want to place the Managed Units, and then click **Import**. In the **Import Directory Objects** dialog box, select the file to which the Managed Units were exported, and click **Open**.

Renaming a Managed Unit

To rename a Managed Unit, right-click the Managed Unit, and click **Rename**. Type the new name, and then press ENTER.

Renaming a Managed Unit does not affect the membership rules, permission settings, or policy settings associated with the Managed Unit.

Steps for renaming a managed Unit

To rename a Managed Unit

1. In the console tree, expand **Active Roles | Configuration | Managed Units**.
2. Under **Managed Units**, locate the Managed Unit you want to rename, right-click it, and click **Rename**.
3. Type a new name, and then press ENTER.

Deleting a Managed Unit

To delete a Managed Unit, right-click the Managed Unit, and then click **Delete**.

When you delete a Managed Unit, the objects held in the Managed Unit are not deleted. The deletion erases the membership rules, permission settings, and policy settings associated with the Managed Unit.

Steps for deleting a Managed Unit

To delete a Managed Unit

1. In the console tree, expand **Active Roles | Configuration | Managed Units**.
2. Under **Managed Units**, locate the Managed Unit you want to delete, right-click it, and then click **Delete**.

NOTE: When you delete a Managed Unit, its members are not deleted. However, the permission settings and the policy settings that were specified via the Managed Unit are no longer in effect after the Managed Unit has been deleted.

Scenario: Implementing role-based administration across multiple OUs

This scenario involves the creation of an administrative view named **Sales** in an organization with an OU-based structure of Active Directory.

Suppose an organization has offices in USA and Canada. The rule for including a user in an OU is the geographical location of the user. Therefore, all users who work in USA reside in the **USA** OU, and those working in Canada reside in the **Canada** OU.

The offices in USA and Canada each have **Marketing**, **Development**, and **Sales** departments. By creating a **Sales** MU, it is possible to manage users from the **Sales** departments in USA and Canada collectively, without changing the actual OU-based structure.

When delegating control of an MU, all users that belong to the MU inherit security settings defined at the level of the Managed Unit. Thus, applying an Access Template to a Managed Unit specifies the security settings for each user in the MU.

To implement this scenario, perform the following steps:

1. Create the **Sales** MU.
2. Add users from the **Sales** department in USA and Canada to the **Sales** MU.
3. Prepare the **Sales** Access Template.
4. Apply the **Sales** Access Template to the **Sales** MU, and designate an appropriate group as a Trustee.

As a result, the members of the group gain control of user accounts that belong to the **Sales** MU. The scope of control is defined by the permissions in the **Sales** Access Template.

The following sections elaborate on the steps to implement this scenario.

Step 1: Creating the Managed Unit

The first step is to create the **Sales** Managed Unit. For information on how to create a Managed Unit, see Creating a Managed Unit earlier in this chapter.

Step 2: Adding users to the Managed Unit

When the **Sales** Managed Unit is prepared, add users from the **Sales** departments across the company.

Suppose that all users from the **Sales** departments (in both USA and Canada) have the **Description** property set to **Sales**.

Create a membership rule of the **Include by Query** type with the following parameters: from the **Find** list, select **Users**; in the **Description** box, type **Sales**. As a result, all users with the description **Sales** will be included in the Managed Unit.

For more information on how to create membership rules, see Adding or removing members from a Managed Unit earlier in this chapter.

Step 3: Preparing the Access Template

To define which rights the Trustee will get for the **Sales** Managed Unit, create a **Sales** Access Template, and add permissions to this Access Template.

For more information on how to create an Access Template, see Creating an Access Template later in this document.

Step 4: Applying the Access Template

To apply the **Sales** Access Template to the **Sales** Managed Unit, right-click the **Sales** Managed Unit and click **Delegate Control**. Then, click the **Add** button and follow the instructions in the Delegation of Control wizard.

On the **Users or Groups** page of the wizard, add the user or group to be designated as a Trustee.

On the **Access Templates** page of the wizard, select the **Sales** Access Template you prepared in Step 3.

For more information on how to apply an Access Template to a Managed Unit, see Applying Access Templates later in this document.

Deployment considerations

Managed Units can best be described as virtual Organizational Units, allowing you to take advantage of delegation and policy application within a logical grouping of objects that may not always correspond with your Active Directory structure. At their rawest form, Managed Units are nothing more than LDAP queries stored in Active Roles' configuration database. As such, Managed Units can only be configured and accessed via Active Roles' interfaces. This section covers the following topics, to help you make the best use of Managed Units:

- Managed Unit membership rules
- Delegation of Managed Units

Managed Unit membership rules

It is membership rules that determine the list of objects to be included in a Managed Unit. Although several types of membership rule are available, there are two that are most commonly used. These are query-based inclusion or exclusion rules and explicit inclusion or exclusion rules.

Typically you would use query-based rules to include objects that span multiple Organizational Units or Organizational Unit structures. An example is a Managed Unit that includes all disabled user accounts or a Managed Unit that includes all user accounts without mailboxes. Query-based rules are also used to build logical structures from a flat Organizational Unit structure.

You have to be careful with query-based rules because in essence these are conditions imposed on object attributes. If the value of an object's attribute does not meet the specified conditions, the object is not included in the Managed Unit. The opposite is also true. If you, for example, configure a Managed Unit to include all users whose name begins with letter A, the Managed Unit would include the Administrator account. If the Helpdesk were delegated control over that Managed Unit, Helpdesk operators could gain control over the Administrator account. This brings in the use for explicit rules.

Explicit rules allow you to include or exclude objects based upon their identifier (GUID). So no matter how the object is changed or renamed, as long as that object exists in the directory, the rule will be in effect. Explicit rules normally complement query-based rules to include an object the query does not cover or to exclude an object that may meet the conditions of the query. Other uses are to statically include an object so no matter what that object is named it will always be included. Most typically this is Organizational Units. You can build a logical structure of Organizational Units from any part of the directory tree by explicitly adding them to a Managed Unit. This makes delegation and policy application much easier, since either can be done at the Managed Unit level instead of each individual Organizational Unit.

The following table lists some useful examples of membership rules. These examples demonstrate how to control membership rules by using LDAP filters. You can apply an LDAP filter under the **Custom Search** option in either of the query-based rule types.

Table 8: Managed Unit membership rules

Managed Unit Contents	LDAP Filter
Groups hidden from the Exchange Address List	(&(objectCategory=group)(mailnickName=*)(msExchHideFromAddressLists=True))
Mail-enabled groups	(&(objectCategory=group)(mailNickName=*))
Mail-enabled users with forwarders set	(&(sAMAccountType=805306368)(mailNickName=*)(altRecipient=*))
Users who do not have an Exchange mailbox	(&(sAMAccountType=805306368)(!(homeMDB=*)))
Distribution groups	(&(objectCategory=group)(!(groupType:1.2.840.113556.1.4.803:=2147483648)))
Security groups	(&(objectCategory=group)(groupType:1.2.840.113556.1.4.803:=2147483648))
Disabled user accounts	(&(objectCategory=user)(userAccountControl:1.2.840.113556.1.4.803:=2))
Users from the Sales department	(&(objectCategory=user)(department=Sales))

Delegation of Managed Units

You can delegate Managed Units exactly like Organizational Units or an entire domain, by applying Access Templates in Active Roles. This can drastically expedite deployment, ease administrative burden for Active Roles itself, and simplify the training and job processes for the administrators using this tool.

For instance, by grouping all disabled and locked out accounts within a single Managed Unit, you can delegate control to a Helpdesk group so that they can quickly and easily perform a large part of their job function by only having to enumerate and look through a single structure. Also, when delegating control of a Managed Unit, you do not have to give your delegated administrators access to any Organizational Unit itself; all objects that meet the membership rules are in the Managed Unit regardless of what Organizational Units hold those objects.

One more example would be where Active Directory has a very flat structure of Organizational Units, however different administrators are responsible for different locations. As long as the location code is stored in an attribute of the objects to be managed, you can create Managed Units based on that attribute, and delegate to each set of administrators a Managed Unit containing their respective objects that meet a particular location code. Since Managed Units are merely groupings of objects based on certain

attributes, the objects will move in and out of Managed Units regardless of how their attributes change, either through Active Roles interfaces or natively.

An important feature of Managed Units is the fact that a single Managed Unit can include objects from any domains managed by Active Roles (managed domains). As long as a given domain is registered with Active Roles, regardless of the domain's forest, any object from that domain can be added to a Managed Unit. When doing delegation of a Managed Unit that holds objects from different domains, it is advisable to use domain local groups from the domain where the Active Roles installation exists, or universal groups. This is because Active Roles allows you to do delegation to any security group within the managed domains; however, if the Active Roles installation exists in domain A and a delegation was done to a domain local group in domain B, an administrator who authenticates against Active Roles in domain A will never have the local group from domain B added to his security token, therefore he will not have his delegated rights. Also global groups can be used as long as all administrative users reside in the domain where Active Roles is installed.

One precaution that must be considered with delegating control of Managed Units is the ability to sync the delegated permissions to Active Directory. When you apply an Access Template to a Managed Unit and do not sync the permissions with Active Directory, the permission settings are only stored within Active Roles' configuration database. Active Roles maintains the parent-child relationship for the objects held in the Managed Unit, thus allowing permission inheritance to work. If you choose to sync the permissions with Active Directory, there is no way to maintain that parent-child relationship in Active Directory since a Managed Unit is not truly an object within Active Directory so to Active Directory that parent does not exist. As a result, every permission entry found in the Access Template will be included into the native Access Control List of every object held in the Managed Unit. Potentially this could cause performance issues.

Working with Federated Authentication

Federated Authentication allows users to access the application or web sites by authenticating them against a certain set of rules, known as claims. The authentication ticket or the token is used to validate the user across multiple application, web sites, or IT systems.

Claim-based authentication is a method to acquire the user identity related information on both on-premises and cloud-based products. A single token is created based on the predefined claims to identify the users trying to access the applications or web site. After the identification of the user is complete, a security token service is used to identify the type of user.

Active Roles supports federated authentication with Security Assertion Markup Language (SAML), through which you can sign in to an application once using the single sign-on option and you are authenticated to access websites.

For more information, see Appendix E: Enabling Federated Authentication

NOTE: While switching between the STS providers, restart IIS and clear the browser cache.

Configuring Federated Authentication settings

To configure the Federated Authentication settings, configure the **Identity provider configuration**, set claims in the **Claim editor**, and provide the **Domain user login credentials**.

Prerequisite

Ensure that the user has appropriate permissions to query the Active Directory. The same user must be provided in the **Domain user login credentials** field.

To set identity provider configuration

1. In the Configuration Center main window, click **Web Interface**.

   The **Web Interface** page displays all the Web interface sites that are deployed on the Web server running the Web interface.
2. To configure the federated authentication settings, click **Authentication**.

   The **Site authentication settings** page is displayed.

   NOTE: By default, the **Default** windows authentication settings is configured.
3. To configure the federated authentication settings, click **Federated**.
4. In the **Identity provider configuration** section, select the security **Identity provider** from the Identity provider drop-down menu. The available options are **Azure**, **ADFS**, **RSTS**, and **Custom**.
5. Select the required additional options from **Options**.
6. Provide a valid URL in the **Federated metadata URL** field.

   NOTE: A federation metadata document is an XML document that conforms to the WS-Federation 1.2 schema. It exposes all data required for an STS implementer.
7. To test the connection, click **Test metadata**.

   If the connection is successful, a message is displayed.
8. To view the metadata URL, click **Yes**. To proceed further with the settings, click **No**.
9. If you select the **Token encryption** from **Options**, you must enter the **certificate thumbprint** manually. If the option is not selected, this field is not available.

   NOTE: The **certificate thumbprint** key must be entered manually. Copying the key and pasting in the field is not supported.

10. Provide the Realm URL of the requesting realm in the **Realm** field.
11. Provide the URL to send a response in the **Reply URL** field. A URL that identifies the address at which the relying party (RP) application receives replies from the Security Token Service (STS).

To set claims in the claim editor

IMPORTANT: By default, the priority of the claim is set based on the order the claims are created. The claim created first has the first priority, the claim created next has the secondary priority, and so on. However, you can move the claims based on the required priority.

1. In the **Claim editor** section, to add claims, click **Add**.

   Add claim window is displayed.
2. Select the type of claim from the **Claim type** drop-down menu.

   IMPORTANT: **UPN**, **SID**, and **email** claims are supported.
3. Select the **Claim value**.
4. Provide a name for the claim in the **Display name** field.
5. Provide a description in the **Claim description** field.
6. Click **Save**.

   The claim is added successfully.

NOTE: You can modify or remove the claims that are created.

To provide the Domain user login credentials

1. In the **Domain user login credentials** section, provide the valid credentials in the **Username** and **Password** fields.
2. Click **Modify** to update the authentication settings.

   A message is displayed about the successful completion of the operation.

After you click **Modify**, the ARSWeb is modified and is ready for federated authentication.

For more information on using the Federated Authentication feature in multihop scenario, see Appendix E: Enabling Federated Authentication

5

Role-based Administration

- Access Templates as administrative roles
- Access Template management tasks
- Examples of use
- Deployment considerations
- Windows claims-based Access Rules

Access Templates as administrative roles

Active Roles provides safe, distributed administration through advanced delegation of rights with very high granularity to individual users or groups. This relieves highly skilled administrators from routine day-to-day tasks, saving time and increasing productivity. For example, an administrator can allow the Help Desk to perform specific tasks, such as resetting passwords or managing group memberships, without granting full administrative privileges.

As you develop your administration and security design, you define delegated administrators (Trustees) and administrative roles (Access Templates). Then, you define Managed Units and apply Access Templates, designating Trustees for each Managed Unit. You can also apply Access Templates to objects and folders in Active Directory, assigning the permissions to the necessary Trustees. This three-way relationship between Trustees, Access Templates, and managed objects is central to the implementation of your role-based administration model.

The Active Directory Users and Computers tool provides the facility to delegate administrative responsibilities. However, every time you want to delegate rights, you need to define a set of permissions. This makes the delegation procedure time-consuming and prone to errors. Active Roles overcomes this problem by consolidating permissions into customizable administrative roles—Access Templates. The logical grouping of permissions simplifies the management of delegation settings.

Access Templates are collections of permissions representing administrative roles. Permissions are used to allow or deny certain administrative operations to a user or group.

You can create an Access Template that incorporates all permissions required to perform a particular administrative role.

To assign the role to a user or group, you should link the Access Template to a Managed Unit, Organizational Unit, domain, or individual object, depending on the scope of the role, and then select a user or group to designate as a Trustee. As a result, the individual user, or each member of the group, acquires the rights specified by the role to administer objects that reside in the collection or folder to which the Access Template has been linked.

How Access Templates work

Active Roles implements delegated administration by linking Access Templates to collections of objects (Managed Units), directory folders (containers), or individual (leaf) objects.

When applied to a directory object, an Access Template specifies permission settings for that object and its child objects. Applying Access Templates to Managed Units is a convenient way to manage permissions on collections of directory objects.

Each Access Template is applied in relation to some users and/or groups (Trustees), and the permissions specified in the Access Template determine their access to managed objects. When an Access Template is modified or no longer applied, permissions set for the directory objects are modified accordingly.

When permissions on a Managed Unit change, Active Roles recalculates the permission settings on all the Managed Unit members. Likewise, the permission information is modified whenever the list of objects in a Managed Unit changes. When objects join or leave a Managed Unit (due to object property changes, for example), all permission settings on those objects are recalculated.

Every object inherits its permission settings from the Managed Units in which it resides. For example, if a Trustee has permissions to access multiple Managed Units that hold a given object, the Trustee's permissions to access that object are simply defined as a union of all permissions specified at the Managed Unit level.

Applying Access Templates to a container object (directory folder) establishes the Trustee's access to both the container and its child objects. The Trustee, having permissions specified over a container, possesses inherited permissions for the child objects residing in the container.

Security synchronization

Permissions defined in an Access Template can be propagated to Active Directory, with all changes made to them in Active Roles being automatically synchronized to Active Directory.

By enabling synchronization from Active Roles security to Active Directory native security, Active Roles provides the facility to specify Active Directory security settings with Access Templates. Access Templates simplify and enhance the management of permissions in

Active Directory, enable the logical grouping of permissions, and providing an efficient mechanism for setting and maintaining access control.

For each permission entry defined in Active Roles and configured with the **Permissions Propagation** option set, Active Roles generates native Active Directory permission entries based on the Active Roles permission entry.

The **Permissions Propagation** option (also referred to as **Sync to Native Security** or **Sync to AD** in the user interface) ensures that every time Active Roles permissions change, the associated native permission entries change accordingly.

Disabling the **Permissions Propagation** option on existing Active Roles permissions, or deleting Active Roles permissions with this option set, deletes all native permission entries specified through those Active Roles permissions.

If a propagated permission entry is deleted or modified in Active Directory, whether intentionally or by mistake, Active Roles restores that entry based on Access Template information, thus ensuring the correct permission settings in Active Directory. The "Sync of Permissions to Active Directory" scheduled task is used in Active Roles to create or update permission entries in Active Directory based on the Access Template links that have the Permissions Propagation option enabled.

Access Template management tasks

This section guides you through the Active Roles console to manage Access Templates. The following topics are covered:

- Using predefined Access Templates
- Creating an Access Template
- Applying Access Templates
- Managing Access Template links
- Synchronizing permissions to Active Directory
- Adding, modifying, or removing permissions
- Nesting Access Templates
- Copying an Access Template
- Exporting and importing Access Templates
- Renaming an Access Template
- Deleting an Access Template

Using predefined Access Templates

Active Roles offers an extensive suite of preconfigured Access Templates that represent typical administrative roles, enabling the correct level of administrative authority to be

delegated quickly and consistently.

The predefined Access Templates are located in containers under **Configuration/Access Templates** in the Active Roles console. You can display a list of Access Templates in the details pane by expanding **Configuration | Access Templates** and then selecting one of these containers in the console tree:

- Active Directory
- Azure
- AD LDS (ADAM)
- Computer Resources
- Configuration
- Exchange
- Starling
- User Interfaces
- User Self-management

Active Directory

You can use Access Templates from the **Active Directory** container to delegate Active Directory data management tasks and Active Directory service management tasks, such as:

- User and group management
- Management of computer, printer queue, or shared folder objects
- Forest and domain configuration management

This container includes templates that allow for a wide range of administrative tasks and templates that limit access to selected properties of Active Directory objects.

Azure

You can use Access Templates from the **Azure** container to delegate management tasks on containers performing Azure related operations, such as:

- Azure Configuration in hybrid environment
- Azure user management tasks in hybrid environment
- Azure contact management tasks in hybrid environment
- Azure group management tasks in hybrid environment
- Office 365 group management tasks in Azure AD

AD LDS (ADAM)

You can use Access Templates from the **AD LDS (ADAM)** container to delegate data management tasks on the following object types in Directory Lightweight Directory Services (AD LDS):

- AD LDS Container
- AD LDS Group
- AD LDS Organizational Unit (OU)
- AD LDS User

For instructions on how to view or set permissions on AD LDS objects, refer to the AD LDS Data Management chapter, later in this document.

Computer Resources

You can use Access Templates from the **Computer Resources** container to delegate management tasks on resources that reside on local computers, such as:

- Local users and groups
- Services
- Network file shares (shared directories)
- Printers and print jobs

This container includes templates for specific administrative roles, such as Printer Operator or Service Operator, and templates that specify access to selected properties of computer local resources.

Configuration

You can use Access Templates from the **Configuration** container to delegate management tasks on Active Roles configuration, such as:

- Administer Managed Units, Policy Objects, or Access Templates
- Configure replication (add or remove replication partners)
- Add or remove managed domains

This container also includes templates governing access to individual properties of Managed Units, Policy Objects and Access Templates.

Exchange

You can use Access Templates from the **Exchange** container to delegate the following administrative tasks on Exchange Server recipients:

- Manage all recipient settings
- Use Exchange Tasks Wizard
- Manage e-mail addresses
- Configure general message settings
- Configure advanced message settings

This container also includes templates governing access to individual Exchange-related properties of users, groups, and contacts.

Starling

You can use Access Templates from the **Starling** container to delegate required permission to perform Starling operations.

User Interfaces

You can use Access Template from the **User Interfaces** container to delegate the control to users in the User Interfaces container under Server Configuration, to log in to the Active Roles MMC interface.

To examine an Access Template in detail

1. Right-click the Access Template and click **Properties**.

   The **Permissions** tab in the **Properties** dialog box lists all permissions entries defined in the Access Template, and allows you to inspect each entry.

2. Select an entry and click the **View** button.

> NOTE: Active Roles does not allow predefined Access Templates to be modified or deleted. If you need to make changes to a predefined Access Template, you should create a copy of the Access Template and then modify the copy as needed. To create a copy, right-click the Access Template and click **Copy**.

To apply an Access Template by using the Delegation of Control wizard, right-click the Access Template, click **Links**, and then, in the **Links** window, click **Add** to start the wizard.

For more information, see Applying Access Templates later in this chapter.

User Self-management

You can use Access Templates from the **User Self-management** container to delegate self-management tasks to end-users (for instance, giving end-users the right to make changes to their own accounts by using the Active Roles Web Interface for self-administration).

You can examine an Access Template in detail by viewing the **Properties** dialog box: right-click the Access Template and click **Properties**. The **Permissions** tab in the **Properties** dialog box lists all permissions entries defined in the Access Template, and allows you to inspect each entry: select an entry and click the **View** button.

> NOTE: Active Roles does not allow predefined Access Templates to be modified or deleted. If you need to make changes to a predefined Access Template, you should create a copy of the Access Template and then modify the copy as needed. To create a copy, right-click the Access Template and click **Copy**.

You can apply an Access Template by using the Delegation of Control wizard: right-click the Access Template, click **Links**, and then, in the **Links** window, click **Add** to start the wizard. For more information, see Applying Access Templates later in this chapter.

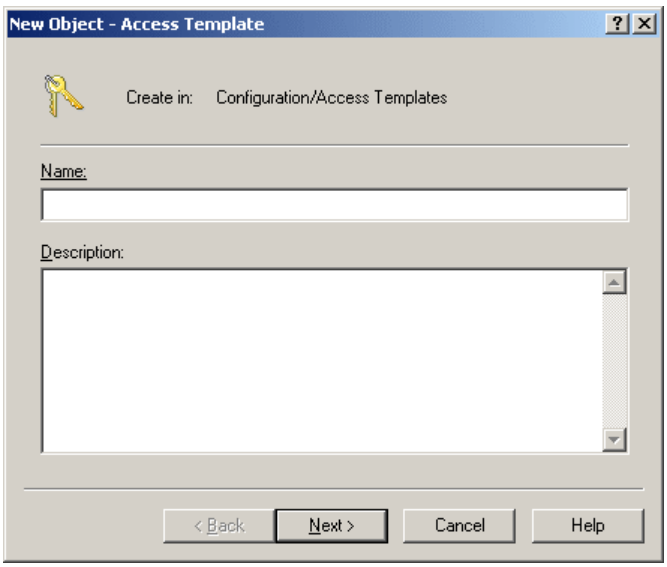
Creating an Access Template

The Active Roles console provides the New Object - Access Template wizard for creating Access Templates. You can start the wizard as follows: right-click **Access Templates** in the console tree, and select **New | Access Template**. In this case, the wizard adds an Access Template to the **Access Templates** container.

> NOTE: It is advisable to store custom Access Templates in a separate container. You can create a container as follows: right-click **Access Templates** in the console tree, and select **New | Access Template Container**. After you have created a container, you can have the wizard add an Access Template to that container rather than directly to **Access Templates**: right-click the container in the console tree and select **New | Access Template**.

The first page of the wizard looks as shown in the following figure.

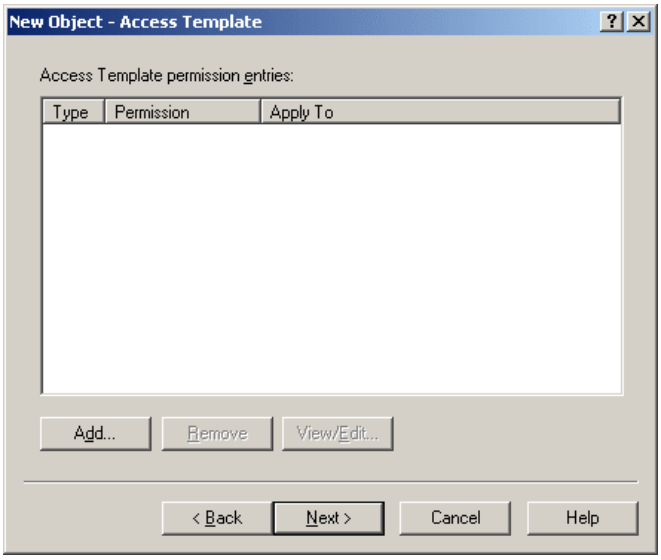
Figure 12: Add new Access template



On this page, type a name and description for the new Access Template. The Active Roles console will display the name and description in the list of Access Templates in the details pane.

Click **Next**. The second page of the wizard looks as shown in the following figure.

Figure 13: Access template permission enteries



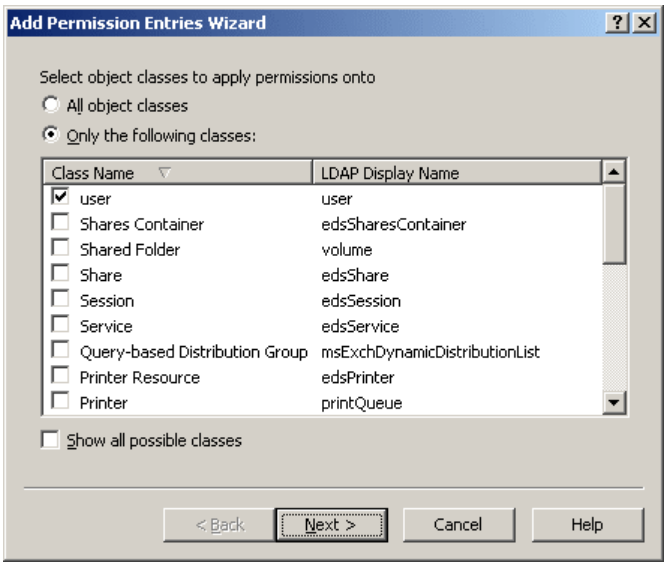
This page prompts you to configure a list of Access Template permission entries. You can use the **Add**, **Remove** and **View/Edit** buttons to add, remove and modify an entry, respectively. Clicking **Add** starts the Add Permission Entries wizard that helps you configure permission entries. The wizard is discussed later in this section.

After you have completed the list of permission entries, click **Next**, and then click **Finish**. The new Access Template is created.

Add Permission Entries wizard

The Add Permission Entries wizard lets you specify the permission to be added into the Access Template. The first page of the wizard looks as shown in the following figure.

Figure 14: Add Permission Entries



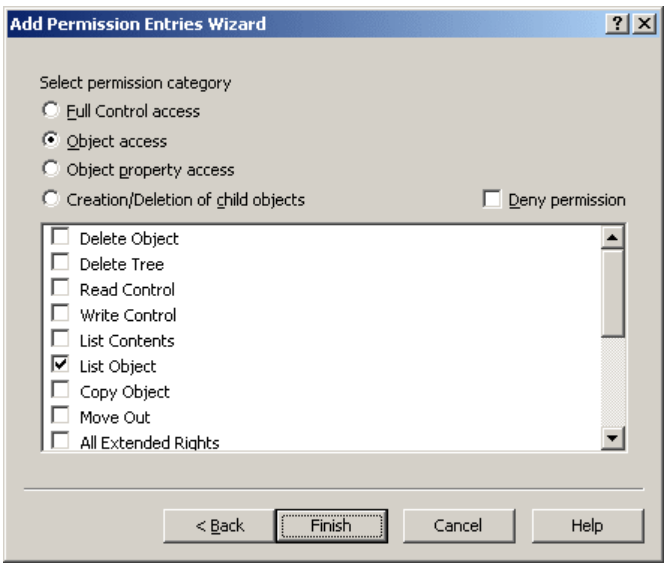
On this page, you select the types of objects to which you want the permission to allow (or deny) access. You can select one of these options:

- **All object classes** With this option, the permission controls access to objects of any type.
- **Only the following classes** With this option, the permission controls access to objects of the type you choose by selecting the appropriate check boxes in the list.

NOTE: By default, all object classes are not displayed in the list. To display all object classes, select the **Show all possible classes** check box.

After you have selected the object classes you want, click **Next**. The next page of the wizard looks as shown in the following figure.

Figure 15: Permission category



On this page, you select a permission category, and specify whether you want the permission to allow or deny certain administrative actions.

You can select one of the following permission categories:

- **Full Control access** Allows or denies all administrative actions on an object
- **Object access** Controls how an object is accessed and controlled.
- **Object property access** Controls access to an object's attributes.
- **Creation/Deletion of child objects** Allows or denies creation or deletion of objects in a container.

If you want the permission to deny certain administrative actions, you select the **Deny permission** check box.

The following sections elaborate on the permission categories you can select in the Add Permission Entries wizard.

Full Control access

Permissions in this category provide for all administrative operations on objects (and their properties) of the classes that you selected in the previous step of the Add Permission Entries wizard.

After you select **Full Control access** and click **Finish**, the permission is added into the newly created Access Template.

Object access

Permissions in this category provide for administrative operations on objects themselves (but not their properties) of the classes that you selected in the previous step of the Add Permission Entries wizard.

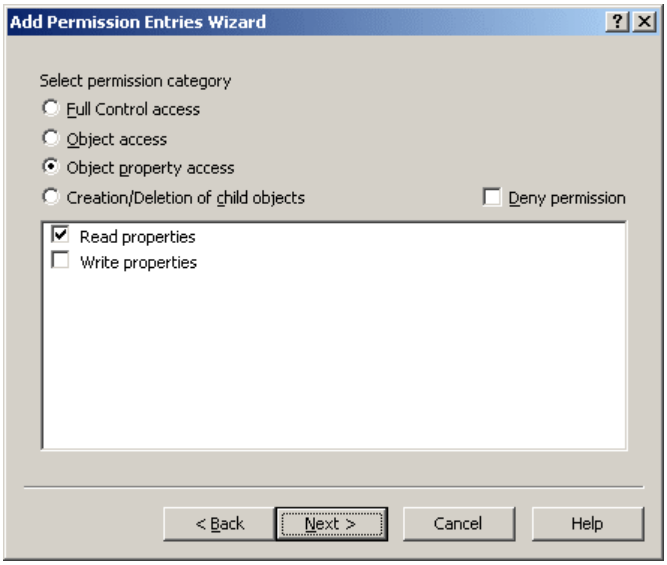
Administrative operations are selected from the list, which is displayed when you select **Object access**. You select the necessary operations by selecting the appropriate check boxes. For example, you might select **List Object** to allow viewing objects of certain types.

After you have selected the operations, click **Finish** to complete the Add Permission Entries wizard. The permission is added into the newly created Access Template.

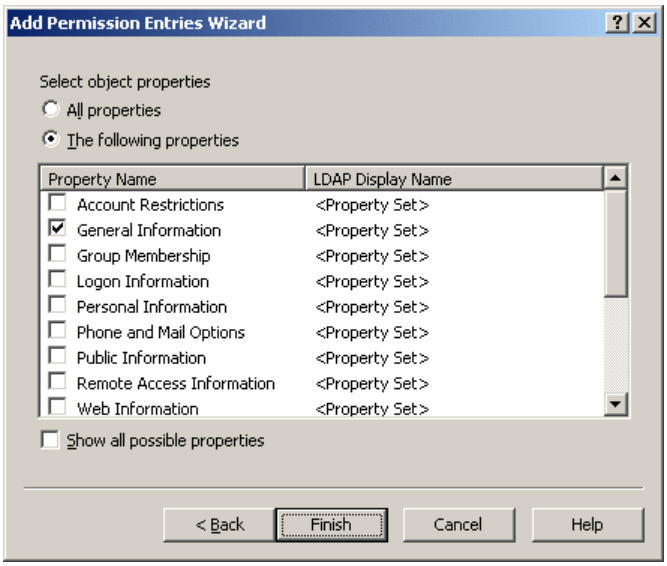
Object property access

Permissions in this category provide for administrative operations on object properties for objects of the classes that you selected in the previous step of the Add Permission Entries wizard.

When you select **Object property access**, you specify access to object properties. You can select **Read properties** and **Write properties**, as shown in the following figure.



After you click **Next**, the wizard displays a page where you can select the properties to which you want the permission to allow (or deny) access. The page is similar to the following figure.



On that page, you can select one of the following options:

- **All properties** With this option, the permission controls access to all properties.
- **The following properties** With this option, the permission controls access to the properties you select from the list by selecting the appropriate check boxes.

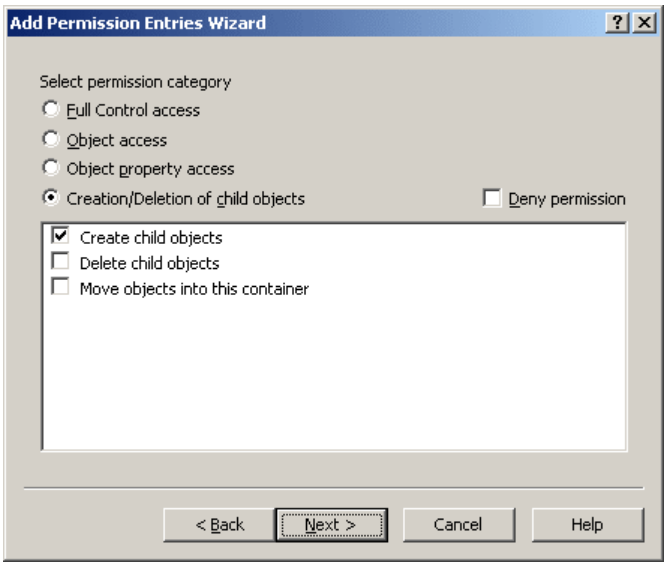
NOTE: By default, all object properties are not displayed in the list. To display all object properties, select the **Show all possible properties** check box.

After you have selected the properties you want, click **Finish** to complete the Add Permission Entries wizard. The permission is added to the Access Template.

Creation/Deletion of child objects permission

Permissions in this category provide for creation and deletion of child objects in container objects of the classes you selected in the previous step of the Add Permission Entries wizard.

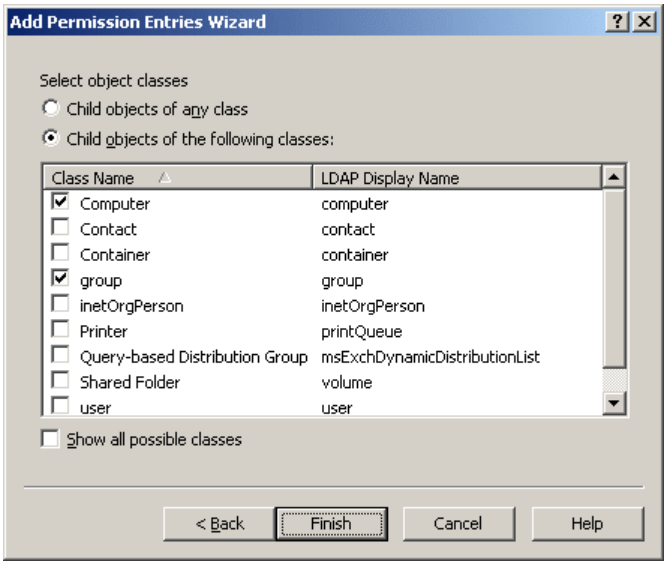
When you select **Creation/Deletion of child objects**, you specify the creation, deletion, and move operations you want the permission to allow (or deny). The list of operations looks as shown in the following figure.



You can select the following operations:

- **Create child objects** Controls the creation of child objects of the classes you select in the next step.
- **Delete child objects** Controls the deletion of child objects of the classes you select in the next step.
- **Move objects into this container** Controls the relocation of object of the classes you select in the next step. This operation assumes moving objects from one container to another without permission to delete existing objects or create new objects.

After you click **Next**, the wizard displays the page where you can select the types of objects on which you want the permission to allow (or deny) the operations you selected in the previous step. The page is similar to the following figure.



On that page, you select the types of objects for which you want the permission to allow (or deny) the creation, deletion, or move operation. You can select one of these options:

- **Child objects of any class** With this option, the permission controls the operations on objects of any type.
- **Child objects of the following classes** With this option, the permission controls the operations on objects of the type you select from the list by selecting the appropriate check boxes.

NOTE: By default, all object classes are not displayed in the list. To display all object classes, select the **Show all possible classes** check box.

After you have selected the object classes, click **Finish** to complete the Add Permission Entries wizard. The permission is added to the Access Template.

Steps for creating an Access Template

To create an Access Template

1. In the console tree, under **Configuration | Access Templates**, locate and select the folder in which you want to add the Access Template.

   You can create a new folder as follows: Right-click **Access Templates** and select **New | Access Template Container**. Similarly, you can create a sub-folder in a folder: Right-click the folder and select **New | Access Template Container**.

2. Right-click the folder, and select **New | Access Template** to start the New Object - Access Template wizard.
3. On the first page of the wizard, do the following, and then click **Next**:
   a. In the **Name** box, type a name for the Access Template.
   b. In the **Description** box, type any optional information about the Access Template.
4. On the second page of the wizard, configure the list of permission entries, and then click **Next**.

   The instructions on how to add, modify, or delete permission entries are given later in this topic.

5. Click **Finish** to create the Access Template that includes the permission entries you have specified.

To add a permission entry to an Access Template

1. On the page that displays a list of permission entries included in the Access Template, click **Add** to start the Add Permission Entries wizard.
2. On the first page of the wizard, select one of these options:
   - **All object classes** The rights defined by this permission entry apply to objects of any class.
   - **Only the following classes** The rights defined by this permission entry apply to objects of specific classes. Select object classes from the list. If the

list does not include the object class you want, select **Show all possible classes**.

3. Click **Next**.
4. On the second page of the wizard, select one of these options:
   - **Full control access** The rights to create or delete child objects, read and write properties, examine child objects and the object itself, add and remove the object from the directory, and read or write with any extended right. This option does not have any configuration parameters.
   - **Object access** The rights to exercise certain generic permissions and extended rights on the objects. Select permissions and extended rights from the list to configure this option as appropriate.
   - **Object property access** The rights to read or write certain properties of the object. Select check boxes to configure this option as appropriate: **Read properties**, **Write properties**. On the next page of the wizard, you can select the properties you want to be controlled by this permission entry.
   - **Creation/Deletion of child objects** The rights to create or delete child objects of the object. Select check boxes to configure this option as appropriate: **Create child objects**, **Delete child objects**, **Move objects into this container**. On the next page of the wizard, you can specify the class or classes of child object you want to be controlled by this permission entry.
5. If you want the Access Template to deny the rights defined by this permission entry, select the **Deny permission** check box. Otherwise, leave the check box cleared.
6. Do the following, depending on the option you selected and configured in Step 4:
   - **Full control access** or **Object access** Click **Finish** to add the permission entry to the Access Template.
   - **Object property access** or **Creation/Deletion of child objects** Click **Next** to continue configuring the option.
7. On the third page of the wizard, continue configuring the option you selected in Step 4, and then click **Finish** to add the permission entry to the Access Template:
   - If you selected **Object property access**, select the properties to be controlled by this permission entry. You have two options: **All properties** and **The following properties**. With the second option, you must select properties from the list. If the list does not include the property you want, select **Show all possible properties**.
   - If you selected **Creation/Deletion of child objects**, specify the class or classes of child object to be controlled by this permission entry. You have two options: **Child objects of any class** and **Child objects of the following classes**. With the second option, you must select one or more object classes from the list. If the list does not include the object class you want, select **Show all possible classes**.

To view or modify a permission entry in an Access Template

1. On the page that displays a list of permission entries included in the Access Template, select the permission entry you want to view or modify, and click **View/Edit** to display the **Modify Permission Entry** dialog box.
2. Examine the **Apply Onto** tab in the **Modify Permission Entry** dialog box. On this tab, you can view or modify the same settings as on the first page of the Add Permission Entries wizard (see Step 2 in the procedure above).
3. Examine the **Permissions** tab in the **Modify Permission Entry** dialog box. This tab provides the same options as the second page of the Add Permission Entries wizard (see Step 4 in the procedure above). The options are read-only, so you cannot change the option that was selected upon creation of the permission entry. However, you can manage the configuration of the option:
   - **Object access** Select generic permissions or extended rights you want to add to the Access Template.
   - **Object property access** Select or clear these check boxes: **Read properties**, **Write properties**.
   - **Creation/Deletion of child objects** Select or clear these check boxes: **Create child objects**, **Delete child objects**, **Move objects into this container**.
4. If you want the Access Template to deny the rights defined by this permission entry, select the **Deny permission** check box on the **Permissions** tab. Otherwise, leave the check box cleared.
5. If **Object property access** is selected on the **Permissions** tab, use the **Object Properties** tab in the **Modify Permission Entry** dialog box to view or modify the settings that determine which properties are controlled by this permission entry (see Step 7 in the procedure above).
6. If **Creation/Deletion of child objects** is selected on the **Permissions** tab, use the **Object Classes** tab in the **Modify Permission Entry** dialog box to view or modify the settings that determine which classes of child object are controlled by this permission entry.

To delete a permission entry from an Access Template

1. On the page that displays a list of permission entries included in the Access Template, select the permission entry you want to delete, and click **Remove**.
2. Click **Yes** to confirm the deletion.

Applying Access Templates

Active Roles allows Access Templates to be applied to any objects—administrative views (Managed Units), directory folders (containers), or individual (leaf) objects.

When applying an Access Template to an object, you designate a Trustee (user or group) and assign permissions to the Trustee for that object. As a result, the Trustee gets access to the object according to permissions defined in the Access Template.

For example, two assistants of a directory administrator might be delegated full control of different domains; Help Desk might be assigned the administrative role to reset passwords.

> NOTE: When you apply Access Templates to a folder, you can configure the permission settings to propagate from the folder to its child objects, down the directory structure.

To apply an Access Template, you need to start and complete the Delegation of Control wizard.

You can start the Delegation of Control wizard from any of the following points:

- **Access Template** Right-click the Access Template, click **Links**, and then click the **Add** button. Access Templates are located in the **Configuration/Access Templates** container.

  When started in this way, the wizard allows you to select directory objects where to apply the Access Template and Trustees for those objects.

- **Securable object** Depending on whether the object is a container or leaf object, do one of the following:

  - For a container or a Managed Unit, right-click it, click **Delegate Control**, and then click the **Add** button.
  - For a leaf object, display the **Properties** dialog box, go to the **Administration** tab, click the **Security** button, and then click the **Add** button.

  When started in this way, the wizard allows you to select Trustees for the object and Access Templates to define the Trustees' rights to the object.

- **Security principal (Trustee)** Right-click the group or user you want to designate as a Trustee, click **Delegated Rights**, and then click the Add button.

  When started in this way, the wizard allows you to select objects for which you want to designate the Trustee and Access Templates to define the Trustees' rights to those objects.

You can also start the Delegation of Control wizard from the advanced details pane (ensure that **Advanced Details Pane** is checked on the **View** menu):

- Select an Access Template, right-click a blank area on the **Links** tab, and then click **Add**.

  When started in this way, the wizard allows you to select directory objects where to apply the Access Template and Trustees for those objects.

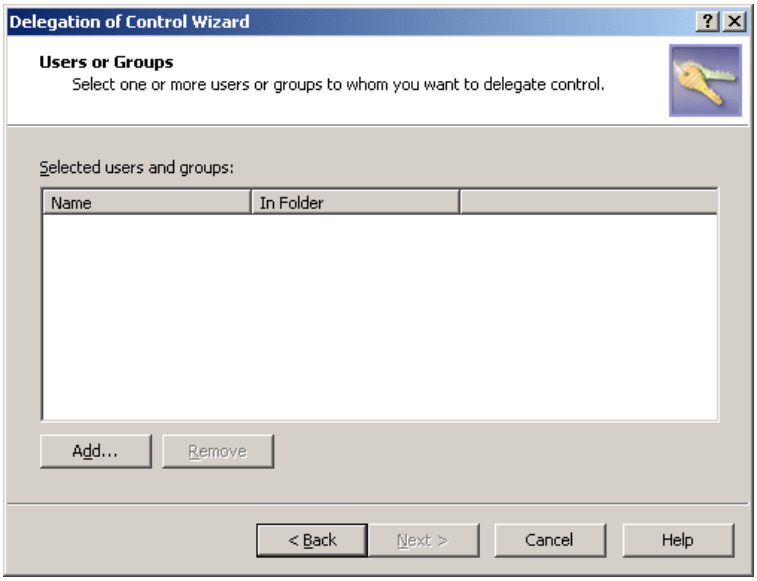
- Select a directory object (securable object), right-click a blank area on the **Active Roles Security** tab, and then click **Add**.

  When started in this way, the wizard allows you to select Trustees for the object and Access Templates to define the Trustees' rights to the object.

The rest of this section provides instructions on how to complete the Delegation of Control wizard, assuming that you start the wizard from the object of which control you want to delegate (securable object). For instructions on how to complete the wizard in the other cases, see Steps for applying an Access Template later in this chapter.

If you start Delegation of Control wizard from a securable object, clicking **Next** on the **Welcome** page displays the **Users or Groups** page, shown in the following figure.

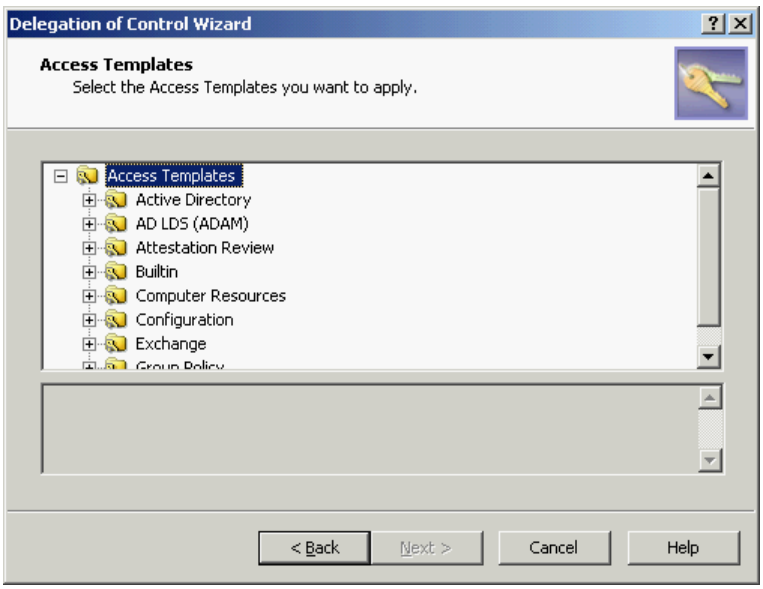
Figure 16: Delegation of control - Users or Groups



On the **Users or Groups** page, click **Add** to display the **Select Objects** dialog box where you can select groups or users to be designated as Trustees. Type or select the names of the users or groups you want to add to the list, and then click **OK**.

After you have completed the list on the **Users or Groups** page, click **Next**. This displays the **Access Templates** page, shown in the following figure.

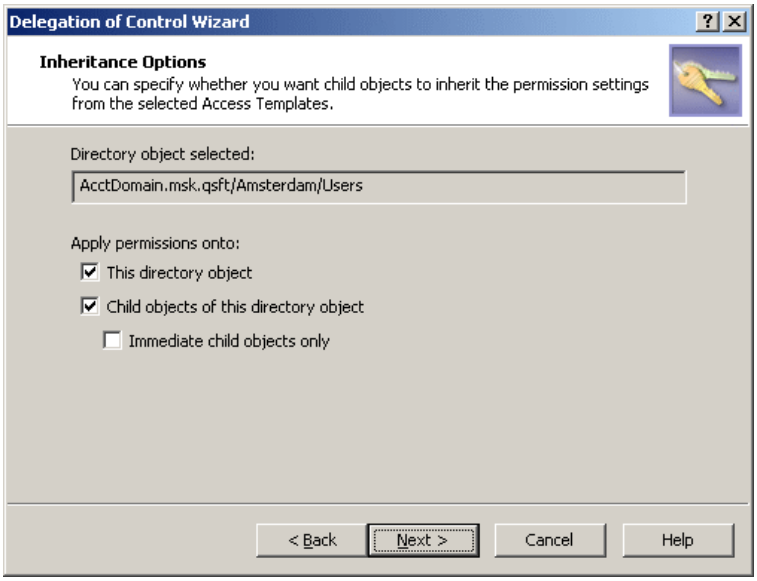
Figure 17: Delegation of control - Access Templates



On the **Access Templates** page, expand containers that hold Access Templates, and select check boxes next to the names of the Access Templates you want to apply.

When you are done with selecting Access Templates, click **Next**. This displays the **Inheritance Options** page, shown in the following figure.

Figure 18: Delegation of Control - Inheritace Options



On the **Inheritance Options** page, you can select the following options to control inheritance of permissions:

- **This directory object** Ensures that the Trustees have administrative rights to the securable object itself.

- **Child objects of this directory object** Ensures that the Trustees have administrative rights to the child objects of securable object, down the directory structure.
- **Immediate child objects only** Limits the Trustees' rights to only immediate child objects of the securable object.

By default, the first two options are selected.

Click **Next**. This displays the **Permissions Propagation** page where you can select the **Propagate permissions to Active Directory** check box. If you do so, the permission settings you are configuring are synchronized to Active Directory. As a result, the Trustees may also exercise their rights outside the Active Roles environment, thus incurring a potential risk of bypassing policies configured and enforced with Active Roles. Therefore, you should use this option carefully.

By default, the **Propagate permissions to Active Directory** check box is cleared. If you choose to select it, you can change this setting at any time by using the **Sync to AD** button in the **Active Roles Security** window or **Sync to AD** command in the advanced details pane (see Synchronizing permissions to Active Directory later in this chapter).

Click **Next**, and then click **Finish** to complete the wizard.

Steps for applying an Access Template

To apply an Access Template

1. In the console tree, under **Configuration | Access Templates**, locate and select the folder that contains the Access Template you want to apply.
2. In the details pane, right-click the Access Template, and click **Links**.
3. In the **Links** dialog box, click **Add** to start the Delegation of Control wizard.
4. On the Welcome page of the wizard, click **Next**.
5. On the **Objects** page, add or remove the objects on which you want to specify permission settings by using the Access Template:
   - To add objects, click **Add**, and then use the **Select Objects** dialog box to locate and select the objects.
   - To remove objects, select them from the list on the **Objects** page, and click **Remove**.
6. Click **Next**.
7. On the **Users or Groups** page, add or remove the users or groups (Trustees) to whom you want to assign the permissions defined by the Access Template on the objects that you have included on the **Objects** page:
   - To add users or groups, click **Add**, and then use the **Select Objects** dialog box to locate and select the users or groups.
   - To remove users or groups, select them from the list on the **Users or Groups** page, and click **Remove**.
8. Click **Next**.

9. On the **Inheritance Options** page, select or clear these check boxes as needed:
    - **This directory object** Specify permission settings on the objects you have included on the **Objects** page.
    - **Child objects of this directory object** Specify permission settings on all the child objects (or members, as applied to a Managed Unit) in the entire hierarchy under each of the objects you have included on the **Objects** page.
    - **Immediate child objects only** Specify permission settings on only the child objects (or members, as applied to a Managed Unit) of which the objects that you have included on the **Objects** page are the direct ancestors.
10. Click **Next**.
11. On the **Permissions Propagation** page, if you want the Access Template-based permission settings to be synchronized to the native Active Directory access controls, select **Propagate permissions to Active Directory**. Doing so causes the authorization information on the objects to be modified in Active Directory based on the permission settings defined within Active Roles.
12. Click **Next**.
13. Click **Finish**.

To specify permission settings on an object by using an Access Template

1. Open the **Active Roles Security** dialog box for the object:
    - Right-click the object, and click **Delegate Control**.

    OR

    - Right-click the object, and click **Properties**. Then, on the **Administration** tab in the **Properties** dialog box, click **Security**.
2. In the **Active Roles Security** dialog box, click **Add** to start the Delegation of Control wizard.
3. On the Welcome page of the wizard, click **Next**.
4. On the **Users or Groups** page, add or remove the users or groups (Trustees) to whom you want to assign permissions on the object:
    - To add users or groups, click **Add**, and then use the **Select Objects** dialog box to locate and select the users or groups.
    - To remove users or groups, select them from the list on the **Users or Groups** page, and click **Remove**.
5. Click **Next**.
6. On the **Access Templates** page, select the Access Template to apply.

    You can select multiple Access Templates to apply.

7. Click **Next**.
8. On the **Inheritance Options** page, select or clear these check boxes as needed:
   - **This directory object** Specify permission settings on the object itself.
   - **Child objects of this directory object** Specify permission settings on all the child objects (or members, as applied to a Managed Unit) in the entire hierarchy under the object.
   - **Immediate child objects only** Specify permission settings on only the child objects (or members, as applied to a Managed Unit) of which the object is the direct ancestor.
9. Click **Next**.
10. On the **Permissions Propagation** page, if you want the Access Template-based permission settings to be synchronized to the native Active Directory access controls, select **Propagate permissions to Active Directory**. Doing so causes the authorization information on the object to be modified in Active Directory based on the permission settings defined within Active Roles.
11. Click **Next**.
12. Click **Finish**.

To specify permissions for a user or group by using an Access Template

1. Right-click the user or group, and click **Delegated Rights**.
2. In the **Delegated Rights** dialog box, click **Add** to start the Delegation of Control wizard.
3. On the Welcome page of the wizard, click **Next**.
4. On the **Objects** page, add or remove the objects on which you want to specify permissions for the user or group:
   - To add objects, click **Add**, and then use the **Select Objects** dialog box to locate and select the objects.
   - To remove objects, select them from the list on the **Objects** page, and click **Remove**.
5. Click **Next**.
6. On the **Access Templates** page, select the Access Template to apply.
7. You can select multiple Access Templates to apply.
8. Click **Next**.
9. On the **Inheritance Options** page, select or clear these check boxes as needed:
   - **This directory object** Specify permissions on the objects you have included on the **Objects** page.
   - **Child objects of this directory object** Specify permissions on all the child objects (or members, as applied to a Managed Unit) in the entire hierarchy under each of the objects you have included on the **Objects** page.

- **Immediate child objects only** Specify permissions on only the child objects (or members, as applied to a Managed Unit) of which the objects that you have included on the **Objects** page are the direct ancestors.

10. Click **Next**.
11. On the **Permissions Propagation** page, if you want the Access Template-based permission settings to be synchronized to the native Active Directory access controls, select **Propagate permissions to Active Directory**. Doing so causes the authorization information on the objects to be modified in Active Directory based on the permission settings defined within Active Roles.
12. Click **Next**.
13. Click **Finish**.

NOTE:

- Active Roles allows Access Templates to be applied to any objects, including Managed Units, directory folders (containers), and individual (leaf) objects.
- When applying an Access Template to an object, you designate a Trustee (user or group) and assign permissions to the Trustee for that object. As a result, the Trustee gains access to the object to the extent of the permissions defined by the Access Template.
- To apply an Access Template, you use the Delegation of Control wizard. You can start the wizard as described in this topic. In addition, you can start the wizard from the **Links** or **Active Roles Security** tab in the advanced details pane: Right-click a blank area on the tab, and click **Add**. To display the advanced details pane, check **Advanced Details Pane** on the **View** menu (see Advanced pane earlier in this document).

Managing Access Template links

When applying an Access Template, Active Roles creates an Access Template link. Thus, administrative rights are specified by linking Access Templates to securable objects—Managed Units, directory folders (containers), or individual (leaf) objects.

Each Access Template link includes the identifier (SID) of the security principal—user or group—to which the specified administrative rights are assigned. When an Access Template link is created, the user or group becomes a Trustee over the collection of objects or the folder to which the Access Template is linked, with permissions specified by that Access Template.

When an Access Template is modified or no longer applied, the permission information on objects affected by the Access Template changes accordingly.

You can display a list of Access Template links starting from one of the following points:

- **Access Template** Right-click an Access Template and click **Links**.

  This displays the links in which the Access Template occurs.

- **Security principal (Trustee)** Right-click a group or user, and click **Delegated Rights**.

  This displays the links in which the group or user occurs as a Trustee either directly or due to group memberships.

- **Securable object** Right-click a container object or Managed Unit and click **Delegate Control**. For a leaf object, open the **Properties** dialog box, go to the **Administration** tab, and click **Security**.

  This displays the links in which the selected object occurs as a securable object (referred to as *Directory Object*).

Another way to see a list of Access Template links is to use the advanced details pane. Ensure that **Advanced Details Pane** is checked on the **View** menu, and then select one of the following:

- Access Template

  The **Links** tab lists the links in which the selected Access Template occurs.

- Other object (Managed Unit, container, or leaf object)

  The **Active Roles Security** tab lists the links in which the selected object occurs as a securable object (referred to as Directory Object).

The Active Roles console displays a list of Access Template links in a separate window. Thus, the **Active Roles Security** window is displayed when you start from a securable object (for example, by clicking a Managed Unit or Organizational Unite and then clicking **Delegate Control**).

Each entry in the list of the Access Template links includes the following information:

- **Trustee** The link defines administrative rights of this security principal (group or user).
- **Access Template** The Access Template that determines the Trustee's rights.
- **Directory Object** The link defines the Trustee's rights to this securable object.
- **Sync to Native Security** Indicates whether the permissions are synced to Active Directory.
- **Disabled** Indicates whether the link is disabled. If a link is disabled, the permissions defined by that link have no effect.
- **Access Rule** Indicates whether an Access Rule is applied to this link (see Windows claims-based Access Rules).

The **Active Roles Security** window (as well as the **Active Roles Security** tab in the advanced details pane) lists the links of these categories:

- **Direct links** Access Template is applied (linked) directly to the securable object you have selected.
- **Inherited links** Access Template is applied (linked) to a container in the hierarchy of containers above the securable object you have selected, or to a Managed Unit to which the securable object belongs.

The links inherited from parent objects can be filtered out of the list:

- When using the **Active Roles Security** window, clear the **Show inherited** check box.
- When using the **Active Roles Security** tab, right-click the list and then click **Show Inherited** to uncheck the menu item.

A window or tab that displays Access Template links allows you to manage links. In a window, you can use buttons beneath the list. In a tab, you can right-click a list entry or a blank area, and then use commands on the shortcut menu. For example, the following buttons appear in the **Active Roles Security** window:

- **Add** Starts the Delegation of Control wizard to create apply Access Templates.
- **Remove** Deletes the selected entries from the list of links. Available for direct links only.
- **View/Edit** Displays the dialog box to view or modify link properties such as permissions inheritance and propagation options.
- **Sync to AD** Toggles the permissions propagation option of the links selected in the list.
- **Disable** Disables or enables the link. If a link is disabled, the permissions specified by the link takes no effect.

> TIP: In the **Active Roles Security** dialog box, the **Remove** button is available on direct links only. When you need to delete links, it is advisable to manage them using the **Links** command on the Access Template.

Steps for managing Access Template links

When you apply an Access Template (see Applying Access Templates earlier in this document), Active Roles creates an object, referred to as an *Access Template link*, that stores information about the Access Template, the directory object on which the Access Template is applied, and the user or group (Trustee) to whom the permissions are assigned. Basically, the management of permission settings in Active Roles comes to the management of Access Templates and Access Template links. This topic provides some instructions you can use to view or modify Access Template links.

To view or modify Access Template links in which a given Access Template occurs

1. Right-click the Access Template, and click **Links**.
2. In the **Links** dialog box, do the following:
    - To create a new link, click **Add** and follow the steps in the Delegation of Control wizard to apply an Access Template (see Steps for applying an Access Templateearlier in this document).
    - To delete a link, select it from the list and click **Remove**.
    - To view or modify the inheritance and synchronization settings for a link, select the link and click **View/Edit**.

- To change the synchronization setting for a link, select the link and click **Sync to AD** or **Desync to AD**.
- To remove or restore the effect of a link, select the link and click **Disable** or **Enable**, respectively.

To view or modify Access Template links on a given object

1. Open the **Active Roles Security** dialog box for the object:
   - Right-click the object, and click **Delegate Control**.

   OR

   - Right-click the object, and click **Properties**. Then, on the **Administration** tab in the **Properties** dialog box, click **Security**.
2. In the **Active Roles Security** dialog box, do the following:
   - To create a new link, click **Add** and follow the steps in the Delegation of Control wizard to specify permission settings on the object by using an Access Template (for instructions, see Steps for applying an Access Template earlier in this document).
   - To delete a link, select it from the list and click **Remove**.
   - To view or modify the inheritance and synchronization settings for a link, select the link and click **View/Edit**.
   - To change the synchronization setting for a link, select the link and click **Sync to AD** or **Desync to AD**.
   - To remove or restore the effect of a link, select the link and click **Disable** or **Enable**, respectively.

To view or modify Access Template links for a given user or group

1. Right-click the user or group, and click **Delegated Rights**.
2. In the **Delegated Rights** dialog box, do the following:
3. To create a new link, click **Add** and follow the steps in the Delegation of Control wizard to specify permissions for the user or group by using an Access Template (for instructions, see Steps for applying an Access Template earlier in this document).
4. To delete a link, select it from the list and click **Remove**.
5. To view or modify the inheritance and synchronization settings for a link, select the link and click **View/Edit**.
6. To change the synchronization setting for a link, select the link and click **Sync to AD** or **Desync to AD**.
7. To remove or restore the effect of a link, select the link and click **Disable** or **Enable**, respectively.

NOTE:

- By default, the **Active Roles Security** dialog box for an object lists all the links that determine the permission settings on the object, regardless of whether a link was created on the object itself or on a container or Managed Unit that holds the object. To change the display of the list, clear the **Show inherited** check box.
- In the **Active Roles Security** dialog box, only direct links can be removed, that is, a link can be removed if the link was created on the object itself (not inherited from a container or Managed Unit). Only direct links are displayed when you clear the **Show inherited** check box, so you can delete them by clicking **Remove**.
- In the **Active Roles Security** dialog box, the **Remove** button is available only on direct links. When you need to delete links, it is advisable to manage this by using the **Links** command on the Access Template or by using the **Delegated Rights** command on the Trustee (user or group). Alternatively, you can delete a link by using **View/Edit**: Select the link and click **View/Edit**; then, click **Properties** next to the **Access Template** box; then, on the **Administration** tab, click **Links**, and, finally, delete the link from the **Links** dialog box.
- In the **Active Roles Security** dialog box, the **Sync to AD** button is available only on direct links. When you need to change synchronization status of a link, it is advisable to manage this by using the **Links** command on the Access Template or by using the **Delegated Rights** command on the Trustee (user or group). Alternatively, you can change the synchronization status of a link by using **View/Edit**: Select the link and click **View/Edit**; then, on the **Synchronization** tab, select or clear **Propagate permissions to Active Directory**.
- Clicking **View/Edit** displays the **Properties** dialog box for the selected link. This dialog box can be considered as a focal point for administration of all elements of the link. Thus, from the **Properties** dialog box, you can access the properties of the directory object, Access Template and Trustee that are covered by the link, view or modify the settings found on the **Inheritance Options** and **Permissions Propagation** pages in the Delegation of Control wizard, and enable or disable the link.
- You can also manage Access Template links on the **Links** or **Active Roles Security** tab in the advanced details pane, which allows you to perform the same tasks as the **Links** or **Active Roles Security** dialog box, respectively. Right-click a link or a blank area on the tab, and use command on the shortcut menu. The **Links** tab is displayed when you select an Access Template. Otherwise, the **Active Roles Security** tab is displayed. To display the advanced details pane, check **Advanced Details Pane** on the **View** menu (see Advanced pane earlier in this document).

Synchronizing permissions to Active Directory

Active Roles provides the option to keep Active Directory native security updated with selected permissions specified using Access Templates. This option, referred to as *permissions propagation*, is intended to provision users and applications with native permissions to Active Directory. The normal operation of Active Roles does not rely on this option.

You can set the permissions propagation option in these ways:

- When applying Access Templates, you can select the **Propagate permissions to Active Directory** check box in the Delegation of Control wizard.
- When managing Access Template links, you can use the **Sync to AD** button in a window that displays a list of links or use the **Sync to AD** command on a tab that displays a list of links in the advanced details pane.

For example, suppose Active Roles defines certain permissions on an Organizational Unit, and you want to synchronize them to Active Directory. You can accomplish this task as follows.

First, right-click the Organizational Unit and click **Delegate Control** to display the **Active Roles Security** window.

Next, in the **Access Template links** list, select the links that define the permissions you want to synchronize.

Finally, click the **Sync to AD** button. The **Sync to Native Security** column in the list displays **Yes** for the links that you are going to synchronize.

After you click **OK**, Active Roles creates permission entries in Active Directory so that the Trustee has the same rights in Active Directory as it has in the Active Roles environment in accordance with the Access Template links you have synchronized.

You can stop synchronization of permissions at any time by clicking the **Desync to AD** button. If you do so, Active Roles deletes all permission entries in Active Directory that were created as a result of synchronization.

> TIP: In the **Active Roles Security** dialog box, the **Sync to AD** button is only available on direct links. When you need to synchronize links, it is advisable to manage them using the **Links** command on the Access Template.

You can also accomplish this task using the advanced details pane as follows:

1. Select the Organizational Unit.
2. On the **Active Roles Security** tab, select the Access Template links that define the permissions you want to synchronize.
3. Right-click the selection and click **Sync to AD**.

You can use the **Sync to AD** command to stop synchronization: right-click the links you want to no longer be synchronized, and click **Desync to AD**.

TIP: On the **Active Roles Security** tab, the **Sync to AD** command is available on direct links only. When you need to synchronize links, it is advisable to manage them using the **Links** tab for the Access Template.

Steps for synchronizing permissions to Active Directory

Active Roles provides the option to keep Active Directory native security updated with selected permission settings that are specified by using Access Templates. This option, referred to as *permissions propagation*, is intended to provision users and applications with native permissions to Active Directory. The normal operation of Active Roles does not rely upon this option.

You can set the permissions propagation option as follows:

- When applying an Access Template, select the **Propagate permissions to Active Directory** check box in the Delegation of Control wizard (see Steps for applying an Access Template earlier in this document).
- When managing Access Template links, use the **Sync to AD** button in the dialog box that displays a list of links (see Steps for managing Access Template links earlier in this document).

As an example, you can use the following instructions to set the permissions propagation option on the permission settings that are defined by applying a certain Access Template to an Organizational Unit:

To synchronize permission settings on an Organizational Unit

1. Right-click the Organizational Unit and click **Delegate Control**.
2. In the **Active Roles Security** dialog box, select the Access Template link that determines the permission settings you want to synchronize to Active Directory, and then click **Sync to AD**.
3. Click **OK** to close the **Active Roles Security** dialog box.

NOTE:

- When synchronizing permissions to Active Directory, Active Roles creates permission entries in Active Directory so that the Trustee has the same rights in Active Directory as it has in the Active Roles environment as per the Access Template links you have synchronized.
- You can stop synchronization of permissions at any time by clicking the **Desync to AD** button. If you do so, Active Roles deletes all permission entries in Active Directory that were created as a result of synchronization.
- You can also manage the permissions propagation option on the **Links** or **Active Roles Security** tab in the advanced details pane, which allows you to perform the same tasks as the **Links** or **Active Roles Security** dialog box, respectively. Right-click the link on which you want to set the permissions propagation option, and click **Sync to AD** to start synchronization or **Desync to AD** to stop synchronization. The **Links** tab is displayed when you select an Access Template. Otherwise, the **Active Roles Security** tab is displayed. To display the advanced details pane, check **Advanced Details Pane** on the **View** menu (see Advanced pane earlier in this document).

Managing Active Directory permission entries

The **Native Security** tab in the advanced details pane lists the native Active Directory permission entries for the securable object (for example, an organizational unit) selected in the console tree.

By analyzing information in the **Type** and **Source** columns on the **Native Security** tab, you can determine whether a given entry is synchronized from Active Roles.

In the **Type** column, the synchronized entries are marked with the icon. This icon changes to if synchronization of the entry is invalid or unfinished. For example, if you delete a synchronized entry from Active Directory, Active Roles detects the deletion and re-creates the entry. Until the entry is re-created, the **Type** column marks the entry with the icon.

For each synchronized entry, the **Source** column displays the name of the Access Template that defines the permissions synchronized to that entry.

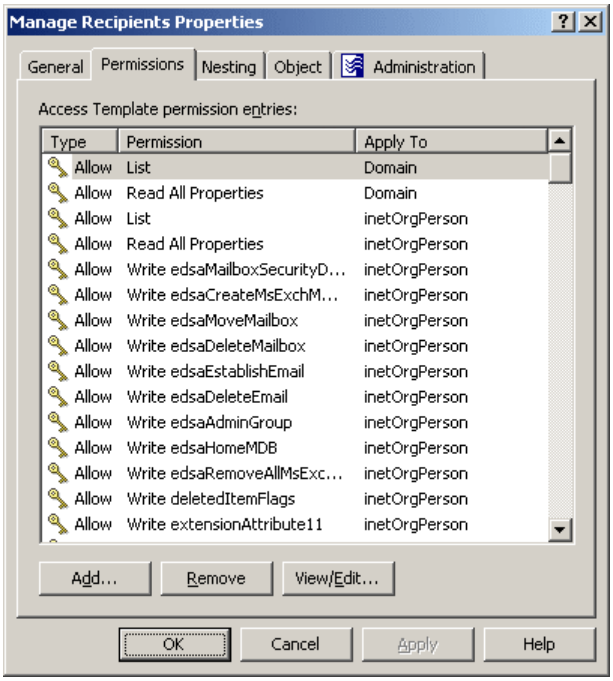
From the **Native Security** tab, you can manage permission entries: right-click an entry, and click **Edit Native Security**. This displays the **Permissions** dialog box where you can add, remove and modify Active Directory permission entries for the securable object you selected.

Adding, modifying, or removing permissions

When you add, remove, or modify permissions in an Access Template, permission settings automatically change on all objects to which the Access Template is applied (linked), including those that are affected by the Access Template because of inheritance.

To add, remove, or modify permissions in an Access Template, open the **Properties** dialog box for the Access Template, and go to the **Permissions** tab.

Figure 19: Access Template - Manage permissions



The **Permissions** tab lists permission entries defined in the Access Template. Each entry in the list includes the following information:

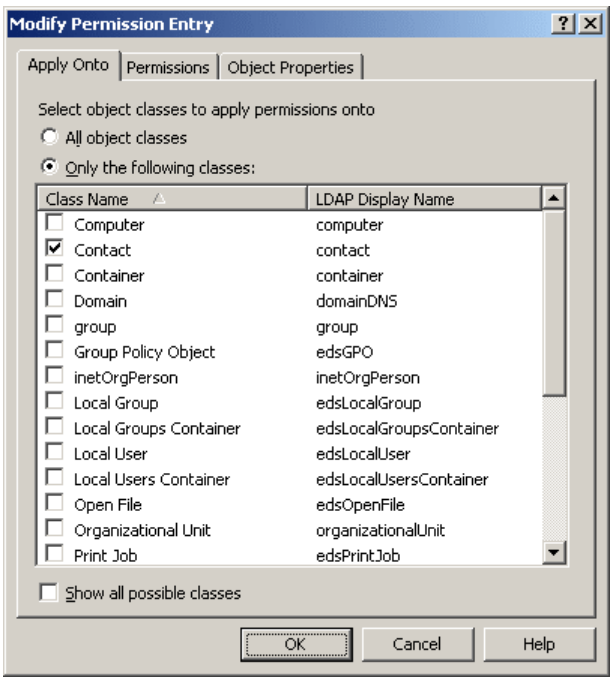
- **Type** Specifies whether the permission allows or denies access.
- **Permission** Name of the permission.
- **Apply To** Type of objects that are subject to the permission.

To add a new permission, click **Add** and complete the Add Permission Entries wizard, as described in Add Permission Entries wizard earlier in this chapter.

To delete permissions, select them from the **Access Template permission entries** list, and click **Remove**.

To modify a permission, select it from the **Access Template permission entries** list, and click **View/Edit**. This displays the **Modify Permission Entry** dialog box, similar to the following figure.

Figure 20: Access Template - Modify permissions



You can use the tabs in that dialog box to modify the permission as needed. The tabs are similar to the pages in the Add Permission Entries wizard, discussed in Add Permission Entries wizard earlier in this chapter.

Steps for adding permissions to an Access Template

To add a permission entry to an Access Template

1. In the console tree, under **Configuration | Access Templates**, locate and select the folder that contains the Access Template you want to modify.
2. In the details pane, right-click the Access Template, and click **Properties**.
3. On the **Permissions** tab, click **Add**, and then use the Add Permission Entries wizard to configure a permission entry.

   For detailed instructions on how to add a permission entry to an Access Template, see Steps for creating an Access Template earlier in this document.

NOTE: The **Permissions** tab lists the permission entries that are configured in the Access Template. You can use the **Permissions** tab to add, modify, or delete permission entries from the Access Template.

Once an Access Template is applied within Active Roles to determine permission settings in the directory, any changes to the list of permission entries in the Access Template causes the permission settings in the directory to change accordingly.

Active Roles includes a suite of pre-defined Access Templates. The list of permission entries in a pre-defined Access Template cannot be modified. If you need to add, modify, or delete permission entries from a pre-defined Access Template, create a copy of that Access Template, and then make changes to the copy. Another option is to create an Access Template and nest the pre-defined Access Template into the newly created Access Template. For instructions, see Steps for creating an Access Template, Steps for copying an Access Template, and Steps for managing nested Access Templates.

Steps for modifying permissions in an Access Template

To modify a permission entry in an Access Template

1. In the console tree, under **Configuration | Access Templates**, locate and select the folder that contains the Access Template you want to modify.
2. In the details pane, right-click the Access Template, and click **Properties**.
3. On the **Permissions** tab, select the permission entry you want to modify, click **View/Edit**, and then use the tabs in the **Modify Permission Entry** dialog box to make changes to the permission entry.

   For detailed instructions on how to view or modify a permission entry in an Access Template, see Steps for creating an Access Template earlier in this document.

NOTE: '

- The **Permissions** tab in the **Properties** dialog box lists the permission entries that are configured in the Access Template. You can use the **Permissions** tab to add, modify, or delete permission entries from the Access Template.
- The options on the **Permissions** tab in the **Modify Permission Entry** dialog box are read-only. If you need to choose a different option for the permission entry, you should delete the permission entry and then add a new permission entry with the option you need. For instructions, see Steps for adding permissions to an Access TemplateSteps for adding permissions to an Access Template.
- Once an Access Template is applied within Active Roles to determine permission settings in the directory, any changes to the list of permission entries in the Access Template causes the permission settings in the directory to change accordingly.
- Active Roles includes a suite of pre-defined Access Templates. The permission entries in a pre-defined Access Template cannot be modified. If you need to modify a permission entry in a pre-defined Access Template, create a copy of that Access Template, and then make changes to the copy. For instructions, see Steps for copying an Access Template.

Steps for removing permissions from an Access Template

To delete a permission entry from an Access Template

1. In the console tree, under **Configuration | Access Templates**, locate and select the folder that contains the Access Template you want to modify.
2. In the details pane, right-click the Access Template, and click **Properties**.
3. On the **Permissions** tab, select the permission entry you want to delete, click **Remove**, and then click **Yes** to confirm the deletion.

NOTE:

- The **Permissions** tab lists the permission entries that are configured in the Access Template. You can use the **Permissions** tab to add, modify, or delete permission entries from the Access Template.
- Once an Access Template is applied within Active Roles to determine permission settings in the directory, any changes to the list of permission entries in the Access Template causes the permission settings in the directory to change accordingly.
- Active Roles includes a suite of pre-defined Access Templates. Permission entries cannot be deleted from a pre-defined Access Template. If you need to modify the list of permission entries found in a pre-defined Access Template, create a copy of that Access Template, and then make changes to the copy. For instructions, see Steps for copying an Access Template.

Nesting Access Templates

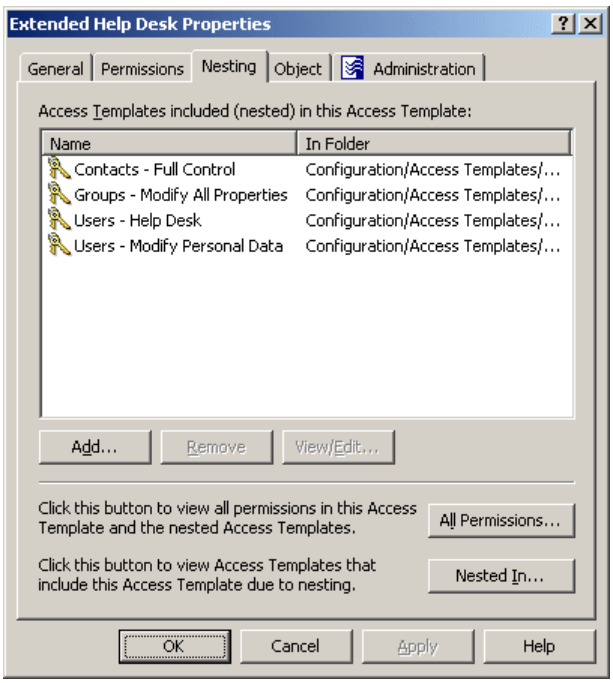
Active Roles makes it possible to define permissions in an Access Template by including (nesting) other Access Templates. This reduces the work required if you need to create a new Access Template that is similar to an existing one. Instead of modifying an existing Template to add new permissions, you can nest it into a new Access Template.

This feature simplifies Access Template management by re-using the existing preconfigured or custom Access Templates. For example, if you need to add permissions to the pre-defined Help Desk Access Template, you can create a new Access Template, nest the Help Desk Access Template into the new Access Template, and add permissions to the new Access Template as needed.

To nest Access Templates to a given Access Template, use the **Nesting** tab in the **Properties** dialog box for that Access Template.

The **Nesting** tab lists all Access Templates that are included (nested) in the selected Access Template, similar to the following figure:

Figure 21: Nesting Access templates



Each entry in the list provides the following information:

- **Name** The name of the nested Access Template.
- **In Folder** Path to the container that holds the nested Access Template.

You can manage the list on the **Nesting** tab by using the button beneath the list:

- **Add** Click this button to select Access Templates you want to nest into the Access Template being administered.
- **Remove** Select Access Templates from the list and click this button to remove them from the Access Template being administered.
- **View/Edit** Select an Access Template from the list and click this button to view or modify the selected Access Template.

From the **Nesting** tab, you can also access the following information:

- **All Permissions** Displays all permissions in the Access Template, including those that come from the nested Access Templates.
- **Nested In** Displays a list of Access Templates in which the Access Template is included due to nesting.

Steps for managing nested Access Templates

To configure an Access Template to include another Access Template

1. In the console tree, under **Configuration | Access Templates**, locate and select the folder that contains the Access Template you want to configure.
2. In the details pane, right-click the Access Template, and click **Properties**.
3. On the **Nesting** tab, click **Add**, and then select the Access Template you want to be included in the Access Template you are configuring.

NOTE:

- Configuring an Access Template to include another Access Template is referred to as *nesting*. The **Nesting** tab provides a list of Access Templates that are nested into the Access Template. You can add Access Templates to the list or remove Access Templates from the list.
- Nesting an Access Template into a target Access Template causes the list of permission entries in the target Access Template to be extended with the permission entries of the nested Access Template. Thus, if Access Template A is nested into Access Template B, all the permission entries found in Access Template A are added to the list of permission entries in Access Template B.
- You can view a consolidated list of permission entries for the Access Template: On the **Nesting** tab, click **All Permissions**. The list includes both the permission entries that are configured in the Access Template and the permission entries found in each Access Template that is nested into the Access Template. Note that the **Permissions** tab in the **Properties** dialog box lists only those permission entries that are configured in the Access Template. The permission entries that are inherited from other Access Templates by reason of nesting are not listed on the **Permissions** tab.
- You can view the Access Templates into which the selected Access Template is nested: On the **Nesting** tab, click **Nested In**. Double-clicking items in the **Nested In** list opens the **Properties** dialog box for each of the Access Templates that the selected Access Template is nested into.
- Nesting allows you to reuse the existing pre-defined or custom Access Templates. For example, if you need to add permission entries to the pre-defined Access Template **Help Desk**, then you can create a new Access Template, nest the **Help Desk** Access Template into the newly created Access Template, and add permission entries to the new Access Template as needed.

Copying an Access Template

With the Active Roles console, you can create copies of Access Templates. This feature helps you re-use existing Access Templates. For example, if you need to modify a

predefined Access Template, you can create a copy of that Access Template and then modify the copy as needed.

To create a copy of an Access Template, right-click the Access Template, and click **Copy**. This opens the Copy Object – Access Template wizard. Type a name and description for the copy, and then click **Next**.

On the next page, the wizard displays a list of permission entries. By default, the list includes all entries defined in the original Access Template. You can modify the list in the same way as on the **Permissions** tab in the **Properties** dialog box for an Access Template (see Adding, modifying, or removing permissions earlier in this chapter). When you are done with the list of permission entries, click **Next**, and then click **Finish** to complete the wizard.

Steps for copying an Access Template

To copy an Access Template

1. In the console tree, under **Configuration | Access Templates**, locate and select the folder that contains the Access Template you want to copy.
2. In the details pane, right-click the Access Template, and then click **Copy** to start the Copy Object - Access Template wizard.
3. On the first page of the wizard, do the following, and then click **Next**:
   a. In the **Name** box, type a name for the new Access Template.
   b. In the **Description** box, type any optional information about the new Access Template.
4. On the second page of the wizard, you can add, modify, and delete the permission entries that were copied from the original Access Template. Do the following, and then click **Next**:
   - To add a permission entry to the new Access Template, click **Add**.
   - To modify a permission entry for the new Access Template, select the entry from the list, and click **View/Edit**.
   - To delete a permission entry from the new Access Template, select the entry from the list, and click **Remove**.

   For detailed instructions on how to add or modify a permission entry, see Steps for creating an Access Template earlier in this document.
5. Click **Finish** to complete the creation of the new Access Template.

Exporting and importing Access Templates

With the Active Roles console, you can export Access Templates to an XML file and then import them from that file to populate another instance of Active Roles. The export and

import operations provide a way to move Access Templates from a test environment to a production environment, and vice versa.

> NOTE: When you export and then import an Access Template, only permission entries are transferred. The Access Template links are not exported, and therefore you need to reconfigure them manually after you have imported the Access Template.

To export Access Templates, select them, right-click the selection, and select **All Tasks | Export**. In the **Export Objects** dialog box, specify the file where you want to save the data, and click **Save**.

To import Access Templates, right-click the container where you want to place the Access Templates, and then click **Import**. In the **Import Directory Objects** dialog box, select the file to which the Access Templates were exported, and click **Open**.

Renaming an Access Template

To rename an Access Template, right-click the Access Template, and click **Rename**. Type the new name, and then press ENTER.

Renaming an Access Template does not affect its links. This is because Access Templates are referenced by immutable identifier rather than by name.

Steps for renaming an Access Template

To rename an Access Template

1. In the console tree, under **Configuration | Access Templates**, locate and select the folder that contains the Access Template you want to rename.
2. In the details pane, right-click the Access Template, and click **Rename**.
3. Type a new name, and then press ENTER.

NOTE:

- If an Access Template is applied within Active Roles to determine permission settings in the directory, renaming the Access Template does not cause any changes to the permission settings in the directory. When applying an Access Template, Active Roles refers to the Access Template by an internal identifier rather than by the name of the Access Template.
- Active Roles includes a suite of pre-defined Access Templates. The name of a pre-defined Access Template cannot be modified. If you need an Access Template with a different name to have the same permission entries as a pre-defined Access Template, create a copy of the pre-defined Access Template, and then make changes to the copy. Another option is to create an Access Template and nest the pre-defined Access Template into the newly created Access Template. For instructions, see Steps for creating an Access Template, Steps for copying an Access Template, and Steps for managing nested Access Templates.

Deleting an Access Template

To delete an Access Template, you must first remove all references to the Access Template:

- Delete the links to the Access Template (see Managing Access Template links earlier in this chapter).
- Remove the Access Template from all Access Templates in which the Access Template is nested (see Nesting Access Templates earlier in this chapter).

Then, you can perform the deletion: right-click the Access Template and click **Delete**.

Steps for deleting an Access Template

To delete an Access Template

1. In the console tree, under **Configuration | Access Templates**, locate and select the folder that contains the Access Template you want to delete.
2. In the details pane, right-click the Access Template, and then click **Delete**.

NOTE:

- Once an Access Template is applied (linked) within Active Roles to determine permission settings in the directory, the Access Template cannot be deleted. You can view the links in which the Access Template participates: Right-click the Access Template, and click **Links**. If you need to delete the Access Template, first remove all items from the **Links** list. For instructions, see Steps for managing Access Template links.
- An Access Template cannot be deleted if it is nested into another Access Template. You can view the Access Templates into which the selected Access Template is nested: On the **Nesting** tab, click **Nested In**. Double-click an item in the **Nested In** list to open a dialog box where you can remove the Access Template from nesting. For instructions, see Steps for managing nested Access Templates.
- Active Roles includes a suite of pre-defined Access Templates and a number of built-in Access Templates. Neither pre-defined Access Templates nor built-in Access Templates can be deleted.

Examples of use

This section discusses scenarios to help you understand and use the role-based administration features available in Active Roles. The following scenarios are covered:

- Scenario 1: Implementing a Help Desk
- Scenario 2: Implementing Self-administration

Scenario 1: Implementing a Help Desk

This scenario shows how to use an Access Template that allows a Help Desk service to perform day-to-day operations on user accounts, such as resetting passwords, viewing user properties, locking and unlocking user accounts.

The scenario also involves a group to hold Help Desk operators. The Access Template is applied so that the group is designated as a Trustee, thus giving the administrative rights to the Help desk operators. When both the Access Template and group are prepared, you can implement a Help Desk administration in your enterprise.

Suppose you need to authorize the Help Desk to manage user accounts in the **Sales** organizational unit. To implement this scenario, you should perform the following steps:

1. Prepare a **Help Desk** Access Template that defines the Help Desk operator permissions on user accounts.
2. Create and populate a **Help Desk** group to hold the Help Desk operators.

3. Apply the **Help Desk** Access Template to the **Sales** organizational unit, selecting the **Help Desk** group as a Trustee.

As a result of these steps, each member of the **Help Desk** group is authorized to perform management tasks on user accounts in the **Sales** organizational unit. The **Help Desk** Access Template determines the scope of the tasks.

The following sections elaborate on each of these steps.

Step 1: Preparing a Help Desk Access Template

For the purposes of this scenario, you can use the predefined Access Template **Users – Help Desk**, located in the folder **Configuration/Access Templates/Active Directory**. The **Users – Help Desk** Access Template specifies the necessary permissions to reset user passwords, unlock user accounts, and view properties of user accounts.

If you want to add or remove permissions from the **Users – Help Desk** Access Template, you need to first create a copy of that Access Template and then modify and apply the copy.

This scenario assumes that you apply the predefined Access Template **Users – Help Desk**.

Step 2: Creating a Help Desk group

To create a group, right-click an organizational unit in the console tree, select **New | Group**, and then follow the instructions in the New Object – Group wizard. The wizard includes the page where you can add members (Help Desk operators) to the group you are creating.

For step-by-step instructions on how to create groups, see "Steps for Creating a Group" in the Active Roles User Guide or Active Roles Help.

Step 3: Applying the Help Desk Access Template

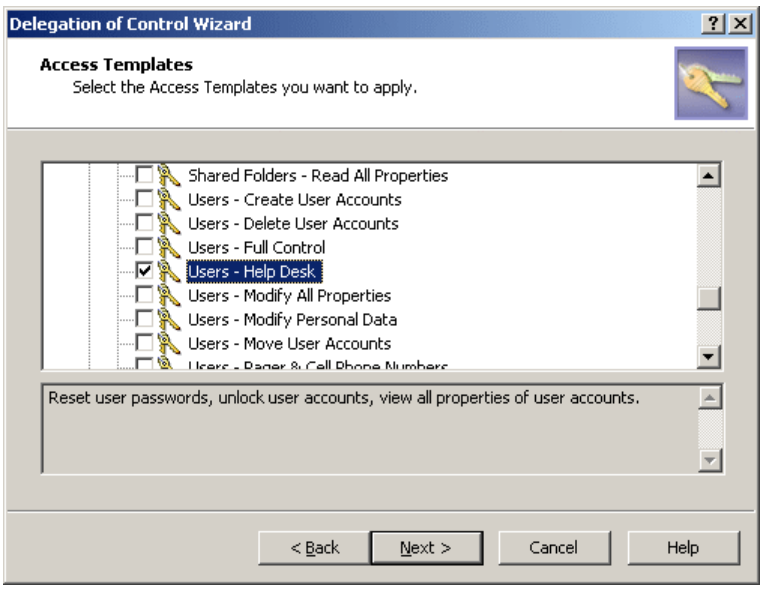
You can apply the Access Template using the Delegation of Control wizard.

First, you start the wizard on the **Sales** organizational unit: right-click the organizational unit, click **Delegate Control**, and then, in the **Active Roles Security** window, click the **Add** button.

Next, on the **Users or Groups** page of the wizard, add the **Help Desk** group to the list.

Next, on the **Access Templates** page of the wizard, expand **Access Templates | Active Directory** and select the check box next to **Users - Help Desk**, as shown in the following figure.

Figure 22: Access Template - Delegation of control



Click **Next** and accept the default settings in the wizard. On the completion page, click **Finish**. Finally, click **OK** to close the **Active Roles Security** window.

For more information about the Delegation of Control wizard, see Applying Access Templates earlier in this chapter.

Scenario 2: Implementing Self-administration

This scenario shows how to use an Access Template that allows users to modify certain portions of their personal information in Active Directory.

The Active Roles Web Interface provides the Site for Self-Administration to manage user accounts. The site displays users their personal information, such as the first and last names, address information, phone numbers, and other data. By default, Web Interface users are only authorized to view their personal information. To enable the users to also modify their personal information, you must give them additional permissions.

Suppose you need to authorize the users in the **Sales** organizational unit to perform self-administration. To implement this scenario, you should perform the following steps:

1. Prepare a **Self-Administration** Access Template that defines the appropriate permissions on user accounts.
2. Apply the **Self-Administration** Access Template to the **Sales** organizational unit, selecting the **Self** object as a Trustee.

As a result of these steps, users from the **Sales** organizational unit are authorized to perform self-management tasks on their personal accounts. The **Self-Administration** Access Template determines what data the users are permitted to modify. Users can

manage their personal information via the Site for Self-Administration. For information about the Site for Self-Administration, refer to the Active Roles Web Interface User Guide.

The following sections elaborate on the steps involved in this scenario.

Step 1: Preparing a self-administration Access Template

For the purposes of this scenario, you can use the predefined Access Template **Self - Account Management**, located in the folder **Configuration/Access Templates/User Self-management**. This Access Template specifies the necessary permissions to view a basic set of user properties and modify telephone numbers.

If you want to add or remove permissions from the **Self - Account Management** Access Template, you need to first create a copy of that Access Template and then modify and apply the copy.

This scenario assumes that you apply the predefined Access Template **Self - Account Management**.

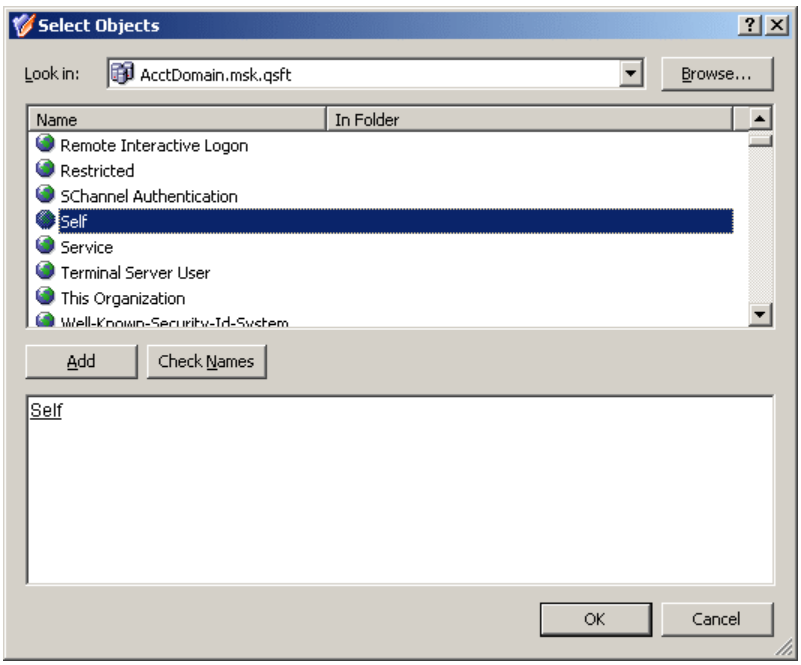
Step 2: Applying the self-administration Access Template

You can apply the Access Template using the Delegation of Control wizard.

First, you start the wizard on the **Sales** organizational unit: right-click the organizational unit, click **Delegate Control**, and then, in the **Active Roles Security** window, click the **Add** button.

Next, on the **Users or Groups** page of the wizard, click the **Add** button. In the **Select Objects** window, select the **Self** object, as shown in the following figure, click **Add**, and then click **OK**.

Figure 23: Access Template - Self administration



Next, on the **Access Templates** page of the wizard, expand **Access Templates | User Self-management** and select the check box next to **Self - Account Management**.

Click **Next** and accept the default settings in the wizard. On the completion page, click **Finish**. Finally, click **OK** to close the **Active Roles Security** window.

For more information about the Delegation of Control wizard, see Applying Access Templates earlier in this chapter.

Deployment considerations

Active Roles utilizes role-based delegation for assigning of administrative permissions. The benefits of this model are that a role can be created once and delegated to multiple groups of users that fit that role. If a change is needed, an update to the role will take effect for everyone. These roles are referred to as "Access Templates."

When doing delegation with Active Roles, you should remember a few rules:

- Active Roles administrators (Active Roles Admins) have full control throughout the system and cannot be denied access anywhere within Active Roles. Everyone else starts with nothing and permissions are added from the ground up.
- Permissions are cumulative, an explicit deny takes precedence over an explicit allow. An explicit allow takes precedence over an inherited deny.
- You should keep your permission model as simple as possible. Sometimes this means giving users all read/write permissions and denying the ability to write a few fields.

- Do not use the default (built-in) Access Templates as they cannot be modified. Instead, copy those Access Templates and move them to a new container. This way all of the Access Templates you are using are stored within a particular structure.

There are three basic types of permissions that can be added to an Access Template:

- First is *object access*. With this permission type, you can set permissions that affect an object as a whole. For instance: Move; List; Deprovision—all these are object permissions.
- Second is *object property access*. These are used to control access to individual attributes of an object, such as an object's description, samAccountName, or homeFolder. With this permission type, you can delegate granular rights over an object. However just because the rights that can be delegated can be granular does not mean that they should. For instance, if a helpdesk operator needs to be able to manage a large set of user properties, it makes more sense to delegate read/write for all properties as this is one permission entry instead of delegating read/write for every individual attribute since each attribute would need to have its own permission entry.
- Third is *child object creation/deletion*. With this permission type, you can set permissions for creation or deletion of objects. For instance, to set up an Access Template that allows creation of users, you should add a permission entry that applies to the Organizational Unit and Container object classes, and contains a "Create child objects" permission for the User object class.

The following sections give a sample set of the permissions necessary for certain delegation scenarios:

- Delegation of Organizational Unit administration
- Delegation of group administration

Delegation of Organizational Unit administration

The following table lists a sample set of permission entries for a scenario of delegating administration of Organizational Units:

Table 9: Permission entries for delegating administration of Organizational Units

Object Class	Permission Type	Attribute or Permission
Domain	Object Access	Allow List
Domain	Object Property Access	Allow Read All Properties
Domain	Object Property Access	Allow Write LDAP Server (permission to change Operational Domain Controller)

Object Class	Permission Type	Attribute or Permission
Organizational Unit	Object Access	Allow List
Organizational Unit	Object Property Access	Allow Read All Properties
Organizational Unit	Child Object Creation/Deletion	Allow Create/Delete Users
User	Object Access	Allow List
User	Object Property Access	Allow Read/Write All Properties
User	Object Property Access	Deny Write Employee ID

This set of permission entries has several important characteristics:

- It allows access to the Domain and the Organizational Unit object classes. This is because without access to the domain and the Organizational Units a delegated administrator cannot see the users beneath. This access should always include the List and Read All Properties permissions.
- It gives a delegated administrator the ability to create and delete user objects. This permission applies to the Organizational Unit object class.
- It gives a delegated administrator the ability to see (List) users and modify any property except Employee ID.

Delegation of group administration

The following table lists a sample set of permission entries for a scenario of delegating administration of groups:

Table 10: Permission entries for delegating administration of groups

Object Class	Permission Type	Attribute or Permission
Domain	Object Access	Allow List
Domain	Object Property Access	Allow Read All Properties
Domain	Object Property Access	Allow Write LDAP Server (permission to change Operational Domain Controller)
Organizational Unit	Object Access	Allow List
Organizational Unit	Object Property Access	Allow Read All Properties
Organizational Unit	Child Object Creation/Deletion	Allow Create/Delete Groups

Object Class	Permission Type	Attribute or Permission
Group	Object Access	Allow List
Group	Object Property Access	Allow Read All Properties
Group	Object Property Access	Allow Write Members
User	Object Access	Allow List
User	Object Property Access	Allow Read All Properties

This set of permission entries has several important characteristics:

- It allows access to the Domain and the Organizational Unit object classes. This is because without access to the domain and the Organizational Units a delegated administrator cannot see the groups and users beneath. This access should always include the List and Read All Properties permissions.
- It gives a delegated administrator the ability to create and delete group objects. This permission applies to the Organizational Unit object class.
- It gives a delegated administrator the ability to see (List) groups, view any property of a group (Read All Properties), and add or remove members from a group (Write Members).
- It gives a delegated administrator the ability to see (List) users and view any property of a user (Read All Properties). This is necessary for a delegated administrator to be able to add users to a group.

Delegation in a functional vs. hosted environment

For your delegation model to work correctly, you need to determine whether you have a functional or hosted environment.

Delegation in a functional environment

In a functional environment there is a separate group of administrators for each function. So there may be a group for managing users, a helpdesk, domain administrators, and Exchange administrators. In case of a functional environment, you need to decide on a certain role for each function. These roles usually cross Organizational Unit boundaries so delegation is typically done at the root of the domain or domains. Typically a delegation model for this scenario would look something like the following:

Table 11: Delegation model in a functional environment

Location / Template	Permission	Delegate (Trustee)
Domain / Read All Objects	• All Objects - List • All Objects - Read All Properties • Domain - Write LDAP Server Property (permission to change Operational Domain Controller)	Authenticated Users
Domain / User Admin	• User Objects - Full Control • Organizational Unit - Create/Delete User Objects	User Admin group
Domain / Group Admin	• Group Objects - Full Control • Organizational Unit - Create/Delete Group Objects	Group Admin group

Delegation in a hosted environment

In a hosted environment there is an admin group or set of admin groups responsible for each top-level Organizational Unit (OU). In this case administrators may not want others to see what is going on in their OU structure. Active Roles can accommodate this easily. Since except for the Active Roles administrators no one has any default rights, a delegation model may look something like the following:

Table 12: Delegation model in a hosted environment

Location / Template	Permission	Delegate (Trustee)
Domain / Read Domain	• Domain - List • Domain - Read All Properties • Domain - Write LDAP Server Property	Authenticated Users
Top-level OU / OU Admin	• All Objects - List • All Objects - Read all Properties • Organizational Unit - Create/Delete User/Group Objects • User Objects - Full Control • Group Objects - Full Control	OU Admin

With this delegation model, everyone can see the domain and change the domain controller they are using for management. However, below that only the OU admin can see their

associated OU. This keeps administrators from seeing or managing anything outside of their control.

More than likely a delegation model would incorporate features of both. For instance, you may have a hosted environment where each business unit is responsible for their own Active Directory management, with a central helpdesk to perform basic user and group management tasks.

Lastly is the issue of syncing permission to Active Directory. Although Active Roles enables you to accomplish this task, it is a better idea to keep all of the permissions within Active Roles for the following reasons:

- This protects your Active Directory. Directory-enabled applications can be modified to use the Active Roles ADSI Provider allowing for granular access to only the data and areas that are needed. Doing so helps prevent malicious software from destroying data in Active Directory.
- This ensures directory integrity. By forcing all administrators to use Active Roles, you ensure that all policies, such as naming standards, are correctly enforced.
- This gives a complete auditing picture. By having all applications and administrators use Active Roles' interfaces, you ensure that Active Roles' Report Data Collector can gather every action that happens in the directory, down to the attribute level.

Windows claims-based Access Rules

Active Roles introduces claims-based authorization rules (access rules) to allow or deny access to Active Directory objects depending on the attributes of the identity attempting to access those objects. Built on the concept of Dynamic Access Control (DAC), this feature enables Active Roles to recognize and evaluate the attribute-based claims of the identity that requests access to data held in Active Directory.

Access rules improve access control management for Active Directory administration. With access rules, Active Roles adds more flexibility and precision in delegating control of Active Directory objects, such as users, computers or groups, through the use of claims—that is, Active Directory user and computer properties—in the Active Roles authorization model.

By using access rules, you can control access to Active Directory objects based on the characteristics of both the objects and the delegated administrators requesting access to the objects. This feature enables you to define and enforce very specific requirements for granting administrative access to Active Directory data. For example, you can easily restrict access of delegated administrators to user accounts whose properties (such as department or country) match the properties of the delegated administrator's account in Active Directory.

Access rules help you create more complete access controls on Active Directory objects by comparing object properties with user and device claims. A domain controller issues claims to an identity that consist of assertions based on the properties of that identity retrieved from Active Directory. When an identity requests access to a particular object, Active Roles evaluates the claims of that identity and the properties of that object against the access

rules, and then, depending upon the evaluation results, applies the appropriate Access Templates to make an authorization decision.

Understanding Access Rules

Access rules enable administrators to apply access-control permissions and restrictions based on well-defined conditions that can include the properties of the target objects, the properties of the user who requests access to target objects, and the properties of the device from which the user requests access to target objects. For example, when the user's role or job changes (resulting in changes to the attributes of the user's account in Active Directory), access rules can cause the user's permissions to change dynamically without additional intervention from the administrator.

An access rule is an expression of authorization rules that can include conditions that involve user groups, user claims, device groups, device claims, and target object properties. When you apply an Access Template, you can use an access rule to determine the conditions that must be satisfied for the permissions resulting from the Access Template to take effect.

Conditional Access Template links

Active Roles enhances its authorization model by introducing conditional Access Template links, and takes advantage of conditional links by inserting user claims, device claims, and target object properties, into conditional expressions specified in access rules. An access rule can be applied to an Access Template link, causing the link to have an effect only if the access rule's condition evaluates to TRUE. During permission check, Active Roles inserts the claims and properties into conditional expressions found in the access rule, evaluates these expressions, and enables or disables the Access Template link based on results of the evaluation. In this way, the access rule determines the results of the permission check.

Access rules, along with conditional Access Template links, enable Active Roles to leverage claims for authorization to securable objects. This authorization mechanism (known as claims-based access control) supplements Access Template based access control to provide an additional layer of authorization that is flexible to the varying needs of the enterprise environment.

Prerequisites for using Access Rules

Before you can use Access Rules, the following conditions must be fulfilled:

- Claim support must be enabled in your Active Directory domain. For details, review the topic Enabling claim support, later in this document.
- For Access Rules to use device claims, Group Policy setting **Computer Configuration\ Policies\Administrative**

Templates\System\Kerberos\Support Compound Authentication with the **Always** option must be enabled on the client computers, in addition to the **Kerberos client support for claims, compound authentication and Kerberos armoring** setting (see Client computer).

- The Active Roles Administration Service must be installed on a computer running Windows Server 2012 or a later version of the Windows Server operating system.
- The Active Roles Administration Service that performs authorization using Access Rules must be installed in the Active Directory forest where the user account of the authorizing user is defined and in which the claim types used by the Access Rules are created. Active Roles does not support the use of Access Rules for cross-forest authorization.
- Group Policy setting **Computer Configuration\Policies\Administrative Templates\ System\Kerberos\Kerberos client support for claims, compound authentication and Kerberos armoring** must be enabled on the computer running the Administration Service.
- The Administration Service must be configured to support Kerberos authentication.

Configuring the Administration Service to support Kerberos authentication

Access Rules require the Active Roles Administration Service to support Kerberos authentication. This is because Windows claims are delivered inside Kerberos tickets. To enable Kerberos authentication, the Service Principal Name (SPN) of the Active Roles Administration Service must be added to the service account (domain user account under which the Administration Service runs). For example, suppose that:

- `arsrv.domain.com` is the FQDN of the computer running the Administration Service
- `arsrv` is the name of the computer running the Administration Service

In this example, the following SPNs must be added to the service account:

- `aradminsvc/arsrv.domain.com`
- `aradminsvc/arsrv`

You can add the SPNs to the service account by using the Setspn command line tool:

1. `setspn -s aradminsvc/<FQDN> <ServiceAccountName>`

   For example, `setspn -s aradminsvc/arsrv.domain.com domain\arsvcacct`
2. `setspn -s aradminsvc/<name> <ServiceAccountName>`

   For example, `setspn -s aradminsvc/arsrv domain\arsvcacct`

Managing Windows claims

Claims are statements about an authenticated user or device, issued by an Active Directory domain controller running Windows Server 2012 or later. Claims can contain information about the user or device retrieved from Active Directory.

Dynamic Access Control (DAC), a feature of Windows Server 2012, employs claims-based authorization to create versatile and flexible access controls on sensitive resources by using access rules that evaluate information about the user who accesses those resources and about the device from which the user accesses those resources. By leveraging claims in the user's authentication token, DAC makes it possible to allow or deny access to resources based on the Active Directory attributes of the user or device.

Active Roles uses claims-based access rules to improve authorization management for Active Directory administration. With claims-based access rules, Active Roles adds more flexibility and precision in delegating control of Active Directory objects, such as users, computers or groups, by extending the Active Roles authorization model to recognize and evaluate the claims specific to the user who requests access to those objects or device used to request access.

Enabling claim support

Claims-based authorization requires:

- Domain controller running Windows Server 2012 or later, with claim support enabled
- Domain-joined client computer running Windows 8, Windows Server 2012 or a later version of the Windows operating system, with claim support enabled (if you need to use device claims)

Domain controller

The claims-based authorization mechanism requires extensions to Active Directory, such as claim type objects intended to store the claim configuration data. By adding a Windows Server 2012 domain controller, you extend the Active Directory schema to provide the object classes and attributes required to support claims-based authorization.

Another requirement is the enhancements in the Kerberos Key Distribution Center (KDC) and Security Accounts Manager (SAM) that enable domain controllers running Windows Server 2012 to recognize claim types, retrieve claim information, and transport claims within Kerberos tickets.

A Windows Server 2012 domain controller that supports claim issuance understands claim types published in Active Directory. Claim types define the claim source attributes. When servicing an authentication request, the domain controller reads the source attribute from the claim type and retrieves the attribute data for the authenticating user. Then, the retrieved attribute data is included in the Kerberos ticket and returned to the requestor.

By default, Windows Server 2012 domain controllers do not support claim issuance. You need to enable claim issuance by using Group Policy. The Group Policy setting that serves

this purpose is **Computer Configuration\Policies\Administrative Templates\System\KDC\KDC support for claims, compound authentication and Kerberos armoring**. Enable this policy setting in a Group Policy object applied to the **Domain Controllers** Organizational Unit (for example, in the **Default Domain Controllers Policy** object), and confirm that this policy setting has the **Supported** option selected.

Claims-based authorization does not impose domain or forest functional requirements. If your Active Directory domain has a sufficient number of Windows Server 2012 domain controllers to service authentication requests that include claim information, then you can make use of Windows claims even though Windows Server 2008 R2 domain controllers exist in your Active Directory domain.

Client computer

Domain-joined client computers running Windows 8 or Windows Server 2012 are required for claims-based authorization when using device claims. A domain controller issues claims in the Kerberos ticket in response to an authentication request created by a client computer, so the computer needs to understand how to request claim information when making authentication requests, and how to locate a claims-aware domain controller. Computers running earlier versions of the Windows operating system don't have such knowledge, so they can't request user or device claims upon user authentication. Although applications and services that require claim information can request user claims on their own, this is not the case with the device claims. If you need to use device claims, the user must log on from a computer running Windows 8, Windows Server 2012, or a later version of the Windows operating system. This requirement does not apply to authorization scenarios that employ user claims only.

By default, Windows 8 and Windows Server 2012 based computers don't request user or device claims upon user authentication. You need to enable claim support on client computers by using Group Policy. The Group Policy setting that serves this purpose is **Computer Configuration\Policies\ Administrative Templates\System\Kerberos\Kerberos client support for claims, compound authentication and Kerberos armoring**. Enable this policy setting in a Group Policy object applied to the Organizational Unit that holds the computer accounts of client computers.

Claim Type management overview

After you enable the **KDC support for claims, compound authentication and Kerberos armoring** Group Policy setting, your Windows Server 2012 domain controllers are ready to issue claims in response to authentication requests. However, you need to configure claim types before the domain controller can issue claims.

You can use Active Roles to create attribute-based claim types that source their information from user and computer attributes. The claim types you create are stored in the configuration partition of the Active Directory forest. All domains within that forest share the claim types and domain controllers from those respective domains issue claim information during user authentication.

It is important that the Active Directory attributes intended to source claim types contain accurate information. Incorrect attribute information can lead to unexpected access to data using claims-based authorization. You can ensure the accuracy of information held in claim source attributes by leveraging property generation and validation policies provided by Active Roles.

You can use the Active Roles console to create, modify and delete user and computer claim types. The claim type objects are stored in the configuration partition of the Active Directory forest, and appear under the **Active Directory | Claim Types** node in the Active Roles console. If you have domains from multiple forests registered with Active Roles, then the console tree provides a separate **Claim Types** node for each forest. The forest to which a given **Claim Types** node applies is identified by the name (or a part of the name) of the forest root domain shown in brackets next to the name of the node.

The Active Roles console provides the following pages for creating and modifying claim types:

- **Source Attribute** On this page you can select the Active Directory attribute from which the claim value is obtained, specify the display name and description for the claim type, and choose whether the claim type applies to user, computer, or both.
- **Suggested Values** This page allows you to configure predetermined selectable values from which you can choose when using the claim type in a conditional expression for an access rule.

On these pages you can view or change the following configuration settings.

Source attribute setting

On the **Source Attribute** page you can select, view or change the source attribute for the claim type. The source attribute is the Active Directory attribute from which the value is obtained for claims of this claim type.

The page provides a list allowing you to select the desired attribute. The list includes the attributes for the User, Computer, InetOrgPerson, ManagedServiceAccount, GroupManagedServiceAccount and Auxiliary classes of object, with the exception of:

- Attributes marked as defunct in the Active Directory schema
- Password attributes such as dBCSPwd, lmPwdHistory, and unicodePwd
- Attributes that are not replicated among domain controllers
- Attributes that are not available on read-only domain controllers
- Attributes with an Active Directory syntax type other than
    - String: DN String, Unicode, NT Security Descriptor, or Object ID
    - Integer or Large Integer
    - Boolean

For an existing claim type, the page displays the claim type's current source attribute, and allows you to select a different attribute of the same syntax type. However, changing the source attribute does not change the claim type's ID.

Claim type identifier setting

The claim type identifier (ID) determines the Common Name (cn) of the claim type object in Active Directory. Normally, Active Roles automatically generates an ID when creating a claim type. The automatically generated ID has the following format:

`ad://ext/attributeName:uniqueHexidecimalNumber`

In this format, `attributeName` stands for the LDAP display name of the claim type's source attribute and `uniqueHexidecimalNumber` is a randomly generated string of hexadecimal characters that ensures the uniqueness of the claim type's ID.

To enable authorization scenarios where claims are used across a forest trust, you need to create claim types in both the trusted forest and trusting forest with the same claim type ID. Domain controllers in a trusting forest receiving claims from a trusted forest cannot understand these claims unless:

- Each claim has a claim type object created in both forests
- The claim type ID in the trusting forest is identical to the claim type ID in the trusted forest
- A Claim Transformation Policy object is applied to allow incoming claims across the forest trust

Therefore, when you create a claim type object, you may need to specify the appropriate claim type ID by hand. The option **Set ID to a semantically identical claim type in a trusted forest** serves this purpose, allowing you to type in an ID instead of having it created automatically. If you choose to enter an ID by hand, ensure that your ID string specifies a unique ID and conforms to the following format:

- Starts with the `ad://ext/` prefix
- The prefix is followed by 1 to 32 characters
- Does not contain space characters or these characters: `\ * ? " < > |`
- If a slash mark (`/`) occurs after the `ad://ext/` prefix, then the slash mark must be surrounded by a character on each side. The surrounding character must not be a colon (`:`) or slash mark.

A valid example of an ID string is `ad://ext/BusinessImpact`.

The option **Set ID to a semantically identical claim type in a trusted forest** is available only when you create a claim type object. The ID should not be changed on existing claim type objects. When you create a claim type object, it is advisable to let an ID be generated automatically unless a business need justifies otherwise, such as the use of claim transformation policies in a multi-forest environment. This ensures that the newly created claim type has a valid, unique ID.

Display name setting

The display name of the claim type object is used to represent the claim type as a choice throughout the user interface. Thus, when you configure a conditional expression for an access rule, the condition builder allows you to select a claim type from a list where each

list item is the display name of a certain claim type object. For this reason, each claim type object must be given a unique display name. The display name accepts alphanumeric characters as valid data.

Description setting

You can use the description of the claim type object to specify a short comment about the claim type. Comments typically include purpose, department usage, or business justification.

User or computer claim issuance setting

You have the option to choose whether claims of the given claim type can be issued for user or computer object class, or both. With the option to issue claims for the user object class, the claim type causes domain controllers to issue user claims based on the attribute of the authenticating user. With the option to issue claims for the computer object class, the claim type causes domain controllers to issue device claims based the attribute of the authenticating user's computer. You can configure a claim type to issue both user and device claims. When you create a conditional expression for an access rule, and choose the claim type to evaluate, the condition builder allows you to distinguish between user and device claims of the same claim type.

Protection from accidental deletion

By default, claim type objects are protected from accidental deletion. This option prohibits all users, including domain and enterprise administrators, from deleting the claim type object. Protection is achieved by adding an explicit permission entry to the claim type object that denies everyone the right to delete the object. When you create a claim type object, the option to protect the object from accidental deletion is selected by default. As a best practice, it is advisable to leave this option selected.

Suggested values setting

The suggested values setting allows you to configure predefined values from which you can choose when using the claim type in a conditional expression. If you create a claim type without suggested values, you will have to type rather than select values in the condition builder. Another option is to create one or more suggested values for the claim type. These values will appear in a list provided by the condition builder.

You can add, edit or remove suggested values for a given claim type when creating or modifying the respective claim type object. When you add or edit a suggested value, you are prompted to complete the following fields:

- **Value** This value data will be used when evaluating conditional expressions that include the suggested value you are configuring.

- **Display name** This is the name of the suggested value that appears in the list when you configure a conditional expression.

Steps for managing Claim Types

Claim types must be created in Active Directory to enable domain controllers to issue claims to users or computers. Claims issued by the domain controller are sourced from attributes of user or computer accounts stored in Active Directory. Claim types specify the attributes from which the claims are sourced, and contain metadata required for using claims.

New claim types are created in the **Claim Types** container under the **Active Directory** node located in the Active Roles console tree. If you have domains from multiple forests registered with Active Roles, then the console displays an individual **Claim Types** container for each forest that has domain controllers running Windows Server 2012 or a later version of the Windows Server operating system. To identify the forest of a given **Claim Types** container, the container's name includes the name (or a part of the name) of the forest root domain.

To create a new claim type

1. Right-click the **Claim Types** container, and select **New | Claim Type**.
2. On the **Source Attribute** page, select the desired source attribute for claims of this type.
3. Review the auto-generated display name and description, and change them if needed.
4. Under **Claims of this type can be issued for the following classes**, select:
   - The **User** check box to enable issuance of this claim type to users
   - The **Computers** check box to enable issuance of this claim type to computers
5. Select the **Set ID to a semantically identical claim type in a trusted forest** check box if the claim type must match an existing claim type in a different forest. Type the claim identifier. Clear this check box to generate the claim identifier automatically.
6. Select the **Protect from accidental deletion** check box to ensure an administrator cannot accidentally delete the claim type. Clear the check box to remove accidental deletion protection.
7. Click **Next** to proceed to the **Suggested Values** page.
8. Click the option you want for suggested values. Create suggested values as needed.
9. Click **Finish**.

To modify an existing claim type

1. Right-click the claim type you want to modify and then click **Properties**.

2. On the **Source Attribute** page, view or change the source attribute, the display name, description, user or computer claim issuance options, and the option to protect the claim type from accidental deletion.
3. Click the **Suggested Values** tab to view or change suggested values.
4. Click **OK** to save the modified claim type.

To delete a claim type

1. Right-click the claim type and then click **Delete**.
2. Confirm the claim type deletion.

If you encounter a message stating that you don't have permission to delete the claim type, then modify the claim type and clear the **Protect from accidental deletion** check box. If this check box is cleared, verify that you have sufficient rights to delete claim type objects.

Enabling and disabling claim types

Windows claim types have two states: disabled and enabled. Disabled claim types are valid claim types, but are unavailable for use in production. Claims of disabled claim types are not issued by domain controllers and disabled claim types are filtered from view in the access rule condition builder. A claim type becomes available for production use once you enable it. Active Roles creates enabled claim types, and allows you to disable and enable claim types as needed.

To disable an enabled claim type

- Right-click the claim type object and click **Disable**.

To enable a disabled claim type

- Right-click the claim type object and click **Enable**.

Populating claim source attributes

Creating a claim type object makes the Active Directory forest aware of the claim type. However, claim type objects do not provide information held in the actual claims. When issuing claims, domain controllers retrieve that information from user and computer objects. Hence, in addition to claim type objects, user and computer objects must contain the information necessary for domain controllers to issue claims.

Attribute-based claim types define the attributes from which to source the claims. These are attributes of user and computer objects. Each claim type object specifies a certain attribute that the domain controller retrieves when creating and issuing claims of that type. During authentication of a user, the claim-aware domain controller reads all enabled claim types from the user's Active Directory forest, and maps them to the attributes of the authenticating user or computer. Then, the domain controller retrieves information from the mapped attributes, and issues claims containing that information.

As domain controllers do not issue blank claims, you may encounter a situation where you have created a valid claim type but the domain controller does not issue the claim during authentication. This is because a claim type object merely maps claims to a certain attribute, directing the domain controller to issue claims based on the information present in that attribute. If the attribute of the authenticating user or computer does not contain information, the domain controller does not issue the claim.

Therefore, it is important that claim source attributes contain information. Additionally, as authorization decisions depend upon information found in claims, claim source attributes must contain valid information. Incorrect attribute information can lead to unexpected access to data using claims-based authorization.

To ensure that claim source attributes contain valid information, you could periodically inspect and, if needed, set or correct the properties of users and computers by using the Active Roles console or Web Interface. However, it would be more practical to leverage property generation and validation policies provided by Active Roles. You can use policies to:

- Auto-generate the appropriate values for user and computer properties upon creation of user and computer objects
- Prevent invalid values from being assigned to user and computer properties, by applying validation rules or creating immutable lists of suggested values

Property generation and validation policies allow you to specify, and enforce, conditions that the property values must meet, and determine default property values. For further information, see Property Generation and Validation in the Active Roles Administration Guide.

Managing and applying Access Rules

Access Rules are used in Active Roles to specify conditions for authorizing user access to securable objects (target objects) that involve user groups, user claims, device groups, device claims, and target object properties. When you apply an Access Template, you can specify an Access Rule to determine the conditions that must be satisfied for the permissions resulting from the Access Template to take effect.

When configuring an Access Rule, you use Conditional expression editor to build a conditional expression for that Access Rule. Conditional expressions are logical expressions that provide a **True** or **False** result. Once an Access Rule has been configured, you can apply the Access Rule to an Access Template link (see Applying an Access Rule), which causes the link to be dynamically enabled or disabled depending upon the evaluation result of the Access Rule's conditional expression during permission check. If the expression evaluates to **True**, the link is enabled and permission check considers the Access Template permission settings; otherwise, the link is disabled and the Access Template permission settings are disregarded.

Conditional expression editor

The Access Rule management pages provide a built-in editor for configuring conditional expressions. Each Access Rule holds a certain conditional expression that evaluates during permission check. A conditional expression is composed of conditions combined using AND/OR logic. Each condition is a certain statement that specifies criteria allowing permission check to determine whether to apply a given Access Template.

When you configure a conditional expression, you need to add at least one condition, but you are not limited in the number of conditions that you can add. You can add, delete, and group conditions using various operators. It is possible to nest condition groups within other condition groups to achieve the results that you want.

A condition group contains one or more conditions connected by the same logical operator. By grouping conditions, you specify that those conditions should be evaluated as a single unit. The effect is the same as if you put parentheses around an expression in a mathematical equation or logic statement.

By default, a single condition group is created when you add a condition. You can create additional condition groups to group a set of conditions and nest grouped conditions within other condition groups.

In a condition group, conditions are connected using the AND or OR logical operator:

- AND group evaluates to TRUE if all conditions in the group are TRUE.
- OR group evaluates to TRUE if any condition in the group is TRUE.

By default, AND is the logical operator between the conditions in a condition group. It is possible to change the logical operator by converting the condition group to a different group type.

When you add a condition, the conditional expression editor first prompts you to specify what you want the condition to evaluate. The following options are available:

- **Device claim** Evaluate a computer claim, or groups the computer account is a member of. You can choose one of the existing computer claim types or, to evaluate groups, you can select the **Group** item in the claim type list provided by the condition builder.
- **Target object property** Evaluate a certain property of the object to which the authorizing user requests access. You can select the desired property from a list provided by the condition builder.
- **User claim** Evaluate a user claim, or groups the user account is a member of. You can select one of the existing user claim types or, to evaluate groups, you can select the **Group** item in the claim type list provided by the condition builder.

Once you have specified what you want the condition to evaluate, you can choose a comparison operator and specify a comparison value. The comparison operator determines the operation of comparing the claim, group membership, or property with the comparison value you specified, and causes the condition to evaluate to TRUE or FALSE depending on the outcome of that operation.

The following comparison operators are available:

- **equals** The condition evaluates to **True** if the comparison value evaluates to the exact value of the claim or property; otherwise, the condition evaluates to **False**.
- **does not equal** The condition evaluates to **False** if the comparison value evaluates to the exact value of the claim or property; otherwise, the condition evaluates to **True**.
- **member of any** The condition evaluates to **True** if the comparison value lists any of the groups the user (or computer) is a member of. If the user (or computer) is not a member of any of the groups listed in the comparison value, the condition evaluates to **False**.
- **member of each** The condition evaluates to **True** if the comparison value lists only the groups the user (or computer) is a member of. If the user (or computer) is not a member of each group listed in the comparison value, the condition evaluates to **False**.
- **not member of any** The condition evaluates to **False** if the comparison value lists any of the groups the user (or computer) is a member of. If the user (or computer) is not a member of any of the groups listed in the comparison value, the condition evaluates to **True**.
- **not member of each** The condition evaluates to **False** if the comparison value lists only the groups the user (or computer) is a member of. If the user (or computer) is not a member of each group listed in the comparison value, the condition evaluates to **True**.

You can choose from the following options to specify a comparison value:

- **Device claim** The comparison value is the value of a certain computer claim. You can select one of the existing computer claim types from the claim type list provided by the condition builder.
- **Target object property** The comparison value is the value of a certain property of the object to which the authorizing user requests access. You can select the desired property from a list provided by the condition builder.
- **User claim** The comparison value is the value of a certain user claim. You can select one of the existing user claim types from the claim type list provided by the condition builder.
- **Value** Depending on what the condition is intended to evaluate, this option allows you to specify a particular text string, integer, Boolean value (True or False), or a list of groups. In case of a claim type that provides a list of suggested values, the condition builder prompts you to select a value from the list.

Applying an Access Rule

You apply Access Rules to Access Template links. A single Access Rule or no Access Rule can be applied to a given link. By default, no Access Rule is applied, which configures an unconditional link. Applying an Access Rule creates a conditional link that has an effect only if the Access Rule's conditional expression evaluates to **True** during permission check.

To apply an Access Rule, the Active Roles console provides the **Access Rule** tab in the **Properties** dialog box for an Access Template link. You can display a list of Access Template links in a number of ways:

- Right-click a container and then click **Delegate Control**. This displays a list of all Access Template links applied to that container or inherited from a higher-level container.
- Right-click a user or group and then click **Delegated Rights**. This displays a list of all Access Template links applied to that user or group or inherited from another security group.
- Right-click an Access Template and then click **Links**. This displays a list of all Access Template links referring to that Access Template.

In the list, double-click a link to open the **Properties** dialog box. The **Access Rule** tab in that dialog box includes the following items:

- **Access Rule** This field identifies the Access Rule that is currently applied to the Access Template link. If no Access Rule is applied, this field is empty; otherwise, the field displays the name of the Access Rule along with the path to the Access Rule object in the **Configuration/Access Rules** container.
- **Change** Click this button to select the Access Rule you want to apply to the link.
- **Properties** Click this button to view or change the Access Rule properties, including the Access Rule's conditional expression.
- **Clear** Click this button if you want to remove the Access Rule from the Access Template link.
- To see if a given link has an Access Rule applied, refer to the **Access Rule** field in the list of Access Template links.

Steps for managing and applying Access Rules

Access Rules allow you to deploy and manage authorization policies that include conditional expressions involving user claims, device claims, and object properties. Claims are assertions about the attributes of the user or device. When authorizing access to a given object, Active Roles can use Access Rules to evaluate the claims of the user or device requesting access along with the properties of that object, and enable the appropriate Access Template links depending upon the evaluation results.

Access Rules are held in the **Access Rules** container under the **Configuration** node in the Active Roles console tree. You can use the Active Roles console to:

- Create or modify an Access Rule
- Configure a conditional expression for an Access Rule
- Apply an Access Rule to Access Template links

Create or modify an Access Rule

You can create a new Access Rule in the **Configuration | Access Rules** container, or modify an existing Access Rule in that container.

To create a new Access Rule

1. Right-click the **Access Rules** container, and select **New | Access Rule**.
2. On the **General** page, type a name and description for the new Access Rule.
3. Click **Next** to proceed to the **Conditions** page.
4. Configure a conditional expression and then click **Finish**.

To modify an existing Access Rule

1. Right-click the Access Rule you want to modify, and then click **Properties**.
2. On the **General** page, view or change the name and description of the Access Rule.
3. On the **Conditions** page, view or change the conditional expression.

Configure a conditional expression for an Access Rule

The **Conditions** page provides an editor for configuring a conditional expression. When you configure an expression, you need to add at least one condition. Initially, you add a condition to the default condition group. You can create additional condition groups to group a set of conditions and nest the grouped conditions within other condition groups.

A condition group contains one or more conditions connected by the same logical operator. By grouping conditions, you specify that those conditions should be evaluated as a single unit. The effect is the same as if you put parentheses around an expression in a mathematical equation or logic statement.

To add a condition to a condition group

- Click the name of the condition group and then click **Insert condition**.

  OR

- Click the plus sign (+) next to the name of the condition group.

You can remove a condition, if needed, by clicking the **Delete condition** button labeled **X** on the right side of the list item representing the condition in the condition builder.

To add a condition group into another condition group

- Click the name of the condition group, point to **Insert condition group**, and then click an option to specify the logical operator:
  - **AND group** The condition group evaluates to TRUE if all conditions in the group are TRUE.
  - **OR group** The condition group evaluates to TRUE if any condition in the group is TRUE.

By default, AND is the logical operator between the conditions in a condition group. It is possible to change the logical operator by converting the condition group to a different group type: Click the name of the group, point to **Convert condition group to**, and then click the option appropriate to the desired logical operator.

You can remove an entire condition group, if needed, by clicking the name of the group and then clicking **Delete condition group**.

Once you have added a condition to a condition group, you can use the following steps to configure the condition.

To configure a condition

1. Click **Configure condition to evaluate**, and then choose from the following options to specify what you want the condition to evaluate:
    - Click **Device claim** to evaluate a computer claim, or groups the computer account is a member of. Then, in the claim type list, select the desired claim type, or click **Group** if you want the condition to evaluate the group membership of the computer account.
    - Click **Target object property** to evaluate a certain property of the object to which the authorizing user requests access. Then, in the property list, select the desired property.
    - Click **User claim** to evaluate a user claim, or groups the user account is a member of. Then, in the claim type list, select the desired claim type, or click **Group** if you want the condition to evaluate the group membership of the user account.
2. Click in the middle field of the condition line to choose the comparison operator you want.
3. Click **Define value to compare to**, and then choose from the following options to specify the desired comparison value:
    - Click **Device claim** to perform comparison with a computer claim. Then, in the claim type list, select the desired claim type.
    - Click **Target object property** to perform comparison with the value of a certain property of the object to which the authorizing user requests access. Then, in the property list, select the desired property.
    - Click **User claim** to perform comparison with a user claim. Then, in the claim type list, select the desired claim type.
    - Click **Value** to perform comparison with a particular text string, integer, Boolean value, or a list of groups. Then, supply the desired value. The value you can supply depends upon the type of data the condition is intended to evaluate. For example, when configuring a condition to evaluate group memberships, then you have to supply a list of groups as a comparison value. If the claim type you have selected to evaluate provides a list of suggested values, then you can only select a comparison value from that list.

When you configure a condition, consider the following:

- Only single-value claim types and object properties are supported. The multi-value claim types and object properties are filtered out from the lists provided by the condition builder.
- To perform comparison, a valid condition requires that values on either side of the comparison operator be of the same or compatible data type. Therefore, when you supply a comparison value, the condition builder restricts you to the options that match the data type of the claim or property you choose to evaluate. If you choose to evaluate a string-value, integer-value, or Boolean claim type or object property, then the comparison value must be a string, integer, or Boolean value, respectively.
- If you choose to evaluate the group membership of a user or device, the comparison value must be a list of groups. Other options are unavailable in this case.

Apply an Access Rule to Access Template links

Access Rules are intended to be applied to Access Template links. A single Access Rule or no Access Rule can be applied to a given link. By default, no Access Rule is applied, which configures an unconditional link. By applying an Access Rule, you create a conditional link that has an effect only if the Access Rule's conditional expression evaluates to True during permission check.

To apply an Access Rule

1. In a list of Access Template links, double-click the Access Template link to which you want to apply the Access Rule.

   You can select Access Template links from various lists provided be the Active Roles console. Thus, you can use the **Delegate Control** command on a container object to display a list of all Access Template links that determine the permission settings for that container. The **Links** command on an Access Template displays all links of that Access Template. The **Active Roles Security** tab in the advanced details pane lists the Access Template links that determine the security settings for the object selected in the Active Roles console.

2. In the **Properties** dialog box that appears, click the **Access Rule** tab.
3. Click the **Change** button, and then select the Access Rule you want to apply.

From the **Access Rule** tab, you can also perform the following tasks:

- Choose a different Access Rule for the selected Access Template link. Click the **Change** button and choose the Access Rule you want.
- View or change the Access Rule applied to the selected Access Template link. Click the **Properties** button and then go to the **Conditions** page to review or modify the Access Rule's conditional expression.
- Remove the Access Rule from the selected Access Template link. Click the **Clear** button to remove the Access Rule.

Deploying an Access Rule (demonstration steps)

This section demonstrates how to implement a security scenario where each delegated administrator is restricted to managing users from a single department. The scenario is implemented by using an Access Rule that enables a delegated administrator to access only those objects whose Department property is identical with the Department claim of that delegated administrator.

Step 1. Prerequisites

In this section, we assume that you already have the following prerequisites:

- An Active Directory domain, with at least one domain controller running Windows Server 2012 (or a later version of the Windows Server operating system).
- The Active Roles Administration Service and MMC Interface (console) of the latest version installed on a member server in your Active Directory domain, with the server running Windows Server 2012 (or a later version of the Windows Server operating system).
- Your Active Directory domain is registered with Active Roles as a managed domain.

Step 2. Enable claim support

Configure Group Policy to enable domain controllers to issue claims:

1. On a domain controller running Windows Server 2012 or later, open the Group Policy Management console.

   To open the console, press **Windows logo key+R** to open the **Run** dialog box, type **gpmc.msc**, and click **OK**.

2. In the console tree, select the **Domain Controllers** OU under your domain.
3. In the details pane, right-click **Default Domain Controllers Policy**, and then click **Edit**.
4. Perform the following steps in the Group Policy Management Editor console that appears:
   a. In the console tree, select **Computer Configuration | Policies | Administrative Templates | System | KDC**.
   b. In the details pane, double-click **KDC support for claims, compound authentication and Kerberos armoring**.
   c. In the **KDC support for claims, compound authentication and Kerberos armoring** dialog box, click **Enabled** and select **Supported** from the **Options** drop-down list. When finished, click **OK** to close the dialog box.

5. Close Group Policy Management Editor.
6. Close Group Policy Management.
7. Open a command prompt and enter the following command: `gpupdate /force`.

Configure Group Policy to enable the Active Roles Administration Service to retrieve claims for clients by using Kerberos protocol transition:

1. On the server running the Active Roles Administration Service, open the Local Group Policy Editor console.

   To open the console, press **Windows logo key+R** to open the **Run** dialog box, type **gpedit.msc**, and click **OK**.

2. In the console tree, select **Computer Configuration | Administrative Templates | System | Kerberos**.
3. In the details pane, double-click **Kerberos client support for claims, compound authentication and Kerberos armoring**.
4. In the **Kerberos client support for claims, compound authentication and Kerberos armoring** dialog box, click **Enabled**, and then click **OK**.
5. Restart the computer to apply the new setting to the Active Roles Administration Service. (Restarting only the Administration Service may not suffice.)

Add the Service Principal Names (SPNs) of the Active Roles Administration Service to the service account, to enable support for Kerberos authentication. Enter the following commands at a command prompt, where `<FQDN>` stands for the fully qualified domain name of the computer running the Administration Service; `<name>` stands for the name of that computer; and `<ServiceAccountName>` stands for the name of the service account (domain user account under which the Administration Service runs):

1. `setspn -s aradminsvc/<FQDN> <ServiceAccountName>`

   For example, `setspn -s aradminsvc/arsrv.domain.com domain\arsvcacct`

2. `setspn -s aradminsvc/<name> <ServiceAccountName>`

   For example, `setspn -s aradminsvc/arsrv domain\arsvcacct`

Step 3. Create Claim Type

Create a Claim Type object for your domain controller to issue user claims sourced from the Department attribute. Log on as an Active Roles Admin and perform the following steps in the Active Roles console. (Assuming the default configuration, you should log on with a domain user account that is a member of the Administrators local group of the member server running the Active Roles Administration Service.)

1. In the console tree, expand the **Active Directory** node, right-click the **Claim Types** container, and select **New | Claim Type**.
2. On the **Source Attribute** page, scroll down the list of attributes, and click **Department**.
3. Click **Next** and then click **Finish**.

Step 4. Create Access Rule

Use the Active Roles console to create an Access Rule object with a conditional expression that evaluates to TRUE if the **Department** claim of the authorizing user evaluates exactly to the **Department** property of the target object:

1. In the console tree, expand the **Configuration** node, right-click the **Access Rules** container, and select **New | Access Rule**.
2. On the **General** page, type **Department Admins** in the **Name** field, and then click **Next**.
3. On the **Conditions** page, configure the conditional expression:
   a. Click the **AND group** item, and then click **Insert condition**.
   b. Click **Configure condition to evaluate**, and then click **User claim**.
   c. On the **Select Claim Type** page that appears, click **Department** in the list of claim types, and then click **OK**.
   d. Verify that the comparison operator reads **equals** (this is the default setting).
   e. Click **Define value to compare to**, and then click **Target object property**.
   f. On the **Select Target Object Property** page that appears, select the **Department** property, and then click **OK**.
4. Click **Finish**.

Step 5. Apply Access Rule

To apply the Access Rule you created in Step 4, you first need to delegate control by using an Access Template, and then attach the Access Rule to the Access Template link. Create a security group to hold your delegated administrators, and perform the following steps in the Active Roles console:

1. In the console tree, under the **Active Directory** node, right-click the name of your domain, and then click **Delegate Control**.
2. On the **Active Roles Security** page that appears, click **Add** to start the Delegation of Control wizard.
3. Follow the wizard pages:
   a. On the **Users or Groups** page, click **Add**, and select the security group that holds your delegated administrators. Click **Next**.
   b. On the **Access Templates** page, expand the **Active Directory** node, and select the **OUs - Read All Properties** and **Users - Modify All Properties** check boxes. Click **Next**.
   c. On the remaining pages, click **Next** to accept the default settings.
   d. On the completion page, click **Finish**.

You will apply the Access Rule to the **Users - Modify All Properties** Access Template link. The **OUs - Read All Properties** Access Template enables the delegated administrators to browse the domain for user objects.

4. Click **OK** to close the **Active Roles Security** page. This will create the Access Template links.
5. Right-click the name of your Active Directory domain and click **Active Roles Security** to open the **Active Roles Security** page again.
6. On the **Active Roles Security** page, select the **Users - Modify All Properties** Access Template link and then click **View/Edit**.
7. On the **Access Rule** tab in dialog box that appears, click the **Change** button, select the **Department Admins** Access Rule, click **OK** to close the **Select an Access Rule** page, and then click **OK** to close the dialog box.
8. Click **OK** to close the **Active Roles Security** page.

After you have completed these steps, Active Roles allows a delegated administrator to make changes to only those user accounts that have the same department setting as the delegated administrator's account.

6

Rule-based AutoProvisioning and Deprovisioning

- About Policy Objects
- Policy Object management tasks
- Policy configuration tasks
- Deployment considerations
- Checking for policy compliance
- Deprovisioning users or groups
- Restoring deprovisioned users or groups
- Container Deletion Prevention policy
- Picture management rules
- Policy extensions

About Policy Objects

Active Directory (AD) supports delegating control with fine granularity. However, simply restricting control, access and permissions may not always be a sufficient or effective way of managing the resources of an organization.

Many directory administration processes (such as creating or disabling user accounts, enforcing user name conventions, resetting passwords, and so on) are based on predefined workflows that often share the same procedures. In practice, this means that administrators have to repeatedly perform configuration tasks with similar steps.

To make the management of such administrative tasks easier, Active Roles provides a policy-based administration solution to automate and speed up repeat procedures when administering on-premises, hybrid and Azure cloud-only objects. This approach is represented with **Policy Objects**, available in the **Configuration** > **Policies** > **Administration** node of the Active Roles MMC console.

NOTE: Policy Object settings specific to Azure cloud-only objects (such as cloud-only Azure users, guest users, or contacts) are available only if your Active Roles deployment is licensed for managing cloud-only Azure objects. Contact One Identity support for more information.

Also, Policy Objects specific to Azure cloud-only objects will work correctly only if an Azure tenant is already configured in the AD of the organization, and Active Roles is already set as a consented Azure application for that Azure tenant. For more information on these settings, see Configuring a new Azure tenant and consenting Active Roles as an Azure application.

Summary of Policy Objects

Each configured Policy Object contains one or more policies, defining either the behavior of the Active Roles system, or the actions that Active Roles performs when certain directory objects are created, modified, or deleted. This way, Active Roles can automate the administrative workflow within the organization.

Policy Objects specify what AD objects to change, how, when, whenever they are created, modified, or deleted. You can also configure policies to have Active Roles accept certain data changes only if they conform to the formatting requirements specified by the policy. This helps maintain control over the data stored in AD, and also keeps network objects in a consistent state with each defined policy.

To offer additional flexibility for configuring policies, Active Roles Policy Objects can also run customizable scripts before or after running a task.

Example: Use case for setting up a policy

A typical use case for an Active Roles policy is to automate the administration of a new employee. When creating a user account for a new employee, you can create a policy that makes Active Roles automatically perform all of the following steps:

1. Retrieve information from the HR database of the organization.
2. Use the retrieved information as the default data for filling user account properties, such as name, contact information, and so on.
3. Create a home folder and home share for the new user account.
4. Add the user account to all relevant groups within the organization.
5. Create an Exchange mailbox for the user account, and add the mailbox to the relevant distribution lists.

With one or more properly configured Policy Objects, this entire procedure can be performed either automatically, or with minimal manual administrator work. Without policies, it would require time-consuming manual administrative actions each time a new user is administered.

NOTE: Active Roles does not automatically check for changes in directory objects, containers or groups specified for provisioning in the configured Policy Objects. This

means that if any changes are made in any directory resources in use in a policy, you must update the impacted policies manually. For example, if a directory group used by a Group Membership AutoProvisioning Policy Group is deleted, the Policy Group must be updated manually to reflect the changes.

Advantages of using Policy Objects

Configuring Policy Objects has the following advantages:

- They reduce the workload and the time needed to perform common administration duties by automating tasks, combining multiple tasks into a single workflow, or even eliminating certain tasks altogether.
- They offer automated (or largely simplified) workflows for provisioning, reprovisioning and deprovisioning directory objects in the organization.
- They improve network security.
- They ensure the consistency of the managed AD objects across the organization.
- They minimize administration errors.

Types of Policy Objects

To help you configure, organize and apply Policy Objects, they are in two main categories in the Active Roles MMC console:

- Provisioning Policy Objects: Use provisioning policy objects to specify provisioning rules, such as:
  - Populating and validating directory data.
  - Creating account resources (such as home folders and mailboxes).
  - Administering access to resources within the organization.
- Deprovisioning Policy Objects: Use deprovisioning policy objects to specify rules upon requests to deprovision a selected user or group. Deprovisioning rules may include:
  - Removing user accounts or email addresses.
  - Revoking group and distribution list memberships.
  - Disabling security permissions and application access rights.

Both categories can contain multiple Policy Objects.

Built-in Policy Objects

To help you get started with configuring policy-based administration in your organization, Active Roles includes a set of built-in Policy Objects that offer provisioning and deprovisioning rules to the most typical administrative use cases. To find the built-in Policy Objects, navigate to the following node of the Active Roles MMC console:

Configuration > **Policies** > **Administration** > **Builtin**

To help you configure Script Execution policies, Active Roles also ships with several built-in **Script Modules** that you can use to set up your own **Script Execution** policies. Find these built-in **Script Modules** in the following node of the Active Roles MMC console:

Configuration > **Script Modules** > **Builtin**

Provisioning Policy Objects

To configure provisioning policies for user name and email generation, group memberships, property generation or script running, use the policies available via the Provisioning Policy Objects.

NOTE: Policy Object settings specific to Azure cloud-only objects (such as cloud-only Azure users, guest users, or contacts) are available only if your Active Roles deployment is licensed for managing cloud-only Azure objects. Contact One Identity support for more information.

Also, Policy Objects specific to Azure cloud-only objects will work correctly only if an Azure tenant is already configured in the AD of the organization, and Active Roles is already set as a consented Azure application for that Azure tenant. For more information on these settings, see Configuring a new Azure tenant and consenting Active Roles as an Azure application.

Table 13: Provisioning Policy Objects

Policy	Description
User Logon Name Generation	Generates a user login name (pre-Windows 2000) for a newly-created user account. Use this policy to: • Add a uniqueness number to the generated logon name. • Apply multiple rules to generate a logon name. • Allow a logon name to be specified manually when creating a new user. TIP: Combine these options to ensure the uniqueness of the user logon name (pre-Windows 2000), which is a schema requirement in Active Directory (AD). For more information on how to set up this policy, see Steps for configuring a User Logon Name Generation policy
Email Alias Generation	Sets up the appropriate email aliases for newly-created user accounts. Use this policy to generate aliases based on: • Pre-selected user properties, such as the first and last names. • A custom selection of properties, not limited to user properties. TIP: Use this policy to make each alias unique by adding a unique-

Policy	Description
	ness number to the alias. For more information on how to set up this policy, see Steps for configuring an E-mail Alias Generation policy
Exchange Mailbox AutoProvisioning	Creates user mailboxes in the appropriate mailbox stores or databases. Use this policy to: • Specify the mailbox stores or databases in which mailboxes can be created. • Apply a rule to distribute mailboxes among multiple stores or databases. TIP: Configure this policy to distribute mailboxes either with the round-robin method, or by selecting a store or database with the least number of mailboxes. For more information on how to set up this policy, see Steps for configuring an E-mail Alias Generation policy
Group Membership AutoProvisioning	Ensures that directory objects (such as users) are assigned to (or unassigned from) the appropriate group(s) if the specified policy criteria are met. TIP: Use this policy to have Active Roles automatically add or remove objects (such as users or guest users) to or from certain groups if the configured group membership rules are met. NOTE: Consider the following when configuring a Group Membership AutoProvisioning Policy: • In case of cloud-only Azure objects, you can use the Group Membership Autoprovisioning policy to automatically assign (or unassign) Azure users and Azure guest users to (or from) the specified O365 group(s) in the same Azure tenant. • Active Roles does not automatically check for changes in directory objects, containers or groups specified for provisioning in the configured Policy Objects. This means that if any changes are made in any directory resources in use in a policy, you must update the impacted policies manually. For example, if a directory group used by a Group Membership AutoProvisioning Policy Group is deleted, the Policy Group must be updated manually to reflect the changes. For more information on how to set up this policy, see Steps for configuring a Group Membership AutoProvisioning policy.
Home Folder AutoProvisioning	Performs provisioning actions to assign home folders and home shares to user accounts. Use this policy to:

Policy	Description
	• Create home folders for newly-created user accounts. • Rename home folders upon renaming user accounts. TIP: Use this policy to specify the server on which to create home folders and shares, determine their naming conventions, and configure their access rights as well. For more information on how to set up this policy, see Steps for configuring a Home Folder AutoProvisioning policy
Property Generation and Validation	Generates and validates directory data, such as user properties. Use this policy to: • Populate a directory with the default data that the organization requires. • Validate the existing data upon checking directory updates. TIP: Consider the following when planning to configure a Property Generation and Validation policy: • To help you get started with configuring policy-based administration in your organization, Active Roles includes a set of built-in Policy Objects that offer provisioning and deprovisioning rules to the most typical administrative use cases. To find the built-in Policy Objects, navigate to the following node of the Active Roles MMC console: **Configuration** > **Policies** > **Administration** > **Builtin** • If the directory of your organization contains cloud-only Azure objects (Azure users, guest users or contacts), then use the built-in **Azure CloudOnly Policy - Default Rules to Generate Properties** Policy Object to provision their default properties and accepted values. For more information on how to set up this policy, see Steps for configuring a Property Generation and Validation policy.
Script Execution	Runs the specified PowerShell (or other custom) script on request to perform certain operations, such as creating a user account or updating its properties. Use this policy to: • Trigger additional actions to perform directory object provisioning. • Regulate object data format and requirements. • Further automate administrative tasks. When linking a custom script to an administrative operation via a **Script Execution** policy, the script will receive control in Active Roles either when the operation is requested or when it is completed.

Policy	Description
	TIP: Consider the following when planning to use custom scripts for your provisioning policies: • To help you configure Script Execution policies, Active Roles also ships with several built-in **Script Modules** that you can use to set up your own **Script Execution** policies. Find these built-in **Script Modules** in the following node of the Active Roles MMC console: **Configuration** > **Script Modules** > **Builtin** • If the directory of your organization contains any cloud-only Azure users, then use the built-in **Generate User Password - Azure only** script module to set up a password generation policy for cloud-only Azure users that meets the password strength criteria of both your organization and Microsoft Azure Active Directory (AD). For more information on how to set up a Script Execution policy, see Steps for configuring a Script Execution policy
Office 365 and Azure Tenant Selection	Enables configuring multiple assignments to Azure objects. Use this policy to: • Validate the selected Azure tenants for Azure users, guest users, O365 Groups and contacts. • Select O365 Licenses for Azure users and guest users. • Select O365 Roles for Azure users and guest users. • Preprovision OneDrive for Azure users. For more information on how to set up this policy, see Configuring an O365 and Azure Tenant Selection policy.
AutoProvisioning for SaaS products	Automates user and group provisioning in the selected SaaS products using Starling Connect connectors. You can specify the Starling Connect connectors to be validated for the users or groups for which the policy is then applied. For more information on how to set up this policy, see Create Provisioning policy for Starling Connect

Deprovisioning Policy Objects

Deprovisioning Policy Objects allows configuration and application of the following policies.

Table 14: Deprovisioning Policy Objects

Policy	Description
User Account Deprovisioning	When deprovisioning a user, this policy modifies the user account so that the user cannot log on. You can configure this policy to: • Disable the user account. • Set the user's password to a random value. • Set the user's logon names to random values. • Rename the user account. You can also select account properties and configure this policy to update them when processing a deprovisioning request.
Group Membership Removal	When deprovisioning a user, this policy removes the user account from groups. You can configure this policy to remove the account from security groups, mail-enabled groups, or both. In this policy, both distribution groups and mail-enabled security groups are collectively referred to as mail-enabled groups. You can also select the groups from which you do not want this policy to remove the user account, or configure the policy not to remove the user account from any security groups or mail-enabled groups.
User Account Relocation	When deprovisioning a user, this policy moves the user account to a different location. You can select the organizational unit to which you want the policy to move the account. You can also configure the policy not to move the user accounts upon user deprovisioning.
Exchange Mailbox Deprovisioning	When deprovisioning a user, this policy makes changes needed to deprovision Microsoft Exchange resources for that user. You can configure this policy to: • Hide the mailbox from the global address list (GAL). • Prevent non-delivery reports (NDR) from being sent. • Grant the user's manager full access to the user's mailbox. • Grant selected users or groups full access to the user's mailbox. • Disallow forwarding messages to alternate recipients. • Forward all incoming messages to the user's manager.
Home Folder Deprovisioning	When deprovisioning a user, this policy makes changes needed to prevent the user from accessing his or her home folder. You can configure this policy to: • Remove the user's permissions on the home folder.

Policy	Description
	• Grant the user's manager read-only access to the user's home folder. • Grant selected users or groups read-only access to the user's home folder. • Make a selected user or group the owner of the user's home folder. • Delete the home folder when the user account is deleted.
User Account Permanent Deletion	When deprovisioning a user, this policy schedules the user account for deletion. You can specify the number of days (retention period) before the account is deleted. Another option is to delete the deprovisioned user accounts immediately to Active Directory Recycle Bin. It is also possible to configure this policy so that the deprovisioned user accounts are not deleted automatically.
Group Object Deprovisioning	When deprovisioning a group, this policy makes changes to the group object in Active Directory in order to prevent the use of the group. You can configure this policy to: • Hide the group from the Global Address List (GAL). • Change the group type from Security to Distribution. • Rename the group. • Remove members from the group. • Change or clear any other properties of the group object.
Group Object Relocation	When deprovisioning a group, this policy moves the group object to a different container in Active Directory. You can select the organizational unit to which you want the policy to move the group object.
Group Object Permanent Deletion	When deprovisioning a group, this policy schedules the group object for deletion in Active Directory. You can specify the number of days (retention period) before the group is deleted. Another option is to delete the deprovisioned groups immediately to Active Directory Recycle Bin. It is also possible to configure this policy so that the deprovisioned groups are not deleted automatically.
Notification Distribution	In the course of a deprovisioning operation, this policy sends a notification message to the e-mail recipients you specify. You can customize both the message subject and message body.
Report Distribution	Upon completion of a deprovisioning operation, this policy sends a report to the e-mail recipients you specify. The report includes a list of actions taken during the deprovisioning operation and the details of the deprovisioning activity. You can customize the

Policy	Description
	subject of the e-mail message containing the report. You can also configure this policy to send the report only if any errors occurred in the course of a deprovisioning operation.
Script Execution	In the course of a deprovisioning operation, this policy runs the script you specify. By using a script, you can implement custom deprovisioning actions.
Office 365 Licenses Retention	When deprovisioning an Azure AD user, this policy automates retention of all or selected Office 365 licenses assigned to the Azure AD user after the Azure AD user is deprovisioned successfully.

How Policy Objects work

A Policy Object is a collection of administrative policies that specifies the business rules to be enforced. A Policy Object includes stored policy procedures and specifications of events that activate each procedure.

A Policy Object associates specific events with its policy procedures, which can be built-in procedures or custom scripts. This provides an easy way to define policy constraints, implement sophisticated validation criteria, synchronize different data sources, and perform a number of administrative tasks as a single batch.

Active Roles enforces business rules by linking Policy Objects to:

- Administrative views (Active Roles Managed Units)
- Active Directory containers (Organizational Units)
- Individual (leaf) directory objects, such as user or group objects

By choosing where to link a Policy Object, you determine the policy scope. For example, if you link a Policy Object to a container, all objects in the container and its sub-containers are normally subject to the Policy Object.

You can link different Policy Objects to different containers to establish container-specific policies. You may need to do so if each organizational unit uses a dedicated Exchange server to store mailboxes or file server to store home folders.

You can also link a Policy Object to a leaf object, such as a user object. As an example, consider a policy that prohibits changes to group memberships when copying a certain user object.

Policy Objects define the behavior of the system when directory objects are created, modified, moved, or deleted within the policy scope. Policies are enforced regardless of administrative rights of a user performing a management task. It is important to understand that even those who have administrator rights to Active Roles itself are forced to abide by administrative policies once they are enforced.

Policy Object management tasks

This section guides you through the Active Roles console to manage Policy Objects. The following topics are covered:

- Creating a Policy Object
- Adding, modifying, or removing policies
- Applying Policy Objects
- Managing policy scope
- Copying a Policy Object
- Renaming a Policy Object
- Exporting and importing Policy Objects
- Deleting a Policy Object

Creating a Policy Object

The Active Roles console provides separate wizards for creating Policy Objects in each category—provisioning and deprovisioning. You can start the wizards from the **Administration** container, located under **Configuration/Policies** in the console tree:

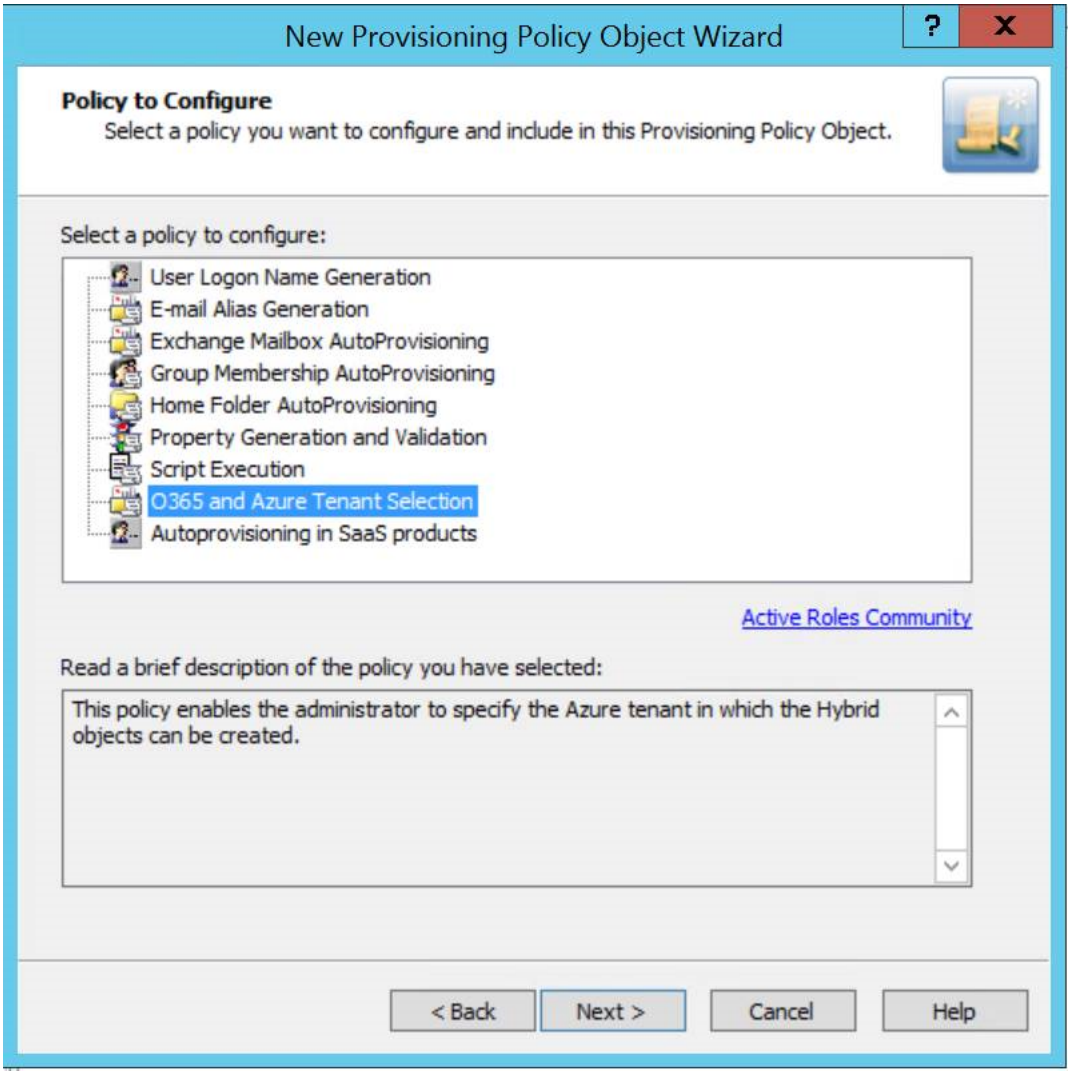
- To configure provisioning policies, right-click **Administration** in the console tree, and select **New | Provisioning Policy**.
- To configure deprovisioning policies, right-click **Administration** in the console tree, and select **New | Deprovisioning Policy**.

If you need to manage a large number of Policy Objects, it is advisable to create containers that hold only specified Policy Objects for easy location: In the console tree, right-click **Administration** and select **New | Container**. Then, you can use wizards to create Policy Objects in that container: Right-click the container and select **New | Provisioning Policy** or **New | Deprovisioning Policy**.

On the **Welcome** page of the wizard, click **Next**. Then, on the **Name and Description** page, type a name and description for the new Policy Object. The Active Roles console will display the name and description in the list of Policy Objects in the details pane.

Click **Next** to continue. This displays a page where you can select the policy you want to configure. The list of policies depends on whether you are creating a Provisioning Policy Object or Deprovisioning Policy Object. For instance, the list of provisioning policies looks as shown in the following figure.

Figure 24: Provisioning policies



On the **Policy to Configure** page, select the type of policy you want to add to the Policy Object. When the type is selected, its description is displayed in the lower box.

Click **Next** to configure the policy. The steps involved in configuring a policy depend on the policy type. For instructions on how to configure policies, see Policy configuration tasks later in this chapter.

When you are done with configuring a policy, the wizard presents you with a page where you can specify the policy scope. You have the option to complete a list of containers or Managed Units on which you want the policy to be enforced. This step is optional because you can configure the policy scope after creating the Policy Object (see Applying Policy Objects later in this chapter).

Click **Next**, and then click **Finish** to complete the wizard. This creates the new Policy Object.

Steps for creating a Policy Object

To create a Policy Object

1. In the console tree, under **Configuration | Policies | Administration**, locate and select the folder in which you want to add the Policy Object.

   You can create a new folder as follows: Right-click **Administration** and select **New | Container**. Similarly, you can create a sub-folder in a folder: Right-click the folder and select **New | Container**.

2. Right-click the folder, point to **New**, and then click **Provisioning Policy** or **Deprovisioning Policy**.
3. On the Welcome page of the wizard, click **Next**.
4. On the **Name and Description** page, do the following:
   a. In the **Name** box, type a name for the Policy Object.
   b. Under **Description**, type any optional information about the Policy Object.

   Click **Next**.
5. On the **Policy to Configure** page, select a policy type, and click **Next** to configure policy settings.
6. On the **Enforce Policy** page, you can specify the objects to which this Policy Object is to be applied:
   - Click **Add**, and use the **Select Objects** to locate and select the objects you want.
7. Click **Next**, and then click **Finish**.

NOTE:

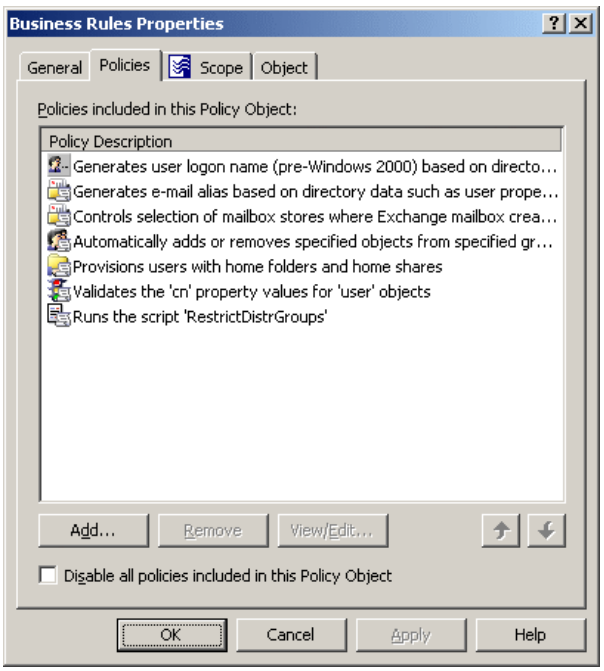
- For information about available policy types, see Provisioning Policy Objects and Deprovisioning Policy Objects earlier in this document.
- For information on how to configure policies, see Policy configuration tasks later in this document.
- To add more policies to the new Policy Object, display the **Properties** dialog box, and click **Add** on the **Policies** tab.

Adding, modifying, or removing policies

Although the New Policy Object wizard makes it possible to configure only one policy, a Policy Object may include multiple policies. You can add policies, remove policies, and modify policy options in an existing Policy Object by managing its properties: Right-click the Policy Object and then click **Properties**.

To add, remove, or edit policies in a Policy Object, go to the **Policies** tab in the **Properties** dialog box. The tab is shown in the following figure.

Figure 25: Policy Objects Management



The **Policies** tab displays a list of policies defined in the Policy Object. Each list entry includes an icon denoting policy type and policy description. The policies are executed in the order shown in the list. To change the order, use the arrows in the lower-right corner of the tab.

On the **Policies** tab, you can perform the following management tasks:

- **Add policy**. Click the **Add** button and follow the instructions in the wizard, which depend on whether you are configuring a Provisioning Policy Object or Deprovisioning Policy Object.

  The wizard prompts you to select the type of policy to add and then guides you through the steps to configure the policy. The steps to configure a policy depend on the policy type. For instructions on how to configure policies, see Policy configuration tasks later in this chapter.

- **Delete policy**. Select policies from the list and click the **Remove** button. This permanently deletes the policies you have selected.

- **Modify policy**. Select a policy from the list and click the **View/Edit** button. This displays the **Properties** dialog box for the policy you have selected.

  The **Properties**. dialog includes several tabs, with each tab containing the same options as the corresponding page of the wizard used to configure the policy. You can manage policy options the same way as you do when initially configuring the policy.

- **Disable all policies**. For troubleshooting purposes, you may need to stop enforcement of the policies without actually deleting them. To accomplish this, select the **Disable all policies included in this policy object** check box.

NOTE: The policies that can be added to a given Policy Object depend on the type of the Policy Object. A Provisioning Policy Object can only include provisioning-related policies whereas a Deprovisioning Policy Object can only include deprovisioning-related policies (see Provisioning Policy Objects and Deprovisioning Policy Objects earlier in this document).

Steps for adding policies to a Policy Object

To add a policy to a Policy Object

1. In the console tree, under **Configuration | Policies | Administration**, locate and select the folder that contains the Policy Object you want to modify.
2. In the details pane, right-click the Policy Object, and then click **Properties**.
3. On the **Policies** tab, click **Add** to start a wizard that helps you configure a policy.
4. On the Welcome page of the wizard, click **Next**.
5. On the **Policy to Configure** page, select the type of the policy you want to add.
6. Configure policy settings. For instructions, see Policy configuration tasks.

NOTE:

- The **Policies** tab lists the policies that are configured in the Policy Object. You can use the **Policies** tab to add, modify, or delete policies from the Policy Object.
- Active Roles processes policies in the order they are listed on the **Policies** tab. To change the order, select a policy and click ↑ or ↓ to move the policy up or down in the list.
- Once a Policy Object is applied within Active Roles to determine policy settings in the directory, any changes to the list of policies in the Policy Object causes the policy settings in the directory to change accordingly.

Steps for modifying policies in a Policy Object

To view or modify a policy in a Policy Object

1. In the console tree, under **Configuration | Policies | Administration**, locate and select the folder that contains the Policy Object you want to examine.
2. In the details pane, right-click the Policy Object, and then click **Properties**.
3. On the **Policies** tab, select the policy you want to view or modify, and click **View/Edit**.
4. Use the tabs in the **Policy Properties** dialog box to view or modify policy settings.

The tabs in the **Policy Properties** dialog box provide the same options as the wizard for configuring the policy. See Policy configuration tasks for information about the options specific to each type of policy.

NOTE:

- The **Policies** tab lists the policies that are configured in the Policy Object. You can use the **Policies** tab to add, modify, or delete policies from the Policy Object.
- Active Roles processes policies in the order they are listed on the **Policies** tab. To change the order, select a policy and click ↑ or ↓ to move the policy up or down in the list.

Steps for removing policies from a Policy Object

To delete a policy from a Policy Object

1. In the console tree, under **Configuration | Policies | Administration**, locate and select the folder that contains the Policy Object you want to modify.
2. In the details pane, right-click the Policy Object, and then click **Properties**.
3. On the **Policies** tab, select the policy you want to delete, click **Remove**, and then click **Yes** to confirm the deletion.

NOTE:

- The **Policies** tab lists the policies that are configured in the Policy Object. You can use the **Policies** tab to add, modify, or delete policies from the Policy Object.
- Once a Policy Object is applied within Active Roles to determine policy settings in the directory, any changes to the list of policies in the Policy Object causes the policy settings in the directory to change accordingly.

Applying Policy Objects

Implementing a policy to enforce business rules is a two-phase process where configuring the policy within a Policy Object is only the first step. When you create a new policy, you select a policy type from the available options and then define the options that make up the policy. The second step is to use the Active Roles console to enforce the policy on the desired areas of the directory.

Active Roles allows policies to be enforced on any directory object—an administrative view (Managed Unit), a directory folder (container), or an individual (leaf) object. Policies are enforced by applying (linking) a Policy Object that holds the policies.

When you apply a Policy Object to a Managed Unit or directory folder, the policies control the objects in that Unit or folder as well as the Unit or folder itself. When you apply a Policy

Object to a leaf object, such as a user or group, the policies only control that object. For example, applying a Policy Object to a group does not affect the members of the group.

The objects that are subject to a given Policy Object, that is, the objects under control of the policies defined in that Policy Object, are collectively referred to as policy scope. For example, if you apply a Policy Object to a Managed Unit, the policy scope is composed of the objects within the Managed Unit.

Thus, the policy scope normally includes all objects that reside in a container or Managed Unit to which the Policy Object is applied. However, sometimes you may need to exclude individual objects or sub-containers from the policy scope, thereby preventing certain objects from being affected by policies.

Active Roles gives you the option to selectively exclude objects or entire containers from the policy scope. You can block policy inheritance on individual objects or containers, refining the policy scope. The option to block policy inheritance is discussed later in this section (see Managing policy scope).

To apply a Policy Object, you can start from any of the following points:

- **Policy Object**. Add Managed Units or containers to the policy scope of the Policy Object.
- **Directory object**. Add the Policy Object to the policy list for the directory object.

The following two sections elaborate on each of these options.

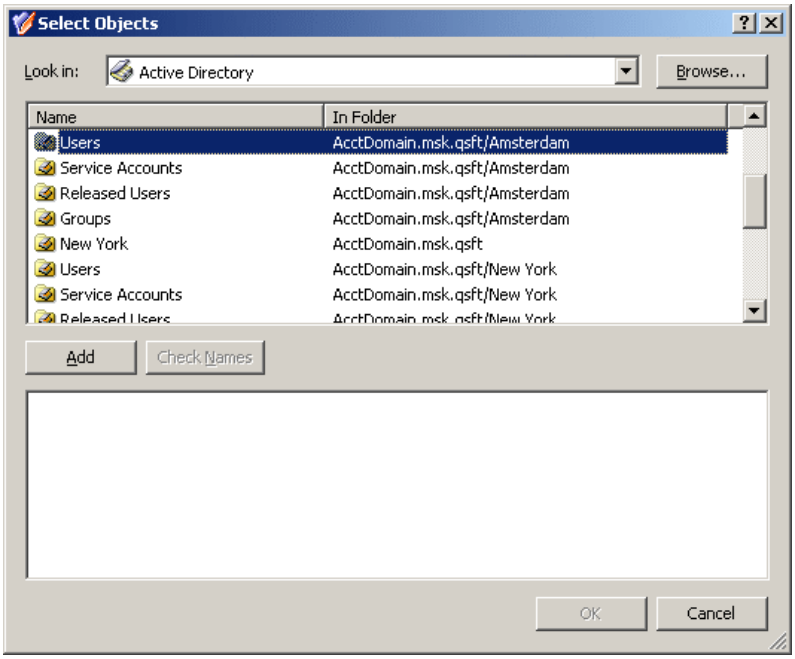
Adding Managed Units or containers to policy scope

You can add administrative views (Managed Units) and directory folders (containers) to the policy scope of a given Policy Object in one of these ways:

- Right-click the Policy Object and click **Policy Scope**. Then, in the **Active Roles Policy Scope** window, click the **Add** button.
- Ensure that **Advanced Details Pane** is checked on the **View** menu. Then, select the Policy Object. On the **Active Roles Policy Scope** tab in the details pane, right-click a blank area and click **Add**.

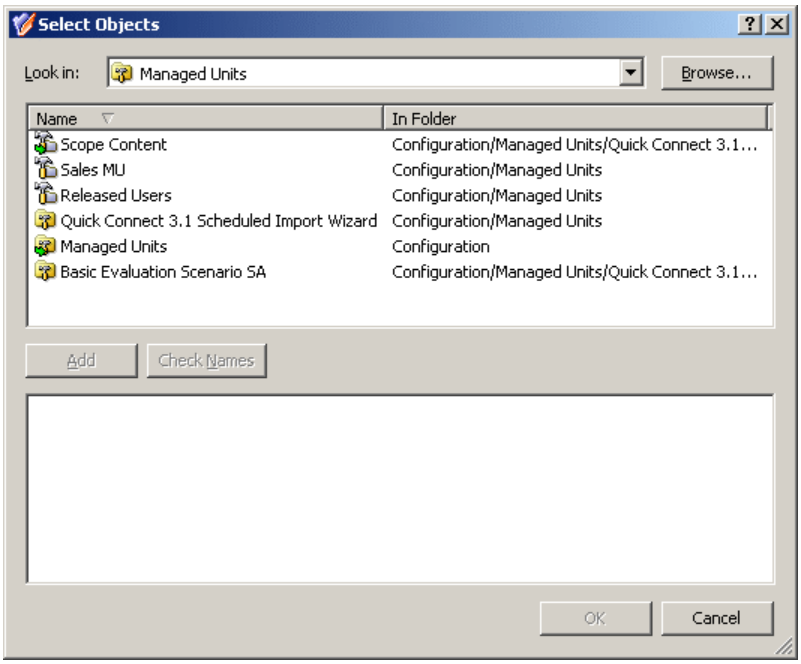
In both cases, clicking **Add** displays the **Select Objects** window where you can select containers and Managed Units. To build a list of containers from which to select, click the **Browse** button and select **Active Directory** or a container in the hierarchy under **Active Directory**. The list is shown in the following figure.

Figure 26: Policy Objects



To build a list of Managed Units, click the **Browse** button and select **Managed Units** or a container in the hierarchy under **Managed Units**. The list looks like the following figure:

Figure 27: Managed Units



In the **Select Objects** window, select containers or Managed Units from the list and click the **Add** button to build the resultant list of items. When finished, click **OK**.

Adding Policy Objects to policy list for directory object

For a given directory object (container, user, group, and so on), a list of Policy Objects that affect the directory object is referred to as policy list. If the directory object is in the policy scope of a given Policy Object, the Policy Object is included in the policy list for that directory object.

The steps to add a Policy Object to the policy list for a directory object depend on whether it is a container or leaf object:

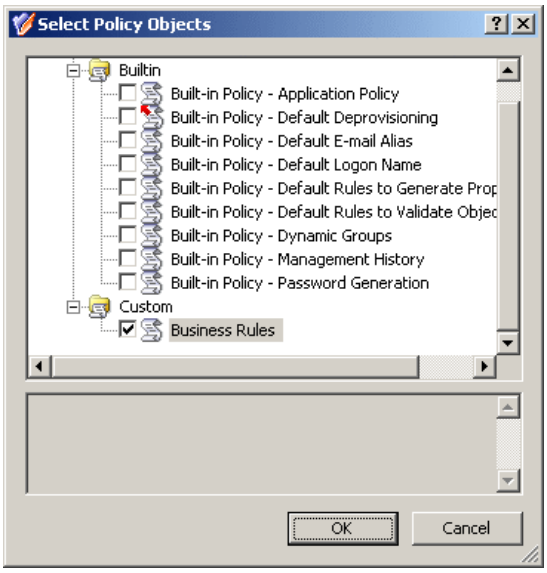
- Right-click a Managed Unit or container and click **Enforce Policy**. Then, in the **Active Roles Policy** window, click the **Add** button.
- Right-click a leaf object (user, group, or the like), click **Properties**, go to the **Administration** tab, and click the **Policy** button. Then, in the **Active Roles Policy** window, click the **Add** button.

If you use the advanced details pane (**Advanced Details Pane** is checked on the **View** menu), you can do this as follows, regardless of the type of the directory object:

- Select the directory object, go to the **Active Roles Policy** tab in the details pane, right-click a blank area on the tab, and then click **Add**.

In all these cases, clicking **Add** displays the **Select Policy Objects** window where you can select Policy Objects to add. Select check boxes next to names of Policy Objects, as shown in the following figure, and then click **OK**.

Figure 28: Policy Objects



Steps for applying a Policy Object

To apply a Policy Object

1. In the console tree, under **Configuration | Policies | Administration**, locate and select the folder that contains the Policy Object you want to apply.
2. In the details pane, right-click the Policy Object, and then click **Policy Scope**.
3. In the **Active Roles Policy Scope** dialog box, click **Add**.
4. Use the **Select Objects** dialog box to locate and select the container, Managed Unit, or a leaf object on which you want to specify policy settings by using the Policy Object.
5. Click **OK** to close the **Active Roles Policy Scope** dialog box.

To specify policy settings on an object by using a Policy Object

1. Open the **Active Roles Policy** dialog box for the object:
   - Right-click the object, and click **Enforce Policy**.

   -OR-

   - Right-click the object, and click **Properties**. Then, on the **Administration** tab in the **Properties** dialog box, click **Policy**.
2. In the **Active Roles Policy** dialog box, click **Add**.
3. Use the **Select Policy Objects** dialog box to locate and select the Policy Object to apply.
4. To select a Policy Object, click the check box next to the name of the Policy Object. You can select multiple Policy Objects.
5. Click **OK** to close the **Active Roles Policy** dialog box.

> TIP: To apply a Policy Object, you can also use the **Active Roles Policy Scope** or **Active Roles Policy** tab in the advanced details pane: Right-click a blank area on the tab, and then click **Add**. To display the advanced details pane, check **Advanced Details Pane** on the **View** menu (see Advanced pane earlier in this document).

To view or modify inheritance options for a Policy Object on a container or Managed Unit

1. Open the **Active Roles Policy Scope** dialog box for the Policy Object: Right-click the Policy Object, and then click **Policy Scope**.
2. In the **Active Roles Policy Scope** dialog box, select the container or Managed Unit to which the Policy Object is applied and on which you want to examine inheritance options, and then click **View/Edit**.
3. On the **General** tab, view or modify the selection of these options, which specifies the scope where the Policy Object determines policy settings:
   - **This directory object**. The scope includes the container or Managed Unit you have selected (this option does not cause the scope to include any child objects

or members of the container or Managed Unit).

- **Child objects of this directory object**. The scope includes all the child objects (or members, as applied to a Managed Unit) in the entire hierarchy under the container or Managed Unit you have selected.
- **Immediate child objects only**. The scope includes only the child objects (or members, as applied to a Managed Unit) of which the container or Managed Unit that you have selected is the direct ancestor.

Managing policy scope

When applying a Policy Object to a directory object, Active Roles creates a Policy Object link. Thus, policies are put in force by linking Policy Objects to directory objects—Managed Units, directory folders (containers), or individual (leaf) objects.

Each Policy Object link includes the following information:

- Policy Object that defines the policies
- Directory object that is the target of the link
- Flag — **Include** or **Exclude** — that specifies whether the directory object is included or excluded from the policy scope

You can display a list of Policy Object links starting from one of the following points:

- **Policy Object**. Right-click a Policy Object and click **Policy Scope**.

  This displays the links in which the Policy Object occurs.
- **Directory object**. First, open a window that lists the Policy Objects that affect this directory object:
  - For a container object or Managed Unit, right-click the object or Unit and click **Enforce Policy**.
  - For a leaf object, right-click the object, click **Properties**, go to the **Administration** tab, and click **Policy**.

  Next, in the window that opens, click the **Advanced** button.

  This displays the links in which the directory object occurs as the target object.

Another way to see a list of Policy Object links is the use of the advanced details pane. Ensure that **Advanced Details Pane** is checked on the **View** menu, and then do one of the following:

- Select a Policy Object.

  The **Active Roles Policy Scope** tab lists the links in which the Policy Object occurs.
- Select a directory object (Managed Unit, container, or leaf object), right-click a blank area on the **Active Roles Policy** tab, and click **Advanced View**.

  This displays the links in which the directory object occurs as the target object.

When you display a list of Policy Object links for a directory object, the list appears in a separate window. Each entry in the list includes the following information:

- **Policy Object** Name of the Policy Object.
- **Directory Object** Canonical name of the object to which the Policy Object is linked, that is, the target object of the link.
- **Include/Exclude** Flag that determines the behavior of the link:
  - **Include Explicitly**. means the link puts the target object within the policy scope, that is, the policies defined in the Policy Object control the target object.
  - **Exclude Explicitly**. means the link puts the target object out of the policy scope, that is, the policies defined in the Policy Object do not control the target object.

The **Exclude** flag takes precedence over the **Include** flag. If there are two links with the same Policy Object, one of which is flagged **Include** while another one is flagged **Exclude**, the object is effectively excluded from the policy scope of the Policy Object.

The list of Policy Object links displays the links of these categories:

- **Direct links**. Policy Object is applied (linked) directly to the object you have selected.
- **Inherited links**. Policy Object is applied (linked) to a container in the hierarchy of containers above the object you have selected, or to a Managed Unit to which the selected object belongs.

The links inherited from parent objects can be filtered out of the list. To do this, clear the **Show inherited** check box.

To manage links, you can use the buttons beneath the list:

- **Add**. Displays the dialog box where you can select Policy Objects, creating the links to the Policy Objects you select.
- **Remove**. Deletes the selected entries from the list of links. Available for direct links only.
- **View/Edit**. Displays the dialog box to view or modify link properties, such as whether the link affects the child objects of the link target object. Available for only those links that are flagged **Include**.
- **Exclude**. Shows up for links flagged **Include**. Available on direct links only. Changes the flag to **Exclude**.
- **Include**. Shows up for links flagged **Exclude**. Available on direct links only. Changes the flag to **Include**.

> TIP: The **Remove** button is only available on direct links. When you need to delete links, it is advisable to manage them using the **Policy Scope** command on the Policy Object.

To simplify the management of policy effect on directory objects, the Active Roles console allows you to manage policy scope without directly managing links to Policy Objects. For a directory object, you can view and modify its policy list—a list of Policy Objects that control (affect) the directory object—instead of having to sort through direct and inherited links.

Given a directory object, you can display its policy list as follows:

- For a container or a Managed Unit, right-click it and click **Enforce Policy**.
- For a leaf object (user, group, or suchlike), right-click it, click **Properties**, go to the **Administration** tab, and click **Policy**.

Each entry in the policy list includes the following information:

- **Policy Object**. The name of the Policy Object. The Policy Object controls this directory object due to a direct link or inherited links.
- **Block Inheritance**. Indicates whether policy effect is blocked on this directory object. If the **Blocked** check box is selected, the Policy Object link flagged **Exclude** is created for this directory object.

You can manage the policy list using the buttons beneath the list:

- **Add**. Displays the dialog box where you can select Policy Objects, putting the directory object under the control of the Policy Objects you select.
- **Remove**. If you select a Policy Object from the policy list and click **Remove**, the direct link of the Policy Object to this object is deleted.

  If the Policy Object is in the list due to an inherited link, the **Remove** button is unavailable. Moreover, if there are both the direct link and an inherited link to the Policy Object, clicking **Remove** deletes the direct link but *does not* remove the Policy Object from the policy list. In this case, the Policy Object remains in the list because the policies are still applied due to inheritance.

  If you need to remove the directory object from the policy scope of a given Policy Object, select the **Blocked** check box in the **Block Inheritance** column. This adds the Policy Object link flagged **Exclude** for the directory object.
- **View/Edit**. Displays the **Properties** dialog box for the Policy Object you select from the list. You can use the **Properties** dialog box to manage policies in the Policy Object and gain access to the list of all links where this Policy Object occurs.
- **Advanced**. Opens the window with the list of Policy Object links for this directory object, discussed earlier in this section.

You can also access the policy list from the advanced details pane. The list is displayed on the **Active Roles Policy** tab when you select a directory object.

On the **Active Roles Policy** tab, you can perform the same management tasks as in the **Active Roles Policy** window: Right-click a list entry or a blank area and use commands on the shortcut menu. The commands act in the same way as the buttons in the **Active Roles Policy** window.

Given a Policy Object, you can also manage its policy scope by using a list of directory objects to which the Policy Object is applied (linked). The list can be displayed in a separate window or on a tab in the advanced details pane:

- To display the list in a window, right-click the Policy Object and click **Policy Scope**.
- To display the list on a tab, ensure that **Advanced Details Pane** is checked on the **View** menu and select the Policy Object.

The list displays all links of the Policy Object. Each entry in the list includes the following information:

- **Name**. Canonical name of the directory object to which the Policy Object is linked, that is, the target object of the link.
- **Include/Exclude**. Flag that determines the behavior of the link:
  - **Include Explicitly**. means the link puts the target object within the policy scope, that is, the policies defined in the Policy Object control the target object.
  - **Exclude Explicitly**. means the link puts the target object out of the policy scope, that is, the policies defined in the Policy Object do not control the target object.

  The **Exclude** flag takes precedence over the **Include** flag. If there are two links with the same target object, one of which is flagged **Include** while another one is flagged **Exclude**, the target object is effectively excluded from the policy scope of the Policy Object.

To manage the list in the **Active Roles Policy Scope** window, you can use the buttons beneath the list: **Add**, **Remove**, **View/Edit**, **Include**, or **Exclude**. The buttons perform basically the same functions as those described earlier in this section. To manage the list in the **Active Roles Policy Scope** tab, you can use the command on the shortcut menu: Right-click a link or a blank area to access the menu. The menu includes the following commands:

- **Add**. Appears when you right-click a blank area. Performs the same action as the **Add** button. Opens the **Select Objects** dialog box where you can select containers or Managed Units to which you want to link the Policy Object (see Applying Policy Objects).
- **Delete**. Appears when you right-click a link. Performs the same action as the **Remove** button. Deletes the link you select from the list.
- **Exclude**. Appears when you right-click a link flagged **Include**. Performs the same action as the **Exclude** button. Changes the flag on the link you select.
- **Include**. Appears when you right-click a link flagged **Exclude**. Performs the same action as the **Include** button. Changes the flag on the link you select.
- **Refresh**. Updates the list with the current information.

Steps for managing Policy Object links

When you apply a Policy Object (see Applying Policy Objects earlier in this document), Active Roles creates an object, referred to as a Policy Object link, that stores information about the Policy Object and about the directory object on which the Policy Object is applied. Basically, the management of policy settings in Active Roles comes to the management of Policy Objects and Policy Object links. This topic provides some instructions you can use to view or modify Policy Object links.

To view or modify Policy Object links in which a given Policy Object occurs

1. Right-click the Policy Object, and click **Policy Scope**.
2. In the **Active Roles Policy Scope** dialog box, do the following:
   - To create a new link, click **Add**, and then use the **Select Objects** dialog box to locate and select the object to which you want to link the Policy Object.
   - To delete a link, select it from the list and click **Remove**.
   - To view or modify the properties of a link, such as the inheritance options, select the link from the list and click **View/Edit**. (For information about inheritance options, see Steps for applying a Policy Object earlier in this document.)
   - To specify whether a link removes or puts the effect of the Policy Object on the object to which the Policy Object is linked, select the link and click **Exclude** or **Include**, respectively.

To view or modify a list of the Policy Objects on a given object

1. Open the **Active Roles Policy** dialog box for the object:
   - Right-click the object, and click **Enforce Policy**.

     -OR-
   - Right-click the object, and click **Properties**. Then, on the **Administration** tab in the **Properties** dialog box, click **Policy**.

   The **Active Roles Policy** dialog box for a given object lists all the Policy Objects that determine the policy settings on that object. Use the following instructions to modify the list, if necessary.
2. In the **Active Roles Policy** dialog box, do the following:
   - To define additional policy settings on the object, click **Add**, and then select one or more Policy Objects that determine the policy settings.
   - To remove the effect of a certain Policy Object on the object you are administering, select the **Blocked** check box next to the name of the Policy Object. Clear the check box if you want the Policy Object to have an effect on the object.
   - To delete a Policy Object link on the object, select the Policy Object and click **Remove**. (This operation can be performed if the Policy Object is linked to the object itself rather than to a container or Managed Unit that holds the object.)
   - To view or modify policies in a Policy Object, select the Policy Object and click **View/Edit**. (For further instructions, see Steps for modifying policies in a Policy Object earlier in this document.)
   - To display a list of the Policy Object links that determine the policy settings on the object, click **Advanced**. Use the following instructions to administer the list, if necessary.

To view or modify Policy Object links that determine the policy settings on a given object

1. In the **Active Roles Policy** dialog box, click **Advanced**.
2. In the **Active Roles Policy - Advanced View** dialog box, do the following:
   - To create a new link, click **Add**, and then select the Policy Object you want.
   - To delete a link, select it from the list and click **Remove**. (This operation can be performed if the Policy Object is linked to the object itself rather than to a container or Managed Unit that holds the object.)
   - To view or modify the properties of a link, such as the inheritance options, select the link from the list and click **View/Edit**.
   - To specify whether a link removes or puts the effect of the Policy Object on the object you are administering, select the link and click **Exclude** or **Include**, respectively.

NOTE:

- By default, the **Active Roles Policy - Advanced View** dialog box for an object lists all the links that determine the policy settings on the object, regardless of whether a link was created on the object itself or on a container or Managed Unit that holds the object. To change the display of the list, clear the **Show inherited** check box.
- Clicking **View/Edit** in the **Active Roles Policy - Advanced View** or **Active Roles Policy Scope** dialog box displays the **Properties** dialog box for the selected link. From the **Properties** dialog box, you can access the properties of both the directory object and Policy Object that are covered by the link, and view or modify the inheritance options for the link (see Steps for applying a Policy Object earlier in this document).
- You can also manage Policy Object links on the **Active Roles Policy Scope** or Active Roles Policy tab in the advanced details pane, which allows you to perform the same tasks as the **Active Roles Policy Scope** or **Active Roles Policy** dialog box, respectively. Right-click a link or a blank area on the tab, and use command on the shortcut menu. The **Active Roles Policy Scope** tab is displayed when you select a Policy Object. Otherwise, the **Active Roles Policy** tab is displayed. To display the advanced details pane, check **Advanced Details Pane** on the **View** menu (see Advanced pane earlier in this document).

Steps for excluding an object from policy scope

The objects on which a given Policy Object has effect are collectively referred to as the policy scope of the Policy Object. When applying a Policy Object, you add objects to the policy scope. You can use the following instructions to exclude certain objects from the policy scope of a Policy Object, in order to remove the effect of the Policy Object on those objects.

To exclude an object from the policy scope of a Policy Object

1. Open the **Active Roles Policy** dialog box for the object:
   - Right-click the object, and click **Enforce Policy**.

     -OR-
   - Right-click the object, and click **Properties**. Then, on the **Administration** tab in the **Properties** dialog box, click **Policy**.
2. In the **Active Roles Policy** dialog box, select the **Blocked** check box next to the name of the Policy Object.
3. Click **OK** to close the **Active Roles Policy** dialog box.

NOTE:

- You can restore the effect of the Policy Object on the object that was excluded from the policy scope: In the **Active Roles Policy** dialog box for that object, clear the **Blocked** check box next to the name of the Policy Object.
- Excluding an object from the policy scope creates a Policy Object link on that object, the link being flagged Exclude Explicitly. Restoring the effect of the Policy Object causes that link to be removed. For instructions on how to manage Policy Object links, see Steps for managing Policy Object links earlier in this document.

Copying a Policy Object

With the Active Roles console, you can create copies of Policy Objects. This feature helps you re-use existing Policy Objects.

To create a copy of a Policy Object, right-click the Policy Object, and click **Copy**. This opens the Copy Object wizard. Type a name and description for the copy, and then click **Next**.

On the next page, the wizard displays a list of policies. The list includes all policies defined in the original Policy Object. Click **Finish** to create the copy.

The copy has the same properties as the original Policy Object, including the policies and their configurations. You can make changes to the copy using the **Properties** dialog box, as described earlier in this chapter (see Adding, modifying, or removing policies).

Steps for copying a Policy Object

To copy a Policy Object

1. In the console tree, under **Configuration | Policies | Administration**, locate and select the folder that contains the Policy Object you want to copy.

2. In the details pane, right-click the Policy Object, and then click **Copy** to start the Copy Object - Policy Object wizard.
3. On the first page of the wizard, do the following:
   a. In the **Name** box, type a name for the new Policy Object.
   b. In the **Description** box, type any optional information about the new Policy Object.

   Click **Next**.
4. Click **Finish** to complete the creation of the new Policy Object.

NOTE: The copy of a Policy Object contains the same policies as the original Policy Object. You can view or modify policies by using the **Properties** dialog box for the newly created Policy Object. To have the console display the **Properties** dialog box, select **Display the object properties when this wizard closes** on the completion page of the Copy Object - Policy Object wizard. For instructions on how to add, modify, and remove policies from a Policy Object, see Adding, modifying, or removing policies earlier in this document.

Renaming a Policy Object

To rename a Policy Object, right-click the Policy Object, and click **Rename**. Type the new name, and then press **ENTER**. Renaming a Policy Object does not affect its links. This is because Policy Objects are referenced by immutable identifier rather than by name.

Steps for renaming a Policy Object

To rename a Policy Object

1. In the console tree, under **Configuration | Policies | Administration**, locate and select the folder that contains the Policy Object you want to rename.
2. In the details pane, right-click the Policy Object, and click **Rename**.
3. Type a new name, and then press ENTER.

NOTE: If a Policy Object is applied within Active Roles to determine policy settings in the directory, renaming the Policy Object does not cause any changes to the policy settings in the directory. When applying a Policy Object, Active Roles refers to the Policy Object by an internal identifier rather than by the name of the Policy Object.

Exporting and importing Policy Objects

With the Active Roles console, you can export Policy Objects to an XML file and then import them from that file to populate another instance of Active Roles. The export and import operations provide a way to move Policy Objects from a test environment to a production environment.

> NOTE: When you export and then import Policy Objects, only policies are transferred. The Policy Object links are not included in the export-import operation. You need to reconfigure them manually after completing the operation.

To export Policy Objects, select them, right-click the selection, and select **All Tasks | Export**. In the **Export Objects** dialog box, specify the file where you want to save the data, and click **Save**.

To import Policy Objects, right-click the container where you want to place the Policy Objects, and then click **Import**. In the **Import Directory Objects** dialog box, select the file to which the Policy Objects were exported, and click **Open**.

Deleting a Policy Object

To delete a Policy Object, you must first delete the links to the Policy Object (see Managing policy scope earlier in this chapter). Then, you can perform the deletion: Right-click the Policy Object and click **Delete**.

Steps for deleting a Policy Object

To delete a Policy Object

1. In the console tree, under **Configuration | Policies | Administration**, locate and select the folder that contains the Policy Object you want to delete.
2. In the details pane, right-click the Policy Object, and then click **Delete**.

> NOTE: Once a Policy Object is applied within Active Roles to determine policy settings in the directory, the Policy Object cannot be deleted. You can view a list of objects to which the Policy Object is applied: Right-click the Policy Object, and click **Policy Scope**. If you need to delete the Policy Object, first remove all items from the list in the **Active Roles Policy Scope** dialog box.

Policy configuration tasks

This section discusses how to configure policies of the following types, grouped by Policy Object category.

Table 15: Policy Configuration Tasks

Policy Object category	Policy type
Provisioning Policy Object	Property Generation and Validation
	User Logon Name Generation
	Group Membership AutoProvisioning
	Email Alias Generation
	Exchange Mailbox AutoProvisioning
	Home Folder AutoProvisioning
	Script Execution
	Office 365 and Azure Tenant Selection
	AutoProvisioning for SaaS products
	Office 365 Licenses Retention
Deprovisioning Policy Object	User Account Deprovisioning
	Group Membership Removal
	Exchange Mailbox Deprovisioning
	Home Folder Deprovisioning
	User Account Relocation
	User Account Permanent Deletion
	Group Object Deprovisioning
	Group Object Relocation
	Group Object Permanent Deletion
	Notification Distribution
	Report Distribution
	Script Execution

Property Generation and Validation

Property Generation and Validation policies help you automate the configuration of directory object properties. Using this policy, you can:

- Automatically generate default property values for new directory objects (for example, when creating new user accounts or groups).
- Automatically check if the configured property values comply with the specified corporate policy rules.

To set up a policy, you can specify conditions that the property values must meet, and can also determine the default value for each property provisioned with the policy. For example, you can configure a policy to enforce a certain type of telephone number formatting in the contact information properties for your directory.

TIP: Consider the following when planning to configure a Property Generation and Validation policy:

- To help you get started with configuring policy-based administration in your organization, Active Roles includes a set of built-in Policy Objects that offer provisioning and deprovisioning rules to the most typical administrative use cases. To find the built-in Policy Objects, navigate to the following node of the Active Roles MMC console:

  Configuration > **Policies** > **Administration** > **Builtin**
- If the directory of your organization contains cloud-only Azure objects (Azure users, guest users or contacts), then use the built-in **Azure CloudOnly Policy - Default Rules to Generate Properties** Policy Object to provision their default properties and accepted values.

NOTE: Policy Object settings specific to Azure cloud-only objects (such as cloud-only Azure users, guest users, or contacts) are available only if your Active Roles deployment is licensed for managing cloud-only Azure objects. Contact One Identity support for more information.

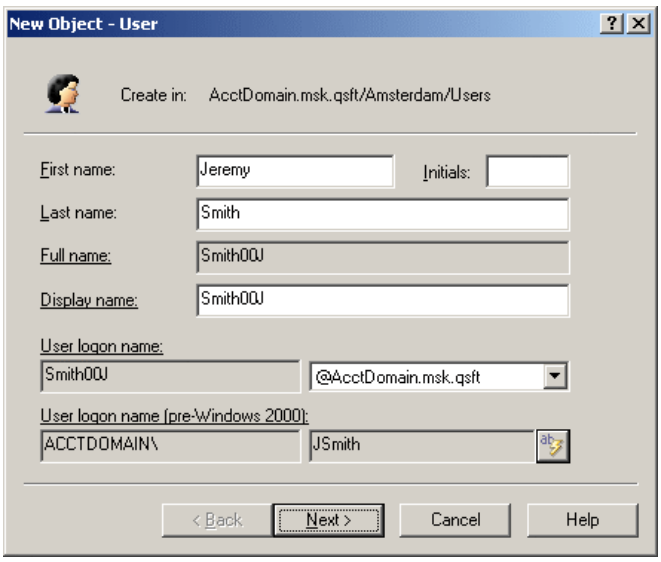
Also, Policy Objects specific to Azure cloud-only objects will work correctly only if an Azure tenant is already configured in the AD of the organization, and Active Roles is already set as a consented Azure application for that Azure tenant. For more information on these settings, see Configuring a new Azure tenant and consenting Active Roles as an Azure application.

How this policy works

When creating or modifying an object, Active Roles checks whether the property values satisfy criteria defined in the policy. If they do not, Active Roles prevents you from the object creation or modification.

In object creation wizards and properties dialog boxes, the properties that are controlled by the policy are displayed as hyperlinks. If you have a policy configured to populate a property with a certain value (generate the default value), the edit box for the property is unavailable for editing, as shown in the following figure.

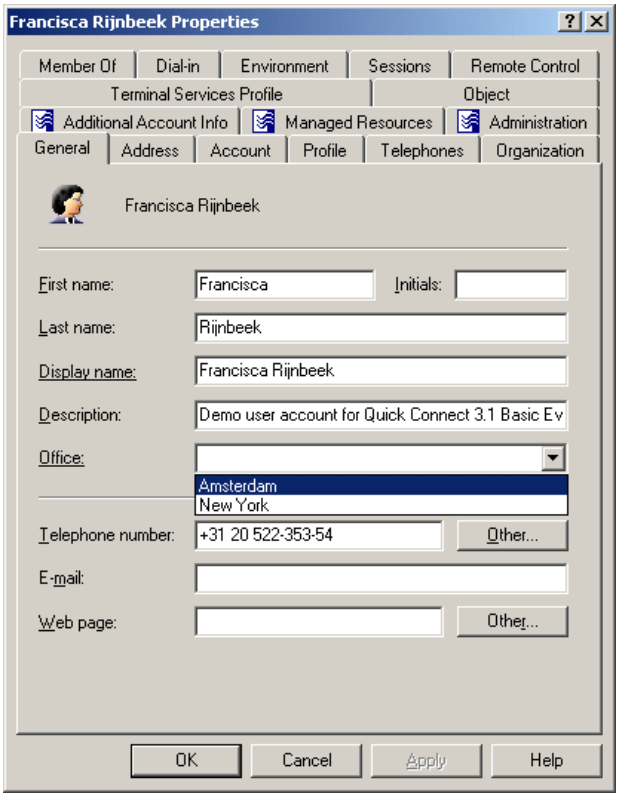
Figure 29: Object creation



You can click a hyperlink to display the policy details.

With a policy configured to define a set of acceptable values for a given property, the Active Roles console provides a drop-down list to select a value when modifying that property. The user of the Active Roles console can choose an acceptable value from the list instead of having to type a value in the edit box. This feature is illustrated in the following figure: The **Office** box provides a list of acceptable values that are prescribed by policy.

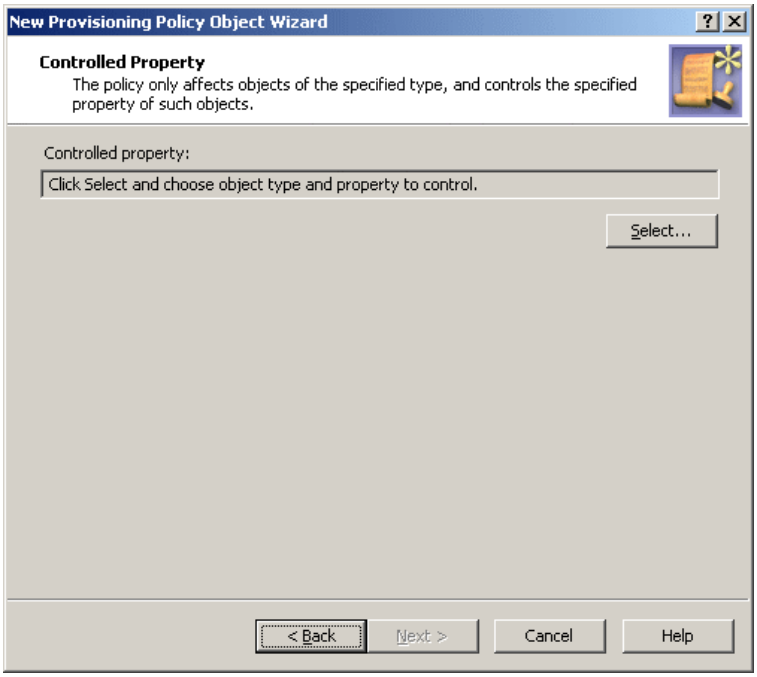
Figure 30: Acceptable values for a policy



How to configure a Property Generation and Validation policy

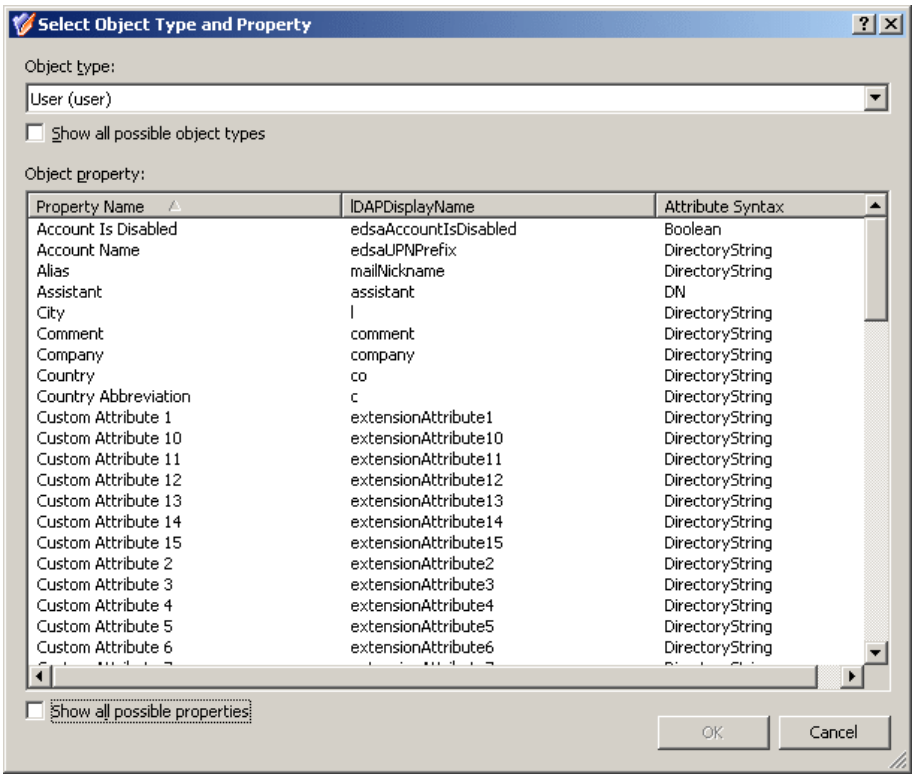
To configure a Property Generation and Validation policy, select **Property Generation and Validation** on the **Policy to Configure** page in the New Provisioning Policy Object wizard or in the Add Provisioning Policy wizard. Then, click **Next** to display the **Controlled Property** page:

Figure 31: New provisioning policy object wizard



Click **Select** to choose the object type and object property you want the policy to control. This displays the **Select Object Type and Property** dialog box.

Figure 32: Select Object type and Property

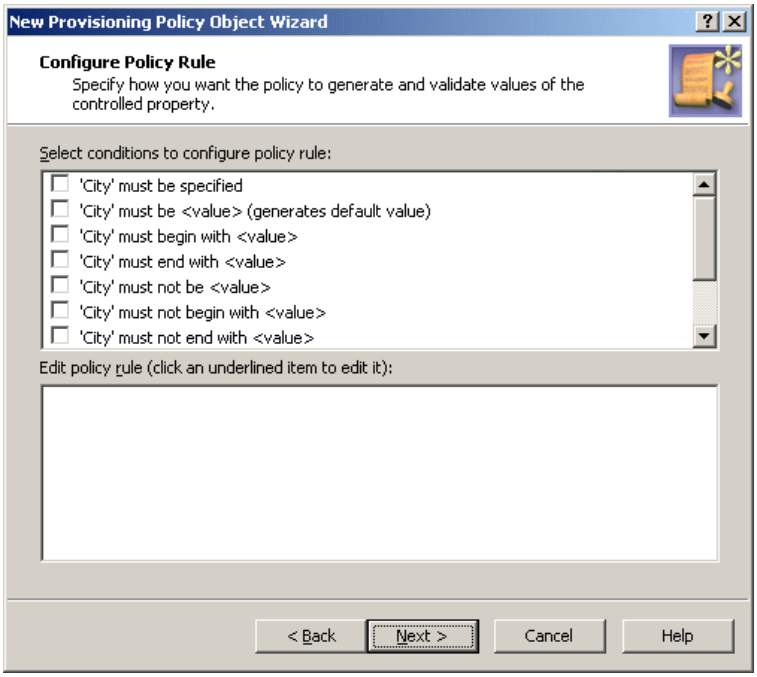


Property Name	lDAPDisplayName	Attribute Syntax
Account Is Disabled	edsaAccountIsDisabled	Boolean
Account Name	edsaUPNPrefix	DirectoryString
Alias	mailNickname	DirectoryString
Assistant	assistant	DN
City	l	DirectoryString
Comment	comment	DirectoryString
Company	company	DirectoryString
Country	co	DirectoryString
Country Abbreviation	c	DirectoryString
Custom Attribute 1	extensionAttribute1	DirectoryString
Custom Attribute 10	extensionAttribute10	DirectoryString
Custom Attribute 11	extensionAttribute11	DirectoryString
Custom Attribute 12	extensionAttribute12	DirectoryString
Custom Attribute 13	extensionAttribute13	DirectoryString
Custom Attribute 14	extensionAttribute14	DirectoryString
Custom Attribute 15	extensionAttribute15	DirectoryString
Custom Attribute 2	extensionAttribute2	DirectoryString
Custom Attribute 3	extensionAttribute3	DirectoryString
Custom Attribute 4	extensionAttribute4	DirectoryString
Custom Attribute 5	extensionAttribute5	DirectoryString
Custom Attribute 6	extensionAttribute6	DirectoryString

*From the **Object type** list, select the object type. This lets you to specify the type of objects that are subject to the policy control. From the **Object property** list, select the object property. This lets you to specify the property you want the policy to control. Click **OK**.*

On the **Controlled Property** page, click **Next** to display the **Configure Policy Rule** page:

Figure 33: Configure Policy Rule



This page allows you to determine criteria used to generate and validate values of the controlled property.

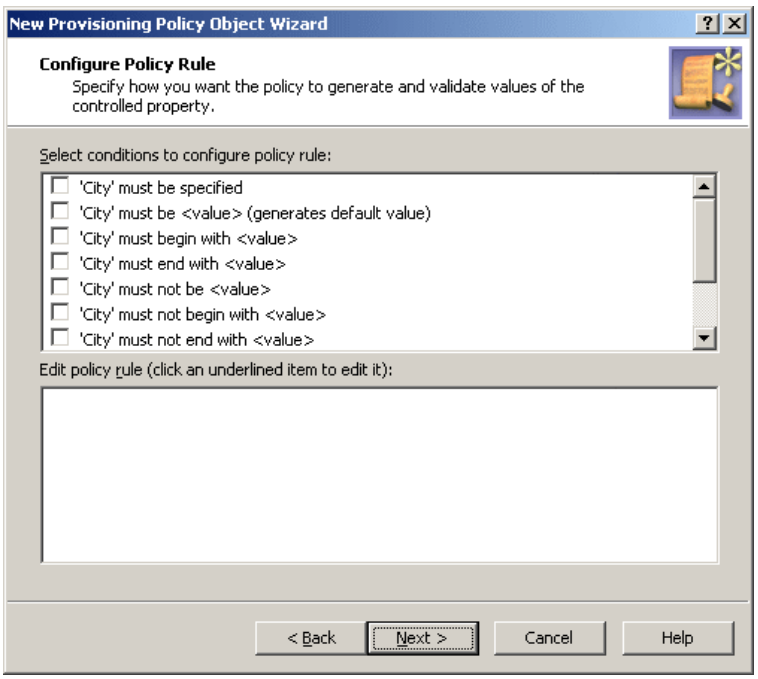
To configure a policy rule, first select appropriate check boxes in the upper box on the **Configure Policy Rule** page. Each check box label is composed of the name of the controlled property followed by a condition. For example, if you select the check box next to **must be specified**, the policy will force a value to be assigned to the property.

If you want the policy to generate a default value for the controlled property, select the check box next to **must be <value> (generates default value)**.

For the policy not to distinguish between uppercase and lowercase letters, select the check box next to **is case insensitive**.

After you selected check boxes in the upper box, the lower box prompts you to configure values, as shown in the following figure.

Figure 34: Edit policy rule



In the lower box, click links labeled **<click to add value>** to configure additional values. If you select several check boxes in the upper box, you must configure value for each condition.

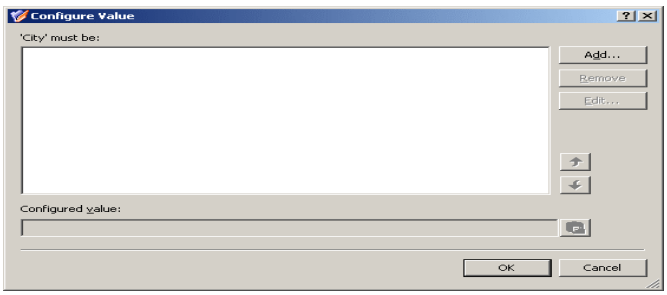
In the **Edit policy rule** box, you can also:

- Modify a value. Right-click the value and click **Edit**. This displays a dialog box similar to the **Add Value** dialog box, discussed later in this section.
- Remove a value. Right-click the value and click **Remove**.
- Rearrange the list of values (provided that multiple values are specified for a particular condition). Right-click a value and use the **Move Up** or **Move Down** command to change position of the value in the list, or click **Sort Items Ascending** or **Sort Items Descending** to sort the list accordingly.
- Import values from a text file. Prepare a text file containing one value per line, right-click any value in the **Edit policy rule** box, click **Import Items**, and then open the file you prepared.
- Export the values to a text file. Right-click any value, click **Export Items**, and specify a text file to write the values to.
- Specify whether you want the rule to generate the default value. Click the **Yes** or **No** link to toggle this option.

To combine criteria into the policy rule, use the AND or OR operator. The policy will pass if the property value meets all of the specified criteria or any one of them depending on the operator you choose. To change the operator, click the link labeled **and** or **or**.

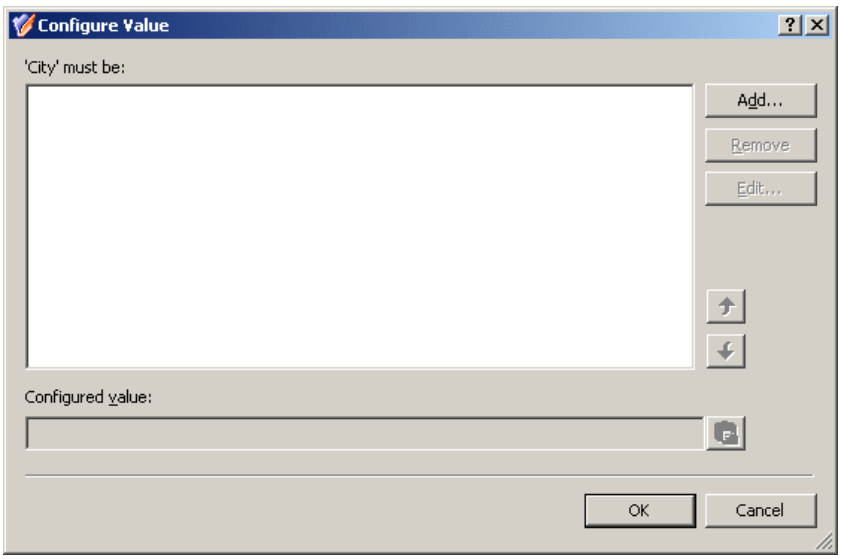
Clicking a link labeled **<click to add value>** displays a dialog box similar to the following figure.

Figure 35: Add Value



The **Add Value** dialog box allows you to specify a value for the selected condition. You can type a value in the edit box or use the point-and-click interface to configure a value. Clicking the **Configure** button displays the **Configure Value** dialog box, shown in the following figure.

Figure 36: Configure Value



Each value is a concatenation of one or more entries. In the **Configure Value** dialog box, you can:

- Add any number of entries to the value. Click **Add** to display the **Add Entry** window, discussed later in this section.
- Remove entries from the value. Select entries from the list and click **Remove**.
- Modify entries included in the value. Select an entry and click **Edit**. This displays a window similar to the **Add Entry** window, where you can view and modify the entry properties.
- Move the selected entry up and down in the list, thereby rearranging the entries in the value. Click an entry in the upper box, and then click the arrow buttons next to the box to move the entry.
- Paste the text from the Clipboard at the end of the value. If the text includes a valid syntax implementing an entry of a type other than **Text** (see the table below), the

syntax is treated accordingly. Copy a text to the Clipboard, and then click the button next to the **Configured value** box to paste the text at the end of the string in that box.

In the **Add Entry** window, you can select the type of the entry to add, and then configure the entry. The following table summarizes the available types of entries.

Table 16: Types of entries

Type of entry	Description
Text	Adds a text string to the value.
<*Object*> Property	Adds a selected property (or a part of a property) of the object being managed by the policy. When displaying this type of entry the console substitutes for the <Object> placeholder the actual category of objects that are subject to the policy control. For example, with the policy configured to control a certain property of user objects, you can use this type of entry to populate that property of a user object with data stored in other properties of that same user object.
Parent OU Property	Adds a selected property (or a part of a property) of an organizational unit in the hierarchy of containers above the object being managed by this policy. For example, with the policy configured to control a certain property of user objects, you can use this type of entry to populate that property of a user object with data stored in properties of the organizational unit containing that user object (immediate parent OU).
Parent Domain Property	Adds a selected property (or a part of a property) of the domain of the object being managed by this policy. For example, with the policy configured to control a certain property of user objects, you can use this type of entry to populate that property of a user object with data stored in properties of the domain in which the user object resides.
Mask	Adds a syntax that determines what characters are allowed in the property controlled by this policy. You can use this type of entry to enforce a data format like numeric, postal/ZIP code, or telephone number.

The steps to configure an entry depend on the type of the entry. The following sections elaborate on the procedures for each of the entry types occurring in the **Add Entry** window.

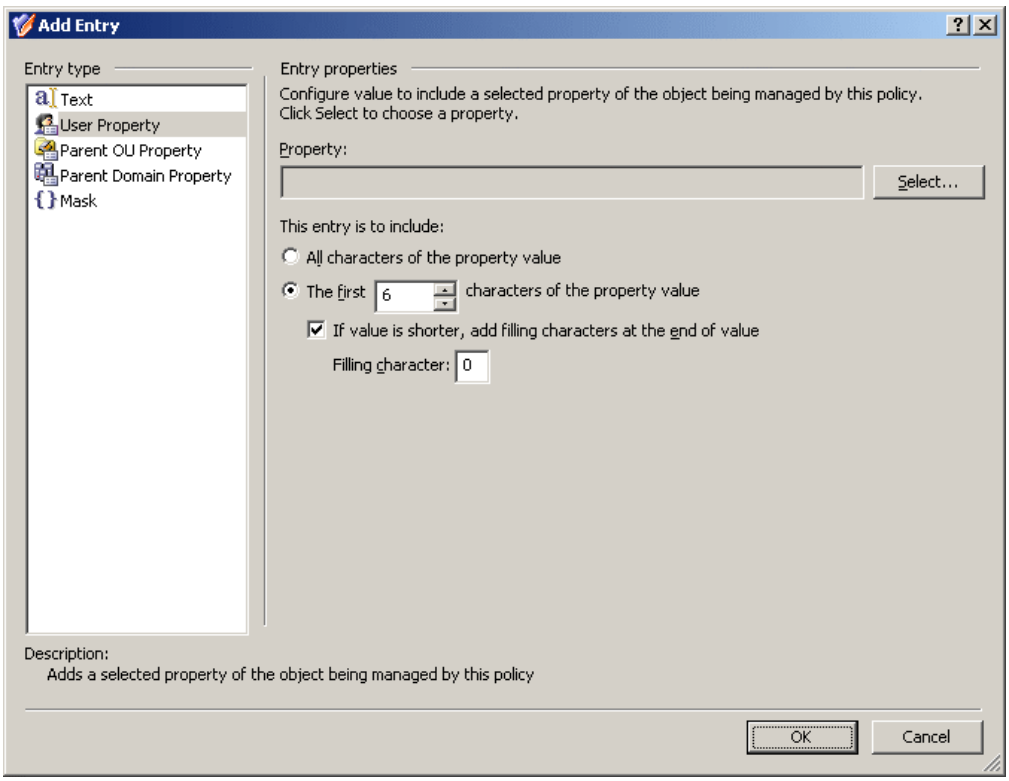
Entry type: Text

When you select **Text** under **Entry type** in the **Add Entry** window, the **Entry properties** area displays the **Text value** box.

In the **Text value** box, type the text you want to include in the value, and then click **OK**.

Entry type: <Object> Property

When you select **<Object> Property** under **Entry type** in the **Add Entry** window, the **Entry properties** area looks like the following figure.



Using this entry type, you can configure a value based on a property of the object itself. To choose a property, click **Select**.

If you want the entry to include the entire value of the property, click **All characters of the property value**. Otherwise, click **The first**, and specify the number of characters to include in the entry.

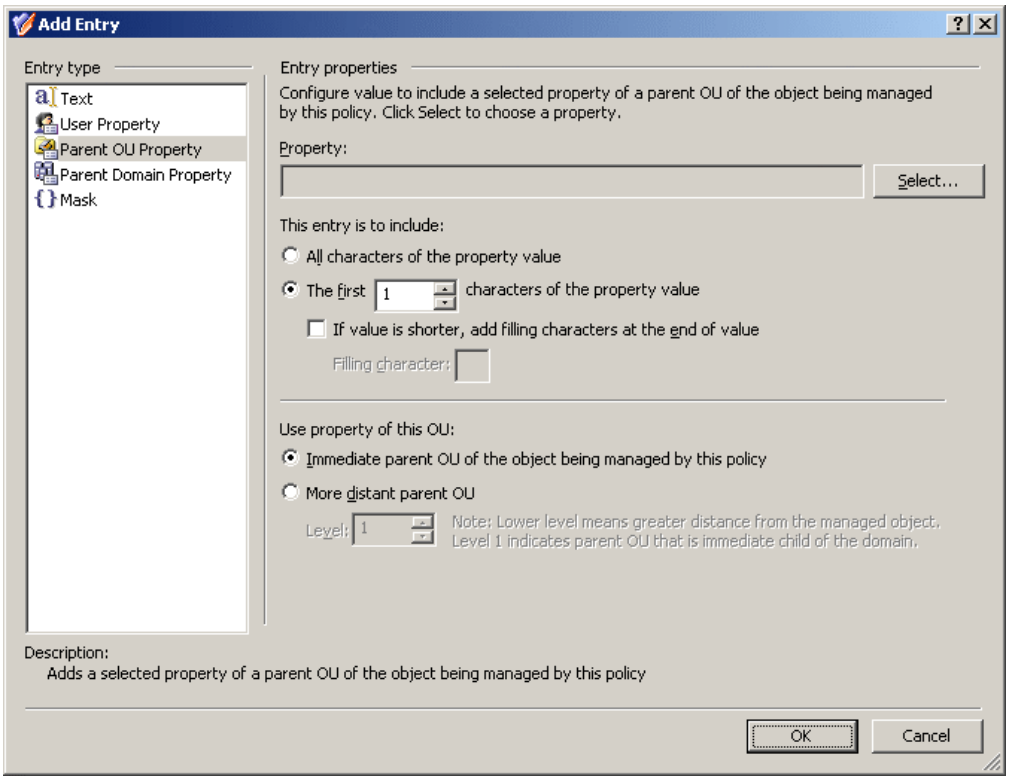
In the latter case, you can select the **If value is shorter, add filling characters at the end of value** check box, and type a character in the **Filling character** box. This character will fill the missing characters in the value of the object property if the value is shorter than specified in the box next to the option **The first**.

When you are done configuring an entry, click **OK** to close the **Add Entry** window. The entry is added to the **Configure Value** dialog box.

Entry type: Parent OU Property

When you select **Parent OU Property** under **Entry type** in the **Add Entry** window, the **Entry properties** area looks like the following figure.

Figure 37: Add Entry: Parent OU Property



Using this entry type, you can configure a value based on a property of a parent organizational unit (OU) of the object being managed by this policy. To choose an OU property, click **Select**.

If you want the entry to include the entire value of the property, click **All characters of the property value**. Otherwise, click **The first**, and specify the number of characters to include in the entry.

In the latter case, you can select the **If value is shorter, add filling characters at the end of value** check box, and type a character in the **Filling character** box. This character will fill the missing characters in the value of the OU property if the value is shorter than specified in the box next to the option **The first**.

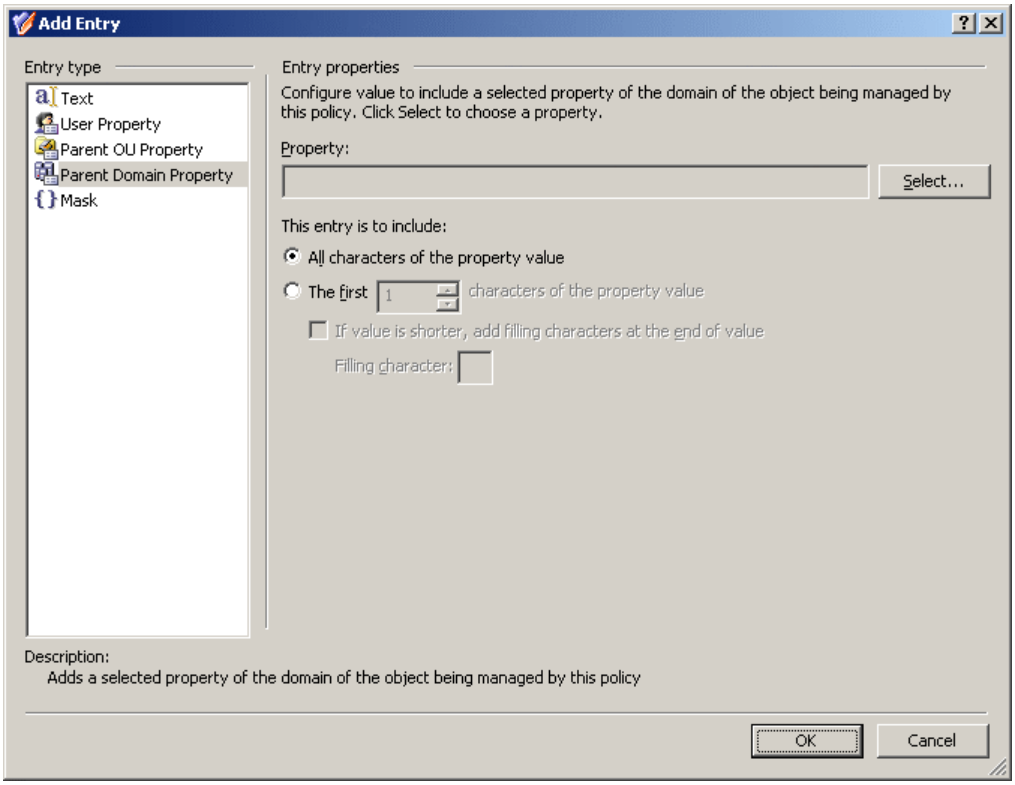
You can also specify the level of the OU you want to the policy to use. To use the property of the OU in which the object resides, click **Immediate parent OU of the object being managed by this policy**. To use the property of a parent OU of a different level, click **More distant parent OU** and then, in the **Level** box, specify the level of the OU. Lower level means greater distance from the managed object in the hierarchy of containers above that object. OU level 1 is an immediate child OU of the domain.

When you are done configuring an entry, click **OK** to close the **Add Entry** window. The entry is added to the **Configure Value** dialog box.

Entry type: Parent Domain Property

When you select **Parent Domain Property** under **Entry type** in the **Add Entry** window, the **Entry properties** area looks like the following figure.

Figure 38: Add Entry: Parent Domain Property



Using this entry type, you can configure a value based on a property of the domain of the object being managed by this policy. To choose a domain property, click **Select**.

If you want the entry to include the entire value of the property, click **All characters of the property value**. Otherwise, click **The first**, and specify the number of characters to include in the entry.

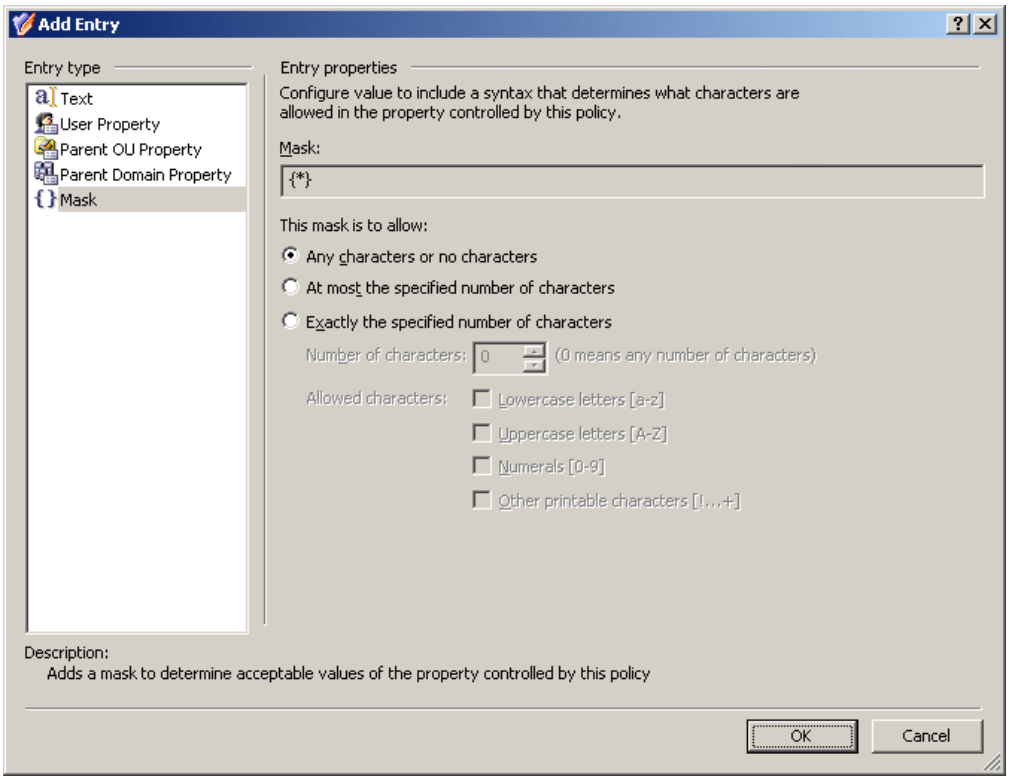
In the latter case, you can select the **If value is shorter, add filling characters at the end of value** check box, and type a character in the **Filling character** box. This character will fill the missing characters in the value of the domain property if the value is shorter than specified in the box next to the option **The first**.

When you are done configuring an entry, click **OK** to close the **Add Entry** window. The entry is added to the **Configure Value** dialog box.

Entry type: Mask

When you select **Mask** under **Entry type** in the **Add Entry** window, the **Entry properties** area looks like the following figure.

Figure 39: Add Entry: Mask



With this entry type, you can define which characters (letters, numerals) are acceptable in the entry you add to the value of the controlled property.

If you want to allow the entry to include any series of characters, click **Any characters or no characters**.

If you want to specify a maximum number of allowed characters the entry may include, click **At most the specified number of characters**. In the **Number of characters** box, specify the number of allowed characters. The entry may include any number of characters not exceeding the specified number. Under **Allowed characters**, select check boxes to specify the allowed characters.

If you want to specify an exact number of allowed characters that the entry must include, click **Exactly the specified number of characters**. In the **Number of characters** box, specify the number of allowed characters. The entry must include exactly the specified number of characters. Under **Allowed characters**, select check boxes to specify the allowed characters.

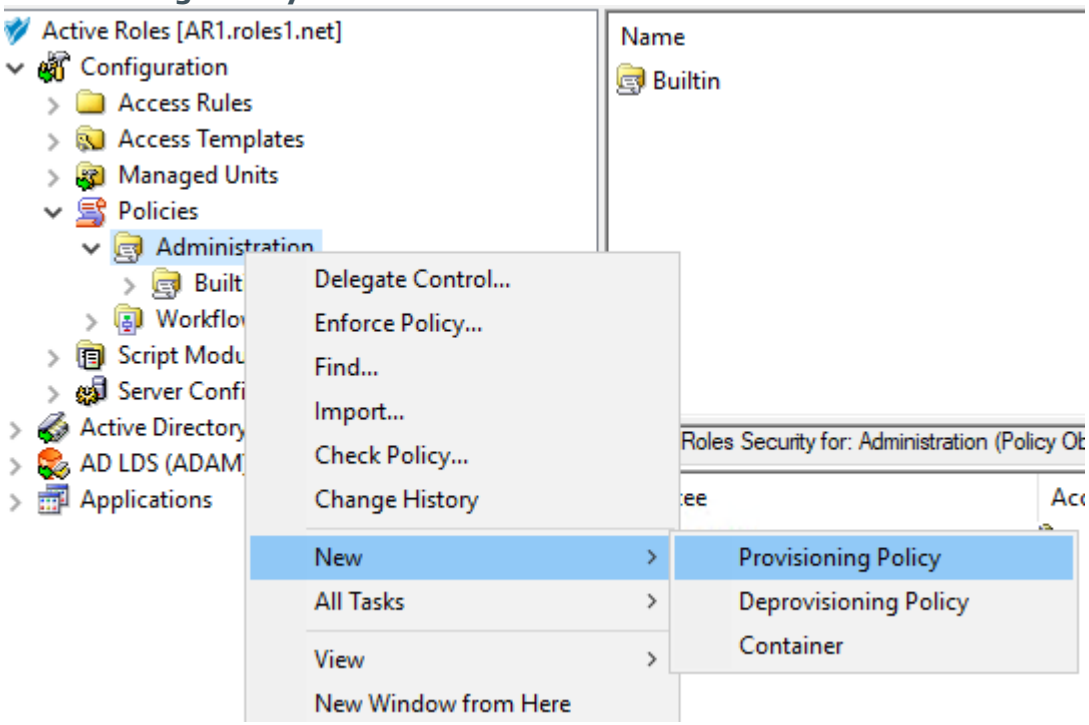
When you are done configuring an entry, click **OK** to close the **Add Entry** window. The entry is added to the **Configure Value** dialog box.

Steps for configuring a Property Generation and Validation policy

To configure a Property Generation and Validation policy via the Active Roles MMC console, perform the following procedure.

To configure a Property Generation and Validation policy

1. Navigate to **Configuration** > **Policies** > **Administration**.
2. To open the **New Provisioning Policy Object Wizard** dialog, right-click in the middle pane to open the context menu, and then select **New** > **Provisioning Policy**.



3. On the **Name and Description** page, provide a unique **Name** for the new policy object. Optionally, also provide a **Description**. To continue, click **Next**.
4. On the **Policy to Configure** page, select **Property Generation and Validation**, and then click **Next**.
5. On the **Controlled Property** page, click **Select** to open the **Select Object Type and Property** dialog.
6. To select the object type and its object property you want the policy to control, use the settings of the **Select Object Type and Property** dialog:
   - Use the **Object type** drop-down menu to select the object type whose property you want to provision.
   - Use either the **Look for Property** search box to manually search for the object property you want to provision, or browse it in the **Object Property**

list.

> TIP: If you do not see the object type you need, expand the list by selecting **Show all possible object types**.

> NOTE: Policy Object settings specific to Azure cloud-only objects (such as cloud-only Azure users, guest users, or contacts) are available only if your Active Roles deployment is licensed for managing cloud-only Azure objects. Contact One Identity support for more information.
>
> Also, Policy Objects specific to Azure cloud-only objects will work correctly only if an Azure tenant is already configured in the AD of the organization, and Active Roles is already set as a consented Azure application for that Azure tenant. For more information on these settings, see Configuring a new Azure tenant and consenting Active Roles as an Azure application.

- Once you selected the object and property, click **OK** to continue.

7. On the **Configure Policy Rule** page, specify the condition(s) you want to configure for the policy by selecting them in the **Select conditions to configure policy rule** list. The selected conditions then appear in the **Edit policy rule** text box.
8. (Optional) If the selected condition supports editing, then click the underlined part of the condition to open the **Add Value** dialog and edit its settings.

   To specify additional configuration for the condition, enter a variable into the **Value** field, then click **OK** to close the **Add Value** dialog.

   Alternatively, click **Configure Value**, then click **Add**, and configure an entry manually in the **Add Entry** dialog. For more information on manual configuration, see Steps for configuring entries. To close the **Add Value** dialog, click **OK**.
9. (Optional) If multiple conditions are selected, switch between the AND and OR logic of the condition relations by clicking **and** or **or**.
10. After selecting and configuring the condition(s), click **Next**.
11. (Optional) On the **Policy Description** page, modify the default description of the policy generated by the wizard. To do so, select **Modify this policy description** to make the description editable. Modify the description, then click **Next**.
12. On the **Enforce Policy** page, specify the objects to which the configured Policy Object will be applied. Click **Add**, and then use the **Select Objects** dialog to locate and select the objects.

    > TIP: When provisioning cloud-only Azure users or guest users, you can either select the respective object category (such as the **Azure user** or **Azure guest user** node) in this step, or the **Azure tenant** that contains the Azure objects.
13. Click **Next** and then **Finish** to complete creating the Policy Object.

Steps for configuring entries

Use the following step-by-step instructions to configure an entry in the **Add Entry** dialog box. The same instructions apply when you are making changes to an existing entry.

To configure a Text entry

1. Under **Entry type**, click **Text**.

   Use a **Text** entry to add a text string to the value you are configuring.

2. In **Text value**, type the text string you want the value to include.
3. Click **OK**.

To configure an <Object> Property entry

1. Under **Entry type**, click **<Object> Property**.

   Use an **<Object> Property** entry when configuring a value to include a certain property (or a part of a property) of the object that is under the control of the policy. In these instructions, **<Object>** stands for the type of object, such as **User**, **Group**, or **Computer**.

2. Click **Select**, click the property to include in the value, and then click **OK**.
3. If you want the entry to include the entire value of the property, click **All characters of the property value**. Otherwise, click **The first**, and specify the number of characters to include in the entry.
4. If you selected **The first**, then, optionally, select **If value is shorter, add filling characters at the end of value**, and type a character in **Filling character**.

   This character will fill the missing characters in the value of the property if the value is shorter than specified in the box next to **The first**.

5. Click **OK**.

To configure a Parent OU Property entry

1. Under **Entry type**, click **Parent OU Property**.

   Use a **Parent OU Property** entry when configuring a value to include a certain property (or a part of a property) of an organizational unit (OU) in the hierarchy of containers above the object being managed by the policy.

2. Click **Select**, click the property to include in the value, and then click **OK**.
3. If you want the entry to include the entire value of the property, click **All characters of the property value**. Otherwise, click **The first**, and specify the number of characters to include in the entry.
4. If you selected **The first**, then, optionally, select **If value is shorter, add filling characters at the end of value**, and type a character in **Filling character**.

   This character will fill the missing characters in the value of the property if the value is shorter than specified in the box next to **The first**.

5. Choose one of these options:
   - To use the property of the OU in which the object resides, click **Immediate parent OU of the object being managed by this policy**.

- To use the property of a parent OU of a different level, click **More distant parent OU** and then, in **Level**, specify the level of the OU.

Lower level means greater distance from the managed object in the hierarchy of containers above that object. OU level 1 is an immediate child OU of the domain.

6. Click **OK**.

To configure a Parent Domain Property entry

1. Under **Entry type**, click **Parent Domain Property**.

   Use a **Parent Domain Property** entry when configuring a value to include a certain property (or a part of a property) of the domain of the object being managed by the policy.

2. Click **Select**, click the property to include in the value, and then click **OK**.
3. If you want the entry to include the entire value of the property, click **All characters of the property value**. Otherwise, click **The first**, and specify the number of characters to include in the entry.
4. If you selected **The first**, then, optionally, select **If value is shorter, add filling characters at the end of value**, and type a character in **Filling character**.

   This character will fill the missing characters in the value of the property if the value is shorter than specified in the box next to **The first**.

5. Click **OK**.

To configure a Mask entry

1. Under **Entry type**, click **Mask**.

   Use a **Mask** entry when configuring a value to include a syntax that determines how many and what characters are allowed in the property controlled by the policy.

2. Select one of these options:
   - **Any characters or no characters** to allow the entry to include any series of characters.
   - **At most the specified number of characters** to specify a maximum number of allowed characters the entry may include.
   - **Exactly the specified number of characters** to specify an exact number of allowed characters that the entry must include.
3. If you selected the second option or the third option in Step 2, do the following:
   - In **Number of characters**, specify the how many characters are allowed in this entry.

   If you selected the second option, the entry may include any number of characters not exceeding the number specified.

   If you selected the third option, the entry must include exactly the specified number of characters.

- Under **Allowed characters**, select check boxes to specify what characters are allowed in this entry.

4. Click **OK**.

To configure a Date and Time entry

1. Under **Entry type**, click **Date and Time**.

   Use a **Date and Time** entry when configuring a value to include the date and time of the operation performed by the policy (for example, the date and time when the user was deprovisioned).

2. In the list under **Date and time format**, click the date or time format you want.
3. Click **OK**.

To configure an Initiator ID entry

1. Under **Entry type**, click **Initiator ID**.

   Use an **Initiator ID** entry when configuring a value to include the ID of the Initiator, that is, the user who initiated the operation performed by the policy (for example, the ID of the user who initiated the deprovisioning operation). You can build the Initiator ID based on a combination of properties of the Initiator.

2. Select one of these options:
   - **User logon name (pre-Windows 2000) of the Initiator, in the form Domain\Name** to set the Initiator ID to the pre-Windows 2000 user logon name of the Initiator.
   - **User logon name of Initiator** to set the Initiator ID to the user logon name of the Initiator.
   - **Initiator ID built using a custom rule** to compose the Initiator ID of other properties specific to the Initiator.
3. If you selected the third option in Step 2, click **Configure**, and use the **Configure Value** dialog box to set up the value to be used as the Initiator ID: Click **Add** and specify the entries for the value as appropriate.

   You can configure entries of these categories: **Text** (any text string), **Initiator Property** (a certain property of the Initiator user object), **Parent OU Property** (a certain property of an organizational unit that holds the Initiator user object), **Parent Domain Property** (a certain property of the domain of the Initiator user object). To configure entries, use the instructions that are given earlier in this topic.

4. Click **OK**.

To configure a Uniqueness Number entry

1. Under **Entry type**, click **Uniqueness Number**.

   Use a **Uniqueness Number** entry when configuring a value to include a number the policy will increment in the event of a naming conflict. For example, in a policy that generates a user logon name or email alias, you can add an entry of this category to

the generation rule in order to ensure the uniqueness of the name or alias generated by the policy.

2. Click one of these options:
   - **Add always**. The value includes this entry regardless of whether or not the policy encounters a naming conflict when applying the generation rule
   - **Add if the property value is in use**. The policy adds this entry to the value in the event of a naming conflict; otherwise the value does not include this entry.
3. Specify how you want the entry to be formatted:
   - To have the entry formatted as a variable-length string of digits, clear the **Fixed-length number, with leading zeroes** check box. In most cases, this will result in a single-digit entry.
   - To have the entry formatted as a fixed-length string of digits, select the **Fixed-length number, with leading zeroes** check box, and then specify the number of digits you want the string to include. This will result in an entry prefixed with the appropriate number of zeroes, such as 001, 002, 003.
4. Click **OK**.

NOTE:

- You may need to configure an entry when configuring a policy such as Property Generation and Validation (see Steps for configuring a Property Generation and Validation policy), User Logon Name Generation (see Steps for configuring a User Logon Name Generation policy), Group Membership AutoProvisioning (see Steps for configuring a Group Membership AutoProvisioning policy), E-mail Alias Generation (see Steps for configuring an E-mail Alias Generation policy), User Account Deprovisioning (see Steps for configuring a User Account Deprovisioning policy), or Group Object Deprovisioning (see Steps for configuring a Group Object Deprovisioning policy).
- The contents of the **Entry Type** list in the **Add Entry** dialog box depends upon the type of the policy you are configuring.

Scenario 1: Using mask to control phone number format

This scenario describes how to configure a policy that forces the user phone number to conform to the format (###) ###-##-##.

To implement this scenario, you must perform the following actions:

1. Create and configure a Policy Object that defines the appropriate policy.
2. Apply the Policy Object to a domain, OU, or Managed Unit.

As a result, when creating or modifying a user object in the container you selected in Step 2, Active Roles checks whether the phone number conforms to the stated format. If not, the policy disallows the creation or modification of the user object.

The following two sections elaborate on the steps to implement this scenario.

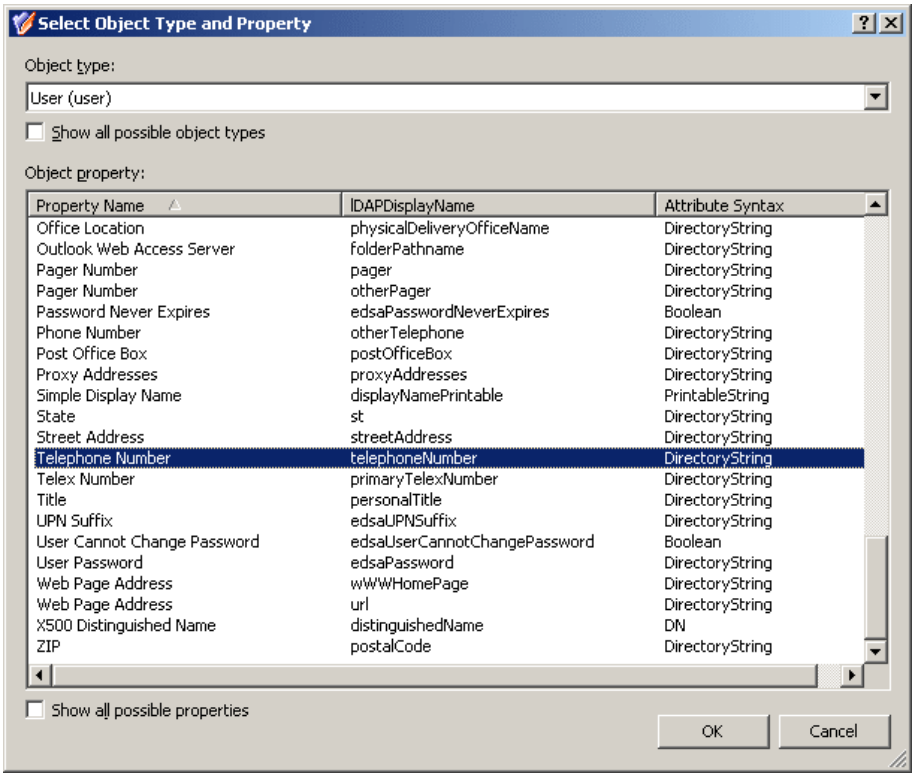
Step 1: Creating and configuring the Policy Object

You can create and configure the Policy Object you need by using the New Provisioning Policy Object wizard. For information about the wizard, see Creating a Policy Object in the Policy Object management tasks section earlier in this chapter.

To configure the policy, click **Property Generation and Validation** on the **Policy to Configure** page of the wizard. Then, click **Next**.

On the **Controlled Property** page, click **Select**. Then, in the **Select Object Type and Property** dialog box, select **User** from the **Object type** list, and click **Telephone number** in the **Object property** list, as shown in the following figure.

Figure 40: Select Object type and property



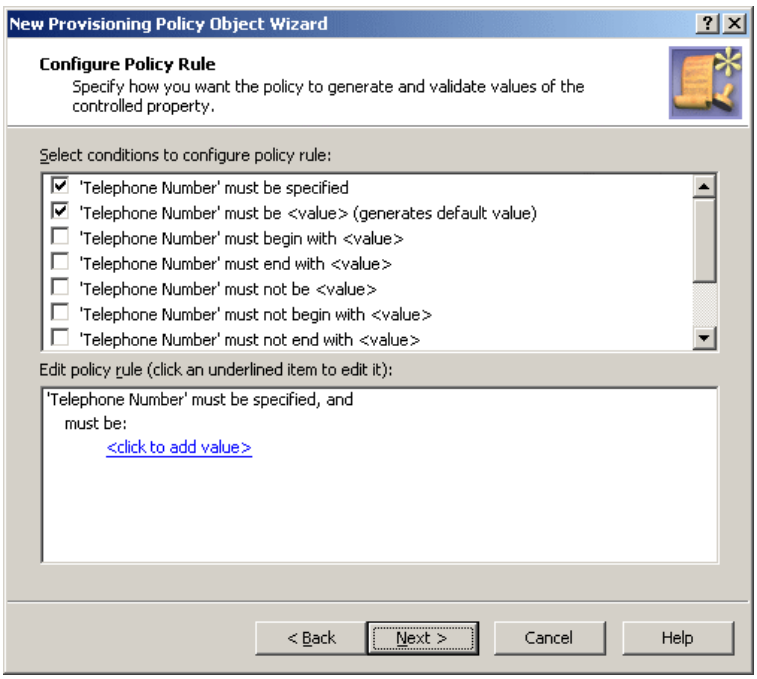
Click **OK**, and then click **Next**.

On the **Configure Policy Rule** page, in the upper box, select the following check boxes:

- **'Telephone Number' must be specified**. This makes the phone number a required property, that is, requires that a phone number be specified in every user account.
- **'Telephone Number' must be <value>**. This allows you configure a mask for the telephone number by adding the appropriate entry to the value for this condition.

At this stage, the **Configure Policy Rules** page looks like the following figure.

Figure 41: Configure policy rules



The next phase is to configure the value.

Click the link labeled **<click to add value>**. In the **Add Value** dialog box, click **Configure**. In the **Configure Value** dialog box, click **Add**. In the **Add Entry** window, under **Entry type**, click **Mask**.

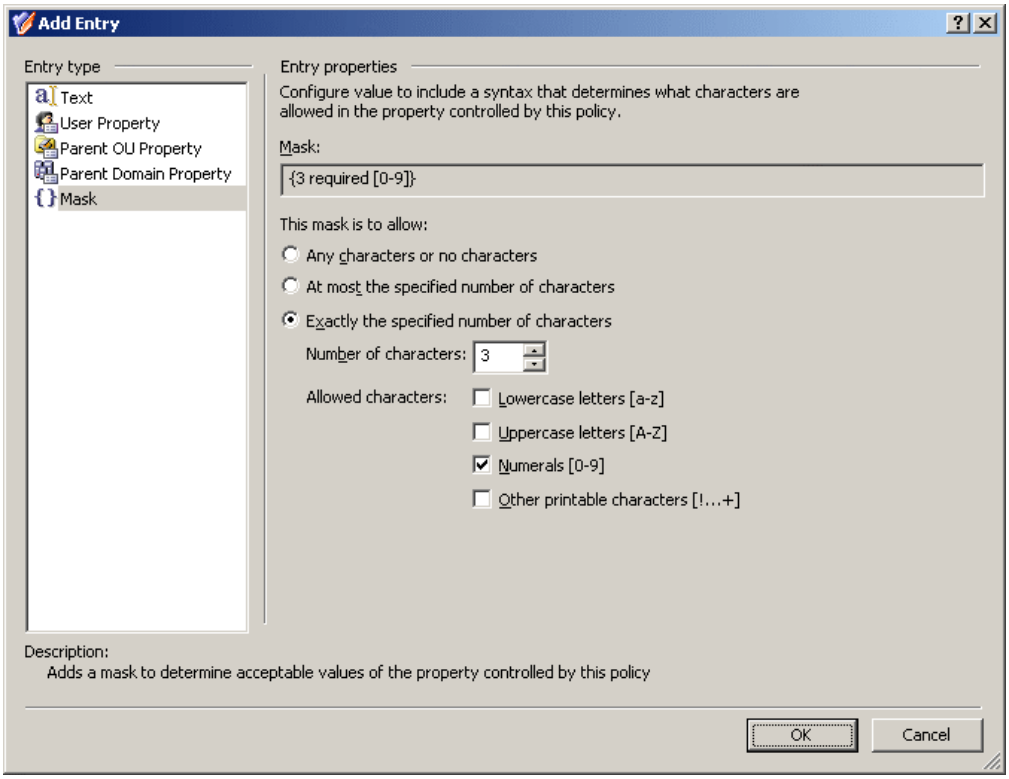
Now you can use the **Entry properties** area in the **Add Entry** window to configure a mask.

The format consists of four groups of numerals divided by certain characters—space character, hyphens, and brackets. First, configure a mask that requires the first three characters to be numerals:

- Select **Exactly the specified number of characters**.
- In the **Number of characters** box, enter **3**.
- Under **Allowed characters**, select the **Numerals** check box.

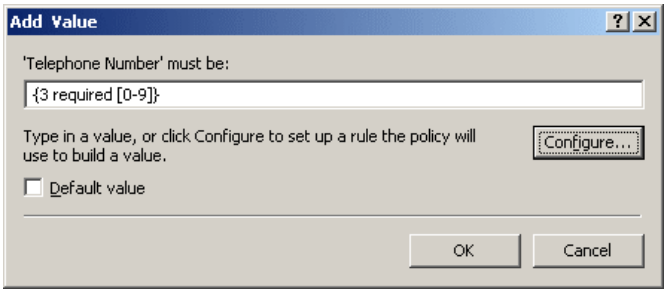
The **Add Entry** window should look as shown in the following figure.

Figure 42: Add entry



Click **OK** to close the **Add Entry** window. Then, click **OK** to close the **Configure Value** dialog box. As a result, the **Add Value** dialog box looks as shown in the following figure.

Figure 43: Add value dialog box



Taking into consideration the mask you have configured, you can guess that the mask for the phone number format you need is as follows:

```
({3 required [0-9]}) {3 required [0-9]}-{2 required [0-9]}-{2 required [0-9]}
```

Type this mask in the **'Telephone Number' must be** box in the **Add Value** dialog box. Pay attention to the round brackets enclosing the first three characters, a space character following the group in the round brackets, and two hyphen characters that separate the groups of characters.

Click **OK** to close the **Add Value** dialog box. Click **Next** and follow the instructions in the wizard to create the Policy Object.

Step 2: Applying the Policy Object

You can apply the Policy Object by using the **Enforce Policy** page in the **New Provisioning Policy Object** wizard, or you can complete the wizard and then use the **Enforce Policy** command on the domain, OU, or Managed Unit where you want to apply the policy.

For more information on how to apply a Policy Object, see Applying Policy Objects and Managing policy scope earlier in this chapter.

Scenario 2: Using regular expressions to control phone number format

This scenario describes how to configure a policy that forces the user phone number to conform to the following format:

- The first character must be "+" .
- The second character(s) must be the country code .

  (This is 1 in the US and Canada, and 61 in Australia for example.)
- Use spaces (instead of dashes or braces) to separate area code.
- Use spaces (instead of dashes) to separate the phone number.
- Optionally, use a lowercase "x" to indicate an extension.

The following table provides some examples to clarify how the phone number should look in accordance with these formatting requirements.

Table 17: Phone number format

Correct	Incorrect	Comment
+1 949 754 8515	949-754-8515	The incorrect entry does not begin with + and country code, and uses dashes instead of space.
+44 1628 606699 x1199	+44 1628 606699 X1199	The incorrect entry uses the upper-case X.

To implement this scenario, you must perform the following actions:

1. Configure the Policy Object that defines the appropriate policy.
2. Apply the Policy Object to a domain, OU, or Managed Unit.

As a result, when creating or modifying a user object in the container you selected in Step 2, Active Roles checks whether the phone number conforms to the stated format. If not, the policy disallows the creation or modification of the user object.

Step 1: Configuring the Policy Object

You can configure the Policy Object you need by modifying the Policy Object that implements the previous scenario, see Scenario 1: Using mask to control phone number format earlier in this section.

Display the **Properties** dialog box for that Policy Object and go to the **Policies** tab. Then, select the policy from the list, and click **View/Edit** to display the **Property Generation and Validation Policy Properties** dialog box.

The **Policy Rule** tab in the **Property Generation and Validation Policy Properties** dialog box looks similar to the **Configure Policy Rule** page in the wizard you used to configure the policy. You can use that tab to modify the policy rules.

First, modify the rule to remove the mask entry. On the **Policy Rule** tab, in the upper box, clear the **'Telephone Number' must be <value>** check box.

Next, choose to configure a rule based on regular expressions. On the **Policy Rule** tab, in the upper box, select the **'Telephone Number' must match regular expression <value>** check box. To access this check box, you need to scroll down the list of check boxes.

Finally, specify the regular expressions that define the policy in question. The regular expressions you need are as follows:

```
^\+([0-9]+ )+[0-9]+$
```

```
^\+([0-9]+ )+x[0-9]+$
```

The following table briefly describes the elements that are used in the two above syntax. For more information about regular expressions, see Appendix A: Using regular expressions later in this document.

Table 18: Regular expressions

This Element	Indicates
^	The beginning of the input string to validate
\+	The escape sequence to represent the plus character (+)
([0-9]+)+	Concatenation of one or more substrings, with each substring consisting of one or more digit characters followed by a space character
[0-9]+	One or more digit characters.
x[0-9]+	A lowercase "x" followed by one or more digit characters
$	The end of the input string to validate

Thus, the policy must be configured to only allow the telephone numbers that match ^\+([0-9]+)+[0-9]+$ (telephone numbers without extensions) or ^\+([0-9]+)+x[0-9]+$ (telephone numbers that include extensions). Proceed with configuring the policy as follows:

1. On the **Policy Rule** tab, in the lower box, click the link labeled **<click to add value>**.
2. In the **Add Value** dialog box, enter **^\+([0-9]+)+[0-9]+$**, and click **OK**.
3. On the **Policy Rule** tab, in the lower box, click the link labeled **<click to add value>**.
4. In the **Add Value** dialog box, enter **^\+([0-9]+)+x[0-9]+$**, and click **OK**.
5. Click **OK** to close the **Property Generation and Validation Policy Properties** dialog box.

Step 2: Applying the Policy Object

You can apply the Policy Object without closing its **Properties** dialog box. Go to the **Scope** tab and do the following:

1. On the **Scope** tab, click the **Scope** button to display the **Active Roles Policy Scope** window for the Policy Object you are managing.
2. Click **Add** and select the domain, OU, or Managed Unit where you want to apply the policy.

   You can also use the **Remove** button to remove items where you want the policy to no longer be applied.
3. Click **OK** to close the **Active Roles Policy Scope** window.
4. Click **OK** to close the **Properties** dialog box for the Policy Object.

For more information on how to apply a Policy Object, see Applying Policy Objects and Managing policy scope earlier in this chapter.

User Logon Name Generation

Policies in this category are intended to automate the assignment of the pre-Windows 2000 user logon name when creating or modifying a user account, with flexible options to ensure uniqueness of the policy-generated name.

The ability to generate a unique name is essential. If Active Roles attempts to assign a policy-generated name when there is an existing user account with the same pre-Windows 2000 user logon name, a naming conflict will occur. Active Directory does not support multiple accounts with the same pre-Windows 2000 user logon name. A policy can be configured to generate a series of names in order to prevent naming conflicts with existing accounts.

When configuring a policy of this category, you can define multiple rules so that the policy applies them successively, attempting to generate a unique name in the event of a naming conflict. You can also configure a rule to include an incremental numeric value to ensure uniqueness of the policy-generated name. You also have the option to allow policy-generated names to be modified by operators who create or update user accounts.

How this policy works

When creating a user account, Active Roles relies on this policy to assign a certain pre-Windows 2000 user logon name to the user account. The policy generates the name based on properties of the user account being created. A policy may include one or more rules that construct the name value as a concatenation of entries that are similar to those you encounter when using a Property Generation and Validation policy.

A special entry—uniqueness number—is provided to help make the policy-generated name unique. A uniqueness number entry represents a numeric value the policy will increment in the event of a naming conflict. For example, a policy may provide the option to change the new name from JSmith to J1Smith if there is an existing user account with the pre-Windows 2000 user logon name set to JSmith. If the name J1Smith is also in use, the new name can be changed to J2Smith, and so on.

The policy configuration provides the option to allow or disallow manual edits of policy-generated names. Permission to modify a policy-generated name can be restricted to the case where the name is in use by another account.

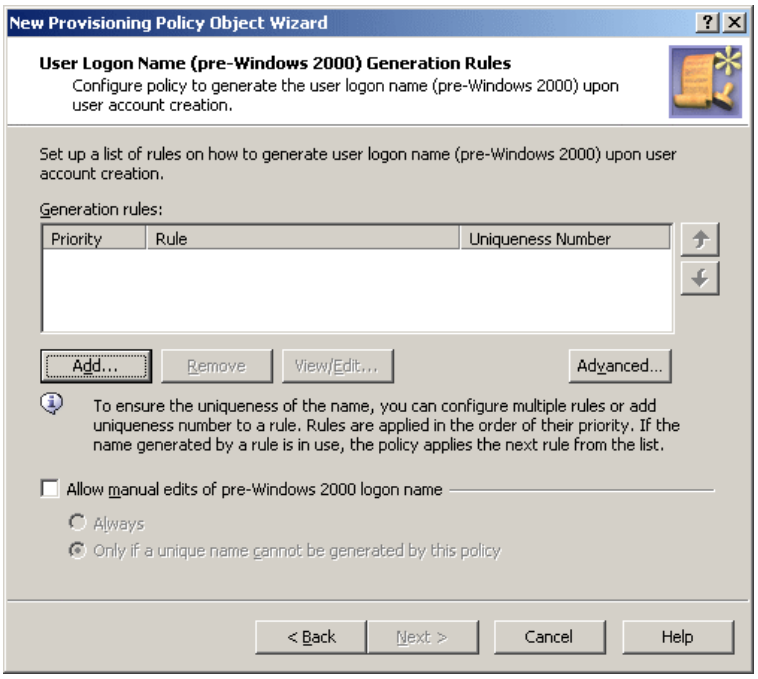
Some specific features of the policy behavior are as follows:

- With a single rule that does not use a uniqueness number, Active Roles simply attempts to assign the generated name to the user account. The operation may fail if the generated name is not unique, that is, the same pre-Windows 2000 user logon name is already assigned to a different user account. If the policy allows manual edits of policy-generated names, the name can be corrected by the operator who creates the user account.
- With multiple rules or with a rule that uses a uniqueness number, Active Roles adds a button at the client side, next to the **User logon name (pre-Windows 2000)** field on the user creation and modification forms.
- To generate a name, the client user (operator) must click that button, which is also the case where the generated name is in use. Clicking the **Generate** button applies a subsequent rule or increases the uniqueness number by one, thereby allowing the name to be made unique.
- The policy defines a list of characters that are unacceptable in pre-Windows 2000 user logon names. The following characters are not allowed: " / \ [ ] : ; | = , + * ? < >
- The policy causes Active Roles to deny processing of operation requests that assign the empty value to the pre-Windows 2000 user logon name.
- When checking user accounts for policy compliance (see later in this document), Active Roles detects, and reports of, the pre-Windows 2000 user logon names that are set up not as prescribed by the user logon name generation policy.

How to configure a User Logon Name Generation policy

To configure a User Logon Name Generation policy, select **User Logon Name Generation** on the **Policy to Configure** page in the New Provisioning Policy Object wizard or in the Add Provisioning Policy wizard. Then, click **Next** to display the **User Logon Name (pre-Windows 2000) Generation Rules** page.

Figure 44: New Provisioning Policy Object wizard



On the **User Logon Name (pre-Windows 2000) Generation Rules** page, you can set up a list of generation rules. Each entry in the list includes the following information:

- **Priority**. The policy applies generation rules in the order of their priority, as they stand in the list: first read, first applied.
- **Rule**. Syntax that defines the rule.
- **Uniqueness Number**. Displays **Yes** or **No**, indicating whether the rule includes a uniqueness number entry.

You can use these buttons manage the list of rules:

- **Add**. Opens the **Configure Value** dialog box, discussed earlier in this chapter (see How to configure a Property Generation and Validation policy). Use that dialog box to configure a value for the **'Logon Name (pre-Windows 2000)' must be** condition, in the same way as you do when configuring a Property Generation and Validation policy. For more information, see Configuring a logon name generation rule later in this section.
- **Remove**. Deletes the rules you select from the list.

- **View/Edit**. Opens the **Configure Value** dialog box for the rule you select from the list. Modify the selected rule by managing the list of entries in that dialog box.
- **Up** and **Down**. Change the order of rules in the list. Click **Up** or **Down** to move a selected rule higher or lower in the list to give the rule a higher or lower priority, respectively.
- **Advanced**. Set certain options that apply to all rules in the list, such as the maximum length of the generated name, whether to format the name as the uppercase or lowercase string, the scope where you want the generated name to be unique, and the characters to be excluded from the generated names.

By selecting the **Allow manual edits of pre-Windows 2000 logon name** check box, you authorize the operator who creates or updates the user account to make changes to the policy-generated name. If this check box is cleared, Active Roles displays the **User logon name (pre-Windows 2000)** field as read-only on the user creation and modification forms.

By selecting the **Always** option, you authorize the operator to modify the pre-Windows 2000 logon name at their discretion. With the **Only if a unique name cannot be generated by this policy** option, you limit manual changes to the situation where a unique name cannot be generated in accordance with the policy rules.

Configuring a logon name generation rule

To configure a generation rule, click the **Add** button beneath the **Generation rules** list. This displays the **Configure Value** dialog box, prompting you to set up a value for the **'Logon Name' must be** condition.

To start configuring a value, click **Add** in the **Configure Value** dialog box. This displays the **Add Entry** window.

A value is a concatenation of one or more entries. In the **Add Entry** window, you can select the type of the entry to add, and then configure the entry. The following table summarizes the available types of entries.

Table 19: Types of entries

Type of entry	Description
Text	Adds a text string to the value.
Uniqueness Number	Adds a numeric value the policy will increment in the event of a naming conflict.
User Property	Adds a selected property (or a part of a property) of the user account to which the policy will assign the logon name.
Parent OU Property	Adds a selected property (or a part of a property) of an organizational unit in the hierarchy of containers above the user account to which the policy will assign the logon name.
Parent Domain Property	Adds a selected property (or a part of a property) of the domain of the user account to which the policy will assign the logon name.

Instructions on how to configure an entry depend on the type of the entry. You can use the instructions outlined in the How to configure a Property Generation and Validation policy section earlier in this chapter to configure an entry of any of these types:

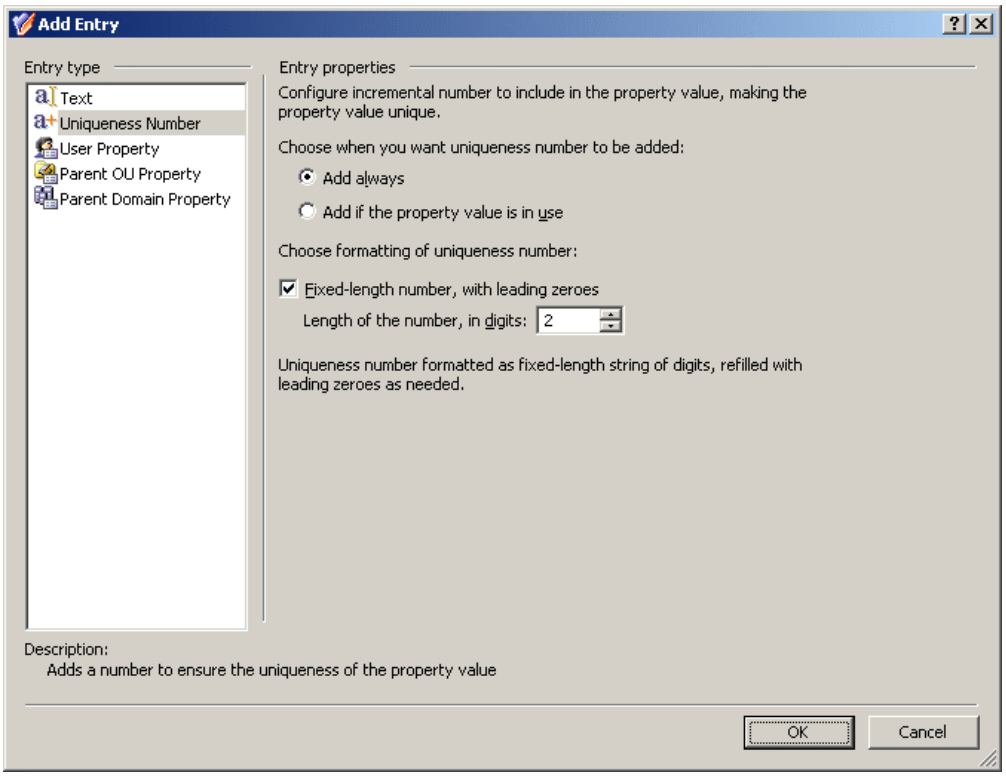
- **Text**. Refer to the Entry type: Text subsection.
- **User Property**. Refer to the Entry type: <Object> Property subsection.
- **Parent OU Property**. Refer to the Entry type: Parent OU Property subsection.
- **Parent Domain Property**. Refer to the Entry type: Parent Domain Property subsection.

The following subsection elaborates on the **Uniqueness Number** entry.

Entry type: Uniqueness Number

When you select **Uniqueness Number** under **Entry type** in the **Add Entry** window, the **Entry properties** area looks likr following figure.

Figure 45: Entry Type: Uniqueness Number



Using this entry type, you can add an entry that represents a number the policy will increment in the event of a naming conflict.

First, you need to choose when you want the policy to employ this entry. You have the following options:

- **Add always**. The value includes this entry regardless of whether or not the policy encounters a naming conflict when applying the generation rule.
- **Add if the property value is in use**. The policy adds this entry to the value in the event of a naming conflict; otherwise the value does not include this entry.

Next, you can specify how you want the entry to be formatted:

- To have the entry formatted as a variable-length string of digits, clear the **Fixed-length number, with leading zeroes** check box. In most cases, this will result in a single-digit entry.
- To have the entry formatted as a fixed-length string of digits, select the **Fixed-length number, with leading zeroes** check box, and then specify the number of digits you want the string to include. This will result in an entry prefixed with the appropriate number of zeroes, such as 001, 002, 003, etc.

When you are done configuring an entry, click **OK** to close the **Add Entry** window. The entry is added to the **Configure Value** dialog box.

Steps for configuring a User Logon Name Generation policy

To configure a User Logon Name Generation policy

1. On the **Policy to Configure** page, select **User Logon Name Generation**, and then click **Next**.
2. On the **User Logon Name (pre-Windows 2000) Generation Rules** page, do the following:
    - Click **Add**, and complete the **Configure Value** dialog box by using the procedure outlined later in this topic, to create a name generation rule.
    - Select a rule and click **Remove** to delete the rule.
    - Select a rule and click **View/Edit** to modify the rule.
    - Select a rule and click **Up** or **Down** to move the rule higher or lower in the list, in order to give the rule a higher or lower priority, respectively.
    - Click **Advanced** to set some options that apply to all rules within the policy. Complete the **Advanced** dialog box by using the procedure outlined later in this topic.
        - If you want the logon name to be allowed for manual edit, select **Allow manual edits of pre-Windows 2000 logon name**. Then, do one of the following:
            - Click **Always** to authorize the operator who creates or updates the user account to modify the pre-Windows 2000 logon name.
            - Click **Only if a unique name cannot be generated by this policy** to allow manual changes only in the situation where a

policy-generated name is already assigned to a different user account.

Click **Next**.

3. On the **Enforce Policy** page, you can specify objects to which this Policy Object is to be applied:
   - Click **Add**, and use the **Select Objects** dialog box to locate and select the objects you want.
4. Click **Next**, and then click **Finish**.

To complete the Configure Value dialog box

1. Click **Add**.
2. Configure an entry to include in the value (for instructions, see Steps for configuring entries).
3. In the **Configure Value** dialog box, add more entries, delete or edit existing ones, and then click **OK**.

To complete the Advanced dialog box

1. In **Maximum length, in characters**, set the maximum length of the generated name.
2. Optionally, select **Adjust the case of characters** to configure case formatting:
   - Click **All UPPERCASE** to format the name as the uppercase string.
   - Click **All lowercase** to format the name as the lowercase string.
3. Specify the scope in which you want the generated name to be unique:
   - Click **Domain** to make the name unique within the domain.
   - Click **Forest** to make the name unique within the forest.
   - Click **All managed domains** to make the name unique across all managed domains.
4. Optionally, in the **Restricted characters** area, specify the characters you want the policy to remove from the generated name.

   The policy always removes the following characters: " @ * + | = \ : ; ? [ ] , < > /

   To specify additional characters, type them one by one, without any separator character, in the provided text box.

Scenario 1: Using uniqueness number

The policy described in this scenario generates the pre-Windows 2000 user logon name in accordance with this rule: the first character of the user first name, optionally followed by a uniqueness number, followed by the user last name. The length of the policy-generated name is at most eight characters. If the name is longer, trailing characters are truncated as needed. Examples of names generated by this policy are as follows:

- JSmitson
- J1Smitso
- J2Smitso

The policy generates the name J1Smitso for the user John Smitson if the name JSmitson is in use. If both JSmitson and J1Smitso are in use, the policy generates the name J2Smitso, and so on.

To implement this scenario, you must perform the following actions:

1. Create and configure the Policy Object that defines the appropriate policy.
2. Apply the Policy Object to a domain, OU, or Managed Unit.

As a result, when assigning a pre-Windows 2000 user logon name to a user account in the container you selected in Step 2, the Active Roles user interfaces provide a **Generate** button to create a name in accordance with the policy rule. In the event of a naming conflict, clicking the **Generate** button causes the policy to add a uniqueness number to the name.

The following two sections elaborate on the steps to implement this scenario.

Step 1: Creating and configuring the Policy Object

You can create and configure the Policy Object you need by using the New Provisioning Policy Object wizard. For information about the wizard, see Creating a Policy Object in the Policy Object management tasks section earlier in this chapter.

To configure the policy, click **User Logon Name Generation** on the **Select Policy Type** page of the wizard. Then, click **Next**.

On the **User Logon Name (pre-Windows 2000) Generation Rules** page, click **Add**. Then, complete the **Configure Value** dialog box as follows:

1. Click **Add**.
2. Configure the entry to include the first character of the user first name:
    a. Under **Entry type**, click **User Property**.
    b. Under **Entry properties**, click **Select**.
    c. In the **Select Object Property** window, click **First Name** in the **Object property** list, and then click **OK**.
    d. Under **Entry properties**, click **The first**, and make sure the box next to that option reads **1**.
    e. Click **OK**.
3. Click **Add**.
4. Configure the entry to optionally include a uniqueness number:
    a. Under **Entry type**, click **Uniqueness Number**.

b. Under **Entry properties**, click **Add if the property value is in use**, and make sure the **Fixed-length number, with leading zeroes** check box is cleared.

c. Click **OK**.

5. Click **Add**.
6. Configure the entry to include the user last name:

a. Under **Entry type**, click **User Property**.

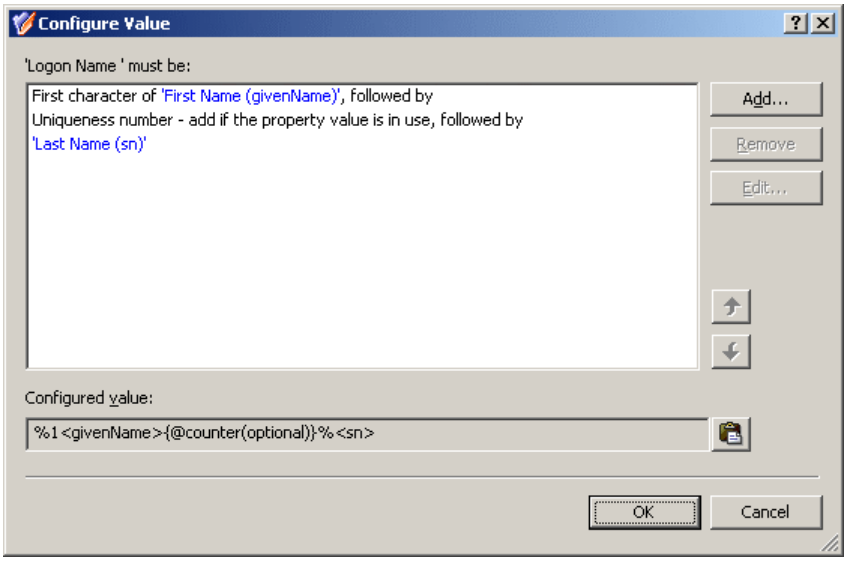
b. Under **Entry properties**, click **Select**.

c. In the **Select Object Property** window, click **Last Name** in the **Object property** list, and then click **OK**.

d. Click **OK**.

After you complete these steps, the list of entries in the **Configure Value** dialog box should look like the following figure.

Figure 46: Configure Value



Click **OK** to close the **Configure Value** dialog box.

You also need to set up the limitation on the length of the name. On the **User Logon Name (pre-Windows 2000) Generation Rules** page, click the **Advanced** button. In the **Advanced** dialog box, in the **Maximum length, in characters** box, type **8**, and then click **OK**.

Click **Next** and follow the instructions in the wizard to create the Policy Object.

Step 2: Applying the Policy Object

You can apply the Policy Object by using the **Enforce Policy** page in the New Provisioning Policy Object wizard, or you can complete the wizard and then use the **Enforce Policy** command on the domain, OU, or Managed Unit where you want to apply the policy.

For more information on how to apply a Policy Object, see Applying Policy Objects and Managing policy scope earlier in this chapter.

Scenario 2: Using multiple rules

The policy described in this scenario uses multiple rules to generate the pre-Windows 2000 user logon name. The rules are as follows:

1. The first character of the user first name, followed by the user last name
2. The first two characters of the user first name, followed by the user last name
3. The first three characters of the user first name, followed by the user last name

The length of the policy-generated name is at most eight characters. If the name is longer, trailing characters are truncated as needed.

Examples of names generated by this policy are as follows:

- JSmitson
- JoSmitso
- JohSmits

The policy generates the name JoSmitso for the user John Smitson if the name JSmitson is in use. If both JSmitson and JoSmitso are in use, the policy generates the name JohSmits.

To implement this scenario, you must perform the following actions:

1. Configure the Policy Object that defines the appropriate policy.
2. Apply the Policy Object to a domain, OU, or Managed Unit.

As a result, when assigning a pre-Windows 2000 user logon name to a user account in the container you selected in Step 2, the Active Roles user interfaces provide a **Generate** button to create the name in accordance with the policy rules. In the event of a naming conflict, clicking the **Generate** button causes the policy to apply a subsequent rule.

The following two sections elaborate on the steps to implement this scenario.

Step 1: Configuring the Policy Object

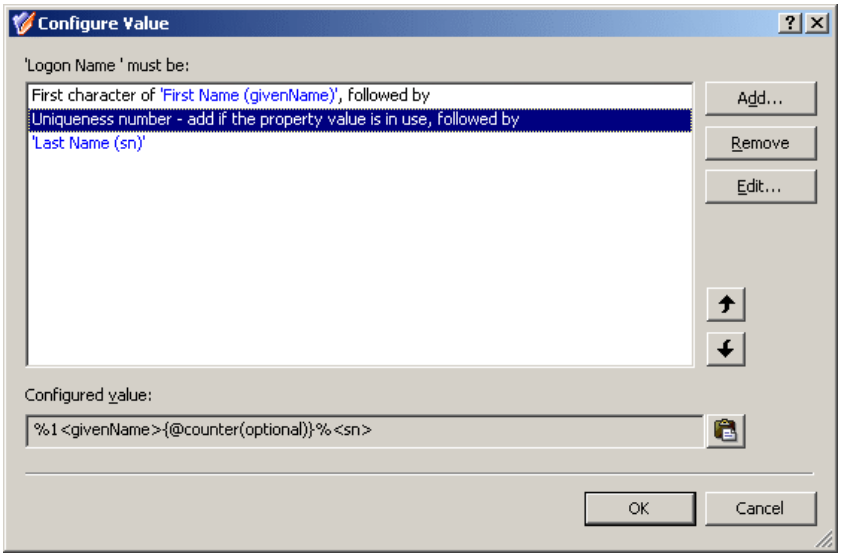
You can configure the Policy Object you need by modifying the Policy Object that implements the previous scenario; see Scenario 1: Using uniqueness number earlier in this section.

Display the **Properties** dialog box for that Policy Object and go to the **Policies** tab. Then, select the policy from the list, and click **View/Edit** to display the **User Logon Name Generation Policy Properties** dialog box.

The **Generation Rules** tab in the **User Logon Name Generation Policy Properties** dialog box looks similar to the **User Logon Name (pre-Windows 2000) Generation Rules** page in the wizard you used to configure the policy. You can use that tab to add or modify policy rules.

First, modify the rule to remove the uniqueness number entry. On the **Generation Rules** tab, select the rule and click **View/Edit** to display the **Configure Value** dialog box. Then, select the uniqueness number entry as shown in the following figure, and click **Remove**.

Figure 47: Configure Value



Click **OK** to close the **Configure Value** dialog box.

Next, configure the additional policy rules as follows.

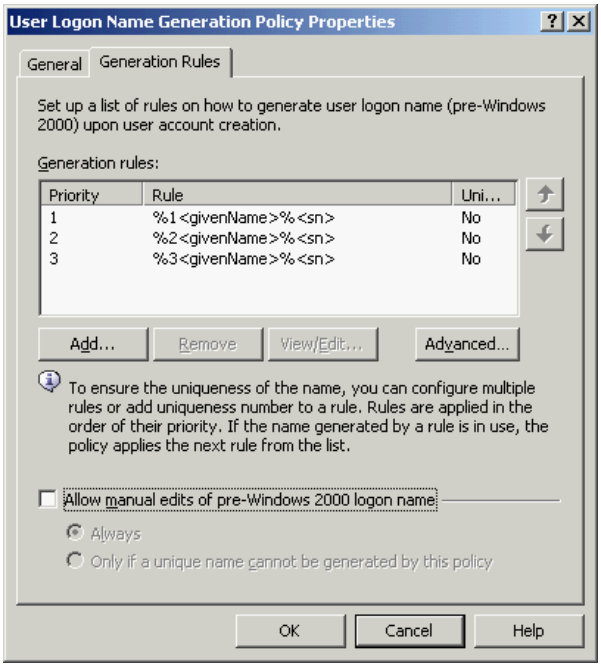
1. On the **Generation Rules** tab, click **Add** to display the **Configure Value** dialog box.
2. In the **Configure Value** dialog box, click **Add** to display the **Add Entry** window.
3. Configure the entry to include the first two character of the user first name:
   a. Under **Entry type**, click **User Property**.
   b. Under **Entry properties**, click **Select**.
   c. In the **Select Object Property** window, click **First Name** in the **Object property** list, and then click **OK**.
   d. Under **Entry properties**, click **The first**, and enter **2** in the box next to that option.
   e. Click **OK** to close the **Add Entry** window.
4. In the **Configure Value** dialog box, click **Add** to display the **Add Entry** window.
5. Configure the entry to include the user last name:
   a. Under **Entry type**, click **User Property**.
   b. Under **Entry properties**, click **Select**.
   c. In the **Select Object Property** window, click **Last Name** in the **Object property** list, and then click **OK**.
   d. Click **OK** to close the **Add Entry** window.

6. Click **OK** to close the **Configure Value** dialog box.
7. Repeat Steps 1 through 6 with the following alteration:

   In Step 3, sub-step d), enter **3** in the box next to the **The first** option.

After you complete these steps, the list of rules on the **Generation Rules** tab should look as follows:

Figure 48: Generation rules



Click **OK** to close the **User Logon Name Generation Policy Properties** dialog box.

Step 2: Applying the Policy Object

You can apply the Policy Object by using the **Scope** tab in the **Properties** dialog box for that Policy Object:

1. On the **Scope** tab, click the **Scope** button to display the **Active Roles Policy Scope** window for the Policy Object you are managing.
2. Click **Add** and select the domain, OU, or Managed Unit where you want to apply the policy.

   You can also use the **Remove** button to remove items where you want the policy to no longer be applied.
3. Click **OK** to close the **Active Roles Policy Scope** window.
4. Click **OK** to close the **Properties** dialog box for the Policy Object.

For more information on how to apply a Policy Object, see Applying Policy Objects and Managing policy scope earlier in this chapter.

Group Membership AutoProvisioning

Group Membership Autoprovisioning policies help you automate adding or removing the specified objects (such as user objects) to or from the specified groups.

In case of cloud-only Azure objects, you can use the Group Membership Autoprovisioning policy to automatically assign (or unassign) Azure users and Azure guest users to (or from) the specified O365 group(s) in the same Azure tenant.

> NOTE: Policy Object settings specific to Azure cloud-only objects (such as cloud-only Azure users, guest users, or contacts) are available only if your Active Roles deployment is licensed for managing cloud-only Azure objects. Contact One Identity support for more information.
>
> Also, Policy Objects specific to Azure cloud-only objects will work correctly only if an Azure tenant is already configured in the AD of the organization, and Active Roles is already set as a consented Azure application for that Azure tenant. For more information on these settings, see Configuring a new Azure tenant and consenting Active Roles as an Azure application.

To set up a policy, select the type of objects you want to provision, select the affected group(s), and then configure the policy rules. Once set up, the policy adds (or removes) directory objects to (or from) the selected groups depending on whether the provisioned objects meet the specified rules.

To help you get started with configuring policy-based administration in your organization, Active Roles includes a set of built-in Policy Objects that offer provisioning and deprovisioning rules to the most typical administrative use cases. To find the built-in Policy Objects, navigate to the following node of the Active Roles MMC console:

Configuration > **Policies** > **Administration** > **Builtin**

> NOTE: Active Roles does not automatically check for changes in directory objects, containers or groups specified for provisioning in the configured Policy Objects. This means that if any changes are made in any directory resources in use in a policy, you must update the impacted policies manually. For example, if a directory group used by a Group Membership AutoProvisioning Policy Group is deleted, the Policy Group must be updated manually to reflect the changes.

How this policy works

A Group Membership AutoProvisioning policy performs provisioning tasks such as adding or removing users from groups. A policy can be configured to define a list of groups and conditions so that a user account is automatically added to, or removed from, those groups depending on whether the properties of the user account meet the policy conditions.

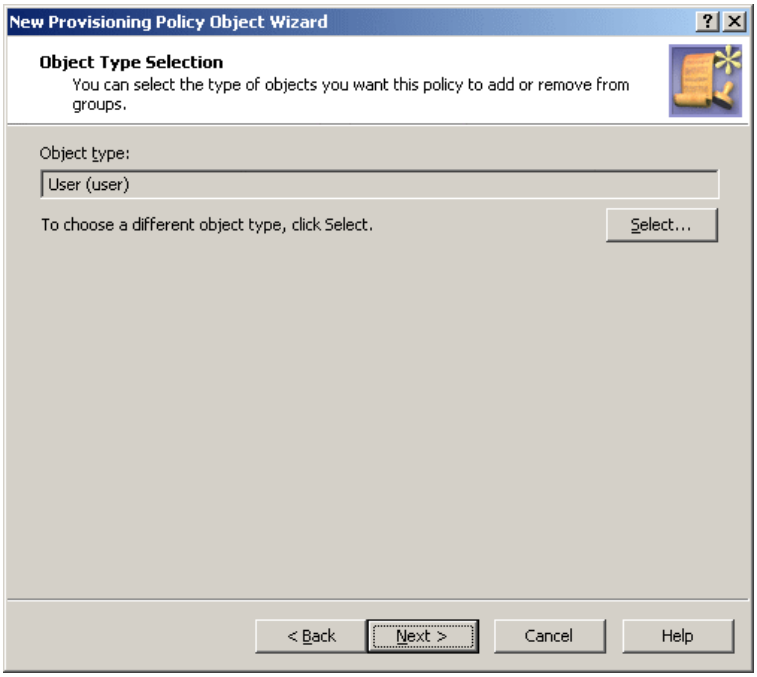
Active Roles automatically checks users against conditions, and adds or removes users from specified groups based on the check results. Although the capabilities of this policy are similar to those provided by Dynamic Groups, a Group Membership AutoProvisioning policy gives the administrator extra flexibility and control over group memberships.

Whereas the Dynamic Groups feature delivers a rules-based mechanism for managing a group membership list as a whole, a Group Membership AutoProvisioning policy allows the administrator to define membership rules on a per-user basis. This policy automates the process of adding particular users to particular groups without affecting the other members of those groups.

How to configure a Group Membership AutoProvisioning policy

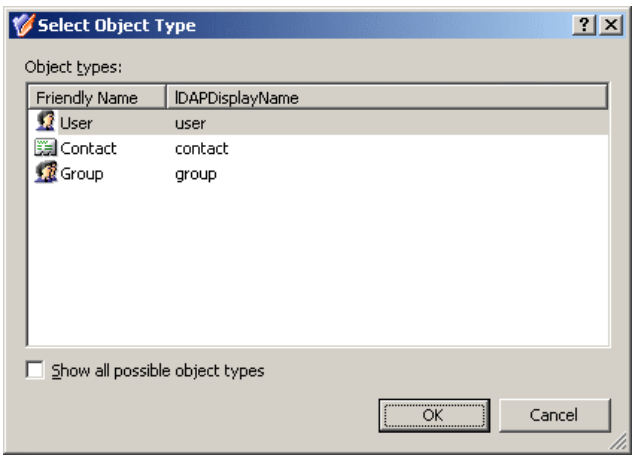
To configure a Group Membership AutoProvisioning policy, select **Group Membership AutoProvisioning** on the **Policy to Configure** page in the New Provisioning Policy Object wizard or in the Add Provisioning Policy wizard. Then, click **Next** to display the **Object Type Selection** page.

Figure 49: Object type selection



On this page, you can choose the type of objects you want the policy to add or remove from groups. By default, the object type is set to User. If you need to change this setting, click **Select** to display the **Select Object Type** dialog box.

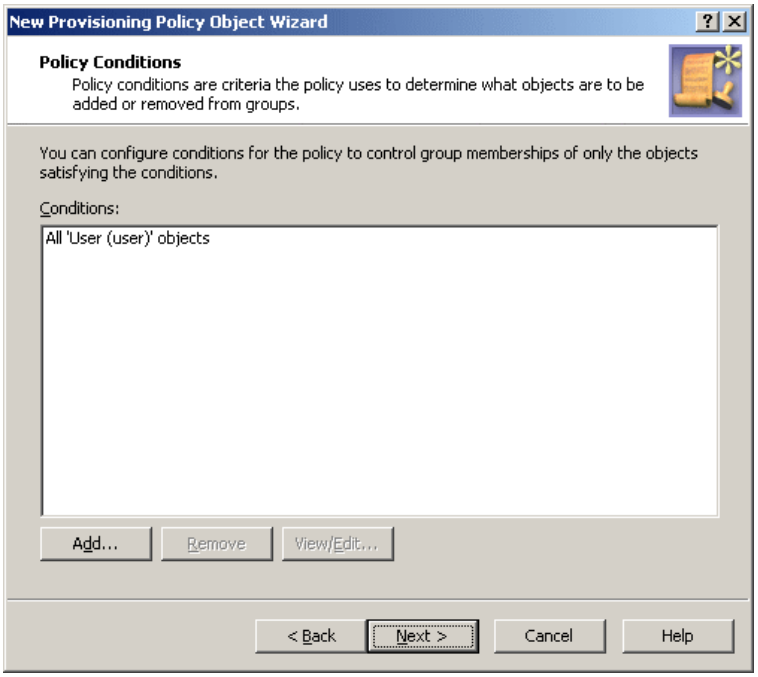
Figure 50: Object types



From the **Object types** list, select the type of objects you want the policy to control. Click **OK**.

On the **Object Type Selection** page, click **Next** to display the **Policy Conditions** page.

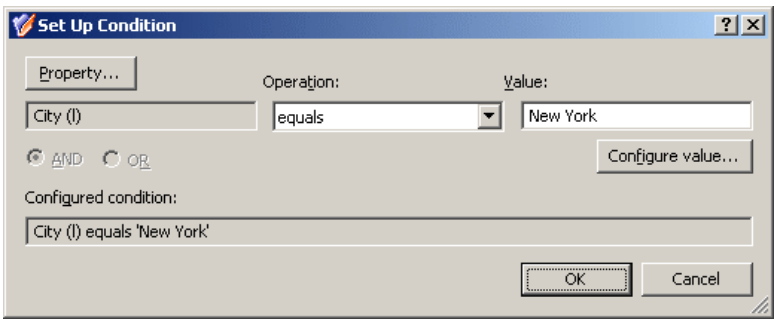
Figure 51: Policy conditions



On this page, you can set up policy conditions—criteria the policy uses to determine what objects are to be added or removed from groups. If you specify no conditions, the policy affects any objects of the type you have selected on the previous page. Otherwise, it only affects the objects matching the conditions you specify using this page.

To configure a condition, click **Add** on the **Policy Conditions** page. This displays the **Set Up Condition** dialog box.

Figure 52: Set up condition



In this dialog, you can configure a condition in the same way as you do for a Property Generation and Validation policy. A condition includes an object property (for example, **City** or **Department**), a requirement (for example, **equals** or **begins with**), and a value. The term value has the same meaning as for a Property Generation and Validation policy.

First, click the **Property** button to display the **Select Object Property** dialog box where you can select the object property you want to include in the condition.

Next, from the **Operation** list, select the requirement you want to apply to the selected property.

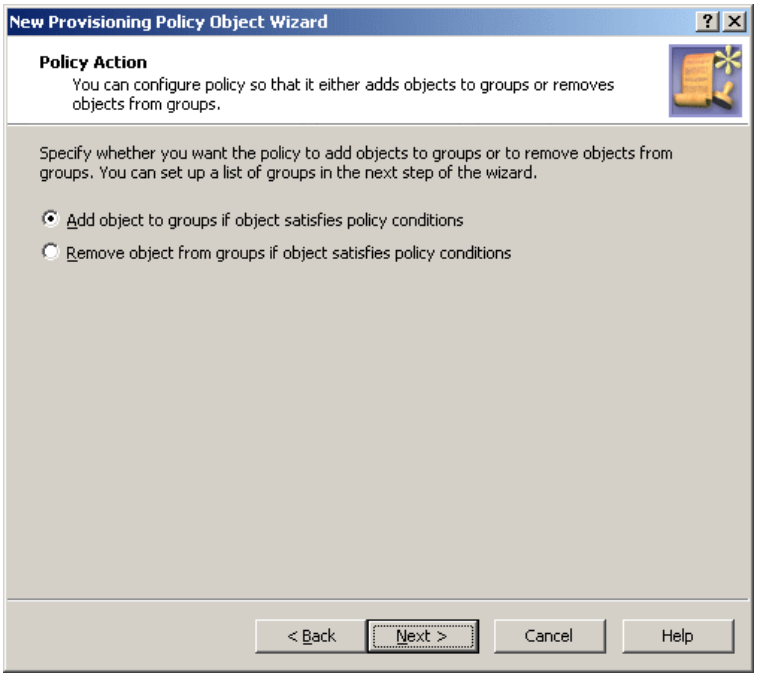
Next, click the **Configure value** button to compose the value for which you want to apply the selected requirement. This displays the **Configure Value** dialog box, discussed earlier in this chapter (see How to configure a Property Generation and Validation policy). You can use that dialog box to set up a value in the same way as for a Property Generation and Validation Policy.

If you specify multiple conditions, you can combine them with a logical AND or OR operator by clicking the **AND** or **OR** option, respectively.

Finally, click **OK** to close the **Set Up Condition** dialog box.

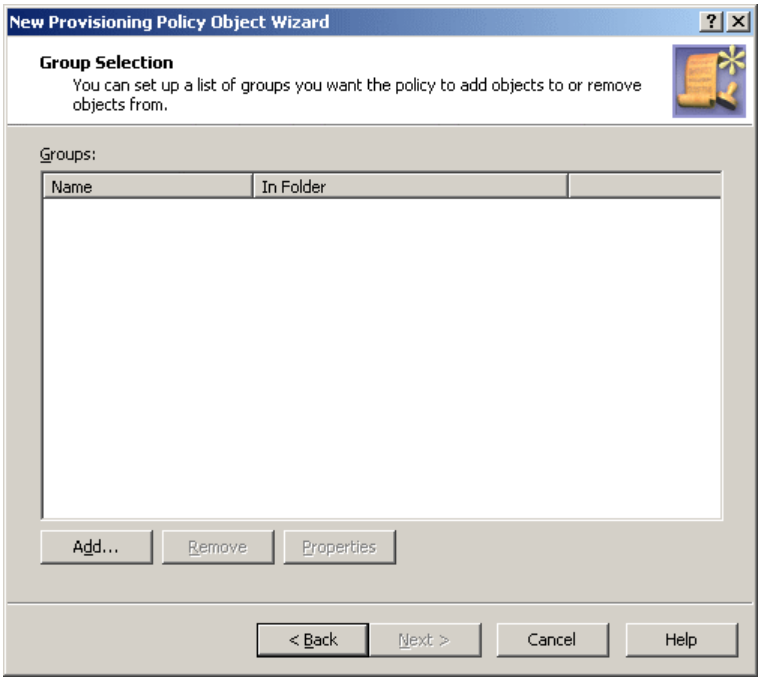
After you complete the list on the **Policy Conditions** page, click **Next** to display the **Policy Action** page.

Figure 53: Policy action



On this page, you can configure the policy to either add objects to groups or remove objects from groups. For example, if you select the option **Add object to groups if object satisfies policy conditions**, the policy populates groups with the objects that match the conditions you set up in the previous step. Click **Next** to specify the groups you want the policy to populate. This displays the **Group Selection** page.

Figure 54: Group selection



On the **Group Selection** page, you can set up a list of groups you want the policy to control. Depending on the option you select in the previous step, the policy either adds or removes objects from each of the groups you specify on this page. You can manage the list by using the **Add** and **Remove** buttons. Clicking **Add** displays the **Select Objects** dialog box to select and add groups to the list. Clicking **Remove** deletes the selected entries from the list.

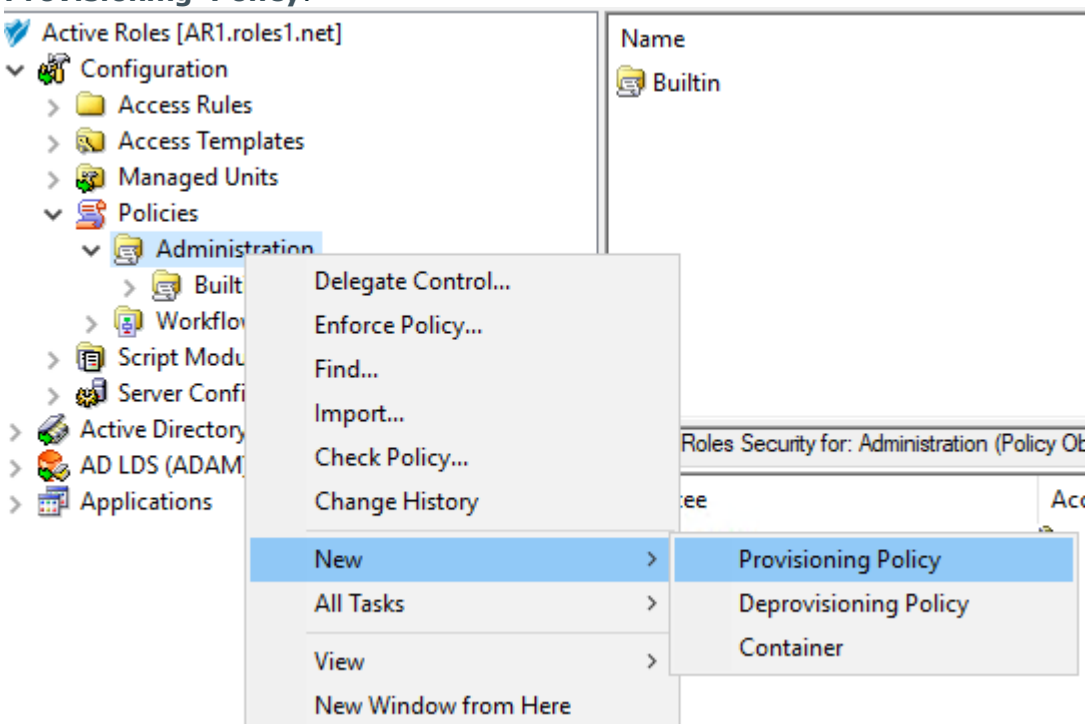
Once you have set up the list of groups, click **Next** and follow the instructions in the wizard to create the Policy Object.

Steps for configuring a Group Membership AutoProvisioning policy

To configure a Group Membership AutoProvisioning policy via the Active Roles MMC console, perform the following procedure.

To configure a Group Membership AutoProvisioning policy

1. Navigate to **Configuration** > **Policies** > **Administration**.
2. To open the **New Provisioning Policy Object Wizard** dialog, right-click in the middle pane to open the context menu, and then select **New** > **Provisioning Policy**.



3. On the **Name and Description** page, provide a unique **Name** for the new policy object. Optionally, also provide a **Description**. To continue, click **Next**.

4. On the **Policy to Configure** page, select **Group Membership AutoProvisioning**, and then click **Next**.
5. On the **Object Type Selection** page, to specify the type of object you want the policy to add or remove from groups, click **Select**, then click **OK**.

   TIP: If you do not see the object type you need, expand the list by selecting **Show all possible object types**.
6. On the **Policy Conditions** page, set up conditions that specify how the policy adds or removes the selected object types to or from groups. To create a new condition with the **Set Up Condition** dialog, click **Add**.
7. To select the object property on which you want to set up the condition, click **Property** to open the **Object property** page.
8. Select the property you want the condition to check, then click **OK**.

   TIP: If you do not see the object type you need, expand the list by selecting **Show all possible object types**.
9. In **Operation**, click the operation type you want to assign to the condition.
10. To specify additional configuration for the condition, enter a variable into the **Value** field, then click **OK** to close the **Add Value** dialog.

    Alternatively, click **Configure Value**, then click **Add**, and configure an entry manually in the **Add Entry** dialog. For more information on manual configuration, see Steps for configuring entries. To close the **Add Value** dialog, click **OK**.
11. (Optional) To modify or remove an existing condition, click **View/Edit** or **Remove** on the **Policy Conditions** page, respectively.
12. Click **Next** on the **Policy Conditions** page to continue onto the **Policy Action** page.
13. On the **Policy Action** page, specify whether you want the policy to add or remove objects if the configured conditions are met.
    - Select **Add object to groups if object satisfies policy conditions** if you want Active Roles to add the object to the specified group(s) if the configured conditions are met.
    - Select **Remove object from groups if object satisfies policy conditions** if you want Active Roles to remove the object from the specified group(s) if the configured conditions are met.

    Click **Next** to continue.
14. On the **Group Selection** page, specify the group(s) you want the policy to add the objects to (or remove from, depending on your choice on the **Policy Action** page). Click **Add** to open the **Select Objects** dialog, and then use either the **Look in:** drop-down or click **Browse** to specify the group(s). Once you are ready, click **Next** to continue.

    NOTE: Consider the following limitations when configuring a Group Membership Autoprovisioning policy for cloud-only Azure objects:

- When provisioning cloud-only Azure users or Azure guest users, you must specify an O365 Group (or O365 Groups) in this step. To do so, click **Browse** to open the **Browse for Container** dialog, and then navigate to the following node for the list of O365 Groups in the organization:

  Azure > **<azure-tenant-name>** > **Office 365 Groups**

- The Group Membership AutoProvisioning policy can only add or remove cloud-only Azure users and guest users to or from O365 Groups that are located in the same Azure tenant as the Azure users and guest users. Selecting O365 Groups located in another Azure tenant causes the configured Policy Object to not work properly.

15. On the **Enforce Policy** page, specify the objects to which the configured Policy Object will be applied. Click **Add**, and then use the **Select Objects** dialog to locate and select the objects.

    TIP: When provisioning cloud-only Azure users or guest users, you can either select the respective object category (such as the **Azure user** or **Azure guest user** node) in this step, or the **Azure tenant** that contains the Azure objects.

16. Click **Next**, and then click **Finish** to create the new policy.

Scenario: Adding users to a specified group

The policy described in this scenario automatically adds user accounts to the specified groups depending on the **Department** property of user accounts. If the **Department** property of a user account is set to **Sales**, the policy adds the account to the **Sales** group.

To implement this scenario, you must perform the following actions:

1. Create and configure a Policy Object that defines the appropriate policy.
2. Apply the Policy Object to a domain, OU, or Managed Unit.

As a result, when a user account in the container you selected in Step 2 has the **Department** property set to **Sales**, Active Roles automatically adds that account in the **Sales** group.

The following two sections elaborate on the steps to implement this scenario.

Step 1: Creating and configuring the Policy Object

You can create and configure the Policy Object you need by using the New Provisioning Policy Object wizard. For information about the wizard, see Creating a Policy Object in the Policy Object management tasks section earlier in this chapter.

To configure the policy, click **Group Memberships AutoProvisioning** on the **Select Policy Type** page of the wizard. Then, click **Next**.

On the **Object Type Selection** page, click **Next** to accept the default setting for the object type—User.

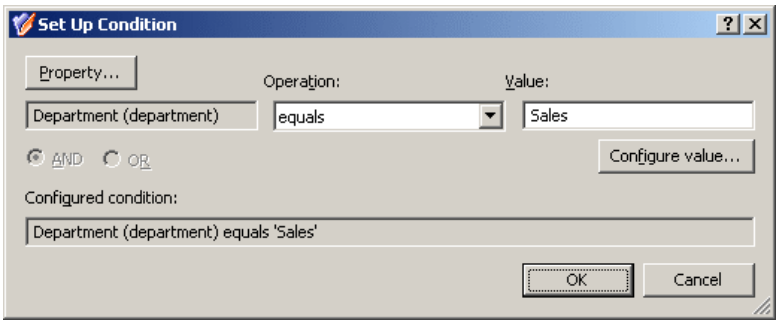
On the **Policy Conditions** page, click **Add** to display the **Set Up Condition** dialog box.

Configure the condition as follows:

1. Click the **Property** button; then, select the **Department** property and click **OK**.
2. In the **Value** box, type **Sales**.

After you complete these steps, the **Set Up Condition** dialog box looks as shown in the following figure.

Figure 55: Set up condition



Click **OK** to close the **Set Up Condition** dialog box.

On the **Policy Conditions** page, click **Next**.

On the **Policy Action** page, click **Add object to groups if object satisfies policy conditions**, and then click **Next**.

On the **Group Selection** page, click **Add** and use the **Select Objects** dialog box to locate the **Sales** group. After you add the **Sales** group to the list on the **Group Selection** page, click **Next** and follow the instructions in the wizard to create the Policy Object.

Step 2: Applying the Policy Object

You can apply the Policy Object by using the **Enforce Policy** page in the New Provisioning Policy Object wizard, or you can complete the wizard and then use the **Enforce Policy** command on the domain, OU, or Managed Unit where you want to apply the policy.

For more information on how to apply a Policy Object, see Applying Policy Objects and Managing policy scope earlier in this chapter.

Email Alias Generation

Policies in this category are intended to automate the assignment of the e-mail alias when designating a user as mailbox-enabled on Microsoft Exchange Server. By default, Microsoft Exchange Server provides for the following recipient e-mail address format: <email alias>@<domain name>

You can use pre-defined rules to generate e-mail aliases, or configure custom rules. For example, you can configure a policy to compose the e-mail alias of the first initial followed by the last name of the user. Custom rules provide for the addition of an incremental

numeric value to ensure uniqueness of the alias. You can also specify whether the alias can be modified by the operator who creates or updates the user account.

How this policy works

When making a user mailbox-enabled, Active Roles relies on this policy to assign a certain e-mail alias to the user account. The policy generates the alias based on user properties, such as the pre-Windows 2000 user logon name, first name, initials, and last name. A custom rule can be configured to use other properties.

A custom rule can also be configured to add so-called uniqueness number. A uniqueness number is a numeric value the policy includes into the alias, incrementing that value in the event of an alias naming conflict. For example, the policy can automatically change the generated alias from John.Smith to John1.Smith if a mailbox with the alias John.Smith already exists. If the alias John1.Smith is also in use, the new alias will be changed to John2.Smith, and so on.

The policy configuration provides the option to allow or disallow manual edits of policy-generated aliases. Permission to modify a policy-generated alias can be restricted to the case where the alias is in use by another mailbox.

Some specific features of the policy behavior are as follows:

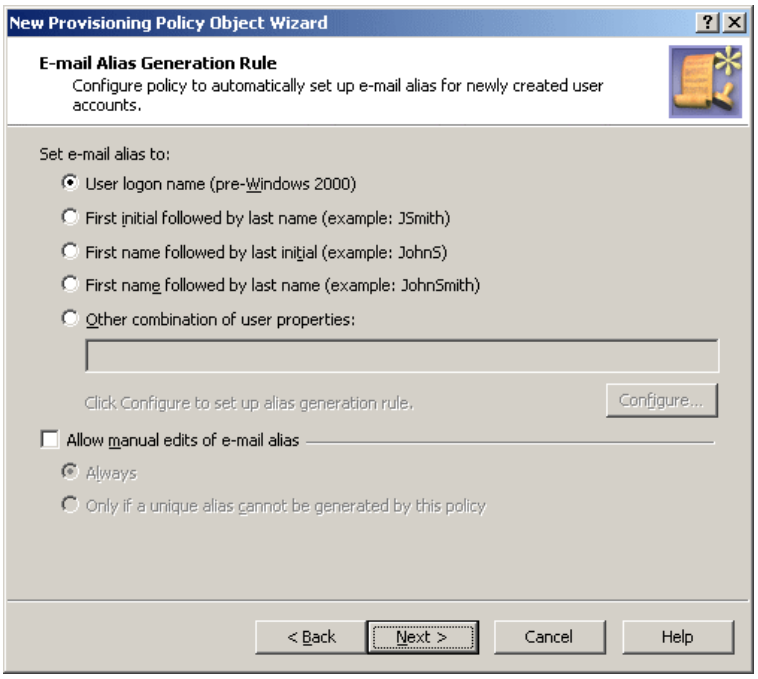
- With a rule that does not use a uniqueness number, Active Roles simply attempts to assign the generated alias to the user account. The operation may fail if the generated alias is not unique, that is, the alias is already assigned to a different user account. If the policy allows manual edits of policy-generated aliases, the alias can be corrected by the operator who creates the user account.
- With a custom rule that uses a uniqueness number, Active Roles adds a button at the client side, next to the **Alias** field on the user creation and modification forms.

  To generate an alias, the client user (operator) must click that button, which also applies if the generated alias is in use. Clicking the **Generate** button increases the uniqueness number by one, thereby allowing the alias to be made unique.
- With a custom rule configured to include user properties that are normally not displayed on the user creation forms, an extra page is added to the New Object - User wizard in the Active Roles console, thus making it possible to specify the user properties required to generate the alias.
- The policy defines a list of characters that are unacceptable in e-mail aliases. Space characters and the following characters are not accepted : @ * + | = \ ; : ? [ ] , < > /
- The policy denies processing of operation requests that assign the empty value to the e-mail alias.
- When checking user accounts for Active Roles policy compliance (described later in this document), Active Roles detects, and reports on, the aliases that are not set up as prescribed by the alias generation policy.

How to configure an E-mail Alias Generation policy

To configure an E-mail Alias Generation policy, select **E-mail Alias Generation** on the **Policy to Configure** page in the New Provisioning Policy Object wizard or in the Add Provisioning Policy wizard. Then, click **Next** to display the **E-mail Alias Generation Rule** page:

Figure 56: E-mail Alias Generation Rule



On the **E-mail Alias Generation Rule** page, you can select a pre-configured rule or create a custom alias-generation rule. The first four options on the page are self-explanatory. For example, the first option makes the e-mail alias the same as the user logon name (pre-Windows 2000). The option **Other combination of user properties**, discussed later in this section, allows you to configure a custom rule, including the addition of uniqueness number.

By selecting the **Allow manual edits of e-mail alias** check box, you authorize the operator who creates or updates the user account to make changes to the policy-generated alias. If this check box is cleared, Active Roles displays the **Alias** field as read-only on the user creation and modification forms.

By selecting the **Always** option, you authorize the operator to modify the alias at their discretion. With the **Only if a unique alias cannot be generated by this policy** option, you limit manual changes to the situation where a unique alias cannot be generated in accordance with the policy rules.

Configuring a custom generation rule

To configure a custom rule, click **Other combination of user properties**, and then click the **Configure** button. This displays the **Configure Value** dialog box, discussed earlier in this chapter (see How to configure a Property Generation and Validation policy). You can use that dialog box to set up a value for the **'Alias' must be** condition, the same way you configure a Property Generation and Validation policy.

To start configuring a value, click **Add** in the **Configure Value** dialog box. This displays the **Add Entry** window.

A value is a concatenation of one or more entries. In the **Add Entry** window, you can select the type of the entry to add, and then configure the entry. The following table summarizes the available types of entries.

Table 20: Available entries

Type of entry	Description
Text	Adds a text string to the value.
Uniqueness Number	Adds a numeric value the policy will increment in the event of an alias naming conflict.
User Property	Adds a selected property (or a part of a property) of the user account to which the policy will assign the alias.
Parent OU Property	Adds a selected property (or a part of a property) of an organizational unit in the hierarchy of containers above the user account to which the policy will assign the alias.
Parent Domain Property	Adds a selected property (or a part of a property) of the domain of the user account to which the policy will assign the alias.

Instructions on how to configure an entry depend on the type of the entry. For each type of an entry, you can find the instructions earlier in this chapter:

- **Text**. Refer to the Entry type: Text subsection in the How to configure a Property Generation and Validation policy section.
- **Uniqueness Number**. Refer to the Entry type: Uniqueness Number subsection in the How to configure a User Logon Name Generation policy section.
- **User Property**. Refer to the Entry type: <Object> Property subsection in the How to configure a Property Generation and Validation policy section.
- **Parent OU Property**. Refer to the Entry type: Parent OU Property subsection in the How to configure a Property Generation and Validation policy section.
- **Parent Domain Property**. Refer to the Entry type: Parent Domain Property subsection in the How to configure a Property Generation and Validation policy section.

When you are done configuring a value, click **OK** to close the **Configure Value** dialog box. This will add the value to the policy rule. If necessary, you can modify the value by clicking

button and then managing the list of entries in the **Configure Value** dialog box.

When you are done configuring the policy rule, click **Next** on the **E-mail Alias Generation Rule** page and follow the instructions in the wizard to create the Policy Object.

Steps for configuring an E-mail Alias Generation policy

To configure an E-mail Alias Generation policy

1. On the **Policy to Configure** page, select **E-mail Alias Generation**, and then click **Next**.
2. On the **E-mail Alias Generation Rule** page, do the following:
   - Select one of the pre-configured generation rules, or create a custom alias-generation rule. To create a custom rule, click **Other combination of user properties**, click **Configure**, and complete the **Configure Value** dialog box by using the procedure outlined later in this topic.
   - If you want the e-mail alias to be allowed for manual edit, select **Allow manual edits of e-mail alias**. Then, do one the following:
     - Click **Always** to authorize the operator who creates or updates the user account to modify the e-mail alias.
     - Click **Only if a unique alias cannot be generated by this policy** to allow manual changes only in the situation where a policy-generated alias is already assigned to a different user account.

   Click **Next**.
3. On the **Enforce Policy** page, you can specify objects to which this Policy Object is to be applied:
   - Click **Add**, and use the **Select Objects** dialog box to locate and select the objects you want.
4. Click **Next**, and then click **Finish**.

To complete the Configure Value dialog box

1. Click **Add**.
2. Configure an entry to include in the value (for instructions, see Steps for configuring entries).
3. In the **Configure Value** dialog box, add more entries, delete or edit existing ones, and then click **OK**.

Scenario: Generating e-mail alias based on user names

The policy described in this scenario generates the e-mail alias in accordance with this rule: user first name, optionally followed by a three-digit uniqueness number, followed by a period, followed by the user last name. Examples of aliases generated by this rule are as follows:

- John.Smith
- John001.Smith
- John002.Smith

The policy generates the alias John001.Smith for the user John Smith if the alias John.Smith is in use. If both John.Smith and John001.Smith are in use, the policy generates the alias John002.Smith, and so on.

To implement this scenario, you must perform the following actions:

1. Create and configure the Policy Object that defines the appropriate policy.
2. Apply the Policy Object to a domain, OU, or Managed Unit.

As a result, when assigning an e-mail alias to a user account in the container you selected in Step 2, the Active Roles user interfaces provide a **Generate** button to create the alias in accordance with the policy rule. In the event of an alias naming conflict, clicking the **Generate** button causes the policy to add a uniqueness number to the alias.

The following two sections elaborate on the steps to implement this scenario.

Step 1: Creating and configuring the Policy Object

You can create and configure the Policy Object you need by using the New Provisioning Policy Object wizard. For information about the wizard, see Creating a Policy Object in the Policy Object management tasks section earlier in this chapter.

To configure the policy, click **E-mail Alias Generation** on the **Select Policy Type** page of the wizard. Then, click **Next**.

On the **E-mail Alias Generation Rule** page, click **Other combination of user properties**, and then click **Configure**.

Complete the **Configure Value** dialog box as follows:

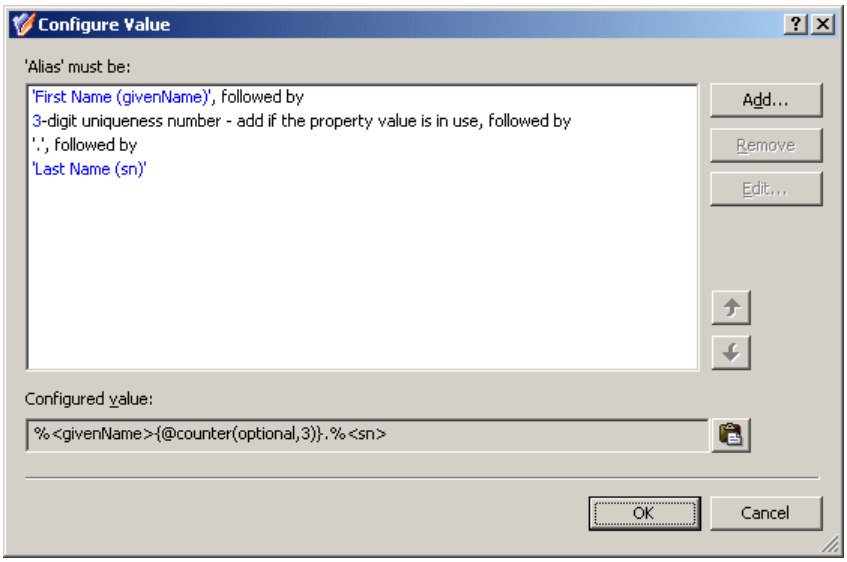
1. Click **Add**.
2. Configure the entry to include the user first name:
   a. Under **Entry type**, click **User Property**.
   b. Under **Entry properties**, click **Select**.
   c. In the **Select Object Property** window, click **First Name** in the **Object**

list, and then click **OK**.

d. Click **OK**.

3. Click **Add**.
4. Configure the entry to optionally include a uniqueness number:
   a. Under **Entry type**, click **Uniqueness Number**.
   b. Under **Entry properties**, set the entry options:
      - Click **Add if the property value is in use**.
      - Select the **Fixed-length number, with leading zeroes** check box.
      - In the box next to **Length of the number, in digits**, type **3**.
   c. Click **OK**.
5. Click **Add**.
6. Configure the entry to include the period character:
   a. In **Text value** under **Entry properties**, type the period character.
   b. Click **OK**.
7. Click **Add**.
8. Configure the entry to include the user last name:
   a. Under **Entry type**, click **User Property**.
   b. Under **Entry properties**, click **Select**.
   c. In the **Select Object Property** window, click **Last Name** in the **Object property** list, and then click **OK**.
   d. Click **OK**.

After you complete these steps, the list of entries in the **Configure Value** dialog box should look like the following figure.



9. Click **OK** to close the **Configure Value** dialog box. Then, click **Next** and follow the instructions in the wizard to create the Policy Object.

Step 2: Applying the Policy Object

You can apply the Policy Object by using the **Enforce Policy** page in the **New Provisioning Policy Object** wizard, or you can complete the wizard and then use the **Enforce Policy** command on the domain, OU, or Managed Unit where you want to apply the policy.

For more information on how to apply a Policy Object, see Applying Policy Objects and Managing policy scope earlier in this chapter.

Exchange Mailbox AutoProvisioning

Policies in this category are intended to automate the selection of a mailbox store or database when designating a user as mailbox-enabled or creating a mailbox on Microsoft Exchange Server.

You can specify Exchange Servers and mailbox stores or databases where mailbox creation is allowed, and specify rules to distribute mailboxes among multiple stores. For example, you can configure a policy to automatically choose a store that holds the least number of mailboxes.

How this policy works

When making a user mailbox-enabled or creating a mailbox, Active Roles relies on this policy to select the mailbox store or database. The policy defines a single store, or a set of stores, in which creation of mailboxes is allowed. Some specific features of the policy behavior are as follows:

- If the policy specifies a single store, mailboxes are created in that store. A different store cannot be selected by the operator who creates or updates the user account.
- If the policy specifies multiple stores, the store is selected either automatically (by Active Roles) or manually (by the operator who creates or updates the user account), depending on policy options.

In case of multiple stores, the policy provides these options to govern the selection of a store:

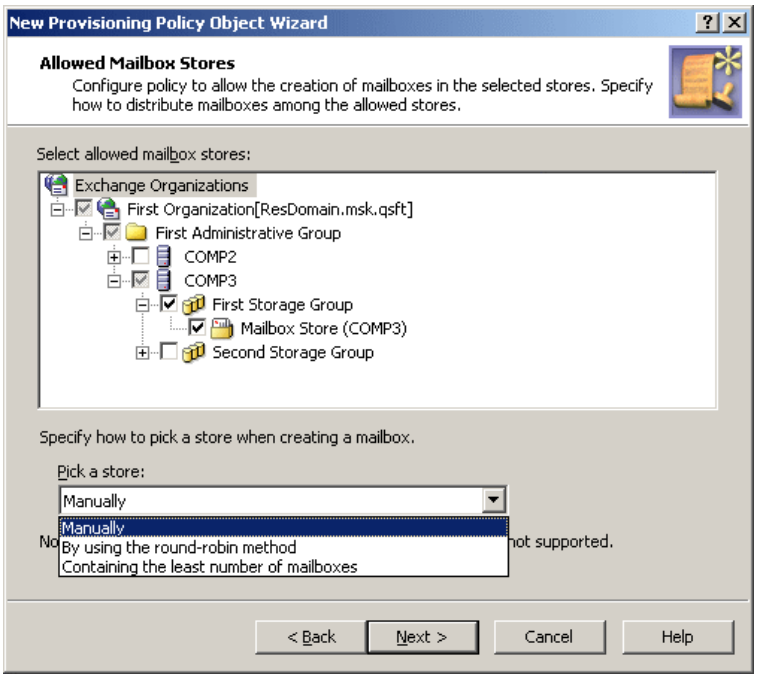
- **Manually**. Allows the operator to select a store from the list defined by the policy.
- **By using the round-robin method**. Redirects mailbox creation requests sequentially across the stores, selecting the first store for the first request, the second store for the second request and so on. After the last store is reached, the next request is passed to the first store in the sequence.

- **Containing the least number of mailboxes**. Forwards mailbox creation requests to the store that holds the least amount of mailboxes.

How to configure an Exchange Mailbox AutoProvisioning policy

To configure an Exchange Mailbox AutoProvisioning policy, select **Exchange Mailbox AutoProvisioning** on the **Policy to Configure** page in the New Provisioning Policy Object wizard or in the Add Provisioning Policy wizard. Then, click **Next** to display the **Allowed Mailbox Stores** page.

Figure 57: Allowed mailbox stores



On this page, you can select the servers and mailbox stores or databases to be allowed for mailbox creation. Select mailbox stores from a single Exchange organization. If you select multiple stores, you can specify how to choose a store upon a mailbox creation request. From the **Pick a store** list, select one of these options:

- Manually
- By using the round-robin method
- Containing the least number of mailboxes

When you are done, click **Next** and follow the instruction in the wizard to create the Policy Object.

Steps for configuring an Exchange Mailbox AutoProvisioning policy

To configure an Exchange Mailbox AutoProvisioning Policy

1. On the **Policy to Configure** page, select **Exchange Mailbox AutoProvisioning**, and then click **Next**.
2. Under **Select allowed mailbox stores**, select servers and stores to be allowed for mailbox creation, and then click **Next**.

   In case of multiple stores, select a method of picking a store from the **Pick a store** list. For information about the methods of picking a store in case of multiple stores, see How this policy works earlier in this chapter.
3. On the **Enforce Policy** page, you can specify objects to which this Policy Object is to be applied:
   - Click **Add**, and use the **Select Objects** dialog box to locate and select the objects you want.
4. Click **Next**, and then click **Finish**.

Scenario: Mailbox store load balancing

The policy described in this scenario allows multiple stores to be used for mailbox creation, and forces Active Roles to automatically select the store that holds the least amount of mailboxes.

To implement this scenario, you must perform the following actions:

1. Create and configure a Policy Object that defines the appropriate policy.
2. Apply the Policy Object to a domain, OU, or Managed Unit.

As a result, when creating a mailbox for a user account that resides in the container you selected in Step 2, Active Roles chooses the least loaded store among those where mailbox creation is allowed.

The following two sections elaborate on the steps to implement this scenario.

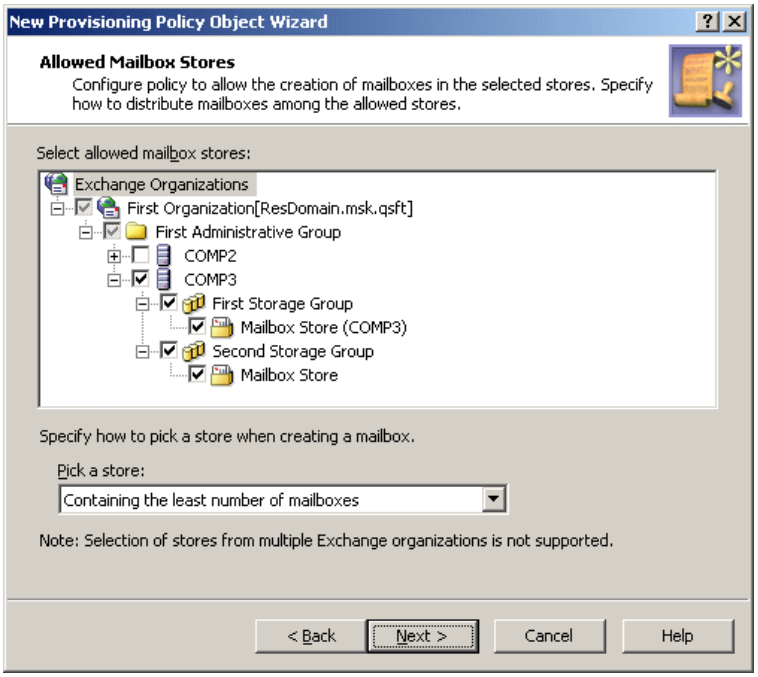
Step 1: Creating and configuring the Policy Object

You can create and configure the Policy Object you need by using the New Provisioning Policy Object wizard. For information about the wizard, see Creating a Policy Object in the Policy Object management tasks section earlier in this chapter.

To configure the policy, click **Exchange Mailbox AutoProvisioning** on the **Select Policy Type** page of the wizard. Then, click **Next**.

On the **Allowed Mailbox Stores** page, select the stores in which you want mailbox creation to be allowed. Then, under **Pick a store**, click **Containing the least number of mailboxes**, as shown in the following figure.

Figure 58: Allowed mailbox stores



Click **Next**, and follow the instructions in the wizard to create the Policy Object.

Step 2: Applying the Policy Object

You can apply the Policy Object by using the **Enforce Policy** page in the New Provisioning Policy Object wizard, or you can complete the wizard and then use the **Enforce Policy** command on the domain, OU, or Managed Unit where you want to apply the policy.

For more information on how to apply a Policy Object, see Applying Policy Objects and Managing policy scope earlier in this chapter.

Default creation options for Exchange mailbox

In the wizard for creating user accounts, whether in the Active Roles console or Web Interface, the **Create an Exchange mailbox** option is selected by default, causing the user mailbox to be created upon creation of a user account. This behavior can be changed by applying an appropriately crafted policy of the Exchange Mailbox AutoProvisioning category.

A policy can be configured so that the **Create an Exchange mailbox** option is not selected by default but the administrator who uses the wizard to create a user account can

select that option if necessary. It is also possible to configure a policy that forces the **Create an Exchange mailbox** option to be selected.

To set default creation options for Exchange mailbox

1. Create a Policy Object containing an Exchange Mailbox AutoProvisioning policy.
2. Open the **Properties** dialog box for the Policy Object you created.
3. On the **Policies** tab in the **Properties** dialog box, double-click the Exchange Mailbox AutoProvisioning policy entry.
4. On the **Mailbox Creation** tab in the **Exchange Mailbox AutoProvisioning Policy Properties** dialog box, set policy options as appropriate for your situation:
   - **Create the user mailbox by default**. Determines whether the **Create an Exchange mailbox** option is selected by default in the wizard for creating user accounts. If you want user mailboxes not to be created by default, unselect this policy option.
   - **Enforce creation of the mailbox**. Causes the **Create an Exchange mailbox** option to be selected and unavailable so that the administrator who creates a user account cannot unselect that option.
5. Click **OK** to close the dialog boxes you opened.
6. Apply the Policy Object to the scope (domains, containers, or Managed Units) where you want this policy to be in effect.

AutoProvisioning for SaaS products

Policies of this category are intended to automate the provisioning of users and groups in the selected SaaS products using Starling Connectors.

You can specify the Starling Connect connectors to be validated for the users or groups for which the policy is applied.

How this policy works

Active Roles relies on this policy during user creation to provision the users for connected systems based on the registered Starling Connectors that are selected based on the configured policy.

Create Provisioning policy for Starling Connect

To create a Policy Object for Starling Connect

1. In the console tree, under **Configuration** | **Policies** | **Administration**, locate and select the folder in which you want to add the Policy Object.

   You can create a new folder as follows: Right-click **Administration** and select **New** | **Container**. Similarly, you can create a sub-folder in a folder: Right-click the folder and select **New** | **Container**.
2. Right-click the folder, point to **New**, and then click **Provisioning Policy**.
3. On the **Welcome** page of the wizard, click **Next**.
4. On the **Name and Description** page, do the following, and then click **Next**:
   a. In the **Name** box, type a name for the Policy Object.
   b. Under **Description**, type any optional information about the Policy Object.
5. On the **Policy to Configure** page, select **Autoprovisioning in SaaS products**, and click **Next** to configure policy settings.
6. On the Object Type Selection page, click **Select**.
   a. On the **Select Object Type**, from the Object types list, select **User** or **Group**, and click **OK**.
   b. Click **Next**.
   c. On the **Policy Conditions** page, from the **Starling Connect Connectors** list, select the connectors to be provisioned for the user or group as part of the policy. Click **Next**.
7. On the **Enforce Policy** page, you can specify the containers on which this Policy Object is to be applied:
   a. Click **Add**, and use the **Select Objects** to locate and select the objects you want.
   b. Click **Next**.
8. Click **Finish**.

IMPORTANT: Starling Connect policy have to be applied on the container for any SaaS operations to take place.

SaaS operations for each connector may vary from each other. Each connector may have a set of mandatory attributes to perform any operation.

The operation will fail in case any of the mandatory attributes are missing in the particular request. The notification will report the information of all the mandatory attributes missing in that event which caused the failure.

In that case, you must create the corresponding virtual attributes, customize the Web Interface to enter the value for the virtual attribute during the specified operation. Using this approach, the attribute value is passed as a part of the request.

OneDrive Provisioning

Policies of this category are intended to provision access to OneDrive for Azure AD users. Provisioning of OneDrive is controlled or restricted by creating a new provisioning policy and applying the policy to the Organizational Unit.

How this policy works

Active Roles relies on this policy during user creation to provision Azure AD users for OneDrive access.

Creating provisioning policy for OneDrive

Provisioning access to OneDrive for Azure AD users is controlled or restricted by creating a new provisioning policy and applying the policy to the Organizational Unit.

To create and apply the new policy

1. From the Active Roles Console, create a Policy Object. For instructions on creating a policy object, see the section **Creating a Policy Object**, in the *Active Roles Administration Guide*.
2. In Active Roles Console, on the **Policy to Configure** page, select **OneDrive Provisioning**.
3. In the New Provisioning Policy Object Wizard > OneDrive folder Management page, enter the SharePoint Admin URL and the storage size, and click **Next**.

   NOTE:

   - If the policy conditions are not satisfied, such as an incorrect SharePoint Admin URL or a storage size that is not within the acceptable range, an error is displayed.
   - Policy accepts a minimum storage size of 1GB and it can span up to a maximum of 10TB.

4. In the **Enforce Policy** page, select the Organizational Unit (OU) on which the policy must be applied.
5. Click **Next**.
6. Click **Finish**.

Home Folder AutoProvisioning

Policies in this category are intended to automate the creation or renaming of user home folders and home shares upon user accounts creation or renaming through Active Roles.

You can specify a server on which to create home folders and home shares, define how to set permissions for new home folders and shares, specify naming conventions for new home folders and home shares, and limit the number of concurrent connections to home shares.

For example, using this type of policy, a corporate rule can be defined so that every time Active Roles creates a user account, it also creates a folder on a network file share, and assigns it as the user's home folder.

How this policy works

When executing a Home Folder AutoProvisioning policy, Active Roles performs various actions depending on whether a user is created, copied, or renamed.

Creating home folders and shares when creating user accounts

When Active Roles creates a user account (whether from scratch or by copying an existing account), the policy can cause Active Roles to create a home folder and, optionally, a home share for the account using the path specified in the policy. The name of the home share is composed of the user name, and the prefix and suffix specified in the policy.

The policy provides the option to enable creation of home folders with paths and names that differ from the path and name prescribed by the policy. For example, a Property Generation and Validation policy can be configured to generate the **Home Drive** and **Home Directory** properties on user accounts. When making changes to those properties, Active Roles verifies that the specified home folder exists, and creates the home folder if necessary.

A special policy is implemented in Active Roles that restricts the folders on the network file shares in which home folders can be created. The Policy Object containing that policy is located in the **Configuration/Policies/Administration/Builtin** container. The name of the Policy Object is **Built-in Policy - Home Folder Location Restriction**. You can access it by using the Active Roles console. The policy settings include a list of the folders on the network file shares in which creation of home folders is allowed. For instructions on how to view or modify that list, see Configuring the Home Folder Location Restriction policy later in this section.

Renaming home folders when renaming user accounts

When Active Roles modifies the user logon name (pre-Windows 2000) of a user account, the policy can rename the home folder and, optionally, re-create the home share for that

user account. The name of the new home share is set up in accordance with the naming convention specified in the policy.

The policy renames the existing home folder based on the new user logon name (pre-Windows 2000). However, if the home folder is in use, Active Roles cannot rename the folder. In this case, Active Roles creates a new home folder with the new name and does not affect the existing home folder.

Option to prevent operation on file server

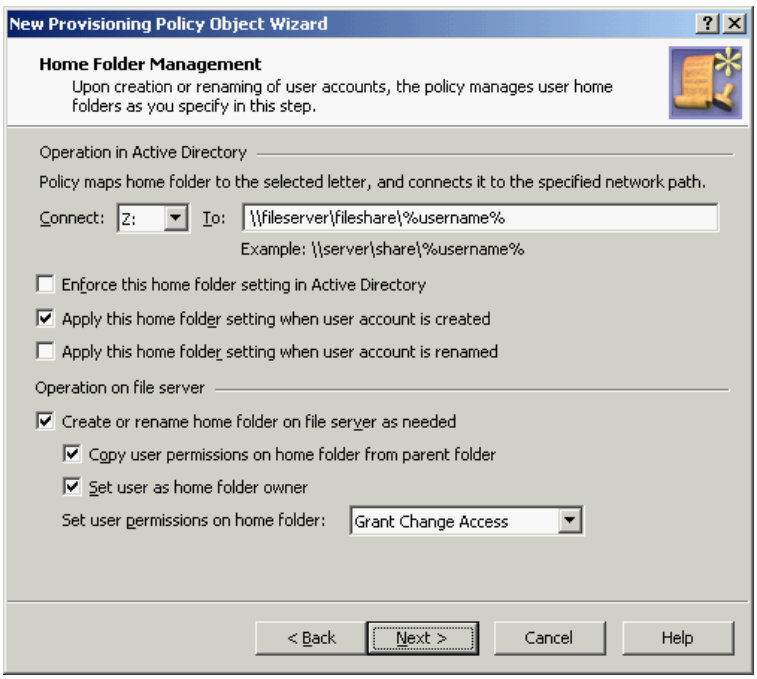
By default, Active Roles attempts to create or rename a (non-local) home folder on the file server when the Home Directory property is set or modified on a user account in Active Directory. If creation or renaming of the home folder fails (for example, because the file server is inaccessible), then the creation or modification of the user account fails, as well. To prevent such an error condition, a Home Folder AutoProvisioning policy can be configured so that Active Roles applies the changes to the **Home Drive** and **Home Directory** properties in Active Directory without attempting an operation on the file server. This policy option enables the use of a tool other than Active Roles for creating home folders on the file server.

Active Roles comes with a pre-configured Policy Object that allows the creation or renaming of home folders when setting home folder properties on user accounts in Active Directory. The Policy Object is located in the **Configuration/Policies/Administration/Builtin** container in Active Roles console tree. The name of the Policy Object is **Built-in Policy - Default Rules to Provision Home Folders**. If you want to prevent Active Roles from attempting to create or rename home folders, you can modify the policy in the built-in Policy Object or configure and apply another Home Folder AutoProvisioning policy with the respective option turned off.

How to configure a Home Folder AutoProvisioning policy

To configure a Home Folder AutoProvisioning policy, select **Home Folder AutoProvisioning** on the **Policy to Configure** page in the New Provisioning Policy Object wizard or in the Add Provisioning Policy wizard. Then, click **Next** to display the **Home Folder Management** page.

Figure 59: Home folder management



On this page, you can configure the following options.

Connect <Drive Letter> To <Network Path>

Upon creation or renaming of a user account, the policy can configure the user account in Active Directory to connect the home folder to the specified network path. From the **Connect** list, select the drive letter to which you want the policy to map the home folder. In the **To** box, specify a network path to the home folder. Ensure that the path meets the following requirements:

- A valid network path must begin with the UNC name of a network file share, such as \\Server\Share\, and should normally include the %username% notation. For example, with the **Connect: Z: To: \\Server\Share\%username%** option, the policy can configure a user account in Active Directory so that the **Home Drive** property of the user account is set to Z: and the **Home Directory** property of the user account is set to \\Server\Share\LogonName where LogonName stands for the pre-Windows 2000 logon name of the user account.
- The path must include a common share at one level above the home folders. For example, if you type **\\Comp\Home\%username%**, the policy creates home folders on the share **Home** on the server **Comp**, with the name of the folder being the same as the user logon name (pre-Windows 2000). The path **\\Comp\%username%** is invalid.
- The folder on the network file share in which you want the policy to create home folders must be listed in the Home Folder Location Restriction policy. For instructions on how to view or modify the list see Configuring the Home Folder Location Restriction policy later in this section.

- If you want the policy to create home shares (see information about the **Home Share Management** page later in this section), you should not specify an administrative share, such as C$, as the common share in the **To** box. Otherwise, the policy may be unable to create home shares when creating home folders. Thus, if you specify **\\Comp\C$\%username%**, the policy can successfully create home folders in the folder **C:** on the computer **Comp**, but it may fail to create home shares.

Enforce this home folder setting in Active Directory

Use this option to have Active Roles verify whether the **Home Drive** and **Home Directory** properties on user accounts in Active Directory are in compliance with the **Connect:** <drive letter> **To:** <network path> setting specified by this policy.

For example, with the **Connect: Z: To: \\Server\Share\%username%** policy setting, this option causes a policy violation condition in Active Roles upon an attempt to modify a user account so that the **Home Drive** property is assigned a drive letter other than Z: or the **Home Directory** property is assigned a network path other than \\Server\Share\LogonName where LogonName stands for the pre-Windows 2000 logon name of the user account.

When this option is turned off, the policy allows a home folder path and name that differs from the path and name prescribed by this policy. A Property Generation and Validation policy can be configured to generate the **Home Drive** and **Home Directory** properties on user accounts, or those properties can be specified manually. In either case, Active Roles updates the user account so that the folder with the specified path and name is set as the user home folder. If necessary, Active Roles creates the folder.

When this option is turned on, the policy behaves as follows:

- It ensures that the path and name of the home folder is in compliance the policy settings. If a different path or name is specified upon creation or modification of a user account, the policy does not allow the changes to the home folder path and name to be committed to the directory.
- The **Check Policy** command causes the policy to verify the existing home folder settings. The policy check results inform about policy violations, if any, and provide the ability to fix the home folder path and name settings on user accounts so as to bring them into compliance with the policy settings.

By selecting the **Enforce this home folder setting in Active Directory** check box, you ensure that the home folders on user accounts are set in compliance with this policy.

By clearing the check box, you get the option of applying a Property Generation and Validation policy in order to generate and validate the Home Drive and Home Directory properties, and thus have Active Roles create and assign home folders in accordance with the flexible, highly customizable rules provided by a Property Generation and Validation policy.

IMPORTANT: When setting the **Home Drive** and **Home Directory** properties, Active Roles does not create the home folder if the network path of the folder to hold the home folder is not listed in the Home Folder Location Restriction policy. The policy defines a list of the folders on network file shares in which creation of home folders is allowed, and prevents Active Roles from creating home folders in other network locations. For instructions on how to view or modify the policy settings, see Configuring the Home Folder Location Restriction policy later in this section.

Apply this home folder setting when user account is created

Upon creation of a user account, this option causes Active Roles to configure the user account in Active Directory in accord with the **Connect:** <drive letter> **To**: <network path> setting specified by this policy.

For example, with the **Connect: Z: To: \\Server\Share\%username%** policy setting, selecting this check box ensures that a newly created user account has the **Home Drive** property set to Z: and the **Home Directory** property set to \\Server\Share\LogonName where LogonName stands for the pre-Windows 2000 logon name of the user account.

Apply this home folder setting when user account is renamed

Upon renaming a user account, this option causes Active Roles to configure the user account in Active Directory in accord with the **Connect:** <drive letter> **To**: <network path> setting specified by this policy.

For example, with the **Connect: Z: To: \\Server\Share\%username%** policy setting, renaming a user account causes the policy to set the **Home Directory** property to \\Server\Share\NewLogonName where NewLogonName stands for the pre-Windows 2000 logon name that is assigned to the user account by the rename operation.

Create or rename home folder on file server as needed

When selected, this option directs Active Roles to attempt the creation or renaming of a (non-local) home folder on the file server when the **Home Directory** property is set or modified on a user account in Active Directory. The renaming of the home folder is attempted if the **Home Directory** property value contains the %username% notation and the changes to the user account include modification of the pre-Windows 2000 logon name of the user account. In other cases, the creation of a new home folder is attempted.

For example, with the **Connect: Z: To: \\Server\Share\%username%** policy setting, selecting this check box together with the option to apply the policy setting upon creation of a user account causes Active Roles to attempt the creation of the home folder for the user account. Active Roles attempts to create the holder with the following network path:

\\Server\Share\LogonName, where LogonName stands for the pre-Windows 2000 logon name of the user account.

Another example is setting the **Home Drive** and **Home Directory** properties on an existing user account in Active Directory: With this check box selected, Active Roles attempts to create the folder specified by the network path that is assigned to the **Home Directory** property.

If creation or renaming of the home folder fails on the file server, then the creation or modification of the user account fails as well. To prevent such an error condition, you could clear this check box.

The result is that Active Roles applies the changes to the **Home Drive** and **Home Directory** properties in Active Directory without attempting an operation on the file server, which allows the use of a different tool for creating home folders on the file server.

Copy user permissions on home folder from parent folder

Upon creation or renaming of a home folder for a particular user account, this option ensures that the user account has the same rights on the home folder as it has on the folder in which the home folder resides.

Set user as home folder owner

Upon creation or renaming of a home folder for a particular user account, this option ensures that the user account is set as the owner of the home folder.

An owner of a folder is authorized to make any changes to permission settings on the folder. For example, an owner can authorize other persons to access the folder.

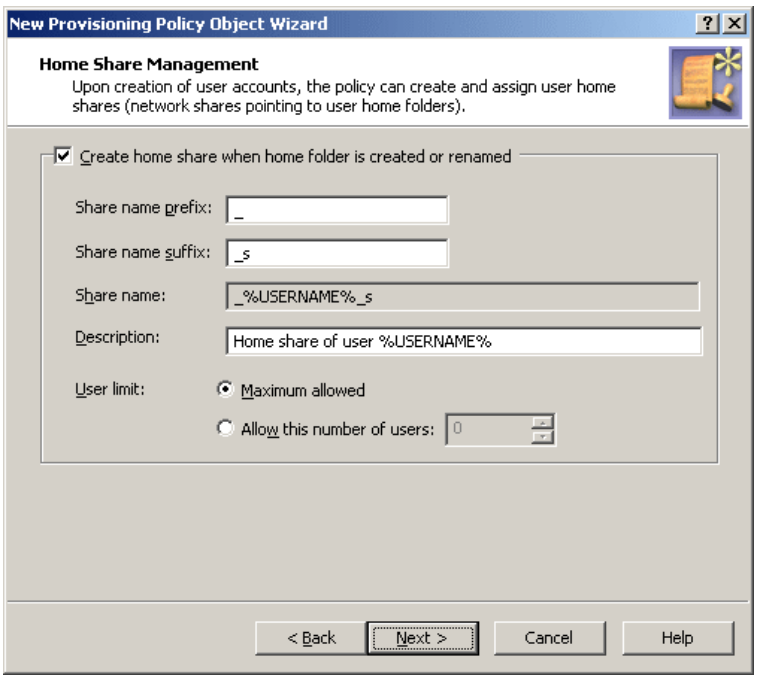
Set user permissions on home folder

Upon creation or renaming of a home folder for a particular user account, this option ensures that the user account has the specified access rights on the home folder.

With the **Grant Full Access** setting, the user account is authorized to perform any operation on the folder and its contents except for making changes to permission settings. With the **Grant Change Access** setting, the user account is authorized to view and modify the contents of the folder.

When finished, click **Next** to display the **Home Share Management** page. This page lets you configure policy options for creating home shares.

Figure 60: Home share management



To have the policy create home shares, select the **Create home share when home folder is created or renamed** check box.

When you configure the policy to create home shares, you can specify the prefix and suffix for the home share names.

Specifying a prefix and suffix allows you to establish a naming convention for home shares. Suppose you want home shares to be displayed at the top of the list of shares. To do so, you can use an underscore as the prefix. You may also assign a suffix to distinguish home shares created by the policy. For example, to distinguish the home shares of users from the Sales department, you could use the suffix _s. Then, when you create a user account with the pre-Windows 2000 logon name set to JohnB, the policy will map the user's home folder to the selected drive and specify **\\Server_JohnB_s** as the path to the home folder. The policy will also create the share **_JohnB_s** that points to the folder **\\Server\Home\JohnB**.

Optionally, in the **Description** box, you can type a comment about the home share. The users will see it when viewing share properties.

You can also limit the number of users that can connect to the share at one time. Click **Maximum allowed** or **Allow this number of users**. With the latter option, specify a number in the box next to the option.

Steps for configuring a Home Folder AutoProvisioning policy

To configure a Home Folder AutoProvisioning policy

1. On the **Policy to Configure** page, select **Home Folder AutoProvisioning**, and then click **Next**.
2. On the **Home Folder Management** page, do the following:
   - From the **Connect** list, select the drive letter to which you want the policy to map the home folder.
   - In the **To** box, specify a network path to the home folder.

     The path must include a common share at one level above the home folders. For example, you might specify \\Ant\Home\%username% for the policy to create home folders on the share Home on the server Ant. The path such as \\SERVER\%username% is not valid.
   - To have the policy verify that the home folder path and name on user accounts are set in compliance with this policy, select **Enforce this home folder setting in Active Directory**.

     When this check box is cleared, the policy allows home folder paths and names that differ from the path and name prescribed by the policy.
   - To have Active Roles automatically set the home folder properties in accord with this policy upon user account creation in Active Directory, select **Apply this home folder setting when user account is created**.
   - To have Active Roles automatically set the home folder properties in accord with this policy upon user account renaming in Active Directory, select **Apply this home folder setting when user account is renamed**.
   - To have Active Roles attempt creation or renaming of a (non-local) home folder on the file server when home folder properties are set or changed on a user account in Active Directory, select **Create or rename home folder on file server as needed**.

     If you want to configure the policy so that it not only sets home folder properties on user accounts in Active Directory but also creates or renames home folders and home shares in accord with the policy settings, you must keep the **Create or rename home folder on file server as needed** check box selected (this is the default setting). If the check box is cleared, then the policy can only set or verify home folder properties on user accounts in Active Directory.
   - Specify how you want the policy to configure permission settings on home folders. You can choose from the following options:
     - **Copy user permissions on home folder from parent folder**. Upon creation or renaming of a home folder for a user account, ensures that the user account has the same rights on the home folder as on the folder in which the home folder resides.

- **Set user as home folder owner**. Upon creation or renaming of a home folder for a user account, ensures that the user account is set as the owner of the home folder.
- **Set user permissions on home folder**. Upon creation or renaming of a home folder for a user account, ensures that the user account has the specified access rights on the home folder (such as Change Access or Full Access).

Click **Next**.

3. On the **Home Share Management** page, specify settings for user home shares. Do the following:
   - Select **Create home share when home folder is created or renamed** for the policy to create or rename the home share when creating or renaming the home folder.
   - Optionally, in **Share name prefix** and **Share name suffix**, type a prefix and suffix for the name of the home share. For details, see How to configure a Home Folder AutoProvisioning policy earlier in this chapter.
   - Optionally, in **Description**, type a comment to add to the home share.
   - If you want to limit the number of users that can connect to the share at a time, click **Allow this number of users** and specify the maximum number of users in the box next to that option. Otherwise, click **Maximum allowed**.

   Click **Next.**
4. On the **Enforce Policy** page, you can specify objects to which this Policy Object is to be applied:
   - Click **Add**, and use the **Select Objects** dialog box to locate and select the objects you want.
5. Click **Next**, and then click **Finish**.

> NOTE: For more information about the Home Folder AutoProvisioning policy configuration options, see How to configure a Home Folder AutoProvisioning policy earlier in this chapter.

Using the built-in policy for home folder provisioning

If you want to configure Active Roles so that setting or changing home folder related properties on any user account in any managed domain does not result in an attempt to create or rename a folder on a file server, then you can use the Active Roles console to modify the built-in Policy Object:

1. In the console tree, select **Configuration | Policies | Administration | Builtin**.
2. In the details pane, double-click **Built-in Policy - Default Rules to Provision Home Folders**.

3. On the **Policies** tab, select the policy from the list and then click **View/Edit**.
4. On the **Home Folder** tab, clear the **Create or rename home folder on file server as needed** check box.
5. Click **OK** to close the dialog boxes you opened.

If you have any other Policy Objects containing policies of the Home Folder AutoProvisioning category, then you need to configure them as appropriate: Select or clear the **Create or rename home folder on file server as needed** check box in each of those policies depending on whether or not Active Roles should attempt creation or renaming of home folders for user accounts that fall within the scope of the respective Policy Object.

Another scenario may require Active Roles to create or rename home folders for user accounts that are outside a certain scope (such as a certain domain, organizational unit, or Managed Unit), whereas creation or renaming of home folders should not be attempted on user accounts that fall within that particular scope. In this scenario, ensure that the **Create or rename home folder on file server as needed** option is selected in the built-in Policy Object. Then, create and configure a Policy Object containing a policy of the Home Folder AutoProvisioning category with the **Create or rename home folder on file server as needed** option un-selected, and apply that Policy Object to the scope in question.

Configuring the Home Folder Location Restriction policy

When creating home folders, Active Roles operates in the security context of the service account under which the Administration Service is running, so the service account must have sufficient rights to create home folders. Normally, the service account has administrative rights on an entire file server, which enables Active Roles to create home folders in any folder on any network file share that exists on that server. The Home Folder Location Restriction is used to restrict to a certain list the network file shares and folders in which Active Roles is authorized to create home folders.

The Home Folder Location Restriction policy determines the folders on the network file shares in which Active Roles is allowed to create home folders, and prevents Active Roles from creating home folders in other locations. The restrictions imposed by this policy do not apply if the home folder creation operation is performed by an Active Roles Admin role holder (normally, these are the users that have membership in the Administrators local group on the computer running the Active Roles Administration Service). Thus, when an Active Roles Admin role holder creates a user account, and a certain policy is in effect to facilitate home folder provisioning, the home folder is created regardless of the Home Folder Location Restriction policy settings.

By default, no network file shares and folders are listed in the policy. This means that Active Roles cannot create a home folder unless the user management operation that involves creation of the home folder is performed by the Active Roles Admin role holder. In order to allow delegated administrators to create home folders, you have to configure the policy so that it lists the folders on the network file shares in which creation of home folders is allowed. You can do this by using the Active Roles console as follows.

To configure the Home Folder Location Restriction policy

1. In the console tree, expand **Configuration | Policies | Administration**, and select **Builtin** under **Administration**.
2. In the details pane, double-click **Built-in Policy - Home Folder Location Restriction**.
3. On the **Policies** tab, double-click the list item under **Policy Description**.
4. On the **Allowed Locations** tab, view or modify the list of folders on the network file shares where creation of home folders is allowed.

   When adding a folder to the list, specify the UNC name of the folder. If you specify the name in the form \\<Server>\<Share>, home folders can be created in any folder on the network file share specified. If you specify the name in the form \\<Server>\<Share>\<PathtoFolder>, home folders can be created in any sub-folder of the folder.

Scenario: Creating and assigning home folders

In this scenario, you configure a policy to create home folders when creating user accounts. The policy assigns home folders to newly created accounts and grants the users change access to their home folders.

To implement this scenario, you must perform the following actions:

1. Verify that the network file share on which you want the policy to create home folders is listed in the Home Folder Location Restriction policy.
2. Create and configure a Policy Object that defines the appropriate policy.
3. Apply the Policy Object to a domain, OU, or Managed Unit.

As a result, when creating a user account in the container you selected in Step 3, Active Roles creates the user home folder and assigns that folder to the user account.

The following sub-sections elaborate on the steps to implement this scenario.

Step 1: Verifying the Home Folder Location Restriction policy

The network file share to hold home folders must be listed in the Home Folder Location Restriction policy. Use the Configuring the Home Folder Location Restriction policy instructions to verify that the policy allows creation of home folders on the network file share.

Step 2: Creating and Configuring the Policy Object

You can create and configure the Policy Object you need by using the New Provisioning Policy Object wizard. For information about the wizard, see Creating a Policy Object in the

Policy Object management tasks section earlier in this chapter.

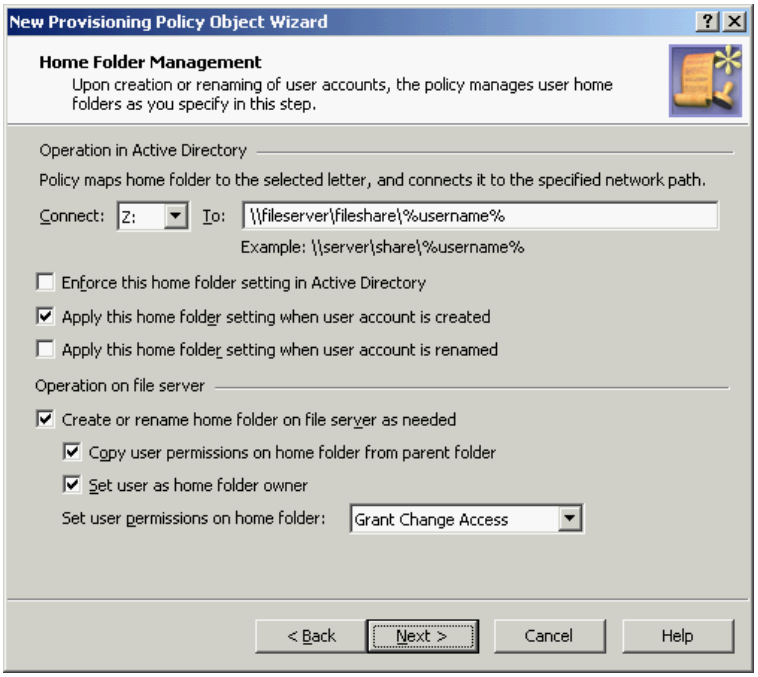
To configure the policy, click **Home Folder AutoProvisioning** on the **Policy to Configure** page of the wizard. Then, click **Next**.

On the **Home Folder Management** page, set up the following options:

- In the **Connect** box, select the drive letter to assign to the home folder (for example, **Z:**).
- In the **To** box, type the path in the form \\server\share\%username% where \\server\share is a valid UNC path to a network file share. For example, if you have a network file share set up on the **comp** server, with the share name set to **home**, you may specify the following path: \\comp\home\%username%
- Select the **Apply this home folder setting when user account is created** check box.

As a result, the **Home Folder Management** page should look like the following figure.

Figure 61: Policy Object: Home folder management



Click **Next** and follow the steps in the wizard to create the Policy Object.

Step 3: Applying the Policy Object

You can apply the Policy Object by using the **Enforce Policy** page in the New Provisioning Policy Object wizard, or you can complete the wizard and then use the **Enforce Policy** command on the domain, OU, or Managed Unit where you want to apply the policy.

For more information on how to apply a Policy Object, see Applying Policy Objects and Managing policy scope earlier in this chapter.

Script Execution

Script execution policies help you run supplementary PowerShell (or other) script modules in Active Roles during or after performing certain administrative operations. When linking a custom script to an administrative operation via a **Script Execution** policy, the script will receive control in Active Roles either when the operation is requested or when it is completed.

Use **Script Execution** policies to set up custom scripts (residing in **Script Modules** in the Active Roles MMC console) to:

- Trigger additional actions when performing directory object provisioning.
- Regulate object data format and requirements (such as for generating user passwords).
- Further automate administrative tasks.

Example use case for a Script Execution policy

Consider a scenario where employees of an organization are frequently transferred among its office branches temporarily due to various projects.

To administer such temporary assignments quickly and efficiently, write and apply a custom script that automatically reassigns the employee's user account from the OU of their original office to the OU of their new office, whenever their **City** or **Office Location** attributes are updated in Active Roles.

For more information on how to set up a Script Execution policy, see Steps for configuring a Script Execution policy

TIP: Consider the following when planning to use custom scripts for your provisioning policies:

- To help you configure Script Execution policies, Active Roles also ships with several built-in **Script Modules** that you can use to set up your own **Script Execution** policies. Find these built-in **Script Modules** in the following node of the Active Roles MMC console:

  Configuration > **Script Modules** > **Builtin**

- If the directory of your organization contains any cloud-only Azure users, then use the built-in **Generate User Password - Azure only** script module to set up a password generation policy for cloud-only Azure users that meets the password strength criteria of both your organization and Microsoft Azure Active Directory (AD).

NOTE: Policy Object settings specific to Azure cloud-only objects (such as cloud-only Azure users, guest users, or contacts) are available only if your Active Roles deployment is licensed for managing cloud-only Azure objects. Contact One Identity support for more information.

Also, Policy Objects specific to Azure cloud-only objects will work correctly only if an Azure tenant is already configured in the AD of the organization, and Active Roles is already set as a consented Azure application for that Azure tenant. For more information on these settings, see Configuring a new Azure tenant and consenting Active Roles as an Azure application.

How this policy works

Active Roles executes the script module specified in the policy when the operation is requested or after the operation is completed. The script module is stored in the Active Roles configuration database.

How to configure Script Execution policy

When configuring a Script Execution policy, you can prepare a script module beforehand. Alternatively, you can create an empty script module when configuring a policy, and later you can edit the module and add a script to be used by the policy.

You can import a script from a file or write a new script using the Active Roles console. The console displays script modules in the **Script Modules** container under **Configuration**.

Importing a script

To import a script file, in the console tree, right-click **Script Modules**, and click **Import**. This displays the **Import Script** dialog box where you can select and open a script file.

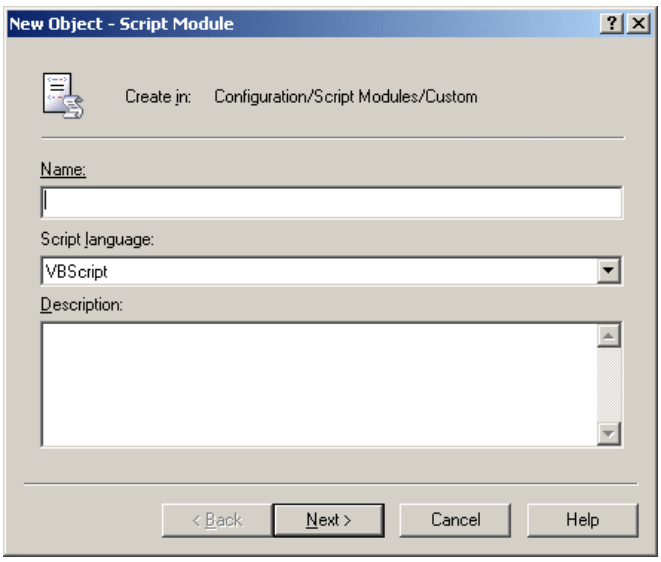
Creating a script

To create a new script module, in the console tree, right-click **Script Modules** and select **New | Script Module**. This opens the New Object - Script Module wizard.

> TIP: It is advisable to store custom script modules in a separate container. You can create a container as follows: Right-click **Script Modules** in the console tree, and select **New | Scripts Container**. After you have created a container, you can have the wizard add a script module to that container rather than directly to **Script Modules**: right-click the container in the console tree and select **New | Script Module**.

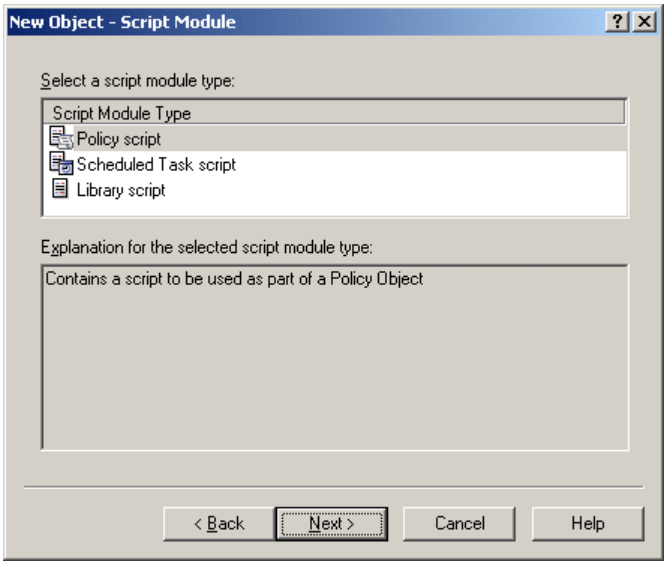
The first page of the wizard looks as shown in the following figure.

Figure 62: Script module: Creating a script



Type a name and description for the new script module, and select script language. Then click **Next**. The next page looks as shown in the following figure.

Figure 63: Script Module: Policy script

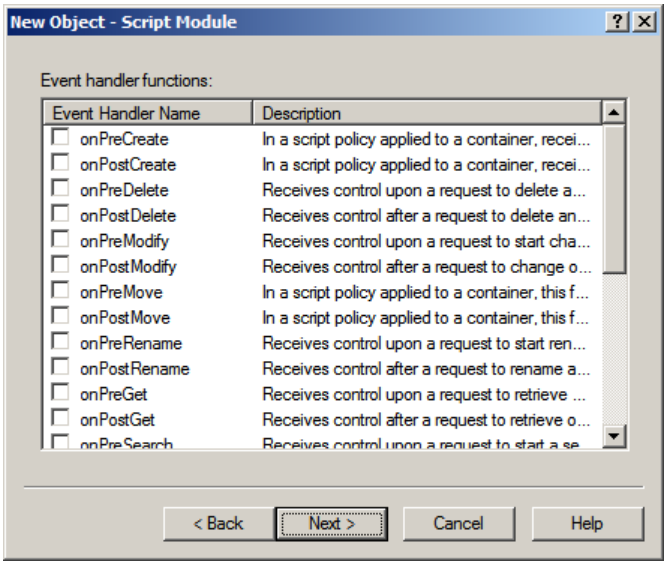


On this page, select a type of the script module. Select **Policy script** to create a script that will be used as part of the Policy Object. The other options are:

- **Scheduled Task script**. Script that you can schedule to run on the Administration Service.
- **Library script**. Script to be used by other script modules. You can collect commonly used functions into a standalone script module and include it in other modules requiring those functions. This allows you to re-use some pieces of existing scripts, thus reducing development effort and time.

Select **Policy script** and click **Next**. This displays the page with a list of event handler functions shown in the following figure.

Figure 64: Script Module: Event handler functions



On this page, select functions to be used in the script, and click **Next**. Then, click **Finish** to create the script module.

For instructions and guidelines on how to develop policy scripts, refer to the Active Roles Software Development Kit (SDK).

In the Active Roles console, you can view and modify scripts, both imported and newly created.

Editing a script

To edit a script, select it in the console tree under **Configuration/Script Modules**. You can view and modify the script in the details pane. To start editing the script, right-click the script module and click **Edit Script**. Then, click **Yes** to confirm the operation. You can make changes to the script in the details pane.

When you are editing the script, a red asterisk is displayed next to the name of the script module in the console tree. This indicates the changes you are making to the script are not saved. You can undo your changes or save them:

- To undo changes, press `CTRL+Z`. (The redo function is also available: press `CTRL+Y`.)
- To undo all unsaved changes, right-click the script module and click **Discard Changes**. (This operation is irreversible: if you perform this command, your changes to the script are lost.)
- To save the changes, right-click the script module and click **Save Script on Server**.

When the script module is ready, you can proceed to configuring a script policy that will use the prepared script module.

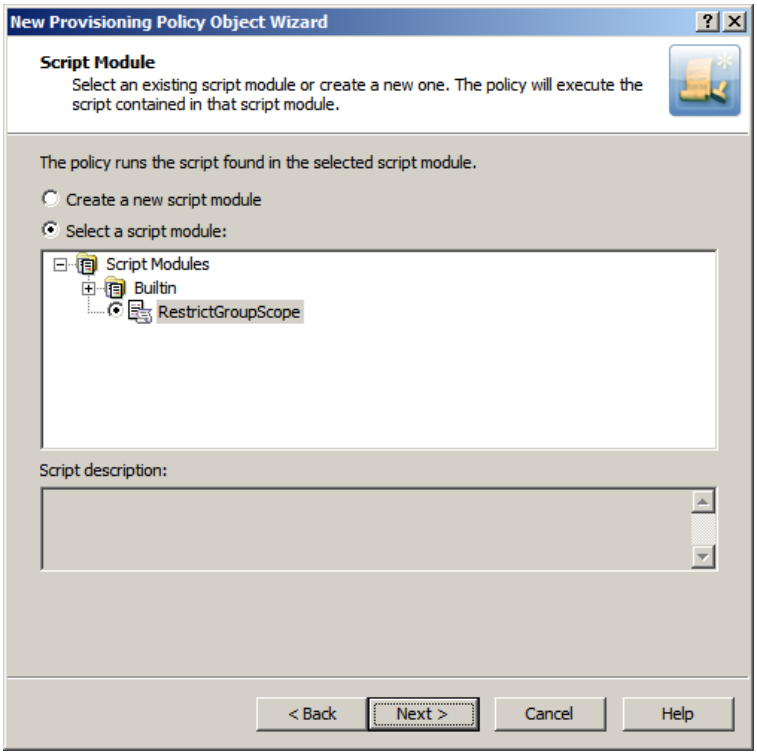
Active Roles allows you to attach a debugger to the Administration Service's script host for a given policy script or scheduled task script. When the script is being executed by the specified Administration Service, the debugger may help you identify and isolate problems, if any, with the policy or task based on that script.

To enable debugging of a script in the Active Roles console, display the **Properties** dialog box for the script module containing the script, go to the **Debugging** tab, and select the **Enable debugging** check box. From the **Debug on server** list, select the Administration Service where you want the debugger to run.

Configuring a policy to execute a script

To configure a script-based policy, select **Script Execution** on the **Select Policy Type** page in the New Provisioning Policy Object wizard, in the New Deprovisioning Policy Object wizard, or in the Add Policy wizard. Then, click **Next** to display the **Script Module** page.

Figure 65: Script Module: Executing a script



On this page, click **Select a script module** and select the script module. Then, click **Next** and follow the instructions in the wizard to create the Policy Object.

Steps for configuring a Script Execution policy

To configure a Script Execution policy

1. On the **Policy to Configure** page, select **Script Execution**, and then click **Next**.
2. On the **Script Module** page, do one of the following:
   - To use an existing script module, click **Select a script module**, and select the script module in the box beneath this option.
   - To create a new script module, click **Create a new script module**, and click **Next**. Then, specify a name for the script module, and click **Next**. Then, select the event handlers you want the script module to include.
3. Click **Next**.
4. On the **Policy Parameters** page, do the following:
   a. If necessary, from the **Function to declare parameters** list, choose the function that defines the parameters specific to this policy.

      The list contains the names of all script functions found in the selected Script Module. The policy has the parameters that are defined by the function specified in the **Function to declare parameters** box. Normally, this is a function named onInit.

   a. Under **Parameter values**, view or change the values of the policy parameters. To change the value of a parameter, select the name of the parameter and click **Edit**.

      Clicking **Edit** displays a page where you can add, remove, or select a value or values for the selected parameter. For each parameter, the function that is used to declare parameters defines the name of the parameter and other characteristics, such as a description, a list of possible values, the default value, and whether a value is required. If a list of possible values is defined, then you can only select values from that list.
5. On the **Enforce Policy** page, you can specify objects to which this Policy Object is to be applied:
   - Click **Add**, and use the **Select Objects** dialog box to locate and select the objects you want.
6. Click **Next**, and then click **Finish**.

To create a script module

1. In the console tree, under **Configuration | Script Modules**, locate and select the folder in which you want to add the script module.

   You can create a new folder as follows: Right-click **Script Modules** and select **New | Scripts Container**. Similarly, you can create a sub-folder in a folder: Right-click the folder and select **New | Scripts Container**.
2. Right-click the folder and select **New | Script Module**.
3. Specify the name and language of the module to create. Then, click **Next**.

4. In **Select a script module type**, click the type of the module to create. Then, click **Next**.
5. If you selected the **Policy script** type for the module, select the event handlers you want the module to include, and then click **Next**.
6. Click **Finish**.

To edit a script module

1. In the console tree, expand **Configuration | Script Modules**.
2. Under **Script Modules**, click the folder that contains the script module you want to edit.
3. In the details pane, right-click the script module, and then click **Edit Script**.
4. Use the details pane to make changes to the script.
5. Right-click the script module in the console tree, and do one of the following:
   - To commit the changes you have made, click **Save Script on Server**.
   - To quit the script editor without saving your changes, click **Discard Changes**.

To import a script module

1. In the console tree, under **Configuration | Script Modules**, locate and select the folder in which you want to add the script module.

   You can create a new folder as follows: Right-click **Script Modules** and select **New | Scripts Container**. Similarly, you can create a sub-folder in a folder: Right-click the folder and select **New | Scripts Container**.
2. Right-click the folder, and click **Import**.
3. Locate and select the file containing the script to import, and click **Open**.

To export a script module

1. In the console tree, expand **Configuration | Script Modules**.
2. Under **Script Modules**, select the folder that contains the script module you want to export.
3. In the details pane, right-click the script module, and select **All Tasks | Export**.
4. Specify the file to which you want to save the script, and then click **Save**.

Scenario: Restricting group scope

This scenario describes how to configure a policy that prevents creation of universal groups. With this policy, the Active Roles console or Web Interface does not allow an administrator to create a new universal group or convert an existing group to a universal group.

To implement this scenario, you must perform the following actions:

1. Prepare the script that implements this scenario.
2. Create and configure the Policy Object to run that script.
3. Apply the Policy Object to a domain, OU, or Managed Unit.

As a result, the Active Roles console or Web Interface cannot be used to set the universal group scope option when creating a new group or changing an existing group in the container you selected in Step 3. For example, if you choose the **Universal** option under **Group scope** and then click **Next** in the New Object - Group wizard, the Active Roles console presents you with an error message stating that creation of universal groups is not allowed.

The following sections elaborate on the steps to implement this scenario.

Step 1: Preparing the script module

The script used in this scenario is installed with the Active Roles SDK. By default, the path and name of the script file is as follows:
`%ProgramFiles%\One Identity\Active Roles\Active Roles\SDK\Samples\RestrictGroupScope\RestrictGroupScope.ps1`.

The script receives control upon a request to check the property values submitted to the Administration Service, and analyzes the value of the groupType attribute to determine if the universal group scope option is attempted. If the script detects that the assumed groupType value would cause a group to be configured as a universal group, it raises a policy violation event in the Administration Service. As a result, the application that initiated the request (such as the Active Roles console or Web Interface) displays an error message provided by the script. For more information, see the "Restricting the Scope of Groups" topic in the Active Roles *SDK documentation*.

To import the script, right-click the **Script Modules** container in the Active Roles console, and click **Import**. Then, select and open the **RestrictGroupScope.ps1** file.

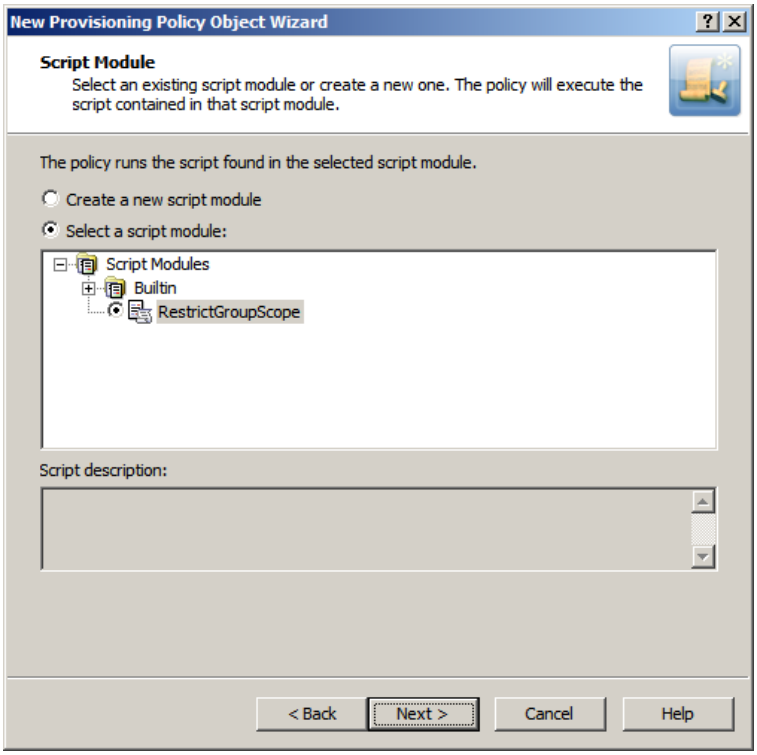
Step 2: Creating and configuring the Policy Object

You can create and configure the Policy Object you need by using the New Provisioning Policy Object wizard. For information about the wizard, see Creating a Policy Object in the Policy Object management tasks section earlier in this chapter.

To configure the policy, click **Script Execution** on the **Policy to Configure** page of the wizard. Then, click **Next**.

On the **Script Module** page, click **Select a script module**, and select **RestrictGroupScope** from the list of script modules, as shown in the following figure.

Figure 66: Script Module: Creating/configuring policy object



Click **Next** and follow the instructions in the wizard to create the Policy Object.

Step 3: Applying the Policy Object

You can apply the Policy Object by using the **Enforce Policy** page in the New Provisioning Policy Object wizard, or you can complete the wizard and then use the **Enforce Policy** command on the domain, OU, or Managed Unit where you want to apply the policy.

For more information on how to apply a Policy Object, see Applying Policy Objects and Managing policy scope earlier in this chapter.

Office 365 and Azure Tenant Selection

Policies in this category are intended to manage the tenant selection, license selection, and roles selection, and OneDrive provisioning for Azure AD.

How this policy works

The provisioning policy **O365 and Azure Tenant Selection** is a unified policy for Azure Office 365 management for users, controlled or restricted by creating a new provisioning policy and applying the policy to the Organizational Unit. This policy is used for tenant

selection, Office 365 license selection, and Office 365 roles selection, and OneDrive provisioning for Azure AD users.

This policy is also used for tenant selection for Groups and contacts.

Configuring an O365 and Azure Tenant Selection policy

You can configure an **O365 and Azure Tenant Selection** policy in the Active Roles Console (also known as the MMC Interface) to:

- Validate the selected Azure tenants for Azure users, guest users, O365 Groups and contacts.
- Select O365 Licenses for Azure users and guest users.
- Select O365 Roles for Azure users and guest users.
- Preprovision OneDrive for Azure users.

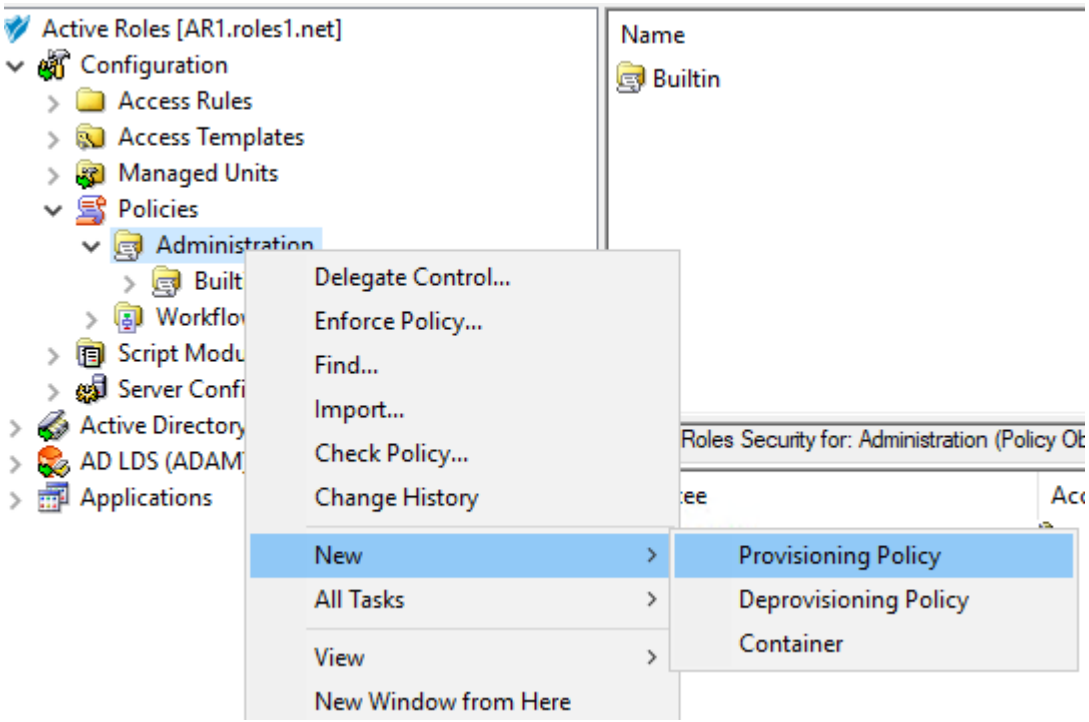
Prerequisites

Consider the following before configuring an **O365 and Azure Tenant Selection** policy:

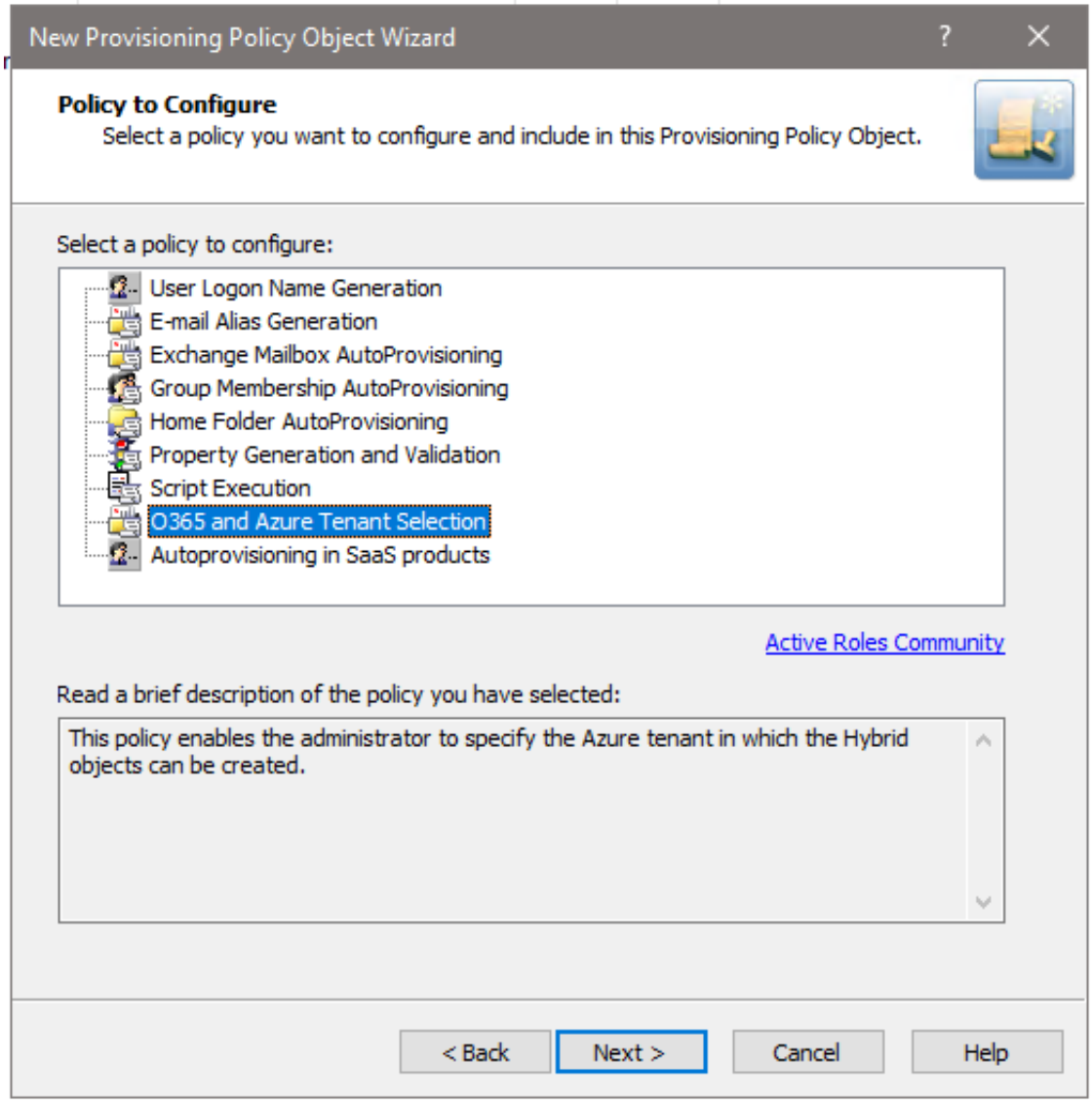
- The OneDrive settings of this policy are applicable to hybrid Azure users only, and will work only if you have already enabled OneDrive for your Azure tenant in the **Azure AD Configuration** > **Modify** (**Tenant details**) window of the Active Roles Configuration Center. For more information on enabling OneDrive for Azure users in an Azure tenant, see Enabling OneDrive in an Azure tenant.
- To configure an **O365 and Azure Tenant Selection** policy, your Organizational Unit (OU) must already have the **Azure - Default Rules to Generate Properties** built-in policy configured. For more information on configuring the policy, see Configuring the Azure - Default Rules to Generate Properties policy.

To configure an O365 and Azure Tenant Selection policy

1. Navigate to **Configuration** > **Policies** > **Administration**.
2. To open the **New Provisioning Policy Object Wizard** dialog, right-click in the middle pane to open the context menu, and then select **New** > **Provisioning Policy**.



3. On the **Name and Description** page, provide a unique **Name** for the new policy object. Optionally, also provide a **Description**. To continue, click **Next**.
4. On the **Policy to Configure** page, select **O365 and Azure Tenant Selection**, and click **Next**.

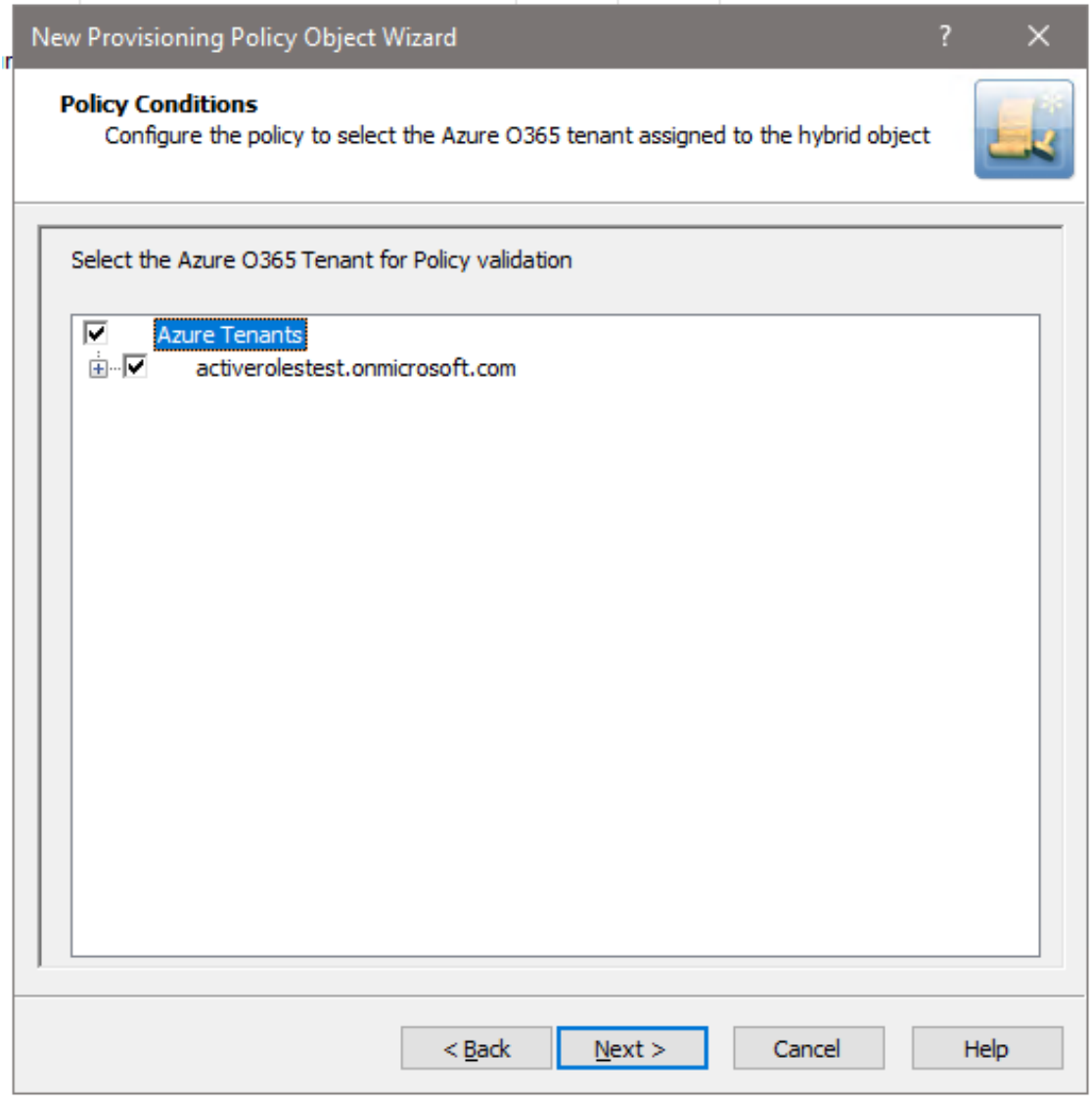


5. On the **Object Type Selection** page, to specify the type of object you want the policy to provision, click **Select**, then click **OK**.

   TIP: If you do not see the object type you need, expand the list by selecting **Show all possible object types**.

   NOTE: If you want to assign and validate Office 365 licenses and roles, or provision OneDrive storage as part of the configured policy, select the **User (user)** object type in this step. Office 365 license and role validation, and OneDrive provisioning are not applicable to Azure Groups and Azure Contacts.

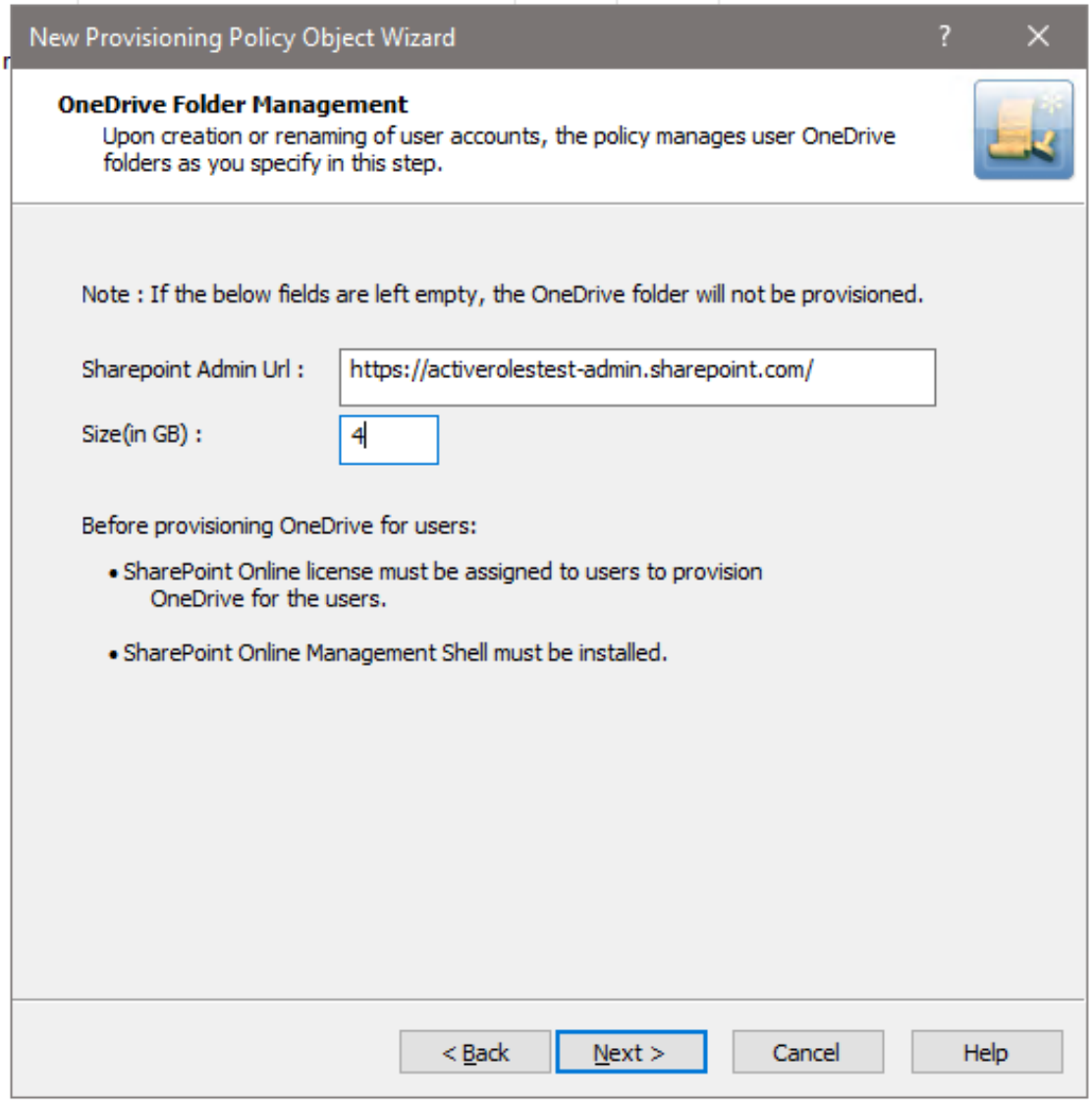
6. On the **Policy Conditions** page, select your Azure tenant for which you want to set up the policy. To continue, click **Next**.



7. (Optional) On the next **Policy Conditions** page, select the licenses to validate and assign to new Azure users in the Azure tenant. To continue, click **Next**.

   NOTE: If OneDrive storage is planned to be provisioned in the selected Azure tenant for Azure users, make sure that you select the **SharePoint Online** license in this step. Otherwise, the configured OneDrive storage cannot be provisioned for Azure users created later. For more information, see Creating a new cloud-only Azure user.

8. (Optional) On the next **Policy Conditions** page, select the Office 365 roles to validate and assign to new Azure users in the Azure tenant. To continue, click **Next**.
9. (Optional) To configure OneDrive storage for the Azure users of the Azure tenant, configure the following attributes on the **OneDrive Folder Management** page:



- **SharePoint Admin URL**: Specify the URL of the SharePoint administration site of your Azure tenant. The URL has the following syntax: `<azure-tenant-name>-admin.sharepoint.com`
- **Size (in GB)**: Specify the default OneDrive storage size allocated for each Azure user in the Azure tenant.

If you do not need to provision OneDrive storage for users in the Azure tenant, leave the settings empty and click **Next**.

NOTE: If the wizard shows an error when clicking **Next** after configuring the OneDrive settings:

- Check that the specified SharePoint Admin URL is correct.
- Make sure that the specified OneDrive storage size is correct (that is, it is within the range of the individual cloud storage allowed for users in your organization).

10. On the **Enforce Policy** page, select the Organizational Unit (OU) for which the policy will be applied. To do so, click **Add** to open the **Select Objects** window, then select the OU from the list. To continue, click **OK** then **Next**.
11. To complete the wizard, click **Finish**.

Applying a new policy

Office 365 user license management

1. From the Web interface, assign, or modify the Office 365 license for an Azure AD User.

   The Policy is triggered for any Azure AD user in the Organization Unit for which the O365 and Azure Tenant selection policy is applied.

   If the policy conditions are not satisfied while assigning or modifying Azure AD User licenses, the following policy violation error is displayed:

   Provisioning policy failure. The 'O365 and Azure Tenant Selection' policy encountered an error. Exception in Azure Tenant Management Policy violation: The Azure user License(s) O365_BUSINESS_ESSENTIALS-PROJECTWORKMANAGEMENT, cannot be assigned. The policy prescribes that this Azure User requires only the specified license in the policy object to be assigned.

2. Right-click and click **Check Policy** to check if there are any policy violations

   For a container object, this displays the Check Policy dialog box.

3. Review the options in the **Check Policy** dialog box and click **OK**.

   The Policy Check Results window is displayed.

   IMPORTANT: Office 365 user license management now allows Administrator to select a subset of the licenses selected in policy during user creation or modification.

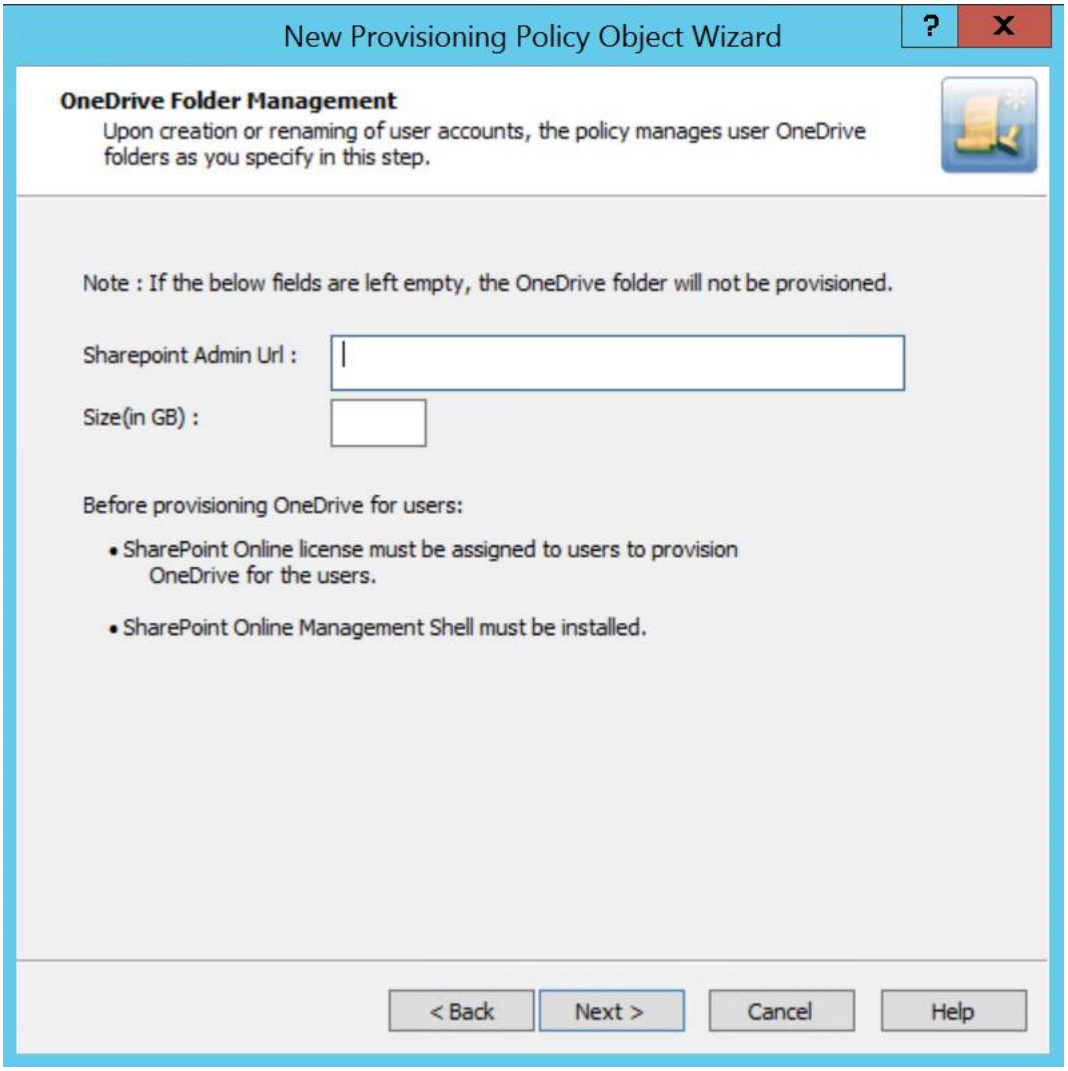
Office 365 user roles management through provisioning policy

From the Web interface, assign or modify the Office 365 roles for an Azure AD User.

While creating an Azure AD user from the Active Roles Web interface, if the policy conditions are not satisfied while assigning Azure AD User roles, the following policy violation error is displayed:

Provisioning policy failure. The 'O365 and Azure Tenant Selection' policy encountered an error. Exception in Azure Tenant Management Policy violation: The Azure user Role(s) cannot be assigned. The policy prescribes that this Azure User requires only the specified role in the policy object to be assigned.

Figure 67: OneDrive folder management wizard



Provisioning OneDrive for Azure AD users

1. From the Web interface, create an Azure AD User, and assign a valid SharePoint Online license.
2. After the user is created, the OneDrive provisioning process is performed in the background and after some time the process is completed.

   NOTE:

   - If the SharePoint Admin URL is incorrect then the OneDrive provisioning is not successful.
   - For an existing Azure AD user, during modification of user properties:

- If OneDrive is not provisioned, then OneDrive provisioning is triggered.
- If OneDrive is provisioned, and any changes are made to the OneDrive provisioning policy, then the policy changes are applied on the user.

3. To check the provisioning result, open Azure Properties window for the user from the Web interface, navigate to OneDrive tab.

   On successful provisioning of the user, the OneDrive URL, the used storage size, and the total storage size are displayed.

   NOTE: The storage size indicated in the policy gets synchronized to the Azure AD user's OneDrive.

User Account Deprovisioning

Policies in this category are intended to automate the following deprovisioning-related tasks on user accounts:

- Disable the user account.
- Set the user password to a random value.
- Set the user's logon names to random values.
- Rename the user account.
- Modify other properties of the user account.

When configuring a policy of this category, you specify how you want Active Roles to modify the user's account in Active Directory upon a request to deprovision a user so that once the deprovision operation is completed, the deprovisioned user cannot log on to the network.

You may also configure a policy to update any user properties, such as those that regulate users' membership in Active Roles Managed Units. In this way, the policy can automate the addition or removal of deprovisioned users from Managed Units.

How this policy works

When processing a request to deprovision a user, Active Roles uses this policy to modify the user's account so that once the user has been deprovisioned, they cannot log on to the network.

A policy can also be configured to update user accounts. Depending on the policy configuration, each policy-based update results in the following:

- Certain portions of account information are removed from the directory by resetting specified properties to empty values.
- Certain properties of user accounts are set to new, non-empty values.

A policy can be configured so that new property values include:

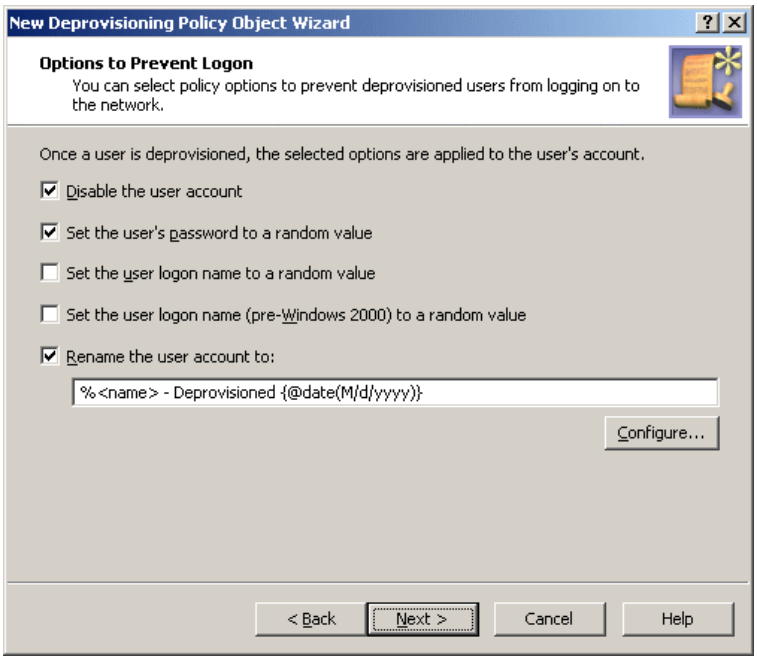
- Properties of the user account being deprovisioned, retrieved from the directory prior to starting the process of the user deprovisioning
- Properties of the user who originated the deprovisioning request
- Date and time when the user was deprovisioned

Thus, when deprovisioning a user, Active Roles modifies the user's account in Active Directory as determined by the User Account Deprovisioning policy that is in effect.

How to configure a User Account Deprovisioning policy

To configure a User Account Deprovisioning policy, select **User Account Deprovisioning** on the **Policy to Configure** page in the New Deprovisioning Policy Object wizard or in the Add Deprovisioning Policy wizard. Then, click **Next** to display the **Options to Prevent Logon** page.

Figure 68: User Account Deprovisioning



On this page, you can select options that make the account ineligible for logon. The option names are self-explanatory:

- Disable the user account
- Set the user's password to a random value
- Set the user logon name to a random value
- Set the user logon name (pre-Windows 2000) to a random value

Select check boxes next to the options you want the policy to apply.

By selecting the **Rename the user account to** check box, you direct the policy to change the user name of the account. This option allows you to configure a property update rule that specifies how to modify the user name. The following subsection provides instructions on how to configure a property update rule, taking the user name as the example.

Configuring a property update rule

To configure a property update rule for the user name, click the **Configure** button. This displays the **Configure Value** dialog box, discussed earlier in this chapter (see How to configure a Property Generation and Validation policy). You can use that dialog box to set up a value for the **'name' must be** condition, in the same way as you do when configuring a Property Generation and Validation policy.

To start configuring a value, click **Add** in the **Configure Value** dialog box. This displays the **Add Entry** window.

A value is a concatenation of one or more entries. In the **Add Entry** window, you can select the type of the entry to add, and then configure the entry. The following table summarizes the available types of entries.

Table 21: Types of entries: Configuring a property update rule

Type of entry	Description
Text	Adds a text string to the value.
User Property	Adds a selected property (or a part of a property) of the user account being deprovisioned.
Parent OU Property	Adds a selected property (or a part of a property) of an organizational unit in the hierarchy of containers above the user account being deprovisioned.
Parent Domain Property	Adds a selected property (or a part of a property) of the domain of the user account being deprovisioned.
Date and Time	Adds the date and time when the account was deprovisioned.
Initiator ID	Adds a string that identifies the Initiator, that is, the user who originated the deprovisioning request. This entry is composed of Initiator-related properties, retrieved from the directory.

Instructions on how to configure an entry depend on the type of the entry. You can use the instructions outlined in the How to configure a Property Generation and Validation policy section earlier in this chapter to configure an entry of any of these types:

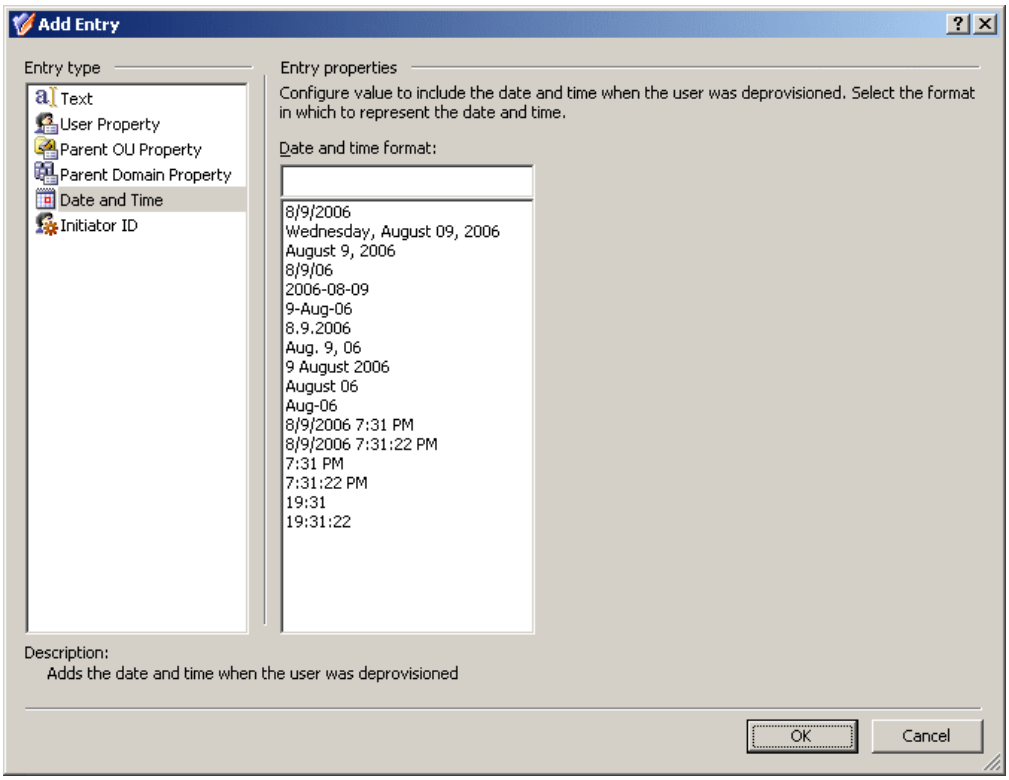
- **Text**. Refer to the Entry type: Text subsection.
- **User Property**. Refer to the Entry type: <Object> Property subsection.
- **Parent OU Property**. Refer to the Entry type: Parent OU Property subsection.
- **Parent Domain Property**. Refer to the Entry type: Parent Domain Property subsection.

The following subsections elaborate on the **Date and Time** and **Initiator ID** entries.

Entry type: Date and Time

When you select **Date and Time** under **Entry type** in the **Add Entry** window, the **Entry properties** area looks like the following figure.

Figure 69: Entry type: Date and Time



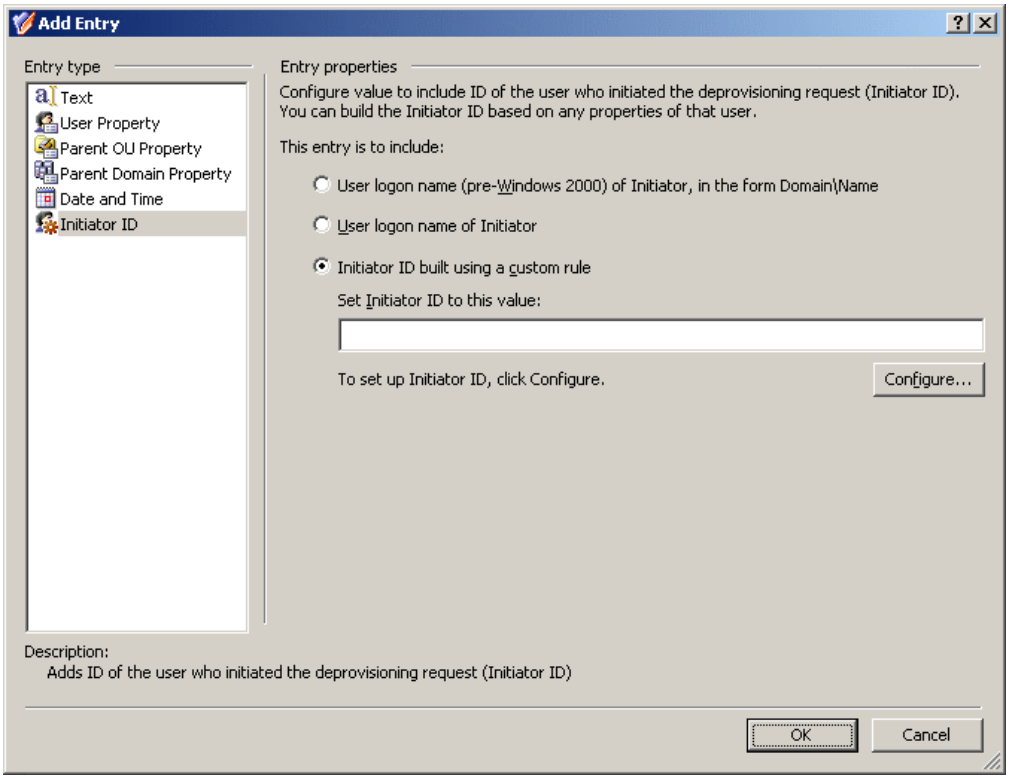
Using this entry type, you can add an entry that represents the date and time when the user account was deprovisioned.

In the list under **Date and time format**, click the date or time format you want. Then, click **OK** to close the **Add Entry** window.

Entry type: Initiator ID

When you select **Initiator ID** under **Entry type** in the **Add Entry** window, the **Entry properties** area looks like the following figure.

Figure 70: Entry type: Initiator ID



With this entry type, you can add a string that identifies the Initiator, that is, the user who originated the deprovisioning request. The policy generates the Initiator ID based on certain properties of the Initiator's account, such as the user logon name. A custom rule can be configured to use other properties.

You can choose a pre-configured rule or configure a custom rule to generate the Initiator ID. The pre-configured rules allow you to set the Initiator ID to one of the following:

- The pre-Windows 2000 user logon name of the Initiator, in the form DomainName\UserName
- The user logon name of the Initiator

A custom rule allows you to compose the Initiator ID of other Initiator-related properties.

Configuring a custom rule to build the Initiator ID

To configure a custom rule for Initiator ID, click the lowermost option under **Entry properties**, and then click the **Configure** button. This displays the **Configure Value** dialog box, discussed earlier in this chapter (see How to configure a Property Generation and Validation policy). You can use that dialog box to set up a value for the **'Initiator ID' must be** condition, in the same way as you do when configuring a Property Generation and Validation policy.

To start configuring a value, click **Add** in the **Configure Value** dialog box. This displays the **Add Entry** window.

A value is a concatenation of one or more entries. In the **Add Entry** window, you can select the type of the entry to add, and then configure the entry. The following table summarizes the available types of entries.

Table 22: Available entries for Configuring a custom rule to build the Initiator ID

Type of entry	Description
Text	Adds a text string to the value.
Initiator Property	Adds a selected property (or a part of a property) of the Initiator's user account.
Parent OU Property	Adds a selected property (or a part of a property) of an organizational unit in the hierarchy of containers above the Initiator's user account.
Parent Domain Property	Adds a selected property (or a part of a property) of the domain of the Initiator's user account.

Instructions on how to configure an entry depend on the type of the entry. For each type of entry, you can find the instructions in the How to configure a Property Generation and Validation policy section, earlier in this chapter:

- **Text**. Refer to the Entry type: Text subsection.
- **Initiator Property**. Refer to the Entry type: <Object> Property subsection.
- **Parent OU Property**. Refer to the Entry type: Parent OU Property subsection.
- **Parent Domain Property**. Refer to the Entry type: Parent Domain Property subsection.

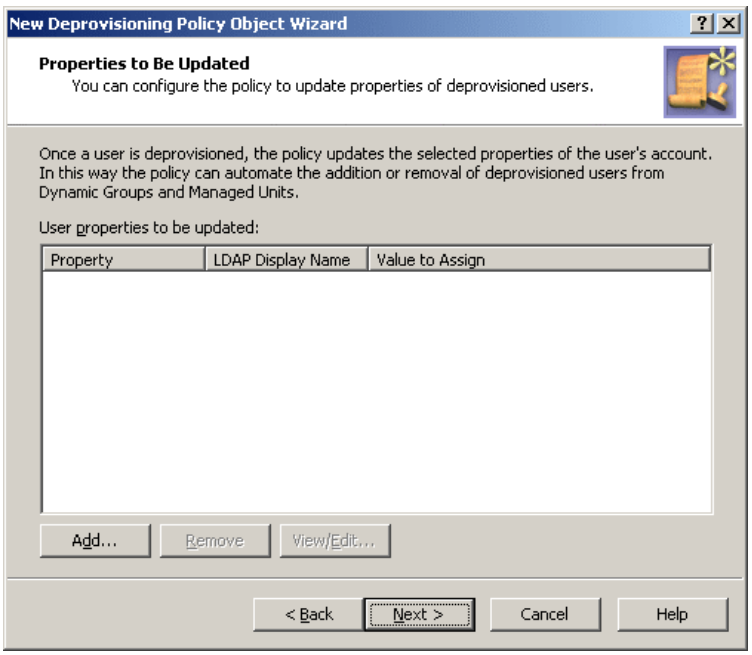
When you are done configuring a value for the **'Initiator ID' must be** condition, click **OK** to close the **Configure Value** dialog box. This will add the value to the Initiator ID entry properties. If necessary, you can modify the value by clicking the **Configure** button in the **Add Entry** window and then managing the list of entries in the **Configure Value** dialog box.

When you are done configuring the **Initiator ID** entry, click **OK** to close the **Add Entry** window. The entry is added to the **Configure Value** dialog box for the **'name' must be** condition.

When you are done configuring a value for the **'name' must be** condition, click **OK** to close the **Configure Value** dialog box. This will add the rule to the **Options to Prevent Logon** page of the wizard. If necessary, you can modify the rule by clicking the **Configure** button on that page and then managing the list of entries in the **Configure Value** dialog box.

Once you have completed the **Options to Prevent Logon** page, click **Next** to display the **Properties to Be Updated** page.

Figure 71: Properties to Be Updated



On this page, you can set up a list of user properties you want the policy to update. Each entry in the list includes the following information:

- **Property**. When deprovisioning a user, Active Roles will update this property of the user's account.
- **LDAP Display Name**. Uniquely identifies the property to be updated.
- **Value to Assign**. After the deprovisioning operation is completed, the property has the value defined by this syntax.

You can use these buttons to manage the list on this page:

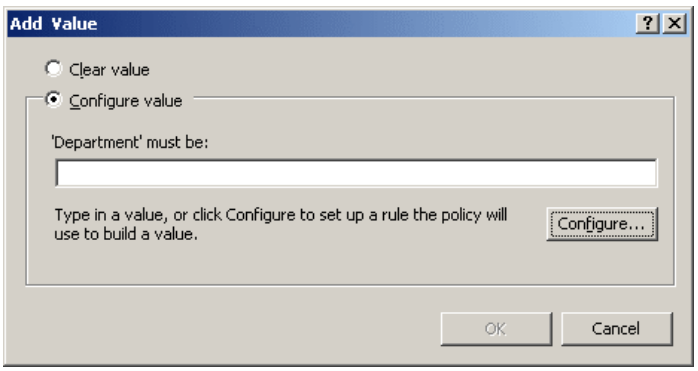
- **Add**. Allows you to select a user property and configure an update rule for that property. A property update rule specifies how to generate the new value to assign to the property.
- **Remove**. If you want the policy to no longer update a given property, select the property from the list and click **Remove**.
- **View/Edit**. Allows you to modify the update rule for the property you select from the list.

Clicking the **Add** button displays the **Select Object Property** dialog box where you can choose user properties you want to the policy to update. To choose a property, select the check box next to the property name, and then click **OK**.

You can select multiple check boxes. If you do so, the properties you have selected are added to the list on the wizard page, with the update rule configured to clear those properties, that is, to assign them the empty value.

If you select a single property in the **Select Object Property** dialog box, you are presented with the **Add Value** dialog box so you can proceed to configuring a property update rule.

Figure 72: Add Value



You can select one of these update options:

- **Clear value**. Causes the policy to assign the empty value to the property.
- **Configure value**. Allows you to configure a value for the **'property' must be** condition.

With the second option, you must configure a value the policy will assign to the property upon the user deprovisioning. You can configure a value in the same way as you do when configuring a property update rule for the user name: Click the **Configure** button and follow the instructions provided earlier in this section (see Configuring a property update rule).

When you are done configuring a value, click **OK** to close the **Add Value** dialog box. The property name along with the property update rule is added to the wizard page. If necessary, you can modify the update rule by clicking the **View/Edit** button beneath the list of properties. This displays a dialog box, similar to the **Add Value** dialog box, allowing you to choose a different update option or set up a different value for the **'property' must be** condition.

Once you have set up the list on the wizard page, click **Next** and follow the instructions in the wizard to create the Policy Object.

Steps for configuring a User Account Deprovisioning policy

To configure a User Account Deprovisioning policy

1. On the **Policy to Configure** page, select **User Account Deprovisioning**, and then click **Next**.
2. On the **Option to Prevent Logon** page, select the options you want the policy to apply when deprovisioning a user account. You can select any combination of these options:
    - Disable the user account
    - Set the user's password to a random value

- Set the user logon name to a random value
- Set the user logon name (pre-Windows 2000) to a random value
- Rename the user account to

3. If you selected **Rename the user account to**, click **Configure**, and then complete the **Configure Value** dialog box by using the procedure outlined later in this topic, in order to specify how you want the policy to update the user name when deprovisioning a user account.
4. Click **Next**.
5. On the **Properties to Be Updated** page, specify how you want the policy to update user properties when deprovisioning a user account:
   - Click **Add**, and then complete the **Select Object Property** dialog box by using the procedure outlined later in this topic, in order to add property update rules.
   - Use **View/Edit** to modify existing rules.
   - Use **Remove** to delete existing rules.
6. Click **Next**.
7. On the **Enforce Policy** page, you can specify objects to which this Policy Object is to be applied:
   - Click **Add**, and use the **Select Objects** dialog box to locate and select the objects you want.
8. Click **Next**, and then click **Finish**.

To complete the Configure Value dialog box

1. Click **Add**.
2. Configure an entry to include in the value (for instructions, see Steps for configuring entries).
3. In the **Configure Value** dialog box, add more entries, delete or edit existing ones, and then click **OK**.

To complete Select Object Property dialog box

1. From the **Object property** list, select an object property, and then click **OK**. The **Add Value** dialog box appears.

   If you select multiple properties, the **Add Value** dialog box is not displayed. The properties you have selected are added to the list on the **Properties to Be Updated** page, with the update rule configured to clear those properties, that is, to assign them the "empty" value.
2. In the **Add Value** dialog box, do one of the following:
   - Select **Clear value** if you want the update rule to assign the empty value to the property.
   - Select **Configure value** if you want the update rule to assign a certain, non-empty value to the property. Then, click **Configure** and complete the

Configure Value dialog box by using the instructions given earlier in this topic.

Scenario 1: Disabling and renaming the user account upon deprovisioning

The policy described in this scenario performs the following functions during the user deprovisioning process:

- Disable the user account.
- Append this suffix to the user name: **- Deprovisioned**, followed by the date that the user account was deprovisioned.

For example, the policy changes the user name **John Smith** to **John Smith - Deprovisioned 12/11/2010**.

To implement this scenario, you must perform the following actions:

1. Create and configure the Policy Object that defines the appropriate policy.
2. Apply the Policy Object to a domain, OU, or Managed Unit.

As a result, when deprovisioning a user account in the container you selected in Step 2, Active Roles disables and renames the user account as prescribed by this policy.

The following two sections elaborate on the steps to implement this scenario.

Step 1: Creating and configuring the Policy Object

You can create and configure the Policy Object you need by using the New Deprovisioning Policy Object wizard. For information about the wizard, see Creating a Policy Object in the Policy Object management tasks section earlier in this chapter.

To configure the policy, click **User Account Deprovisioning** on the **Select Policy Type** page of the wizard. Then, click **Next**.

On the **Options to Prevent Logon** page, select these check boxes:

- Disable the user account
- Rename the user account to

Then, click the **Configure** button, and use the following instructions to complete the **Configure Value** dialog box.

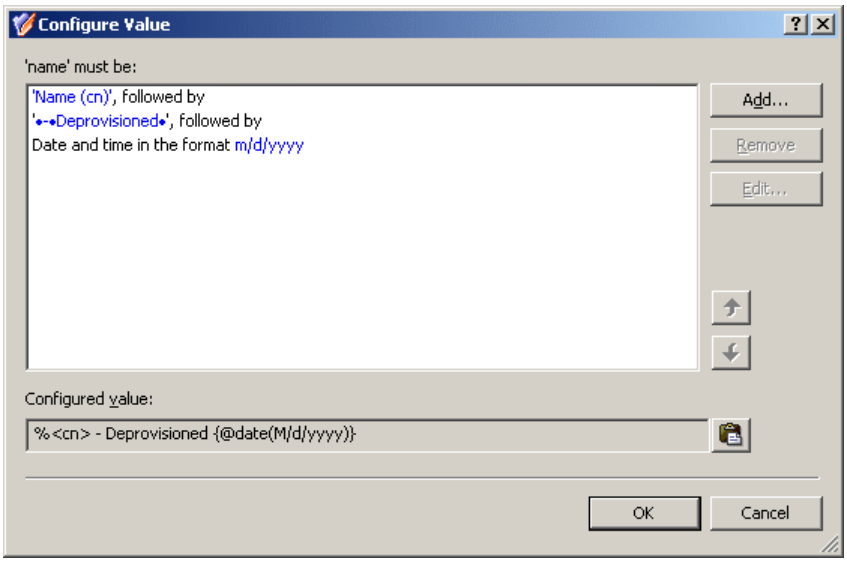
1. Click **Add**.
2. In the **Add Entry** window, click **User Property** under **Entry type**, and configure the entry as follows:
   a. Click **Select** and choose the **Name** property.
   b. Click **All characters of the property value**.

c. Click **OK**.

3. Click **Add**.
4. In the **Add Entry** window, click **Text** under **Entry type**, and configure the entry as follows:
   a. In the **Text value** box, type **- Deprovisioned**.
   b. Click **OK**.
5. Click **Add**.
6. In the **Add Entry** window, click **Date and Time** under **Entry type**, and configure the entry as follows:
   a. From the list under **Date and time format**, select the format **m/d/yyyy**.
   b. Click **OK**.

After you complete these steps, the list of entries in the **Configure Value** dialog box should look like the following figure.

Figure 73: Configure Value



Click **OK** to close the **Configure Value** dialog box. Then, click **Next** and follow the instructions in the wizard to create the Policy Object.

Step 2: Applying the Policy Object

You can apply the Policy Object by using the **Enforce Policy** page in the New Provisioning Policy Object wizard, or you can complete the wizard and then use the **Enforce Policy** command on the domain, OU, or Managed Unit where you want to apply the policy.

For more information on how to apply a Policy Object, see Applying Policy Objects and Managing policy scope earlier in this chapter.

Scenario 2: Managed Unit for deprovisioned user accounts

This scenario describes how to configure a Managed Unit and a User Account Deprovisioning policy so that the Managed Unit includes all the deprovisioned user accounts. The policy sets the **Notes** property to **Deprovisioned** upon the user deprovisioning, whereas the Managed Unit is configured to include user accounts that have the **Notes** property set to **Deprovisioned**.

To implement this scenario, you must perform the following actions:

1. Create and configure the Managed Unit.
2. Configure the Policy Object that defines the appropriate policy.
3. Apply the Policy Object to a domain, OU, or Managed Unit.

As a result, after deprovisioning a user account in the container you selected in Step 3, Active Roles automatically adds that account to the Managed Unit you created in Step 1.

The following sections elaborate on the steps to implement this scenario.

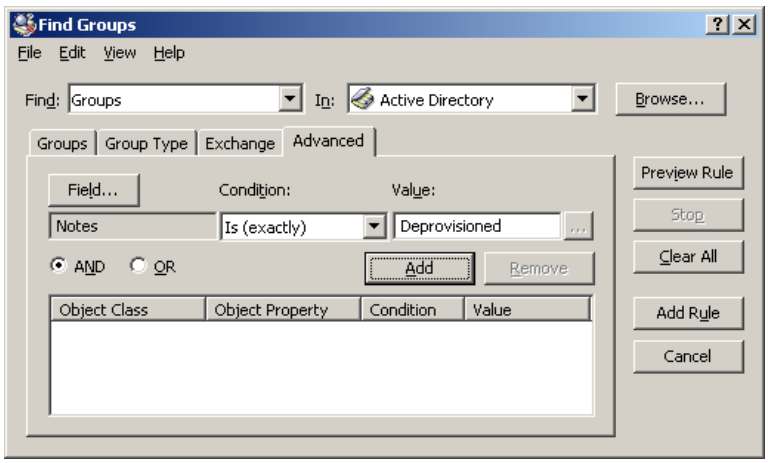
Step 1: Creating and configuring the Managed Unit

You can create and configure the Managed Unit by using the Active Roles console:

1. In the console tree, under **Configuration**, right-click **Managed Units**, and select **New | Managed Unit**.
2. In **Name**, type a name for the Managed Unit. For example, you might type **Deprovisioned Users**.
3. Click **Next**.
4. Configure the membership rule to have the Managed Unit include the deprovisioned user accounts from all domains that are registered with Active Roles (managed domains):
   a. On the wizard page, click **Add**.
   b. In the **Membership Rule Type** dialog box, click **Include by Query**, and then click **OK**.
   c. Use the **Create Membership Rule** window to set up the rule:
      - In **Find**, click **Users**.
      - Click **Browse** and select **Active Directory**.
      - Click the **Advanced** tab.
      - Click the **Field** button, and then click **Notes**.
      - In **Condition**, click **Is (exactly)**.
      - In **Value**, type **Deprovisioned**.

   At this stage, the window should look like the following figure.

Figure 74: Find Groups



- Click the **Add** button.
- Click the **Add Rule** button.

d. On the wizard page, click **Add**.

e. In the **Membership Rule Type** dialog box, click **Retain Deprovisioned**, and then click **OK**.

5. Click **Next**, click **Next**, and then click **Finish**.

Step 2: Configuring the Policy Object

You can configure the Policy Object you need by modifying the Policy Object that implements the previous scenario, see Scenario 1: Disabling and renaming the user account upon deprovisioning earlier in this section.

Display the **Properties** dialog box for that Policy Object and go to the **Policies** tab. Then, select the policy from the list, and click **View/Edit** to display the **Group Object Deprovisioning Policy Properties** dialog box. Click the **Change Properties** tab.

The **Change Properties** tab looks similar to the page of the same name in the wizard you used to create the Policy Object. You can use that tab to add the update rule for the **Notes** property:

1. Click **Add** to display the **Select Object Property** dialog box.
2. Select the check box next to the **Notes** property, and then click **OK**.
3. In the **Add Value** dialog box, type **Deprovisioned** in the **'Notes' must be** box, and then click **OK**.

Click **OK** to close the **Group Object Deprovisioning Policy Properties** dialog box.

Step 3: Applying the Policy Object

You can apply the Policy Object by using the **Scope** tab in the **Properties** dialog box for that Policy Object:

1. On the **Scope** tab, click the **Scope** button to display the **Active Roles Policy Scope** window for the Policy Object you are managing.
2. Click **Add** and select the domain, OU, or Managed Unit where you want to apply the policy.
3. Click **OK** to close the **Active Roles Policy Scope** window.
4. Click **OK** to close the **Properties** dialog box for the Policy Object.

For more information on how to apply a Policy Object, see Applying Policy Objects and Managing policy scope earlier in this chapter.

Office 365 Licenses Retention

This policy is intended to automate retention of all or selected Office 365 licenses assigned to an Azure AD user after the Azure AD user is deprovisioned successfully.

How this policy works

When processing a request to deprovision an Azure AD user, Active Roles uses this policy to determine if the licenses assigned to the Azure AD user must be retained.

When an Azure AD User is deprovisioned, this policy ensures that the administrator assigned Office 365 licenses are retained based on the policy configuration.

You can configure the Office 365 Licenses Retention policy to specify how you want Active Roles to modify the Azure AD user's licenses in Azure AD upon a request to deprovision the Azure AD user.

When an Azure user is deprovisioned from the Active Roles Console, Web Interface, or Management Shell, the Office 365 licenses that were assigned to the user during user provisioning are retained based on the Office 365 Licenses Retention policy configuration. As per the policy set, all the licenses or only selected licenses are retained upon the user deprovision.

The changes that take effect after deprovisioning the user are reflected in the Azure portal and the **Azure Properties** | **Licenses** tab of the Azure AD user in the Web interface

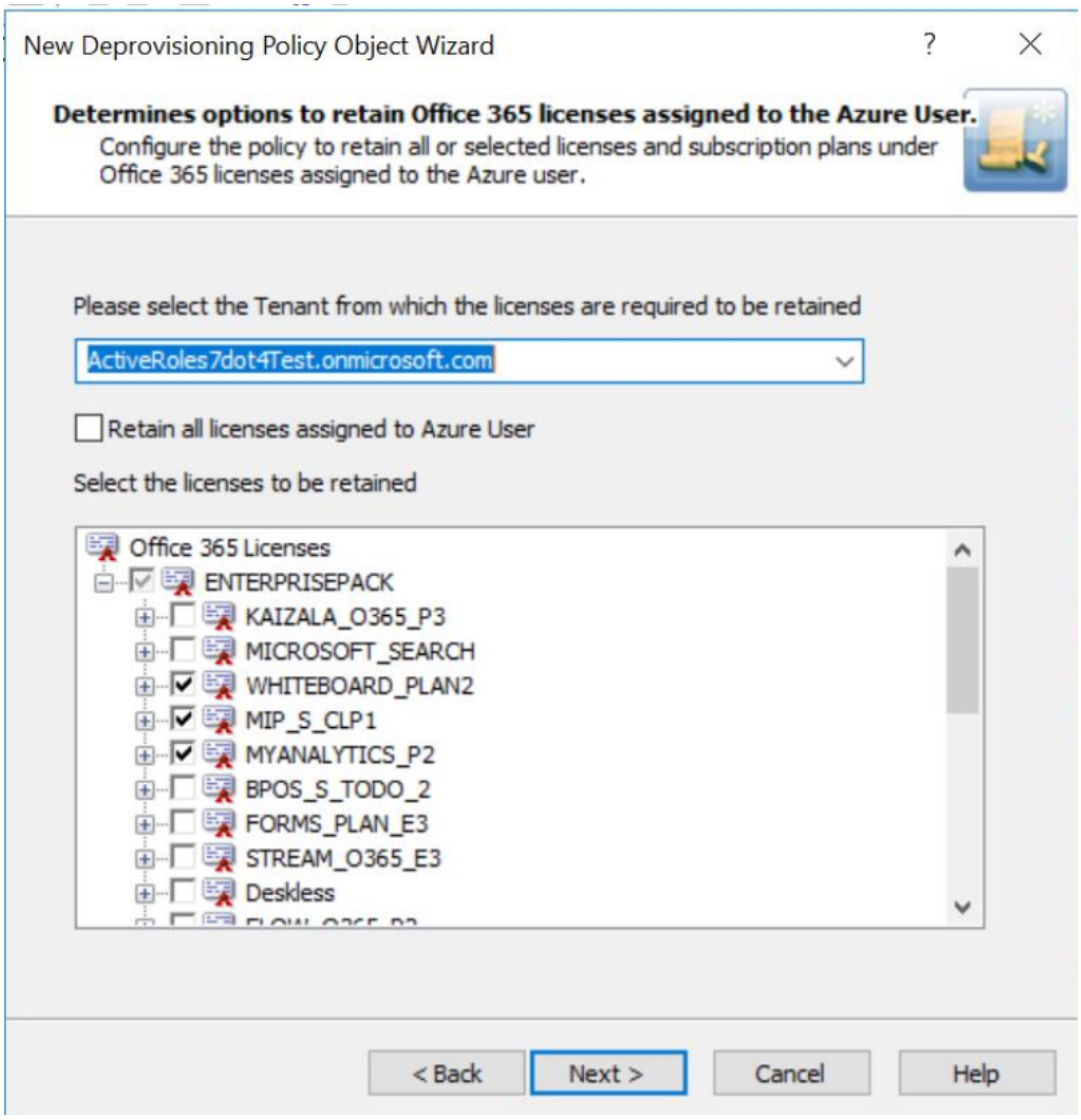
Active Roles Console enables you to create a new Deprovisioning Policy Object or Add to the existing Built-in Policy – User Default Deprovisioning policy. For instructions on how to create a Deprovisioning policy object, see the section **Creating a Policy Object**, in the *Active Roles Administration Guide*. The Office 365 Licenses Retention policy from the User Deprovisioning Policies must be selected to enable retention of the required Office 365 licenses upon Azure AD user deprovisioning.

NOTE: The Office 365 Licenses Retention policy is enabled only when Azure AD is configured.

How to configure Office 365 License Retention policy

To configure an Office 365 Licenses Retention policy, select **Office 365 Licenses Retention** on the Policy to Configure page in the New Deprovisioning Policy Object wizard or in the Add Deprovisioning Policy wizard. Then, click **Next** to display the Office 365 Licenses Retention page.

Figure 75: Office 365 Licenses Retention page



On the Office 365 Licenses Retention page, you can select the tenant from which the licenses are required to be retained. You can also configure the policy to retain all the licenses or selected licenses.

Select the check box corresponding to **Retain all the licenses** option to enable the deprovisioned Azure AD user to retain all the Office 365 licenses after successful deprovisioning.

Select the check boxes corresponding to the specific Office 365 subscription plans and licenses that the deprovisioned Azure AD must retain after successful deprovisioning.

After configuring the required options for retaining the licenses, click **Next** and follow the instructions on the wizard to create the Policy Object.

NOTE:

- After performing an Undo Provisioning operation on the deprovisioned Azure AD User, the original licenses assignment made to the user at the time of User provisioning is restored to the user.
- In Active Roles with **Office365 Licenses Rention** policy applied, when a deprovisioned Azure AD user tries to set licenses, a policy violation error is displayed.
- For more information on deprovisioning policy objects and creating new deprovisioning policies see the sections on Deprovisioning Policy Objects and Creating a Policy Object in the *Active Roles Administration Guide*.

Steps for configuring an Office 365 License Retention policy

To configure an Office 365 License Retention policy:

1. On the Policy to Configure page, select **Office 365 License Retention**, and then click **Next**.
2. On the Office 365 Licenses Retention page, select the options you want the policy to apply when deprovisioning the Azure AD user.
   - Select the tenant from which the licenses have to be retained for the user from the drop-down list.
   - Select the check box corresponding to Retain all the licenses option to enable the deprovisioned Azure AD user to retain all the Office 365 licenses after successful deprovisioning.
   - Select the check boxes corresponding to the specific Office 365 subscription plans and licenses that the deprovisioned Azure AD must retain after successful deprovisioning.
3. Click **Next**.

The Enforce Policy page is displayed, which enables you to specify objects to which this Policy Object is to be applied.

4. Click **Add**, and use the Select Objects dialog box to locate and select the objects on which you want to enforce the policy.
5. Click **Next**, and then click **Finish**.

Report on deprovisioning results

The Deprovisioning Results window displays the deprovision operation results pertaining to the Office 365 Licenses Retention policy. The results display a report of the success or failure of the policy.

Table 23: Office 365 License Retention policy

Report item (success)	Report item (failure)
In accordance with the policy, the Azure AD user's Office 365 licenses are retained.	N/A
Azure User Office 365 licenses are retained.	N/A

Group Membership Removal

Policies in this category are intended to automate the removal of deprovisioned user accounts from groups. A policy can be configured to remove user accounts from all groups with optional exceptions. Individual policy rules can be applied to security groups and to mail-enabled groups of both the security and distribution type.

How this policy works

When processing a request to deprovision a user, Active Roles uses this policy to determine what changes are to be made to group memberships of the user's account. By removing the account from security groups, the policy revokes user access to resources. By removing the account from mail-enabled groups, the policy prevents erroneous situations where e-mail is sent to the deprovisioned mailbox.

> IMPORTANT: The deprovisioned users are automatically removed from all Dynamic Groups, regardless of the Group Membership Removal policy settings.

A Group Membership Removal policy includes separate rules for security groups and for mail-enabled groups. For each category of groups, a rule can instruct Active Roles to perform one of the actions that are summarized in the following table.

Table 24: Group Membership Removal policy includes separate rules

Category	Action	Result
Security groups	Do not remove from groups.	The deprovisioned user remains in all security groups it was a member of as of the time of deprovisioning, except for the Dynamic Groups.
	Remove from all groups.	The deprovisioned user is removed from all security groups.
	Remove from all groups except for the specified ones.	The deprovisioned user is not removed from the specified security groups, with the exception of Dynamic Groups. The user is removed from all the other security groups.
Mail-enabled groups	Do not remove from groups.	The deprovisioned user is not removed from distribution groups or mail-enabled security groups, except for the Dynamic Groups.
	Remove from all groups.	The deprovisioned user is removed from all distribution groups and from all mail-enabled security groups.
	Remove from all groups except for the specified ones.	The deprovisioned user is not removed from the specified distribution or mail-enabled security groups, with the exception of Dynamic Groups. The user is removed from all the other distribution and mail-enabled security groups.

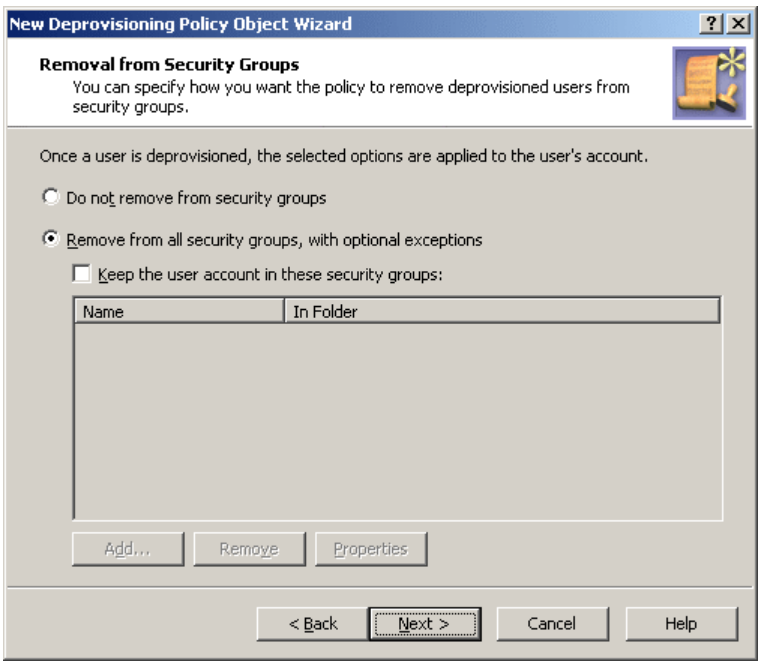
In the event of a conflict in policy implementation, the remove action takes precedence. For example, with a rule configured to remove the user account from all security groups, the user account is removed from all security groups even if there is another rule according to which Active Roles does not remove the user account from mail-enabled security groups.

Another conflict may occur in the situation where a policy of this category attempts to remove a deprovisioned user from a group that is configured as Active Roles' Dynamic Group (see the Dynamic Groups chapter, later in this document). The Dynamic Group policy detects the removal, and might add the deprovisioned user back to the Dynamic Group. To avoid such a situation, Active Roles does not allow Dynamic Groups to hold deprovisioned users. Once a user is deprovisioned, the user's account is removed from all Dynamic Groups.

How to configure a Group Membership Removal policy

To configure a Group Membership Removal policy, select **Group Membership Removal** on the **Policy to Configure** page in the New Deprovisioning Policy Object wizard or in the Add Deprovisioning Policy wizard. Then, click **Next** to display the **Removal from Security Groups** page.

Figure 76: Removal from Security Groups



On this page, you can configure a rule on how to remove a deprovisioned user from security groups.

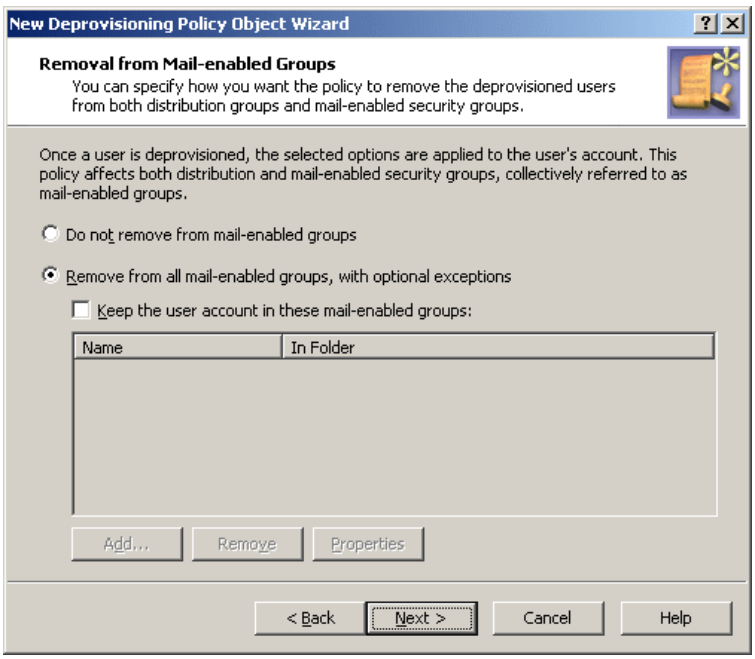
Select one of these options:

- Click **Do not remove from security groups** for the policy not to make changes to security group memberships of the user account.
- Click **Remove from all security groups, with optional exceptions** for the policy to remove the user account from all security groups.

With the second option, you can specify whether you want the policy not to remove the user account from certain security groups. To set up a list of such groups, select the **Keep the user account in these security groups** check box, and then click the **Add** button and select the groups you want to include in the list.

When you are done configuring the rule for security groups, click **Next** to display the **Removal from Mail-enabled Groups** page.

Figure 77: Removal from Mail-enabled Groups



This page is similar to the previous one. It allows you to configure a rule on how to remove a user account from both distribution groups and mail-enabled security groups, which are collectively referred to as mail-enabled groups.

Select one of these options:

- Click **Do not remove from mail-enabled groups** for the policy not to make changes to mail-enabled group memberships of the user account.
- Click **Remove from all mail-enabled groups, with optional exceptions** for the policy to remove the user account from all mail-enabled groups.

With the second option, you can specify whether you want the policy not to remove the user account from certain mail-enabled groups. To set up a list of such groups, select the **Keep the user account in these mail-enabled groups** check box, and then click the **Add** button and select the groups you want to include in the list.

When you are done configuring the rule for mail-enabled groups, click **Next** and follow the instructions in the wizard to create the Policy Object.

Steps for configuring a Group Membership Removal policy

To configure a Group Membership Removal policy

1. On the **Policy to Configure** page, select **Group Membership Removal**, and then click **Next**.

2. On the **Removal from Security Groups** page, do one of the following:
   - Click **Do not remove from security groups** for the policy not to make changes to security group memberships of the user account.
   - Click **Remove from all security groups, with optional exceptions** for the policy to remove the user account from all security groups.
3. If you selected the second option in Step 2, specify whether you want the policy not to remove the user account from certain security groups. Do one of the following:
   - Select the **Keep the user account in these security groups** check box and set up the list of security groups from which you want the policy not to remove the user account.
   - If you want the policy to remove the user account from all security groups, leave the check box cleared.
4. Click **Next**.
5. On the **Removal from Mail-enabled Groups** page, do one of the following:
   - Click **Do not remove from mail-enabled groups** for the policy not to make changes to mail-enabled group memberships of the user account.
   - Click **Remove from all mail-enabled groups, with optional exceptions** for the policy to remove the user account from all mail-enabled groups.
6. If you selected the second option in Step 5, specify whether you want the policy not to remove the user account from certain mail-enabled groups. Do one of the following:
   - Select the **Keep the user account in these mail-enabled groups** check box and set up the list of mail-enabled groups from which you want the policy not to remove the user account.
   - If you want the policy to remove the user account from all mail-enabled groups, leave the check box cleared.
7. Click **Next**.
8. On the **Enforce Policy** page, you can specify objects to which this Policy Object is to be applied:
   - Click **Add**, and use the **Select Objects** dialog box to locate and select the objects you want.
9. Click **Next**, and then click **Finish**.

Scenario: Removing deprovisioned users from all groups

The policy described in this scenario, removes the deprovisioned users from all groups, both security and distribution.

To implement this scenario, you must perform the following actions:

1. Create and configure the Policy Object that defines the appropriate policy.
2. Apply the Policy Object to a domain, OU, or Managed Unit.

As a result, when deprovisioning a user account in the container you selected in Step 2, Active Roles removes the user account from all groups.

The following two sections elaborate on the steps to implement this scenario.

Step 1: Creating and configuring the Policy Object

You can create and configure the Policy Object you need by using the New Deprovisioning Policy Object wizard. For information about the wizard, see Creating a Policy Object in the Policy Object management tasks section earlier in this chapter.

To configure the policy, click **Group Membership Removal** on the **Select Policy Type** page of the wizard. Then, click **Next** and follow these steps:

1. On the **Removal from Security Groups** page:
   a. Click **Remove from all security groups, with optional exceptions**.
   b. Verify that the **Keep the user account in these security groups** check box is cleared.
   c. Click **Next**.
2. On the **Removal from Mail-enabled Groups** page:
   a. Click **Remove from all mail-enabled groups, with optional exceptions**.
   b. Verify that the **Keep the user account in these mail-enabled groups** check box is cleared.
   c. Click **Next**.
3. Click **Next** and follow the instructions in the wizard to create the Policy Object.

Step 2: Applying the Policy Object

You can apply the Policy Object by using the **Enforce Policy** page in the New Provisioning Policy Object wizard, or you can complete the wizard and then use the **Enforce Policy** command on the domain, OU, or Managed Unit where you want to apply the policy.

For more information on how to apply a Policy Object, see Applying Policy Objects and Managing policy scope earlier in this chapter.

Exchange Mailbox Deprovisioning

Policies of this category are intended to automate the following tasks on deprovisioning Microsoft Exchange resources for deprovisioned users:

- Hide deprovisioned users from address lists.
- Prevent non-delivery reports from being sent.
- Grant designated persons full access to deprovisioned mailboxes.
- Redirect e-mail addressed to deprovisioned users.
- Force the mailbox of the deprovisioned user to send automatic replies (requires Exchange 2013 or later).

When configuring a policy of this category, you specify how you want Active Roles to modify the user's account and mailbox upon a request to deprovision a user. The purpose is to reduce the volume of e-mail sent to the mailbox of the deprovisioned user, and to authorize designated persons to monitor such e-mail.

How this policy works

When processing a request to deprovision a user, Active Roles uses this policy to determine the Exchange mailbox deprovisioning options, and then updates the user's account and mailbox accordingly.

The available mailbox-deprovisioning options are summarized in the following table. For each option, the table outlines the policy effect on a user's mailbox.

Table 25: Policy effect on a user's mailbox

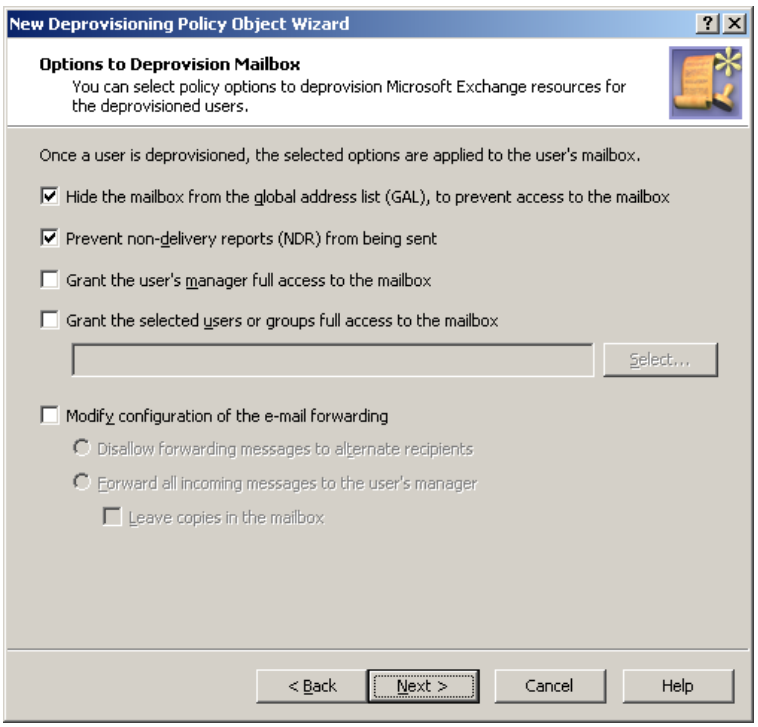
Option	Policy effect
Hide the mailbox from the Global Address List (GAL)	Prevents the deprovisioned user from appearing in your Exchange organization's address lists. If you select this option, the deprovisioned user is hidden from all address lists. This option renders the mailbox inaccessible. You cannot log on to Exchange Server as the mailbox user or otherwise access the hidden mailbox.
Prevent non-delivery reports (NDR) from being sent	Prevents non-delivery reports from being generated when e-mails are sent to the deprovisioned mailbox. (Non-delivery report is a notice that a message was not delivered to the recipient.)
Grant the user's manager full access to the mailbox	Provides the person designated as the deprovisioned user's manager with full access to the mailbox of that user. The manager is determined based on the **Manager** attribute of the deprovisioned user account in Active Directory.
Grant the selected users or groups full access to the mailbox	Provides the specified users or groups with full access to the deprovisioned user's mailbox.
Disallow forwarding messages to alternate	E-mail addressed to the deprovisioned user is not forwarded to an alternate recipient.

Option	Policy effect
recipients	
Forward all incoming messages to the user's manager	E-mail addressed to the deprovisioned user is forwarded to the user's manager. The manager is determined based on the **Manager** attribute of the deprovisioned user account in Active Directory.
Leave copies in the mailbox	E-mail addressed to the deprovisioned user is delivered to both the mailbox of the user's manager and the mailbox of the deprovisioned user. If you do not select this option, such e-mail is only delivered to the manager's mailbox.
Don't change the mailbox autoreply settings	Active Roles makes no changes to the Automatic Replies configuration of the mailbox. Thus, if the mailbox is configured to send automatic replies, deprovisioning the mailbox user does not cause the mailbox to stop sending automatic replies.
Automatically reply with the following messages (once for each sender)	Active Roles configures the mailbox to send the Automatic Replies messages specified by the policy. This option provides for the following policy settings: • The Automatic Replies message that is sent to senders within the organization. • Whether to send an Automatic Replies message to senders outside of the organization (external senders). • Whether to send an Automatic Replies message to all external senders or only to the user's contacts. • The Automatic Replies message that is sent to external senders.

How to configure an Exchange Mailbox Deprovisioning policy

To configure an Exchange Mailbox Deprovisioning policy, select **Exchange Mailbox Deprovisioning** on the **Policy to Configure** page in the New Deprovisioning Policy Object wizard or in the Add Deprovisioning Policy wizard. Then, click **Next** to display the **Options to Deprovision Mailbox** page.

Figure 78: Options to Deprovision Mailbox



On this page, you can select the Exchange mailbox deprovisioning options you want Active Roles to apply when deprovisioning a user. The names of the first four options are self-explanatory (also see the table above):

- Hide the mailbox from the global address list (GAL), to prevent access to the mailbox.
- Prevent non-delivery reports (NDR) from being sent.
- Grant the user's manager full access to the mailbox.
- Grant the selected users or groups full access to the mailbox.

Select check boxes next to the options you want to apply. The fourth option requires that you click the **Select** button to choose users or groups. The users and groups you select will be authorized to access the mailboxes of the deprovisioned users.

You can also configure the policy to modify the forwarding address for the deprovisioned users. If you select the **Modify configuration of the e-mail forwarding** check box, the policy sets the forwarding address to one of the following:

- **None**. To specify that e-mail messages addressed to a deprovisioned user are not to be forwarded, click **Disallow forwarding messages to alternate recipients**.
- **User's manager**. To specify that e-mail messages addressed to a deprovisioned user are to be sent to the manager of that user, click **Forward all incoming messages to the user's manager**.

The second option allows you to specify whether e-mail messages addressed to the deprovisioned user should be delivered to the mailbox of that user:

- If you select the **Leave copies in the mailbox** check box, the messages are delivered to both the user's mailbox and the manager's mailbox.
- If you clear the **Leave copies in the mailbox** check box, the messages are only delivered to the manager's mailbox.

Click **Next** to display the **Automatic Replies**, and choose from the following options on that page:

- **Don't change the mailbox autoreply settings**. Leaves the Automatic Replies configuration of the mailbox intact. Thus, if the mailbox is configured to send automatic replies, deprovisioning the mailbox user does not cause the mailbox to stop sending automatic replies.
- **Automatically reply with the following messages (once for each sender)**. Changes the Automatic Replies configuration of the mailbox to send automatic replies. You can specify separate autoreply messages for people inside and outside the user's organization.

The latter option enables the following policy settings:

- **Inside organization**. Specifies the Automatic Replies message that is sent to senders within the organization.
- **Auto-reply to people outside organization**. Determines whether to send an Automatic Replies message to senders outside of the organization (external senders). If you enable this setting, you must specify the autoreply message for external senders, and choose whether to send the message to the user's contacts only or to anyone outside organization.

When you are done, click **Next** and follow the instructions in the wizard to create the Policy Object.

Steps for configuring an Exchange Mailbox Deprovisioning policy

To configure an Exchange Mailbox Deprovisioning policy

1. On the **Policy to Configure** page, select **Exchange Mailbox Deprovisioning**, and then click **Next**.
2. On the **Options to Deprovision Mailbox** page, select the options you want the policy to apply when deprovisioning a user account. You can select any combination of these options to deprovision Microsoft Exchange resources for the deprovisioned user account:
    - Hide the mailbox from the Global Address List (GAL).
    - Prevent non-delivery reports (NDR) from being sent
    - Grant the user's manager full access to the mailbox
    - Grant the selected users or groups full access to the mailbox
    - Modify configuration of the e-mail forwarding

3. If you selected the **Grant the selected users or groups full access to the mailbox** check box, click **Select** to specify the users or groups you want.
4. If you selected the **Modify configuration of the e-mail forwarding** check box, do one of the following:
   - Click **Disallow forwarding messages to alternate recipients** to specify that the e-mail messages sent to the deprovisioned user are not to be forwarded.
   - Click **Forward all incoming messages to the user's manager** to specify that the e-mail messages sent to the deprovisioned user are to be forwarded to the manager of that user. Then, select or clear the **Leave copies in the mailbox** check box to specify whether you want the messages to be delivered to both the user's mailbox and the manager's mailbox or only to the manager's mailbox.
5. Click **Next**.
6. On the **Automatic Replies** page, choose from the following options:
   - Don't change the mailbox autoreply settings.
   - Automatically reply with the following messages (once for each sender).
7. If you selected the **Automatically reply with the following messages (once for each sender)** option, then do the following:
   - In the **Inside organization** box, specify the autoreply message to be sent to senders within the user's organization.
   - If you want the mailbox to send an autoreply message to external senders, select the **Auto-reply to people outside organization** check box, and specify the message in the area beneath that check box.
   - Select the **User's contacts only** or **Anyone outside organization** option depending on whether you want the mailbox to auto-reply only to external senders that are in the user's Contacts folder or to all external senders, respectively.
8. Click **Next**.
9. On the **Enforce Policy** page, you can specify objects to which this Policy Object is to be applied:
   - Click **Add**, and use the **Select Objects** dialog box to locate and select the objects you want.
10. Click **Next**, and then click **Finish**.

Scenario: Hide mailbox and forward e-mail to manager

The policy described in this scenario performs the following functions during the user deprovisioning process:

- Hides the deprovisioned user from the Exchange organization's address lists.
- Configures e-mail forwarding so that e-mail messages addressed to the deprovisioned user are sent to the user's manager, without delivering them to the user's mailbox.

To implement this scenario, you must perform the following actions:

1. Create and configure the Policy Object that defines the appropriate policy.
2. Apply the Policy Object to a domain, OU, or Managed Unit.

As a result, when deprovisioning a user account in the container you selected in Step 2, Active Roles hides the deprovisioned user from the Exchange address lists and configures the forwarding address for that user as prescribed by this policy.

The following two sections elaborate on the steps to implement this scenario.

Step 1: Creating and configuring the Policy Object

You can create and configure the Policy Object you need by using the New Deprovisioning Policy Object wizard. For information about the wizard, see Creating a Policy Object in the Policy Object management tasks section earlier in this chapter.

To configure the policy, click **Exchange Mailbox Deprovisioning** on the **Select Policy Type** page of the wizard. Then, click **Next**.

On the **Options to Deprovision Mailbox** page, select these check boxes:

- Hide the mailbox from the global address list (GAL), to prevent access to the mailbox.
- Modify configuration of the e-mail forwarding.

Make sure that no other check boxes on the page are selected. Then, click **Forward all incoming messages to the user's manager** and clear the **Leave copies in the mailbox** check box.

When you are done, click **Next** and follow the instructions in the wizard to create the Policy Object.

Step 2: Applying the Policy Object

You can apply the Policy Object by using the **Enforce Policy** page in the New Deprovisioning Policy Object wizard, or you can complete the wizard and then use the **Enforce Policy** command on the domain, OU, or Managed Unit where you want to apply the policy.

For more information on how to apply a Policy Object, see Applying Policy Objects and Managing policy scope earlier in this chapter.

Home Folder Deprovisioning

Policies in this category are intended to automate the following tasks on deprovisioning home folders for deprovisioned users:

- Revoke access to home folders from deprovisioned user accounts.
- Grant designated persons read access to deprovisioned home folders.
- Change ownership on deprovisioned home folders.
- Delete deprovisioned home folders.

When configuring a policy in this category, you specify how you want Active Roles to modify security on the user's home folder upon a request to deprovision a user, and whether you want Active Roles to delete home folders upon user account deletion. The purpose is to prevent deprovisioned users from accessing their home folders, and to authorize designated persons to access deprovisioned home folders.

How this policy works

When processing a request to deprovision a user, Active Roles uses this policy to determine the home folder deprovisioning options, and then updates the configuration of the user's home folder accordingly.

The available home folder deprovisioning options are summarized in the following table. For each option, the table outlines the policy effect on the user's home folder.

Table 26: Policy effect on the user's home folder

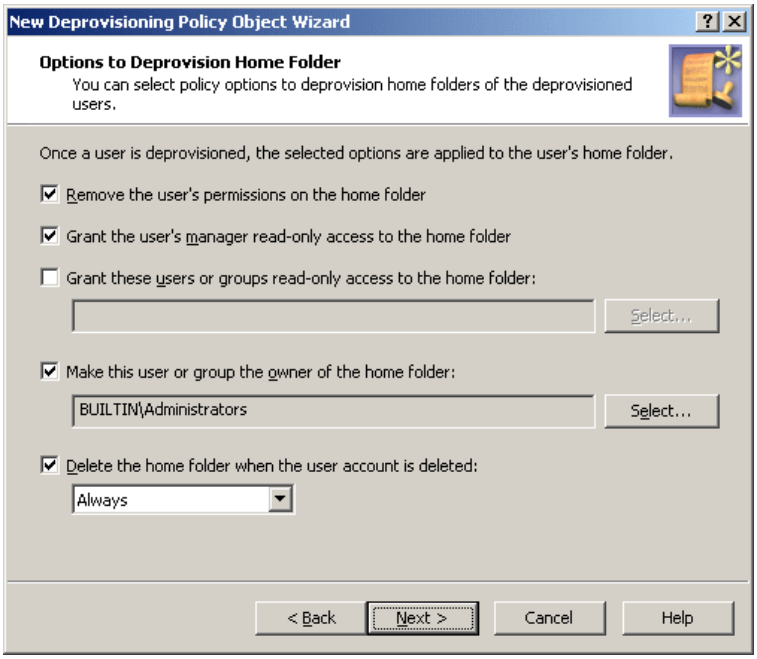
Option	Policy effect
Remove the user's permissions on the home folder	Modifies the home folder security so that the deprovisioned user cannot access his or her home folder.
Grant the user's manager read access to the home folder	Makes it possible for the person designated as the deprovisioned user's manager to view and retrieve data from the home folder of that user. The manager is determined based on the **Manager** attribute of the deprovisioned user account in Active Directory.
Grant selected users or groups read access to the home folder	Makes it possible for the specified users or groups to view and retrieve data from the deprovisioned user's home folder.
Make the selected user or group the owner of the home folder	Designates the specified user or group as the owner of the deprovisioned user's home folder. The owner is authorized to control how permissions are set on the folder, and can grant permissions to others.

Option	Policy effect
Delete the home folder when the user account is deleted	Upon the deletion of a user account, analyzes whether the user's home folder is empty, and then deletes or retains the home folder, depending on the policy configuration. A policy can be configured to only delete empty folders. Another option is to delete both empty and non-empty folders.

How to configure a Home Folder Deprovisioning policy

To configure a Home Folder Deprovisioning policy, select **Home Folder Deprovisioning** on the **Policy to Configure** page in the New Deprovisioning Policy Object wizard or in the Add Deprovisioning Policy wizard. Then, click **Next** to display the **Options to Deprovision Home Folder** page.

Figure 79: Options to Deprovision Home Folder



On this page, you can select the home folder deprovisioning options you want Active Roles to apply when deprovisioning a user.

The names of the first four options are self-explanatory. These refer to the policy options summarized in the table above:

- Remove the user's permissions on the home folder.
- Grant the user's manager read-only access to the home folder.

- Grant these users or groups read-only access to the home folder.
- Make this user or group the owner of the home folder.

Select check boxes next to the options you want to be applied.

The third option requires that you click the **Select** button to choose users or groups. The users or groups you select will be authorized to view and retrieve data from the home folders of the deprovisioned users.

The fourth option requires that you click the **Select** button to choose one user or group. The user or group you select will be authorized to control permissions on the home folders of the deprovisioned users.

You can also configure the policy to delete home folders. If you select the **Delete the home folder when the user account is deleted** check box, the policy causes Active Roles to delete the home folder once the user account associated with that folder is deleted. You can refine this behavior by selecting one of these options from the list beneath the check box:

- **If home folder is empty**. Prevents Active Roles from deleting not empty home folders. If a given home folder contains any data, the policy does not delete that folder.
- **Always**. Allows Active Roles to delete both empty and not empty home folders. Regardless of whether a given home folder contains any data, the policy deletes that folder upon user account deletion.

When you are done, click **Next** and follow the instructions in the wizard to create the Policy Object.

Steps for configuring a Home Folder Deprovisioning policy

To configure a Home Folder Deprovisioning policy

1. On the **Policy to Configure** page, select **Home Folder Deprovisioning**, and then click **Next**.
2. On the **Options to Deprovision Home Folder** page, select the options you want the policy to apply when deprovisioning a user account. You can select any combination of these options to deprovision the home folder for the deprovisioned user account:
    - Remove the user's permissions on the home folder.
    - Grant the user's manager read-only access to the home folder.
    - Grant these users or groups read-only access to the home folder.
    - Make this user or group the owner of the home folder.
    - Delete the home folder when the user account is deleted.

3. If you selected the **Grant these users or groups read-only access to the home folder** check box, click **Select** and use the **Select Objects** dialog box to specify the users or groups you want.
4. If you selected the **Make this user or group the owner of the home folder** check box, click **Select** and use the **Select Objects** dialog box to specify the user or group you want.
5. If you selected the **Delete the home folder when the user account is deleted** check box, select one of these options:
   - **Always** to have the policy delete the home folder regardless of whether the folder contains any files or sub-folders.
   - **If home folder is empty** to prevent the home folder from being deleted if it contains any files or sub-folders.
6. Click **Next**.
7. On the **Enforce Policy** page, you can specify objects to which this Policy Object is to be applied:
   - Click **Add**, and use the **Select Objects** dialog box to locate and select the objects you want.
8. Click **Next**, and then click **Finish**.

Scenario: Removing access to home folder

The policy described in this scenario performs the following functions during the user deprovisioning process:

- Removes all permissions the user had to his or her home folder.
- Designates the Administrators group as the owner of deprovisioned home folders.

To implement this scenario, you must perform the following actions:

1. Create and configure the Policy Object that defines the appropriate policy.
2. Apply the Policy Object to a domain, OU, or Managed Unit.

As a result, when deprovisioning a user account in the container you selected in Step 2, Active Roles modifies the security on the user's home folder as prescribed by this policy.

The following two sections elaborate on the steps to implement this scenario.

Step 1: Creating and configuring the Policy Object

You can create and configure the Policy Object you need by using the New Deprovisioning Policy Object wizard. For information about the wizard, see Creating a Policy Object in the Policy Object management tasks section earlier in this chapter.

To configure the policy, click **Home Folder Deprovisioning** on the **Select Policy Type** page of the wizard. Then, click **Next**.

On the **Options to Deprovision Home Folder** page, select both the **Remove the user's permissions on the home folder** and **Grant the user's manager read-only access to the home folder** check boxes.

Make sure that no other check boxes on the page are selected. Then, click **Next** and follow the instructions in the wizard to create the Policy Object.

Step 2: Applying the Policy Object

You can apply the Policy Object by using the **Enforce Policy** page in the New Deprovisioning Policy Object wizard, or you can complete the wizard and then use the **Enforce Policy** command on the domain, OU, or Managed Unit where you want to apply the policy.

For more information on how to apply a Policy Object, see Applying Policy Objects and Managing policy scope earlier in this chapter.

User Account Relocation

Policies in this category automate the movement of deprovisioned user accounts to specified organizational units. This removes such accounts from the control of administrators who are responsible for management of the organizational units in which those accounts originally reside. A policy in this category can also be configured not to move deprovisioned user accounts.

How this policy works

When processing a request to deprovision a user, Active Roles uses this policy to determine whether to move the deprovisioned user account to a different organizational unit.

A policy configured to move user accounts also specifies the destination organizational unit to which Active Roles moves deprovisioned user accounts.

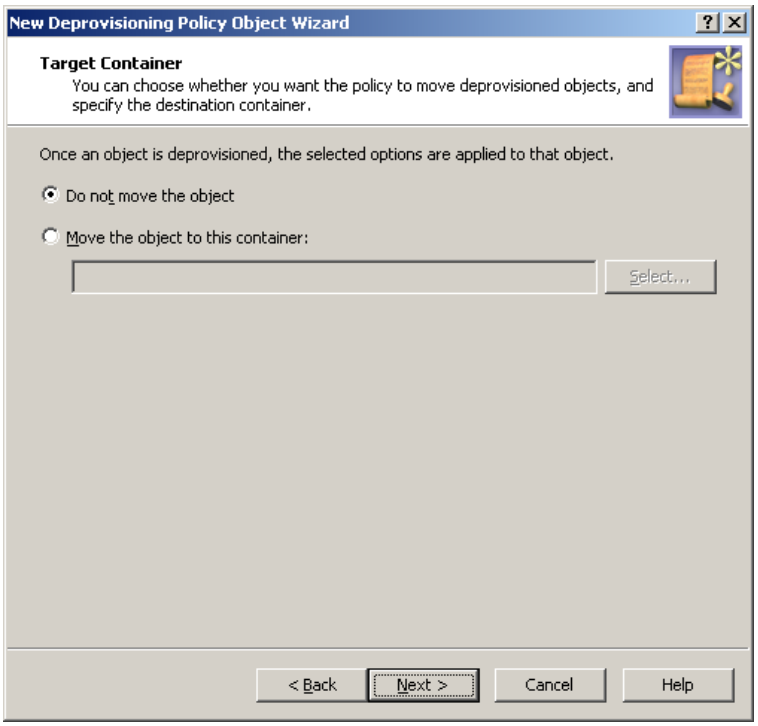
A policy can be configured not to move user accounts. When applied at a certain level of the directory hierarchy, such a policy overrides any other policy of this category applied at a higher level of the directory hierarchy.

Let us consider an example to clarify this behavior. Suppose you configure a policy to move accounts and apply that policy to a certain parent container. In general, the policy is passed down from parent to child containers, that is, the policy applies to all child containers beneath the parent container, causing Active Rolests from each container. However, if you configure a different policy not to move accounts and apply that new policy to a child container, the child container policy overrides the policy inherited from the parent container. Active Roles does not move deprovisioned user accounts from that child container or any container beneath that child container.

How to configure a User Account Relocation policy

To configure a User Account Relocation policy, select **User Account Relocation** on the **Policy to Configure** page in the New Deprovisioning Policy Object wizard or in the Add Deprovisioning Policy wizard. Then, click **Next** to display the **Target Container** page.

Figure 80: Target container



On this page, you can choose whether you want the policy to move deprovisioned user accounts, and select the destination container for the move operation.

First, select one of these options:

- Click **Do not move the object** for the policy to leave deprovisioned user accounts in their original locations. With this option, each deprovisioned user account remains in the organizational unit it was in when it was deprovisioned.
- Click **Move the object to this container** for the policy to place deprovisioned user accounts to a certain container. With this option, each deprovisioned user account is moved from its original location to a specified organizational unit.

The second option requires that you specify the organizational unit to which you want the policy to move deprovisioned user accounts. Click the **Select** button, and then choose the organizational unit you want.

When you are done, click **Next** and follow the instructions in the wizard to create the Policy Object.

Steps for configuring a User Account Relocation policy

To configure a User Account Relocation policy

1. On the **Policy to Configure** page, select **User Account Relocation**, and then click **Next**.
2. On the **Target Container** page, do one of the following, and then click **Next**:
   - Click **Do not move the object** if you want the policy to keep deprovisioned user accounts in their original locations.
   - Click **Move the object to this container** if you want the policy to move deprovisioned user accounts to a certain container. Then, click **Select**, and select the container you want.
3. On the **Enforce Policy** page, you can specify objects to which this Policy Object is to be applied:
   - Click **Add**, and use the **Select Objects** dialog box to locate and select the objects you want.
4. Click **Next**, and then click **Finish**.

Scenario: Organizational Unit for deprovisioned user accounts

This scenario describes how to configure a policy so that a certain organizational unit contains all the deprovisioned user accounts.

To implement this scenario, you must perform the following actions:

1. Create and configure the Policy Object that defines the appropriate policy.
2. Apply the Policy Object to a domain, OU, or Managed Unit.

As a result, after deprovisioning a user account in the container you selected in Step 2, Active Roles automatically moves that account to the organizational unit determined by the policy configuration. The following two sections elaborate on the steps to implement this scenario.

Step 1: Creating and configuring the Policy Object

You can create and configure the Policy Object you need by using the New Deprovisioning Policy Object wizard. For information about the wizard, see Creating a Policy Object in the Policy Object management tasks section earlier in this chapter.

To configure the policy, click **User Account Permanent Deletion** on the **Select Policy Type** page of the wizard. Then, click **Next**.

On the **Deletion Options** page, click **Delete the object after retention period**. Then, in the box beneath that option, type **90**.

When you are done, click **Next** and follow the instructions in the wizard to create the Policy Object.

Step 2: Applying the Policy Object

You can apply the Policy Object by using the **Enforce Policy** page in the New Deprovisioning Policy Object wizard, or you can complete the wizard and then use the **Enforce Policy** command on the domain, OU, or Managed Unit where you want to apply the policy.

For more information on how to apply a Policy Object, see Applying Policy Objects and Managing policy scope earlier in this chapter.

User Account Permanent Deletion

Policies in this category are intended to automate the deletion of deprovisioned user accounts. Deprovisioned user accounts are retained for a specified amount of time before they are permanently deleted. A policy in this category can also be configured not to delete deprovisioned accounts.

How this policy works

When processing a request to deprovision a user, Active Roles uses this policy to determine whether to schedule the deprovisioned user account for deletion. When scheduled for deletion, a user account is permanently deleted after a certain time period, referred to as a retention period.

A policy configured to delete user accounts specifies the number of days to retain deprovisioned user accounts. With such a policy, Active Roles permanently deletes a user account after the specified number of days has passed since the user was deprovisioned.

A policy can be configured not to delete user accounts. When applied at a certain level of the directory hierarchy, such a policy overrides any other policy of this category applied at a higher level of the directory hierarchy.

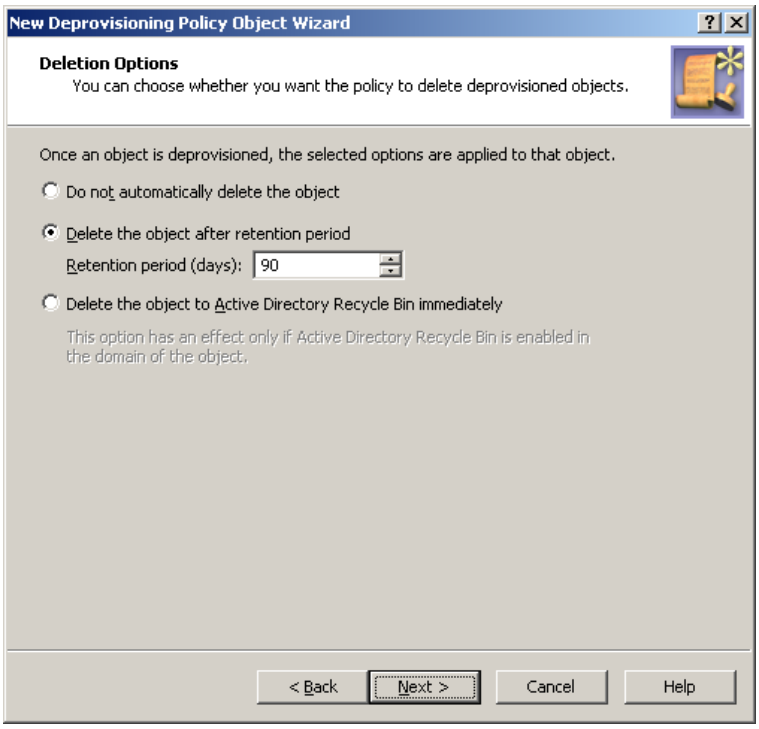
Let us consider an example to clarify this behavior. Suppose you configure a policy to delete accounts and apply that policy to a certain container. In general, the policy is passed down from parent to child containers, that is, the policy applies to all child containers beneath the parent container, causing Active Roles to delete deprovisioned user accounts in each container. However, if you configure a different policy not to delete accounts and apply that new policy to a child container, the child container policy overrides the policy inherited from the parent container. Active Roles does not delete deprovisioned user accounts in that child container or any container beneath that child container.

One more option of this policy is intended for domains where Active Directory Recycle Bin is enabled. The policy can be configured so that once a user account is deprovisioned, the account is moved to Recycle Bin (which effectively means that the account will be deleted immediately, without any retention period). Moving deprovisioned user accounts to the Recycle Bin may be required for security reasons, as an extra security precaution. The Active Directory Recycle Bin ensures that the account can be restored, if necessary, without any loss of data. Active Roles provides the ability to un-delete and then un-deprovision user accounts that were deprovisioned to the Recycle Bin.

How to configure a User Account Permanent Deletion policy

To configure a User Account Permanent Deletion policy, select **User Account Permanent Deletion** on the **Policy to Configure** page in the New Deprovisioning Policy Object wizard or in the Add Deprovisioning Policy wizard. Then, click **Next** to display the **Deletion Options** page.

Figure 81: Deletion options



On this page, you can choose whether you want the policy to schedule deprovisioned user accounts for deletion, and specify the number of days to retain deprovisioned user accounts.

First, select one of these options:

- Click **Do not automatically delete the object** if you want the policy not to delete deprovisioned user accounts.
- Click **Delete the object after retention period** if you want the policy to schedule deprovisioned user accounts for deletion.
- Click **Delete the object to Active Directory Recycle Bin immediately** if you want the policy to move deprovisioned user accounts to Recycle Bin.

If you select the second option, you must specify a number of days in the box beneath that option. Once a user account has been deprovisioned, and the specified number of days has passed, the policy causes Active Roles to delete the user account in Active Directory.

If you select the third option, you should apply this policy to domains that have Active Directory Recycle Bin enabled; otherwise, the policy will have no effect. With this option, once a user account has been deprovisioned, the policy causes Active Roles to delete the user account immediately. In a domain where Active Directory Recycle Bin is enabled, this deletion merely means that the account is marked as deleted and moved to a certain container from which it can be restored, if necessary, without any data loss.

Steps for configuring a User Account Permanent Deletion policy

To configure a User Account Permanent Deletion policy

1. On the **Policy to Configure** page, select **User Account Permanent Deletion**, and then click **Next**.
2. On the **Deletion Options** page, do one the following, and then click **Next**:
   - Click **Do not automatically delete the object** if you want the policy not to delete deprovisioned user accounts.
   - Click **Delete the object after retention period** if you want the policy to schedule deprovisioned user accounts for deletion. Then, in **Retention period (days)**, specify the number of days to retain the deprovisioned user account before it is deleted.
   - Click **Delete the object to Active Directory Recycle Bin immediately** if you want the policy to move deprovisioned user accounts to Recycle Bin.

   Click **Next**.

   If you select the third option, you should apply this policy to domains that have Active Directory Recycle Bin enabled; otherwise, the policy will have no effect. With this option, once a user account has been deprovisioned, the policy causes Active Roles to delete the user account immediately. In a domain where Active Directory Recycle Bin is enabled, this deletion merely means that the account is marked as deleted and moved to a certain container from which it can be restored, if necessary, without any data loss.
3. On the **Enforce Policy** window, you can specify objects to which this Policy Object is to be applied:

- Click **Add**, and use the **Select Objects** dialog box to locate and select the objects you want.

4. Click **Next**, and then click **Finish**.

Scenario: Deleting deprovisioned user accounts

This scenario describes how to configure a policy so that Active Roles permanently deletes deprovisioned user accounts after the 90-day retention period.

To implement this scenario, you must perform the following actions:

1. Create and configure the Policy Object that defines the appropriate policy.
2. Apply the Policy Object to a domain, OU, or Managed Unit.

As a result, after deprovisioning a user account in the container you selected in Step 2, Active Roles retains the deprovisioned account for 90 days and then it deletes that account.

Step 1: Creating and configuring the Policy Object

You can create and configure the Policy Object you need by using the New Deprovisioning Policy Object wizard. For information about the wizard, see Creating a Policy Object in the Policy Object management tasks section earlier in this chapter.

To configure the policy, click **User Account Permanent Deletion** on the **Select Policy Type** page of the wizard. Then, click **Next**.

On the **Deletion Options** page, click **Delete the object after retention period**. Then, in the box beneath that option, type **90**.

When you are done, click **Next** and follow the instructions in the wizard to create the Policy Object.

Step 2: Applying the Policy Object

You can apply the Policy Object by using the **Enforce Policy** page in the New Deprovisioning Policy Object wizard, or you can complete the wizard and then use the **Enforce Policy** command on the domain, OU, or Managed Unit where you want to apply the policy.

For more information on how to apply a Policy Object, see Applying Policy Objects and Managing policy scope earlier in this chapter.

Group Object Deprovisioning

Group object deprovisioning policy specifies the changes to make to the group object in Active Directory in order to prevent the use of the group. It is intended to perform the following tasks when deprovisioning a group:

- Hide the group from the Global Address List (GAL) to prevent access to the group from Exchange Server client applications such as Microsoft Outlook.
- Change the type of the group from Security to Distribution to revoke access rights from the group.
- Rename the group, to distinguish deprovisioned groups by name.
- Remove members from the group to revoke user access to resources controlled by the group. This task has the option to specify the members that should not be removed from the group.

In addition, the policy can be configured to change or clear any other properties of a group, such as the pre-Windows 2000 name, e-mail addresses, or description.

How this policy works

When processing a request to deprovision a group, Active Roles uses this policy to modify the group object in Active Directory so that once the group has been deprovisioned it cannot be used.

A policy can also be configured to update individual properties of groups. Depending on the policy configuration, each policy-based update results in the following:

- Certain portions of group information, such as information about group members, are removed from the directory.
- Certain properties of groups are changed or cleared.

A policy can be configured so that new property values include:

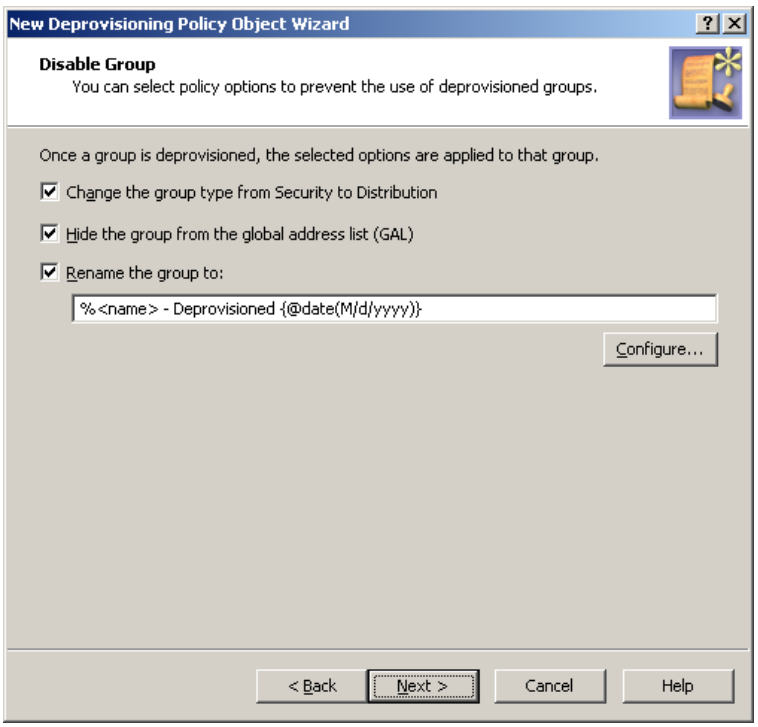
- Properties of the group being deprovisioned, retrieved from the directory prior to starting the process of the group deprovisioning
- Properties of the user who originated the deprovisioning request
- Date and time when the group was deprovisioned

Thus, when deprovisioning a group, Active Roles modifies the group object in Active Directory as determined by the Group Object Deprovisioning policy that is in effect.

How to configure a Group Object Deprovisioning policy

To configure a Group Object Deprovisioning policy, select **Group Object Deprovisioning** on the **Policy to Configure** page in the New Deprovisioning Policy Object wizard or in the Add Deprovisioning Policy wizard. Then, click **Next** to display the **Disable Group** page.

Figure 82: Disable Group



On this page, you can choose from the following options:

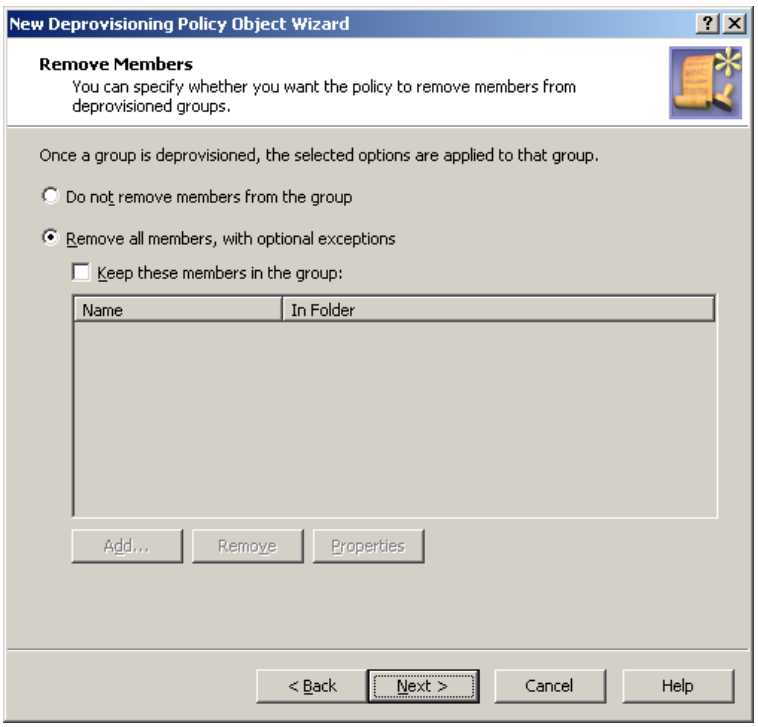
- **Change the group type from Security to Distribution**. Revokes access rights from deprovisioned groups. This option is applicable only to security groups.
- **Hide the group from the Global Address List (GAL)**. Prevents access to deprovisioned groups from Exchange Server client applications. This option is applicable to distribution groups or mail-enabled security groups.
- **Rename the group to**. Changes the name of the group.

Select the check box next to each option you want the policy to apply.

By selecting the **Rename the group to** check box, you direct the policy to change the name of the group. This option allows you to configure a property update rule that specifies how to modify the group name. Click the **Configure** button and follow the instructions provided in the Configuring a property update rule section, earlier in this chapter.

Once you have completed the **Disable Group** page, click **Next** to display the **Remove Members** page.

Figure 83: Remove members



On this page, you can configure a rule on how to remove members from deprovisioned groups.

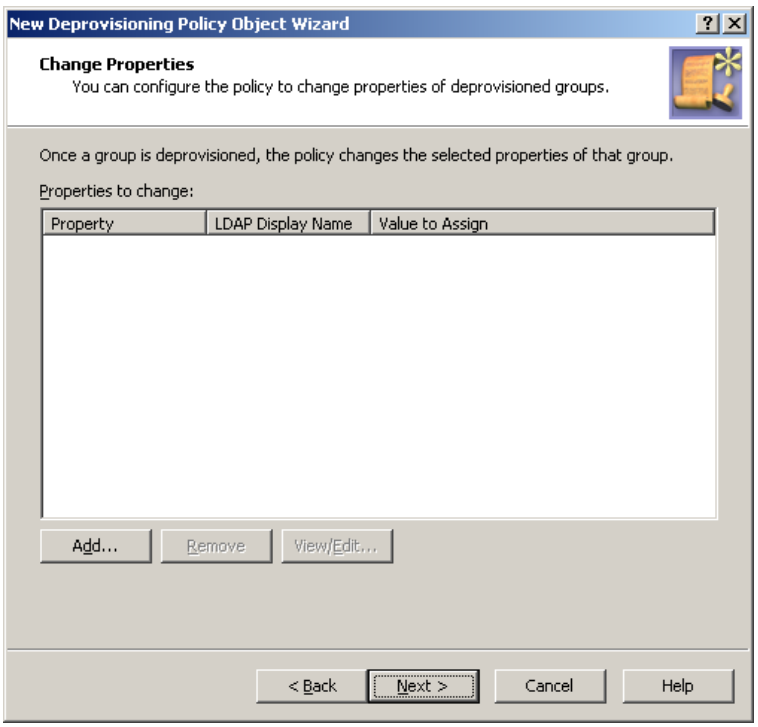
Select one of these options:

- Click **Do not remove members from the group** if you want the policy not to remove members from deprovisioned groups.
- Click **Remove all members, with optional exceptions** if you want the policy to remove members from deprovisioned groups.

With the second option, you can specify whether you want the policy not to remove certain objects from deprovisioned groups. To set up a list of such objects, select the **Keep these objects in the group** check box, and then click the **Add** button and select the objects you want to include in the list.

Once you have completed the **Remove Members** page, click **Next** to display the **Change Properties** page.

Figure 84: Change Properties



On this page, you can set up a list of group properties you want the policy to update. Each entry in the list includes the following information:

- **Property**. When deprovisioning a group, Active Roles will update this property of the group object in Active Directory.
- **LDAP Display Name**. Uniquely identifies the property to be updated.
- **Value to Assign**. After the deprovisioning operation is completed, the property has the value defined by the rule specified.

You can use these buttons to manage the list on this page:

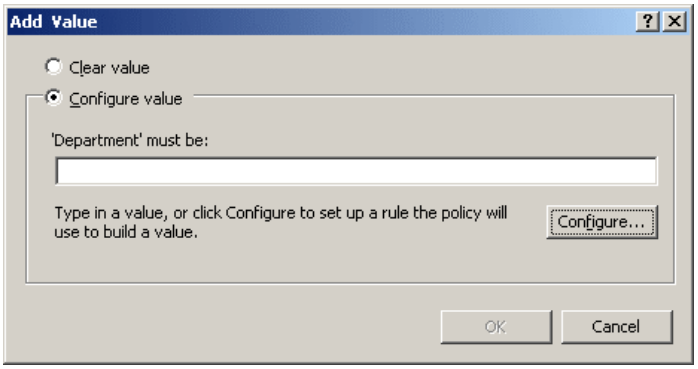
- **Add**. Allows you to select a property and configure an update rule for that property. A property update rule specifies how to generate the new value to assign to the property.
- **Remove**. If you want the policy to no longer update a given property, select the property from the list and click **Remove**.
- **View/Edit**. Allows you to modify the update rule for the property you select from the list.

Clicking the **Add** button displays the **Select Object Property** dialog box where you can choose group properties you want to the policy to update. To choose a property, select the check box next to the property name, and then click **OK**.

You can select multiple check boxes. If you do so, the properties you have selected are added to the list on the wizard page, with the update rule configured to clear those properties, that is, to assign them the empty value.

If you select a single property in the **Select Object Property** dialog box, you are presented with the **Add Value** dialog box so you can proceed to configuring a property update rule.

Figure 85: Add value



You can select one of these update options:

- **Clear value**. Causes the policy to assign the "empty" value to the property.
- **Configure value**. Allows you to configure a value for the **'property' must be** condition.

With the second option, you must configure a value the policy will assign to the property upon the group deprovisioning. You can configure a value in the same way as you do when configuring a property update rule for the user name: Click the **Configure** button and follow the instructions provided in the Configuring a property update rule section, earlier in this chapter.

When you are done configuring a value, click **OK** to close the **Add Value** dialog box. The property name along with the property update rule is added to the wizard page. If necessary, you can modify the update rule by clicking the **View/Edit** button beneath the list of properties. This displays a dialog box, similar to the **Add Value** dialog box, allowing you to choose a different update option or set up a different value for the **'property' must be** condition.

Once you have set up the list on the wizard page, click **Next** and follow the instructions in the wizard to create the Policy Object.

Steps for configuring a Group Object Deprovisioning policy

To configure a Group Object Deprovisioning policy

1. On the **Policy to Configure** page, select **Group Object Deprovisioning**, and then click **Next**.
2. On the **Disable Group** page, select the options you want the policy to apply when deprovisioning a group. You can select any combination of these options to prevent

the use of the group:

- Change the group type from Security to Distribution.
- Hide the group from the Global Address List (GAL).
- Rename the group.

3. If you selected **Rename the group to**, click **Configure**, and then complete the **Configure Value** dialog box by using the procedure outlined later in this topic, in order to specify how you want the policy to update the group name when deprovisioning a group.
4. Click **Next**.
5. On the **Remove Members** page, do one of the following:
   - Click **Do not remove members from the group** for the policy not to make changes to the membership list of the group.
   - Click **Remove all members, with optional exceptions** for the policy to remove the members from the group.
6. If you selected the second option in Step 5, specify whether you want the policy not to remove certain objects from deprovisioned groups. Do the following:
   - Select the **Keep these objects in the group** check box and set up the list of the objects you want the policy not to remove from deprovisioned groups.
   - Leave the check box cleared if you want the policy to remove all members from deprovisioned groups.
7. On the **Change Properties** page, specify how you want the policy to update properties of the group object when deprovisioning a group:
   - Click **Add**, and then complete the **Select Object Property** dialog box by using the procedure outlined later in this topic, in order to add property update rules.
   - Use **View/Edit** to modify existing rules.
   - Use **Remove** to delete existing rules.
8. Click **Next**.
9. On the **Enforce Policy** page, you can specify objects to which this Policy Object is to be applied:
   - Click **Add**, and use the **Select Objects** dialog box to locate and select the objects you want.
10. Click **Next**, and then click **Finish**.

To complete the Configure Value dialog box

1. Click **Add**.
2. Configure an entry to include in the value (for instructions, see Steps for configuring entries).
3. In the **Configure Value** dialog box, add more entries, delete or edit existing ones, and then click **OK**.

To complete the Select Object Property dialog box

1. From the **Object property** list, select an object property, and then click **OK**. The **Add Value** dialog box appears.

   If you select multiple properties, the **Add Value** dialog box is not displayed. The properties you have selected are added to the list on the **Change Properties** page, with the update rule configured to clear those properties, that is, to assign them the empty value.

2. In the **Add Value** dialog box, do one of the following:
   - Select **Clear value** if you want the update rule to assign the empty value to the property.
   - Select **Configure value** if you want the update rule to assign a certain, non-empty value to the property. Then, click **Configure** and complete the **Configure Value** dialog box by using the instructions given earlier in this topic.

Scenario 1: Disabling and renaming the group upon deprovisioning

The policy described in this scenario performs the following functions during the group deprovisioning process:

- When deprovisioning a security group, change the type of the group to Distribution.
- When deprovisioning a distribution group, remove the group from the Global Address List.
- Append this suffix to the group name: **- Deprovisioned**, followed by the date when the group was deprovisioned.

For example, the policy changes the group name of **Partner Marketing** to **Partner Marketing - Deprovisioned 12/11/2013**.

To implement this scenario, you must perform the following actions:

1. Create and configure the Policy Object that defines the appropriate policy.
2. Apply the Policy Object to a domain, OU, or Managed Unit.

As a result, when deprovisioning a group in the container you selected in Step 2, Active Roles disables and renames the group as prescribed by this policy.

The following two sections elaborate on the steps to implement this scenario.

Step 1: Creating and configuring the Policy Object

You can create and configure the Policy Object you need by using the New Deprovisioning Policy Object wizard. For information about the wizard, see Creating a Policy Object in the Policy Object management tasks section earlier in this chapter.

To configure the policy, click **Group Object Deprovisioning** on the **Select Policy Type** page of the wizard. Then, click **Next**.

On the **Disable Group** page, select these check boxes:

- **Change the group type from Security to Distribution**
- **Hide the group from the Global Address List (GAL)**
- **Rename the group to**

Then, type the following text in the box beneath the **Rename the group** to option:

```
%<name> - Deprovisioned {@date(M/d/yyyy)}
```

Click **Next** and follow the instructions in the wizard to create the Policy Object.

Step 2: Applying the Policy Object

You can apply the Policy Object by using the **Enforce Policy** page in the New Deprovisioning Policy Object wizard, or you can complete the wizard and then use the **Enforce Policy** command on the domain, OU, or Managed Unit where you want to apply the policy.

For more information on how to apply a Policy Object, see Applying Policy Objects and Managing policy scope earlier in this chapter.

Scenario 2: Managed Unit for deprovisioned groups

This scenario describes how to configure a Managed Unit and a Group Object Deprovisioning policy so that the Managed Unit includes all deprovisioned groups. The policy sets the **Notes** property to **Deprovisioned** upon the deprovisioning of a group, whereas the Managed Unit is configured to include the groups that have the **Notes** property set to **Deprovisioned**.

To implement this scenario, you must perform the following actions:

1. Create and configure the Managed Unit.
2. Configure the Policy Object that defines the appropriate policy.
3. Apply the Policy Object to a domain, OU, or Managed Unit.

As a result, after deprovisioning a group in the container you selected in Step 3, Active Roles automatically adds that group to the Managed Unit you created in Step 1.

The following sections elaborate on the steps to implement this scenario.

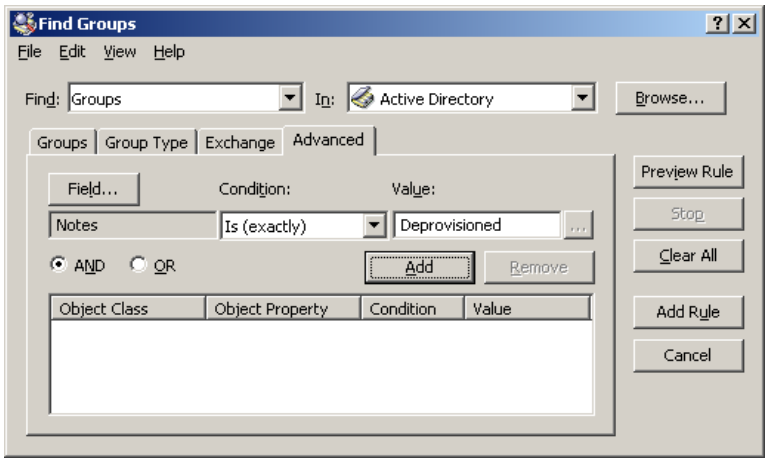
Step 1: Creating and configuring the Managed Unit

You can create and configure the Managed Unit by using the Active Roles console:

1. In the console tree, under **Configuration**, right-click **Managed Units**, and select **New | Managed Unit**.
2. In **Name**, type a name for the Managed Unit. For example, you might type **Deprovisioned Users**.
3. Click **Next**.
4. Configure the membership rule to have the Managed Unit include the deprovisioned user accounts from all domains that are registered with Active Roles (managed domains):
   a. On the wizard page, click **Add**.
   b. In the **Membership Rule Type** dialog box, click **Include by Query**, and then click **OK**.
   c. Use the **Create Membership Rule** window to set up the rule:
      - In **Find**, click **Users**.
      - Click **Browse** and select **Active Directory**.
      - Click the **Advanced** tab.
      - Click the **Field** button, and then click **Notes**.
      - In **Condition**, click **Is (exactly)**.
      - In **Value**, type **Deprovisioned**.

   At this stage, the window should look like the following figure.

   Figure 86: Find Groups



      - Click the **Add** button.
      - Click the **Add Rule** button.
   d. On the wizard page, click **Add**.
   e. In the **Membership Rule Type** dialog box, click **Retain Deprovisioned**, and then click **OK**.
5. Click **Next**, click **Next**, and then click **Finish**.

Step 2: Configuring the Policy Object

You can configure the Policy Object you need by modifying the Policy Object that implements the previous scenario, see Scenario 1: Disabling and renaming the user account upon deprovisioning earlier in this section.

Display the **Properties** dialog box for that Policy Object and go to the **Policies** tab. Then, select the policy from the list, and click **View/Edit** to display the **Group Object Deprovisioning Policy Properties** dialog box. Click the **Change Properties** tab.

The **Change Properties** tab looks similar to the page of the same name in the wizard you used to create the Policy Object. You can use that tab to add the update rule for the **Notes** property:

1. Click **Add** to display the **Select Object Property** dialog box.
2. Select the check box next to the **Notes** property, and then click **OK**.
3. In the **Add Value** dialog box, type **Deprovisioned** in the **'Notes' must be** box, and then click **OK**.

Click **OK** to close the **Group Object Deprovisioning Policy Properties** dialog box.

Step 3: Applying the Policy Object

You can apply the Policy Object by using the **Enforce Policy** page in the New Provisioning Policy Object wizard, or you can complete the wizard and then use the **Enforce Policy** command on the domain, OU, or Managed Unit where you want to apply the policy. For more information on how to apply a Policy Object, see Applying Policy Objects and Managing policy scope earlier in this chapter.

Group Object Relocation

Policies in this category are intended to automate the movement of deprovisioned group objects to specified organizational units. This removes such groups from the control of administrators that are responsible for management of the organizational units in which those groups originally reside. A policy in this category can also be configured not to move deprovisioned group objects.

How this policy works

When processing a request to deprovision a group, Active Roles uses this policy to determine whether to move the deprovisioned group object to a different organizational unit.

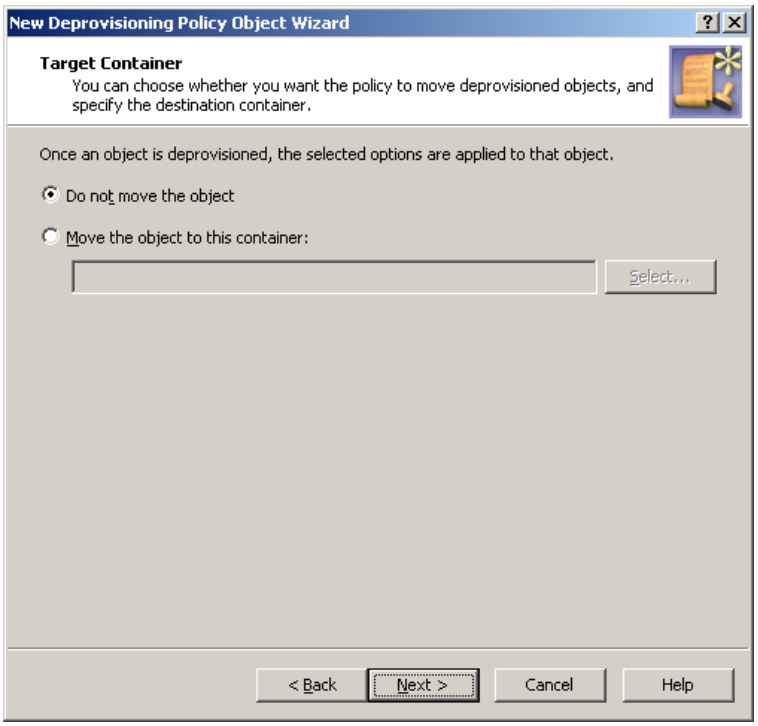
A policy configured to move group objects also specifies the destination organizational unit to which Active Roles moves deprovisioned group objects.

A policy can be configured not to move group objects. When applied at a certain level of the directory hierarchy, such a policy overrides any other policy of this category applied at a higher level of the directory hierarchy.

How to configure a Group Object Relocation policy

To configure a Group Object Relocation policy, select **Group Object Relocation** on the **Policy to Configure** page in the New Deprovisioning Policy Object wizard or in the Add Deprovisioning Policy wizard. Then, click **Next** to display the **Target Container** page.

Figure 87: Target container



On this page, you can choose whether you want the policy to move deprovisioned group objects, and select the destination container for the move operation.

First, select one of these options:

- Click **Do not move the object** for the policy to leave deprovisioned group objects in their original locations. With this option, each deprovisioned group object remains in the organizational unit it was in when it was deprovisioned.
- Click **Move the object to this container** for the policy to place deprovisioned group objects to a certain container. With this option, each deprovisioned group object is moved from its original location to a specified organizational unit.

The second option requires that you specify the organizational unit to which you want the policy to move deprovisioned group objects. Click the **Select** button, and then choose the organizational unit you want.

When you are done, click **Next** and follow the instructions in the wizard to create the Policy Object.

Steps for configuring a Group Object Relocation policy

To configure a Group Object Relocation policy

1. On the **Policy to Configure** page, select **Group Object Relocation**, and then click **Next**.
2. On the **Target Container** page, do one of the following, and then click **Next**:
   - Click **Do not move the object** if you want the policy to keep deprovisioned group objects in their original locations.
   - Click **Move the object to this container** if you want the policy to move deprovisioned group objects to a certain container. Then, click **Select**, and select the container you want.
3. On the **Enforce Policy** page, you can specify objects to which this Policy Object is to be applied:
   - Click **Add**, and use the **Select Objects** dialog box to locate and select the objects you want.
4. Click **Next**, and then click **Finish**.

Scenario: Organizational Unit for deprovisioned groups

This scenario describes how to configure a policy so that a certain organizational unit contains all the deprovisioned groups.

To implement this scenario, you must perform the following actions:

1. Create and configure the Policy Object that defines the appropriate policy.
2. Apply the Policy Object to a domain, OU, or Managed Unit.

As a result, after deprovisioning a group in the container you selected in Step 2, Active Roles automatically moves that group to the organizational unit determined by the policy configuration. The following two sections elaborate on the steps to implement this scenario.

Step 1: Creating and configuring the Policy Object

You can create and configure the Policy Object you need by using the New Deprovisioning Policy Object wizard. For information about the wizard, see Creating a Policy Object in the Policy Object management tasks section earlier in this chapter.

To configure the policy, click **Group Object Relocation** on the **Select Policy Type** page of the wizard. Then, click **Next**.

On the **Target Container** page, click **Move the object to this container**. Then, click the **Select** button to display the **Browse for Container** dialog box. Locate and select the organizational unit to which you want the policy to move deprovisioned groups, and then click **OK**.

When you are done, click **Next** and follow the instructions in the wizard to create the Policy Object.

Step 2: Applying the Policy Object

You can apply the Policy Object by using the **Enforce Policy** page in the New Deprovisioning Policy Object wizard, or you can complete the wizard and then use the **Enforce Policy** command on the domain, OU, or Managed Unit where you want to apply the policy.

For more information on how to apply a Policy Object, see Applying Policy Objects and Managing policy scope earlier in this chapter.

Group Object Permanent Deletion

Policies in this category are intended to automate the deletion of deprovisioned groups. Deprovisioned group objects are retained for a specified amount of time before they are permanently deleted. A policy in this category can also be configured not to delete deprovisioned group objects.

How this policy works

When processing a request to deprovision a group, Active Roles uses this policy to determine whether to schedule the deprovisioned group object for deletion. When scheduled for deletion, a group object is permanently deleted after a certain time period, referred to as a retention period.

A policy configured to delete groups specifies the number of days to retain deprovisioned group objects. With such a policy, Active Roles permanently deletes a group after the specified number of days has passed since the group was deprovisioned.

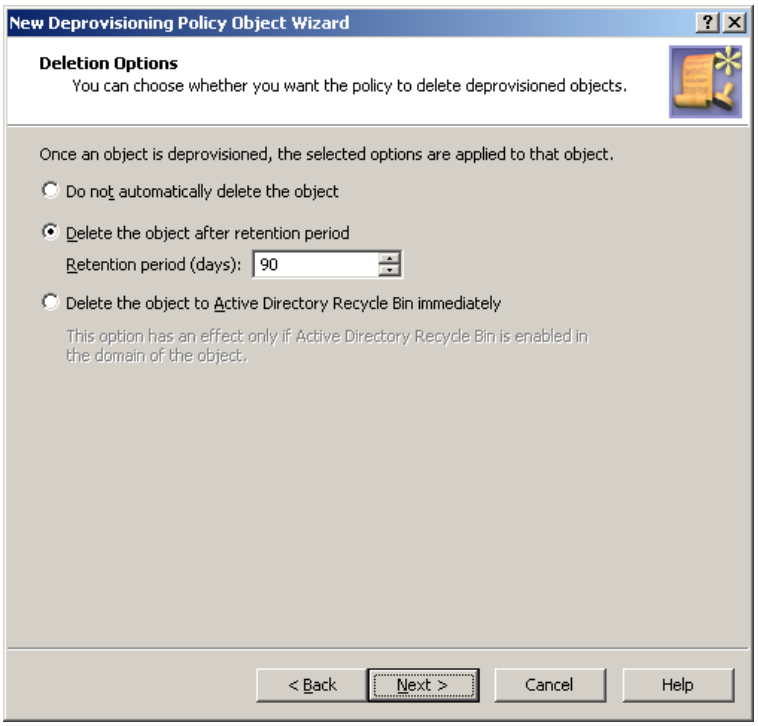
A policy can be configured not to delete groups. When applied at a certain level of the directory hierarchy, such a policy overrides any other policy of this category applied at a higher level of the directory hierarchy.

One more option of this policy is intended for domains where Active Directory Recycle Bin is enabled. The policy can be configured so that once a group is deprovisioned, the group object is moved to the Recycle Bin (which effectively means that the group will be deleted immediately, without any retention period). Moving deprovisioned group objects to the Recycle Bin may be required for security reasons, as an extra security precaution. The Active Directory Recycle Bin ensures that the group object can be restored, if necessary, without any loss of data. Active Roles provides the ability to un-delete and then un-deprovision groups that were deprovisioned to the Recycle Bin.

How to configure a Group Object Permanent Deletion policy

To configure a Group Object Permanent Deletion policy, select **Group Object Permanent Deletion** on the **Policy to Configure** page in the New Deprovisioning Policy Object wizard or in the Add Deprovisioning Policy wizard. Then, click **Next** to display the **Deletion Options** page.

Figure 88: Deletion Options



On this page, you can choose whether you want the policy to schedule deprovisioned groups for deletion, and specify the number of days to retain deprovisioned group objects.

First, select one of these options:

- Click **Do not automatically delete the object** if you want the policy not to delete deprovisioned groups.
- Click **Delete the object after retention period** if you want the policy to schedule deprovisioned groups for deletion.
- Click **Delete the object to Active Directory Recycle Bin immediately** if you want the policy to move deprovisioned group objects to Recycle Bin.

If you select the second option, you must specify a number of days in the box beneath that option. Once a group has been deprovisioned, and the specified number of days has passed, the policy causes Active Roles to delete the group object in Active Directory.

If you select the third option, you should apply this policy to domains that have Active Directory Recycle Bin enabled; otherwise, the policy will have no effect. With this option, once a group has been deprovisioned, the policy causes Active Roles to delete the group object immediately. In a domain where Active Directory Recycle Bin is enabled, this deletion merely means that the object is marked as deleted and moved to a certain container from which it can be restored, if necessary, without any data loss.

Steps for configuring a Group Object Permanent Deletion policy

To configure a Group Object Permanent Deletion policy

1. On the **Policy to Configure** page, select **Group Object Permanent Deletion**. and the click **Next**.
2. On the **Deletion Options** page, do one the following:
   - Click **Do not automatically delete the object** if you want the policy not to delete deprovisioned groups.
   - Click **Delete the object after retention period** if you want the policy to schedule deprovisioned groups for deletion. Then, in **Retention period (days)**, specify the number of days to retain the deprovisioned group before it is deleted.
   - Click **Delete the object to Active Directory Recycle Bin immediately** if you want the policy to move deprovisioned group objects to Recycle Bin.

   Click **Next**.

   If you select the third option, you should apply this policy to domains that have Active Directory Recycle Bin enabled; otherwise, the policy will have no effect. With this option, once a group has been deprovisioned, Active Roles deletes the deprovisioned group immediately. In a domain where Active Directory Recycle Bin is enabled, this means that the group object is marked as deleted and moved to a certain container from which it can be restored, if necessary, without any data loss.
3. On the **Enforce Policy** window, you can specify objects to which this Policy Object is to be applied:

- Click **Add**, and use the **Select Objects** dialog box to locate and select the objects you want.

4. Click **Next**, and then click **Finish**.

Scenario: Deleting deprovisioned groups

This scenario describes how to configure a policy so that Active Roles permanently deletes deprovisioned groups after the 90-day retention period.

To implement this scenario, you must perform the following actions:

1. Create and configure the Policy Object that defines the appropriate policy.
2. Apply the Policy Object to a domain, OU, or Managed Unit.

As a result, after deprovisioning a group in the container you selected in Step 2, Active Roles retains the deprovisioned group object for 90 days and then it deletes that object.

Step 1: Creating and configuring the Policy Object

You can create and configure the Policy Object you need by using the New Deprovisioning Policy Object wizard. For information about the wizard, see Creating a Policy Object in the Policy Object management tasks section earlier in this chapter.

To configure the policy, click **Group Object Permanent Deletion** on the **Select Policy Type** page of the wizard. Then, click **Next**.

On the **Deletion Options** page, click **Delete the object after retention period**. Then, in the box beneath that option, type **90**.

When you are done, click **Next** and follow the instructions in the wizard to create the Policy Object.

Step 2: Applying the Policy Object

You can apply the Policy Object by using the **Enforce Policy** page in the New Deprovisioning Policy Object wizard, or you can complete the wizard and then use the **Enforce Policy** command on the domain, OU, or Managed Unit where you want to apply the policy.

For more information on how to apply a Policy Object, see Applying Policy Objects and Managing policy scope earlier in this chapter.

Notification Distribution

Policies in this category are intended to automatically generate and send e-mail notifications upon deprovisioning requests. The primary purpose of such a policy is to notify designated persons about a request to deprovision a given object so as to take

additional deprovisioning-related actions on that object, if necessary. When configuring a policy in this category, you can specify a list of notification recipients, and customize the subject and body of the notification message.

How this policy works

When processing a deprovisioning request, Active Roles uses this policy to determine whether anyone must be notified of the deprovisioning operation that is requested. Then, it generates a notification message and sends it to the recipients, if any specified in the policy configuration.

When a deprovisioning operation is requested, Active Roles issues a notification message regardless of operation results. Hence, a notification message cannot be considered as an indication of success or failure of the operation. Rather, it only indicates that deprovisioning has been requested. If you need to inform anybody of deprovisioning results, you should use a policy of the Report Distribution category, discussed in the next section.

Notification performs on a per-object basis: Each notification message contains information about a request to deprovision one object. When deprovisioning multiple objects, Active Roles sends multiple notification messages, one message per object.

Active Roles sends notification messages via an SMTP server. The policy configuration specifies the outbound SMTP server by using Active Roles e-mail settings that include the name of the SMTP server and information required to connect to the SMTP server.

How to configure a Notification Distribution policy

To configure a Notification Distribution policy, select **Notification Distribution** on the **Policy to Configure** page in the New Deprovisioning Policy Object wizard or in the Add Deprovisioning Policy wizard. Then, click **Next** to display the **Notification Recipients and Message** page.

On the **Notification Recipients and Message** page, you can set up a list of notification recipients, and make any necessary changes to the message subject and body.

To specify notification recipients, click the button next to the **Notification recipients** box, and then type one or more e-mail addresses. Use a semicolon to separate the addresses of the recipients.

If you need to modify the notification message, type in the **Message subject** or **Message body** box. You can use macros to enter information about the object being deprovisioned, to make the message more meaningful to the recipients.

Macros have the same syntax and semantics as values for policy conditions in Property Generation and Validation policies: An attribute's LDAP display name enclosed in angle brackets (<>) and prefixed with the percent character (%) represents the value of that attribute. For instance, before sending a message, Active Roles replaces %<name> with the name of the object to deprovision.

When you are done, click **Next** to display the **Outgoing Mail Server** page.

On the **Outgoing Mail Server** page, you can select the e-mail configuration you want the policy to use, and view or modify e-mail settings in the selected configuration.

First, from the **Outgoing mail server (SMTP)** list, select the e-mail configuration you want the policy to use.

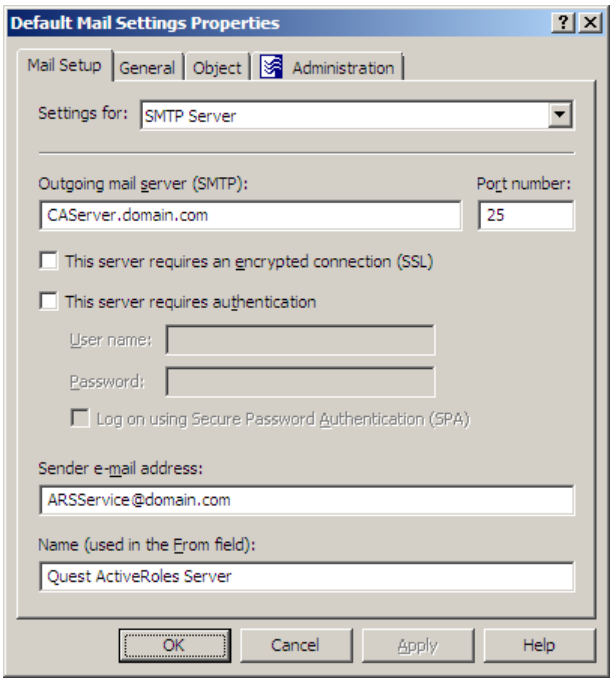
> NOTE: By default, the **Outgoing mail server (SMTP)** list includes a single entry. You can add more entries to the list using the Active Roles console. In the console tree, expand **Configuration/Server Configuration**, right-click **Mail Configuration**, select **New | Mail Configuration**, and then follow the instructions in the wizard.

Each e-mail configuration specifies an SMTP server and provides information required to connect to that server. You can view and modify configuration parameters by clicking the **Settings** button.

Configuring e-mail settings

When you click the **Settings** button, the console displays the **Properties** dialog box for the selected e-mail configuration, with the **Mail Setup** tab that looks like the following figure.

Figure 89: Mail Setup



On this tab, you can configure the following e-mail settings:

- **Outgoing mail server (SMTP)**. Specify the fully qualified address of the SMTP server to use, such as smtp.mycompany.com.
- **Port number** Specify the port number to connect to on the SMTP server. Normally, the SMTP server has this port number set to 25.

- **This server requires an encrypted connection (SSL)**. Select this check box if the SMTP server requires that its clients use Secure Sockets Layer (SSL) when posting messages over the network.
- **This server requires authentication**. Select this check box if the SMTP server is configured to use Basic Authentication or Integrated Windows Authentication. Then, type the user name and password in the boxes beneath this option. By default, the **Outgoing mail server (SMTP)** list includes a single entry. You can add more entries to the list using the Active Roles console. In the console tree, expand **Configuration/Server Configuration**, right-click **Mail Configuration**, select **New | Mail Configuration**, and then follow the instructions in the wizard. passes these credentials to the SMTP server when establishing a connection.
- **Log on using Secure Password Authentication (SPA)**. Select this check box if the SMTP server is configured to use Integrated Windows Authentication, in order not to transmit the actual user password across the network.
- **Sender email address**. The default e-mail address of the message sender. A valid e-mail address must be specified. Normally, this is the email address of the service account used by the Administration Service.
- **Name (used in the From field)**. Specify the default name of the message sender, to be displayed in the **From** field of messages sent by using this e-mail configuration.

When you are done configuring the e-mail server-related settings, click **OK** to close the **Properties** dialog box for the e-mail configuration. Then, click **Next** and follow the instructions in the wizard to create the Policy Object.

Steps for configuring a Notification Distribution policy

To configure a Notification Distribution policy

1. On the **Policy to Configure** page, select **Notification Distribution Policy**, and then click **Next**.
2. On the **Notification Recipients and Message** page, do the following, and then click **Next**:
    - Click the button next to **Notification recipients**, and select one or more e-mail recipients.
    - In **Message Subject**, type the subject of the message that the specified recipients will receive upon a request to perform a deprovisioning operation.
    - Under **Message Body**, type any information regarding the deprovisioning operation.
3. On the **Outgoing Mail Server** page, select the e-mail configuration you want the policy to use. In the **Outgoing mail server (SMTP)** list, click the appropriate mail settings.

4. If you want to view or modify the selected mail settings, click **Settings**, and use the **Mail Setup** tab (see Configuring e-mail settings earlier in this chapter).
5. Click **Next**.
6. On the **Enforce Policy** page, you can specify objects to which this Policy Object is to be applied:
    - Click **Add**, and use the **Select Objects** dialog box to locate and select the objects you want.
7. Click **Next**, and then click **Finish**.

Scenario: Sending deprovisioning notification

This scenario describes how to configure a policy so that the administrator is notified of deprovisioning objects in any domain registered with Active Roles (managed domain).

To implement this scenario, you must perform the following actions:

1. Create the appropriate e-mail configuration.
2. Create, configure, and apply the Policy Object that defines the appropriate policy.

As a result, upon a request to deprovision an object such as a user or group in any managed domain, the administrator receives an e-mail message informing of the deprovisioning request. The message includes the name of the object to deprovision.

The following two sections elaborate on the steps to implement this scenario.

Step 1: Creating the e-mail configuration

This scenario assumes that your SMTP server:

- Runs on the server **smtp.mycompany.com**.
- Uses the **default port number (25)**.
- Allows **anonymous access**.
- Allows **non-encrypted connections**.

Additionally, the service account of the Administration Service is assumed to have a mailbox with the e-mail address of **ARSService@mycompany.com**.

Create the e-mail configuration by using the Active Roles console:

1. In the console tree, expand **Configuration | Server Configuration**, right-click **Mail Configuration**, and then select **New | Mail Configuration** to start the New Mail Configuration wizard.
2. Click **Next**.
3. In **Name**, type **Deprovisioning Notification Distribution**.
4. Click **Next**.
5. In **Outgoing mail server (SMTP)**, type **smtp.mycompany.com**.

6. In **Sender e-mail address**, type the e-mail address of the service account: **ARSService@mycompany.com**.
7. In **Name (used in the From field)**, type **Active Roles**.
8. Click **Next**, and then click **Finish**.

Step 2: Creating, configuring, and applying the Policy Object

You can create, configure, and apply the Policy Object you need by using the New Deprovisioning Policy Object wizard. For information about the wizard, see Creating a Policy Object in the Policy Object management tasks section earlier in this chapter.

To configure the policy, click **Notification Distribution** on the **Select Policy Type** page of the wizard. Then, click **Next**.

On the **Notification Recipients and Message** page, click the button next to the **Notification recipients** box to display the **Deprovisioning Notification Recipients** dialog box. In that dialog box, type the administrator's e-mail address, such as administrator@mydomain.com, and then click **OK**.

Then, customize the message subject and the message body as necessary. For example, you might enter the following subject and body:

Message subject

Deprovisioning of %<objectClass> '%<name>' Requested

Message body

Deprovisioning of %<objectClass> '%<name>' is in progress. Please take any additional deprovisioning actions, if necessary, to complete the deprovisioning of that %<objectClass>.

This notification was generated automatically by Active Roles according to corporate deprovisioning rules.

Click **Next** to display the **Outgoing Mail Server** page.

From the list in the **Outgoing mail server (SMTP)** box, select **Deprovisioning Notification Distribution**—the e-mail configuration you created in Step 1. Then, click **Next** to display the **Enforce Policy** page.

Add the **Active Directory** folder to the list on the **Enforce Policy** page:

1. Click the **Add** button to display the **Select Objects** window.
2. In the **Select Objects** window, click the **Browse** button to display the **Browse for Container** dialog box.
3. In the **Browse for Container** dialog box, click **Active Directory**, and then click **OK**.
4. From the upper list in the **Select Objects** window, select **Active Directory**.
5. Click **Add**, and then click **OK** to close the **Select Objects** window.

Click **Next**, and then click **Finish** to close the wizard.

You can also use the **Enforce Policy** command on the **Active Directory** folder in the console tree to apply the policy to that folder. For more information on how to apply a Policy Object, see Applying Policy Objects and Managing policy scope earlier in this chapter.

Report Distribution

Policies in this category are intended to automatically send a report on deprovisioning results upon completion of a deprovisioning operation. The primary purpose of such a policy is to inform designated persons about problems, if any encountered, when processing deprovisioning requests. These reports are discussed later in this chapter (see the Report on deprovisioning results section).

Reports are delivered via e-mail. When configuring a Report Distribution policy, you can set up a list of report recipients, customize the subject of report messages, and specify whether to send a report if no errors occurred.

How this policy works

Upon completion of a deprovisioning operation, Active Roles uses this policy to determine if the report on deprovisioning results must be sent. Then, Active Roles generates the report message and sends it to the recipients specified in the policy configuration. The report includes a list of actions taken during the deprovisioning operation. For each action, the report informs of whether the action is completed successfully, and provides information about the action results.

With the Report Distribution policy configured not to send reports if no errors occurred, Active Roles examines the deprovisioning results for errors. If there are no errors, the report is not sent.

Active Roles generates deprovisioning reports on a per-object basis: Each report message contains information on the deprovisioning of one object. When deprovisioning multiple objects, Active Roles sends multiple report messages, one message per deprovisioned object.

Active Roles sends report messages via an SMTP server. The policy configuration specifies the outbound SMTP server by using Active Roles e-mail settings that include the name of the SMTP server and information required to connect to the SMTP server.

How to configure a Report Distribution policy

To configure a Report Distribution policy, select **Report Distribution** on the **Policy to Configure** page in the New Deprovisioning Policy Object wizard or in the Add Deprovisioning Policy wizard. Then, click **Next** to display the **Report Recipients and Message** page.

On the **Report Recipients and Message** page, you can set up a list of report recipients, make any necessary changes to the message subject, and specify whether you want the policy to send out the report if no errors occurred during the deprovisioning operation.

To specify report recipients, click the button next to the **Report recipients** box, and then type one or more e-mail addresses. Use a semicolon to separate the addresses of the recipients.

If you need to modify the message subject, type in the **Message subject** box. You can use macros to enter information about the deprovisioned object, to make the message more meaningful to the recipients.

Macros have the same syntax and semantics as values for policy conditions in Property Generation and Validation policies: An attribute's LDAP display name enclosed in angle brackets (<>) and prefixed with the percent character (%) represents the value of that attribute. For instance, before sending a message, Active Roles replaces %<name> with the original name of the object that has been deprovisioned.

> NOTE: Active Roles retrieves the attribute value prior to starting the deprovisioning operation so the value is current as of the time the deprovisioning process begins. Even if you have a deprovisioning policy configured to update a given attribute, the message reads the original rather than updated value of that attribute.

If you want the policy to send out the report regardless of whether or not the deprovisioning operation is completed without any errors, clear the **Send out the report only if any errors occur** check box; otherwise, the report is not sent if the object was deprovisioned without errors.

When you are done, click **Next** to display the **Outgoing Mail Server** page.

This page is similar to the respective wizard page for Notification Distribution policies (see How to configure a Notification Distribution policy). You can select the e-mail configuration you want the policy to use, and view or modify e-mail settings in the selected configuration.

First, from the **Outgoing mail server (SMTP)** list, select the e-mail configuration you want the policy to use.

> NOTE: By default, the **Outgoing mail server (SMTP)** list includes a single entry. You can add more entries to the list using the Active Roles console. In the console tree, expand **Configuration/Server Configuration**, right-click **Mail Configuration**, select **New | Mail Configuration**, and then follow the instructions in the wizard.

Each e-mail configuration specifies an SMTP server and provides information required to connect to that server. You can view and modify configuration parameters by clicking the **Settings** button. For instructions, see Configuring e-mail settings earlier in this chapter.

When you are done, click **Next** and follow the instructions in the wizard to create the Policy Object.

Steps for configuring a Report Distribution policy

To configure a Report Distribution policy

1. On the **Policy to Configure** page, select **Report Distribution Policy**, and then click **Next**.
2. On the **Report Recipients and Message** page, do the following, and then click **Next**:
   - Click the button next to **Report recipients**, and then select one or more e-mail recipients.
   - In **Message Subject**, type the subject of the message that the specified recipients will receive upon completion of a deprovisioning operation.
   - Select the **Send out the report only if any errors occur** check box if you want the policy not to send the report if no errors occurred during the deprovisioning operation. Clear the check box if you want the policy to send the report regardless of whether or not any errors occurred.
3. On the **Outgoing Mail Server** page, select the e-mail configuration you want the policy to use. In the **Outgoing mail server (SMTP)** list, click the appropriate mail settings.
4. If you want to view or modify the selected mail settings, click **Settings**, and use the **Mail Setup** tab (see Configuring e-mail settings earlier in this document).
5. Click **Next**.
6. On the **Enforce Policy** page, you can specify objects to which this Policy Object is to be applied:
   - Click **Add**, and use the **Select Objects** dialog box to locate and select the objects you want.
7. Click **Next**, and then click **Finish**.

Scenario: Sending deprovisioning report

This scenario describes how to configure the following policy to monitor deprovisioning operations in all domains registered with Active Roles (managed domains):

- When a deprovisioning operation is completed, verify if any errors occurred during the operation.
- If any errors occurred, send the report on the deprovisioning results to the administrator.

To implement this scenario, you must perform the following actions:

1. Create the appropriate e-mail configuration.
2. Create, configure, and apply the Policy Object that defines the appropriate policy.

As a result, upon completion of a deprovisioning operation in any managed domain, the administrator receives a report in the event of any error during that operation. The message subject includes the name of the object that has been deprovisioned.

The following two sections elaborate on the steps to implement this scenario.

Step 1: Creating the e-mail configuration

You can use the instructions in the previous section to create the e-mail configuration (see Scenario: Sending deprovisioning notification). When prompted to specify a name for the new configuration, type **Deprovisioning Report Distribution**.

Step 2: Creating, configuring, and applying the Policy Object

You can create, configure, and apply the Policy Object you need by using the New Deprovisioning Policy Object wizard. For information about the wizard, see Creating a Policy Object in the Policy Object management tasks section earlier in this document.

To configure the policy, click **Report Distribution** on the **Select Policy Type** page of the wizard. Then, click **Next**.

On the **Report Recipients and Message** page, click the button next to the **Report recipients** box to display the **Deprovisioning Report Recipients** dialog box. In that dialog box, type the administrator's e-mail address, such as administrator@mydomain.com, and then click **OK**.

Then, customize the message subject as necessary. For example, you might enter the following subject: Deprovisioning of %<objectClass> '%<name>' Completed with Errors. Verify that the **Send out the report only if any errors occur** check box is selected and then click **Next** to display the **Outgoing Mail Server** page.

From the list in the **Outgoing mail server (SMTP)** box, select **Deprovisioning Report Distribution**—the e-mail configuration you have created in Step 1, and then click **Next** to display the **Enforce Policy** page.

On the **Enforce Policy** page, click the **Add** button and select the **Active Directory** folder to add to the list. Click **Next**, and then click **Finish** to close the wizard.

You can also use the **Enforce Policy** command on the **Active Directory** folder in the console tree to apply the policy to that folder. For more information on how to apply a Policy Object, see Applying Policy Objects and Managing policy scope earlier in this chapter.

Deployment considerations

Active Roles enforces policies by applying Policy Objects to promote data integrity throughout the directory. This is done by generating and validating the data entered into the directory. Each Policy Object is basically a container that holds one or more policy

entries (also referred to as policies). There are several types of policy entries that can be configured within a Policy Object. The two major ones are Property Generation and Validation, and Script Execution. Property Generation and Validation policy entries provide a point-and-click interface for creating basic rules for attribute population. Script Execution policy entries enable the use of scripting for a broad range of custom actions that could supplement, extend, or replace the policy types included with Active Roles out of the box.

Just as with Group Policy Objects in Active Directory, the location that Active Roles' Policy Objects are linked to is critical:

- Any policies that are intended to affect the entire domain should be included into a Policy Object linked at the domain level. If needed, filtering can be used to exclude specific objects or containers (Organizational Units) from being processed by these policies.
- If more than one object or containers needs to be excluded from the effect of a domain-wide policy, it is best to include those objects or containers explicitly into a Managed Unit and then apply policy filtering to that Managed Unit by using the **Block Inheritance** option.

From here, the best way to apply policies is at the top level of the directory tree they will affect. Usually, however policies are only needed to affect certain Organizational Units within the tree. In this case, a Managed Unit is the most effective way to apply the policies. Include the desired Organizational Units explicitly into a Managed Unit, and then link the Policy Object to that Managed Unit.

A policy consists of three major components. These are:

- A policy entry that defines the policy
- A Policy Object containing that policy entry
- A Policy Object link that determines where the policy is applied in the directory

Typically, a single Policy Object includes all the entries for a specific set of policies. It is not efficient to create one entry per Policy Object since this defeats the purpose of having separation between the Policy Object and policy entries.

A policy cannot be filtered for specific sets of administrators. Once applied to a given object or container, a policy will be in effect for every administrator under every condition. This is unless a Script Execution policy is included as a policy entry that utilizes the **IEDSEffectivePolicyRequest** interface to override the policies determined by other policy entries. This interface is documented in Active Roles SDK.

Script Execution polices are policy entries that utilize scripts written in a scripting language such as Microsoft Windows PowerShell or VBScript. Policy scripts use event handles that are executed before or after every action that can happen in the directory. See the following table for a list of these handlers.

Table 27: Event handlers

Name	Description
onPreCreate	In a script policy applied to a container; receives control upon a request to create an object in that container. This enables a

Name	Description
	script to perform custom actions prior to creating an object.
onPostCreate	In a script policy applied to a container; receives control after a request to create an object in that container is completed. This enables a script to perform custom actions further to creating an object.
onPreDelete	Receives control upon a request to delete an object. Enables a script to perform custom actions prior to deleting an object.
onPostDelete	Receives control after a request to delete an object is completed. Enables a script to perform custom actions further to deleting an object.
onPreModify	Receives control upon a request to start changing object properties. Enables a script to perform custom actions prior to applying changes to an object.
onPostModify	Receives control after a request to change object properties is completed. Enables a script to perform custom action further to changing an object's property values.
onPreMove	In a script policy applied to a container, this function receives control upon a request to start moving an object from that container. This enables a script to perform custom actions prior to moving an object.
onPostMove	In a script policy applied to a container, this function receives control after a request to move an object to that container is completed. This enables a script to perform custom actions further to moving an object.
onPreRename	Receives control upon a request to start renaming an object. Enables a script to perform custom actions prior to renaming an object.
onPostRename	Receives control after a request to rename an object is completed. Enables a script to perform custom actions further to renaming an object.
onPreGet	Receives control upon a request to retrieve object properties. Enables a script to perform custom actions prior to starting the retrieval of an object's property values.
onPostGet	Receives control after a request to retrieve object properties is completed. Enables a script to perform custom actions following the retrieval of an object's property values.
onPreSearch	Receives control upon a request to start a search. Enables a script to perform custom actions prior to starting a search.

Name	Description
onPreDeprovision	Receives control upon a request to execute the Deprovision operation. Enables a script to perform custom actions prior to starting the operation.
onDeprovision	Receives control in the course of processing a request to execute the Deprovision operation. Enables the use of a script for customizing the behavior of the operation.
onPostDeprovision	Receives control after a request to execute the Deprovision operation is completed. Enables a script to perform custom actions following the operation.
onPreUnDeprovision	Receives control upon a request to execute the Undo Deprovisioning operation. Enables a script to perform custom actions prior to starting the operation.
onUnDeprovision	Receives control in the course of processing a request to execute the Undo Deprovisioning operation. Enables the use of a script for customizing the behavior of the operation.
onPostUnDeprovision	Receives control after a request to execute the Undo Deprovisioning operation is completed. Enables a script to perform custom actions following the operation.
onPreUnDelete	Receives control upon a request to execute the Undelete operation. Enables a script to perform custom actions prior to starting the operation.
onPostUnDelete	Receives control after a request to execute the Undelete operation is completed. Enables a script to perform custom actions following the operation.
onCheckPropertyValues	Receives control upon a request to verify and validate the changes that are going to be made to an object. Enables a script to perform custom actions further to normal validity checks on an object.
onGetEffectivePolicy	Receives control upon a request to retrieve the policy settings that are in effect on a particular object (such as policy constraints on property values). Enables a script to perform custom actions further to retrieval of policy settings.
onInit	Receives control when the Administration Service retrieves the definition of the script parameters, enabling the script to manifest the name and other characteristics of each parameter.
onFilter	Boolean-valued function that is evaluated during execution of the onPreSearch event handler, allowing search results to be filtered based on properties of objects returned by the search.

Name	Description
	For details, see "IEDSRequestParameters Properties" in the Active Roles SDK documentation.

Basically, when an action happens, Active Roles looks to see if there are any Policy Objects applied that hold Script Execution policies. If so, the policy script is checked to see if it has an event handler for the specific action being performed. The object being acted upon is passed into the event handler for further actions. These event handlers are normally run in the security context of the service account, so even if a user does not have rights to perform the actions outlined in the policy script, it will still execute correctly. If any errors occur during the execution of a policy script, the errors can be found in the Active Roles event log for post-action handlers and are displayed to the client for pre-action handlers.

Policy scripts are typically written in a scripting language such as Windows PowerShell or VBScript. Many examples of scripts based on Windows PowerShell and VBScript, along with instructions on how to use the Active Roles ADSI Provider both for policy scripts and for standalone scripts, can be found in the Active Roles SDK documentation.

It is also important to note that policy scripts can pick up and take action upon directory changes made natively, as well. To turn on this behavior, you should choose the option that directs in the policy script to handle directory changes reported by the directory synchronization function (select the **Handle changes from DirSync control** check box on the **Script Module** tab in the **Properties** dialog box for the policy entry), and use the **IEDSRequestParameters** interface in a post-action event handler. More on this topic can be found in the Active Roles SDK documentation.

Checking for policy compliance

Checking for policy compliance provides information on directory data that is out of compliance with the policies, such as user or group naming conventions, defined with Active Roles. If you define some policies when data has already been entered, you can check the data, and modify it accordingly, in order to ensure that the data meets the policy requirements.

Although business rules and policies normally cannot be bypassed once they have been configured, there are situations where the actual directory data may violate some of the prescribed policies or business rules. For example, when applying a new policy, Active Roles does not automatically verify the existing directory data in order to determine whether that data conforms to the new policy. Another example is a process that automatically creates new objects, such as user or group objects, by directly accessing Active Directory without the use of Active Roles.

The Active Roles Report Pack includes a number of reports that help detect policy violations in directory data by collecting and analyzing information on the state of directory objects as against the prescribed policies. However, as retrieving such information may take much time and effort, the reports on policy compliance sometimes do not allow policy-related issues to be resolved in a timely fashion.

In order to address this problem, Active Roles makes it possible to quickly build and examine policy check results on individual objects or entire containers. The policy check results provide a list of directory objects violating policies, and describe the detected violations. From the policy check results, you can make appropriate changes to objects or policies:

- Modify object properties in conformity with policies.
- Prevent individual objects from being affected by particular policies.
- Modify Policy Objects as needed.
- Perform an administrative task—for example, disable or move user objects that violate policies.

In addition, you can save policy check results to a file, print them out, or send them to an e-mail recipient.

To check an object for policy compliance, right-click the object and click **Check Policy**. For a container object, this displays the **Check Policy** dialog box. Review the options in the **Check Policy** dialog box and click **OK**.

The **Policy Check Results** window appears and the operation starts. The check results are displayed in the right pane of the window. The objects that violate a policy are displayed in the left pane. When you click an object in the left pane, the right pane describes the policy violation in detail.

By default, the right pane in the **Policy Check Results** window only displays basic options. You can display more choices by clicking the **Details** column heading.

By using links in the right pane, you can perform the following tasks:

- Modify the property value violating the policy: Click the **edit** link next to the **Property value** label.
- Remove the object from the policy scope: Click the **block policy inheritance** link next to the **Policy Object** label. If you do so, the policy no longer controls the object.
- Modify the policy: Click the **properties** link next to the **Policy Object** label. This displays the **Properties** dialog box for the Policy Object, described in Adding, modifying, or removing policies earlier in this chapter.
- Administer the object violating the policy: Click the **Properties** button in the upper-right corner of the right pane.
- Administer the object to which the Policy Object is applied: Click the **properties** link next to the **Applied to** label.

You can use the following instructions to see how checking for policy compliance works in the Active Roles console:

1. Create and configure a Policy Object with the property validation and generation policy for the **Department** property of user objects, specifying the policy rule as follows: Value must be specified and must be Sales or Production.
2. Apply (link) that Policy Object to an organizational unit that already holds some user objects with no department specified.

3. Right-click the organizational unit and click **Check Policy**. In the **Check Policy** dialog box, click **OK**.

   Once you have performed these steps, the **Policy Check Results** window is displayed. Its left pane lists objects violating the policy.

4. Wait while the list in the left pane is being populated. Then, select a user object from the list.

   The right pane, next to the **Violation** label, displays the prompt **You must specify a value for the property 'department'**.

5. In the right pane, click the **edit** link next to the **Property value** label.
6. In the **Properties** dialog box, select one of the acceptable values (**Production** or **Sales**) from the **Department** combo-box.

Steps to check for policy compliance

Checking for policy compliance provides information on directory data that is out of compliance with the policies, such as user or group naming conventions, defined with Active Roles. If you define some policies when data has already been entered, you can check the data, and modify it accordingly, in order to ensure that the data meets the policy requirements.

To check an object for policy compliance

1. Right-click the object, and click **Check Policy**.
2. If the object is a container or Managed Unit, select the appropriate combination of these check boxes to specify the scope of the operation:
   - **This directory object**. The scope includes the container or Managed Unit you have selected (this option does not cause the scope to include any child objects or members of the container or Managed Unit).
   - **Child objects of this directory object**. The scope includes all the child objects (or members, as applied to a Managed Unit) in the entire hierarchy under the container or Managed Unit you have selected.
   - **Immediate child objects only**. The scope includes only the child objects (or members, as applied to a Managed Unit) of which the container or Managed Unit that you have selected is the direct ancestor.

   Click **OK**.

   The progress and results of the policy check operation are displayed in the **Policy Check Results** window. The left pane of the window lists the objects for which a policy violation has been detected.

3. Click an object in the left pane of the **Policy Check Results** window.

   When you click an object in the left pane, the right pane describes the policy violation in detail. By default, the right pane in the **Policy Check Results** window only

displays basic options. You can display more choices by clicking the **Details** column heading.

4. Use hypertext links in the right pane to perform the following tasks:
   - Modify the property value violating the policy: Click the **edit** link next to the **Property value** label.
   - Remove the object from the policy scope: Click the **block policy inheritance** link next to the **Policy Object** label. If you do so, the policy no longer controls the object.
   - Modify the policy: Click the **properties** link next to the **Policy Object** label. This displays the **Properties** dialog box for the Policy Object. For instructions on how to add, modify, or remove policies in the **Properties** dialog box, see Adding, modifying, or removing policies earlier in this document.
   - View or modify the properties of the object that violates the policy: Click the **Properties** button in the upper-right corner of the right pane.
   - View or modify the properties of the object to which the Policy Object is applied (linked): Click the **properties** link next to the **Applied to** label.

NOTE: The **Check Policy** command on a Policy Object performs a check on all the objects found in the policy scope of the Policy Object. Use the **Check Policy** command on a Policy Object to find all objects that are not in compliance with the policies defined by that Policy Object.

Deprovisioning users or groups

The Active Roles user interfaces, both Active Roles console and Web Interface, provide the **Deprovision** command on user and group objects. This command originates a request to deprovision the selected objects. When processing the request, Active Roles performs all operations prescribed by the deprovisioning policies.

Default deprovisioning options

Active Roles ships with two built-in Policy Objects that specify the operations to perform when deprovisioning a user or group. You can find those Policy Objects in the Active Roles console by selecting the **Configuration | Policies | Administration | Builtin** container**.**

The Built-in Policy - User Default Deprovisioning Policy Object determines the default effect of the **Deprovision** command on user accounts; the **Built-in Policy - Group Default Deprovisioning Policy Object determines the default effect of that command on groups. Both objects are** applied to the **Active Directory** container, taking effect in all domains that are registered with Active Roles.

The following tables summarize the default deprovisioning policy options. If you do not add, remove, or change deprovisioning policies, Active Roles operates in accordance with these options when carrying out the **Deprovision** command on a user or group.

The following table summarizes the default deprovisioning policy options for users, defined by the **Built-in Policy - User Default Deprovisioning** Policy Object.

Table 28: Policy options for users: Built-in Policy - User Default Deprovisioning

Policy	Options
User Account Deprovisioning	• Disable the user account. • Set the user's password to a random value. • Change the user name to include the suffix "deprovisioned" followed by the date when the user was deprovisioned. • Fill in the user description to state that this user account is deprovisioned. • Clear certain properties of the user account, such as city, company, and postal address.
Group Membership Removal	• Remove the user account from all security groups. • Remove the user account from all distribution groups.
Exchange Mailbox Deprovisioning	• Hide the user mailbox from Exchange address lists, thus preventing access to the mailbox.
Home Folder Deprovisioning	• Revoke access to the user home folder from the user account. • Give the user's manager read access to the user home folder. • Designate Administrators as the home folder owner.
User Account Relocation	• Do not move the user account from the organizational unit in which the account was located at the time of deprovisioning.
User Account Permanent Deletion	• Do not delete the user account.

The following table summarizes the default deprovisioning policy options for groups, defined by the **Built-in Policy - Group Default Deprovisioning** Policy Object.

Table 29: Policy options for groups: Built-in Policy - User Default Deprovisioning

Policy	Options
Group Object Deprovisioning	• Change the group type from Security to Distribution. • Hide the group from the Global Address List (GAL) • Change the group name to include the suffix "deprovisioned"

Policy	Options
	followed by the date when the group was deprovisioned • Remove all members from the group • Fill in the group description to state that this group is deprovisioned
Group Object Relocation	• Do not move the group from the organizational unit in which the group was located at the time of deprovisioning
Group Object Permanent Deletion	• Do not delete the group

Delegating the Deprovision task

Deprovisioning is, by default, a right of Active Roles Admin, the administrative account specified during Active Roles installation, but the task of deprovisioning can be delegated to any group or user. A dedicated Access Template is provided for this purpose so that you can delegate the use of the **Deprovision** command without delegating the create or delete operation.

To delegate the task of deprovisioning users or groups in a certain container, such as an organizational unit or a Managed Unit, you should apply the Access Template as follows.

To delegate the Deprovision task

1. In the Active Roles console, right-click the container and click **Delegate Control** to display the **Active Roles Security** window.
2. In the **Active Roles Security** window, click **Add** to start the Delegation of Control wizard. Click **Next**.
3. On the **Users or Groups** page, click **Add**, and then select the users or groups to which you want to delegate the deprovision task. Click **Next**.
4. On the **Access Templates** page, expand the **Active Directory** folder and then do the following:
   - To delegate the task of deprovisioning users, select the check box next to **Users - Perform Deprovision Tasks**.
   - To delegate the task of deprovisioning groups, select the check box next to **Groups - Perform Deprovision Tasks.**
5. Click **Next** and follow the instructions in the wizard, accepting the default settings.

After you complete these steps, the users and groups you selected in Step 3 are authorized to deprovision users or groups in the container you selected in Step 1, as well as in any sub-container of that container.

Using the Deprovision command

The **Deprovision** command is available in both the Active Roles console and Web Interface. By using the **Deprovision** command, you start the deprovisioning operation on the objects you have selected.

The operation progress and results are displayed in the **Deprovisioning Results** window. When the operation is completed, the window displays the operation summary, and allows you to examine operation results in detail.

The left pane of the **Deprovisioning Results** window lists the objects that have been deprovisioned. The right pane displays the operation status and error messages, if any.

To view operation results, select an object in the left pane. The right pane shows a report on all actions taken during the deprovisioning of the selected object. A typical report is discussed in the next section.

Report on deprovisioning results

For each deprovisioned object, the **Deprovisioning Results** window can be used to examine the deprovision operation results on that object.

The Active Roles console or Web Interface opens the **Deprovisioning Results** window when carrying out the **Deprovision** command. You can also open this window by using the **Deprovisioning Results** command, which is available on deprovisioned objects.

The **Deprovisioning Results** window displays a report of the deprovisioning operation. The report organizes operation results into sections named after policy categories, with each section containing report items specific to a certain policy category. When you click the heading at the top of the report, the report is fully expanded and all report items are shown. Alternatively, you can expand and contract individual sections within the report by clicking the heading for each section.

For certain items, the report provides the option to further expand the view and display additional information. By clicking the **List** option, you can display a list of items, such as user or group properties, involved in the operation. By clicking the **Details** option, you can examine the operation result in more detail.

The **Deprovisioning Results** window also meets some common reporting requirements including the ability to document all the operation results to a file for printing or viewing. Using the shortcut menu, you can export the report to a file as either HTML or XML, print the report, or send it out via e-mail.

Report contents

The following tables list the possible report items, one table per report section. The items in each section describe results of the actions that were taken in accord with the respective deprovisioning policy. Report items also inform about success or failure of each action. In the event of a failure, the report item includes an error description.

Not all the listed items must necessarily be present in a report. An actual report only includes the report items corresponding to the configured policy options. For example, if the policy is not configured to disable user accounts, the report does not include the item regarding this action.

Report section: User Account Deprovisioning

Table 30: User Account Deprovisioning

Report item (Success)	Report item (Failure)
The user account is disabled.	Failed to disable the user account.
The user password is reset to a random value.	Failed to reset the user password.
The user logon name is changed to a random value.	Failed to change the user logon name.
The user logon name (pre-Windows 2000) is changed to a random value.	Failed to change the user logon name (pre-Windows 2000).
The user name is changed. Original name: name New name: name	Failed to change the user name. Current name: name Failed to set this name: name
User properties are changed. List: • User properties, old and new property values	Failed to change user properties. List: • User properties, error description

Report section: Group Membership Removal

Table 31: Group Membership Removal

Report item (Success)	Report item (Failure)
In accord with policy, the user is not removed from security groups, except for Dynamic Groups and groups controlled by Group Family. Details: • Security groups to which the user belongs	N/A
The user is removed from all security groups. Details: • Security groups from which the user is removed	Failed to remove the user from some security groups. Details: • Security groups from which the user is removed

Report item (Success)	Report item (Failure)
	• Security groups from which the user is not removed due to an error
In accord with policy, the user is retained in some security groups. Details: • Security groups from which the user is removed • Security groups from which the user is not removed in accord with policy	Failed to remove the user from some security groups. Details: • Security groups from which the user is removed • Security groups from which the user is not removed in accord with policy • Security groups from which the user is not removed due to an error
In accord with policy, the user is not removed from distribution groups or mail-enabled security groups, except for Dynamic Groups and groups controlled by Group Family. Details: • Distribution groups and mail-enabled security groups to which the user belongs	N/A
The user is removed from all distribution groups and mail-enabled security groups. Details: • Distribution groups and mail-enabled security groups from which the user is removed	Failed to remove the user from some distribution groups or mail-enabled security groups. Details: • Distribution groups and mail-enabled security groups from which the user is removed • Distribution groups or mail-enabled security groups from which the user is not removed due to an error
In accord with policy, the user is retained in some distribution groups or mail-enabled security groups. Details: • Distribution groups and mail-enabled security groups from which the user is removed • Distribution groups or mail-enabled security groups from which the user is not removed in accord with policy	Failed to remove the user from some distribution groups or mail-enabled security groups. Details: • Distribution groups and mail-enabled security groups from which the user is removed • Distribution groups or mail-enabled security groups from which the user is not removed in accord with policy • Distribution groups or mail-enabled security groups from which the user is not removed due to an error

Report section: Exchange Mailbox Deprovisioning

Table 32: Exchange Mailbox Deprovisioning

Report item (Success)	Report item (Failure)
Mailbox deprovisioning is skipped because the user does not have an Exchange mailbox.	N/A
The user mailbox is removed (hidden) from the Global Address List (GAL).	Failed to remove (hide) the user mailbox from the Global Address List (GAL).
The user mailbox is configured to suppress non-delivery reports (NDR).	Failed to configure the user mailbox to suppress non-delivery reports (NDR).
The user's manager is provided with full access to the user mailbox. Manager name: name	Failed to provide the user's manager with access to the user mailbox. Manager name: name
N/A	Failed to provide the user's manager with access to the user mailbox. Reason: The user's manager is not specified in the directory.
The required users and groups are provided with full access to the user mailbox. List: • Users and groups	Failed to provide the required users or groups with access to the user mailbox. List: • Users and groups
Forwarding messages to alternate recipients is disallowed on the user mailbox.	Failed to disallow forwarding messages to alternate recipients on the user mailbox.
The user mailbox is configured to forward incoming messages to the user's manager.	Failed to configure the user mailbox to forward incoming messages to the user's manager.
The user mailbox is configured to forward incoming messages to the user's manager, with the option to leave message copies in the mailbox.	Failed to configure the user mailbox to forward incoming messages to the user's manager.
Failed to configure the user mailbox to forward incoming messages to the user's manager. Reason: the user's manager is not specified in the directory.	N/A
Automatic replies turned on.	Failed to turn on automatic replies.

Report section: Home Folder Deprovisioning

Table 33: Home Folder Deprovisioning

Report item (Success)	Report item (Failure)
Home folder deprovisioning is skipped because the user does not have a home folder.	N/A
The user's rights on the home folder are removed.	Failed to remove the user's rights on the home folder.
The user's manager is provided with read-only access to the user home folder. Manager name: name	Failed to provide the user's manager with read-only access to the user home folder. Manager name: name
N/A	Failed to provide the user's manager with read-only access to the user home folder. Reason: The user's manager is not specified in the directory.
In accord with policy, the user home folder will be deleted when the user account is deleted. Home folder name: name	N/A
The required users and groups are provided with read-only access to the user home folder. List: • Users and groups	Failed to provide the required users or groups with read-only access to the user home folder. List: • Users and groups
The new owner is assigned to the user home folder. *Owner name: name*	Failed to assign the new owner to the user home folder. Failed to set this owner name: name

Report section: User Account Relocation

Table 34: User Account Relocation

Report item (Success)	Report item (Failure)
In accord with policy, the user account is not moved from its original location: name of container	N/A
The user account is moved to new location. Original location: name of container New location: name of container	Failed to move the user account to new location. Original location: name of container Failed to move to this location: name of container

Report section: User Account Permanent Deletion

Table 35: User Account Permanent Deletion

Report Item (Success)	Report Item (Failure)
In accord with policy, the user account is not scheduled for deletion.	*Not applicable*
The user account is scheduled for deletion. Will be deleted on this date: *date*	Failed to schedule the user account for deletion.
The user account is deleted to Active Directory Recycle Bin.	Failed to delete the user account to Active Directory Recycle Bin. Verify that Active Directory Recycle Bin is enabled.

Report section: Group Object Deprovisioning

Table 36: Group Object Deprovisioning

Report Item (Success)	Report Item (Failure)
The type of the group is changed from Security to Distribution.	Failed to change the type of the group from Security to Distribution.
The type of the group cannot be changed from Security to Distribution because this is not a security group.	Not applicable
The group is removed (hidden) from the Global Address List (GAL).	Failed to remove (hide) the group from the Global Address List (GAL).
The group cannot be removed (hidden) from the Global Address List (GAL) because this is not a mail-enabled group.	Not applicable
The group name is changed. Original name: *name* New name: *name*	Failed to change the group name. Current name: *name* Failed to set this name: *name*
In accord with policy, the members are not removed from the group. Details: • List of the members retained in the group	Not applicable
The members are removed from the group. Details: • List of the members removed from the group	Failed to remove some members from the group. Details: • List of the members removed from the group • List of the members that are not

Report Item (Success)	Report Item (Failure)
	removed from the group due to an error
In accord with policy, some members are retained in the group. Details: • List of the members removed from the group • List of the members retained in the group	Failed to remove some members from the group. Details: • List of the members removed from the group • List of the members retained in the group in accord with policy • List of the members that are not removed from the group due to an error
Group properties are changed. List: • *Property names, old and new property values*	Failed to change group properties. List: • *Property names, error description*

Report section: Group Object Relocation

Table 37: Group Object Relocation

Report Item (Success)	Report Item (Failure)
In accord with policy, the group is not moved from its original location: *name of container*	*Not applicable*
The group is moved to new location. Original location: *name of container* New location: *name of container*	Failed to move the group to new location. Original location: *name of container* Failed to move to this location: *name of container*

Report section: Group Object Permanent Deletion

Table 38: Group Object Permanent Deletion

Report Item (Success)	Report Item (Failure)
In accord with policy, the group is not scheduled for deletion.	*Not applicable*
The group is scheduled for deletion. Will be deleted on this date: *date*	Failed to schedule the group for deletion.
The group is deleted to Active Directory Recycle Bin.	Failed to delete the group to Active Directory Recycle Bin. Verify that Active Directory Recycle Bin is enabled.

Report section: Notification Distribution

Table 39: Notification Distribution

Report Item (Success)	Report Item (Failure)
Deprovisioning notification will be sent to the listed recipients (not sent so far). List: • *Recipients*	*Not applicable*
Deprovisioning notification was sent to the listed recipients. List: • *Recipients*	Due to an error, deprovisioning notification was not sent to the listed recipients. List: • *Recipients*

Report section: Report Distribution

Table 40: Report Distribution

Report Item (Success)	Report Item (Failure)
Deprovisioning report will not be sent out since no errors occurred.	*Not applicable*
Deprovisioning report will be sent to the listed recipients (not sent so far). List: • *Recipients*	*Not applicable*
Deprovisioning report was sent to the listed recipients. List: • *Recipients*	Due to an error, deprovisioning report was not sent to the listed recipients. List: • *Recipients*

Restoring deprovisioned users or groups

Active Roles provides the ability to restore deprovisioned objects, such as deprovisioned users or groups. The purpose of this operation, referred to as the Undo Deprovisioning operation, is to roll back the changes that were made to an object by the Deprovision operation. When a deprovisioned object needs to be restored (for example, if a user account has been deprovisioned by mistake), the Undo Deprovisioning operation allows the object to be quickly returned to the state it was in before the changes were made.

The Undo Deprovisioning operation rolls back the changes that were made to the object in accord with the standard Deprovisioning policies. For example, assume a User Account Deprovisioning policy is configured so that a deprovisioned user account:

- Is disabled.
- Is renamed.
- Has the Description changed.
- Has a number of properties cleared out.
- Has the password set to a random value.

In this case, the Undo Deprovisioning operation:

- Enables the user account.
- Sets the Description, Name, and other properties to the original values on the user account.
- Can provide the option to reset the password so as to enable the user to log on.

Similar behavior is in effect for the other policies of the Deprovisioning category:

- If the Deprovision operation revokes user access to resources such as the home folder or Exchange mailbox, then the Undo Deprovisioning operation attempts to restore user access to the resources.
- If the Deprovision operation removes a user account from certain groups, the Undo Deprovisioning operation can add the user account to those groups, restoring the original group memberships of the user account.

To offer another example, suppose the deprovisioning policy is configured so that Deprovision operation on a group:

- Removes all members from the group
- Renames the group
- Moves the group to a certain container

In this case, the Undo Deprovisioning operation:

- Restores the original membership list of the group, as it was at the time of deprovisioning
- Renames the group, restoring the original name of the group
- Moves the group to the container that held the group at the time of deprovisioning

Similar behavior is in effect for the other group deprovisioning policy options:

- If the Deprovision operation hides the group from the Global Address List (GAL), Undo Deprovisioning restores the visibility of the group in the GAL.
- If the Deprovision operation changes the group type from Security to Distribution, Undo Deprovisioning sets the group type back to Security.
- If the Deprovision operation changes any other properties of the group, Undo Deprovisioning restores the original property values.

Both the Active Roles console and Web Interface provide the **Undo Deprovisioning** command on deprovisioned users or groups. When selected on a deprovisioned object, this command originates a request to restore the object. Upon receipt of the request, Active Roles performs all necessary actions to undo the results of deprovisioning on the object,

and provides a detailed report of the actions that were taken along with information about success or failure of each action.

Policy options to undo user deprovisioning

The behavior of the Undo Deprovisioning operation is determined by a configurable policy contained in a built-in Policy Object. This is the Policy Object named **Built-in Policy - Default Rules to Undo User Deprovisioning** and located in the **Builtin** container under **Configuration/Policies/Administration**. The Policy Object is applied to the **Active Directory** folder, thus taking effect in all domains that are registered with Active Roles (managed domains).

The option provided by this policy can be used to prevent restoration of group memberships and resetting of the user password:

- **Restore group memberships**. When selected, causes the Undo Deprovisioning operation on a deprovisioned user account to add the account to the distribution and security groups from which the account was removed in accord with the Group Membership Removal policy. If you do not want restored accounts to be automatically added to groups, clear this option.

  Note that regardless of whether this option is selected, once a deprovisioned user account is restored, Active Roles automatically adds the account to the appropriate Dynamic Groups and Group Families depending on properties of the account.

- **Leave password unchanged**. Causes the Undo Deprovisioning operation on a deprovisioned user account to prevent resetting of the password for the restored account. Select this option if you want the password to be reset by the HelpDesk or by using a self-service password management solution after the account is restored.
- **Prompt to reset password**. Causes the Undo Deprovisioning operation on a deprovisioned user account to enable resetting of the password for the restored account. If this option is selected, the **Undo Deprovisioning** command displays a dialog box in which the password can be reset.

To view or modify the policy options

1. Open the Active Roles console.
2. In the console tree, expand **Configuration | Policies | Administration**, and select **Builtin** under **Administration**.
3. In the details pane, double-click **Built-in Policy - Default Rules to Undo User Deprovisioning**.
4. On the **Policies** tab in the **Properties** dialog box, click the policy in the list, and then click **View/Edit** to access the policy options.

Since the built-in Policy Object is normally applied to the Active Directory node in the Active Roles namespace, the policy options are in effect on any deprovisioned user account. If you need different policy options for different domains or containers, create a copy of the built-in Policy Object, and then configure and apply the copy as appropriate.

The Undo Deprovisioning operation is normally enabled in all domains that are registered with Active Roles. It is possible to prohibit this operation in individual domains or containers, or in all domains, by blocking or disabling the policy that governs the operation. In case of disabling the built-in Policy Object, an enabled copy of that Policy Object can be applied in order to allow the Undo Deprovisioning operation in individual domains or containers.

Delegating the task to undo deprovisioning

Restoring deprovisioned users or groups is, by default, a right of Active Roles Admin, the administrative account specified during Active Roles installation, but this task can be delegated to any group or user. A dedicated Access Template is provided for this purpose so you can delegate the use of the **Undo Deprovisioning** command without delegating the create or delete operation.

To delegate the task of restoring deprovisioned users or groups held in a certain container, such as an organizational unit or a Managed Unit, you should apply the Access Template as follows.

To delegate the Undo Deprovisioning task

1. In the Active Roles console, right-click the container and click **Delegate Control** to display the **Active Roles Security** window.
2. In the **Active Roles Security** window, click **Add** to start the Delegation of Control wizard. Click **Next**.
3. On the **Users or Groups** page, click **Add**, and then select the users or groups to which you want to delegate the task. Click **Next**.
4. On the Access Templates page, expand the Active Directory folder and then do the following:
   a. To delegate the task of restoring deprovisioned users, select the check box next to **Users - Perform Undo Deprovision Tasks**.
   b. To delegate the task of restoring deprovisioned groups, select the check box next to **Groups - Perform Undo Deprovision Tasks**.
5. Click **Next** and follow the instructions in the wizard, accepting the default settings.

After you complete these steps, the users and groups you selected in Step 3 are authorized to restore deprovisioned users in the container you selected in Step 1, as well as in any sub-container of that container.

Using the Undo Deprovisioning command

The **Undo Deprovisioning** command is available in both the Active Roles console and Web Interface to those who are authorized to restore deprovisioned users or groups. By

using this command, you start the Undo Deprovisioning operation on the objects you have selected, causing Active Roles to undo the results of deprovisioning on those objects.

To restore a deprovisioned user account

1. In the Active Roles console, right-click the user account, and then click **Undo Deprovisioning**.
2. In the **Password Options** dialog box, choose the options to apply to the password of the restored account, and then click **OK**.

   For information about each option, open the **Password Options** dialog box, and then press F1.

3. Wait while Active Roles restores the user account.

To restore a deprovisioned group

1. In the Active Roles console, right-click the group, and then click **Undo Deprovisioning**.
2. Wait while Active Roles restores the group.

The operation progress and results are displayed in the **Results of Undo Deprovisioning** window, which is similar to the **Deprovisioning Results** window discussed earlier in this chapter. When the operation is completed, the window displays the operation summary, and allows you to examine operation results in detail.

Report on results of undo deprovisioning

For each of the restored objects, the **Results of Undo Deprovisioning** window can be used to examine the restore operation results on that object. The Active Roles console or Web Interface opens the **Results of Undo Deprovisioning** window when carrying out the **Undo Deprovisioning** command.

The **Results of Undo Deprovisioning** window displays a report of the Undo Deprovisioning operation, which is similar to a deprovisioning-related report discussed earlier in this chapter. The report organizes operation results into sections, with each section containing report items specific to a certain category of deprovisioning policy. The report items within a particular section inform of the actions performed to roll back the changes that were made by the deprovisioning policy of the respective category.

When you click the heading at the top of the report, the report is fully expanded and all report items are shown. Alternatively, you can expand and contract individual sections within the report by clicking the heading for each section.

For certain items, the report provides the option to further expand the view and display additional information. By clicking the **List** option, you can display a list of items, such as user or groups properties, involved in the operation. By clicking the **Details** option, you can examine the operation result in more detail.

The **Results of Undo Deprovisioning** window also provides the ability to document all the operation results to a file for printing or viewing. Using the shortcut menu, you can export the report to a file as either HTML or XML, print the report, or send it out via e-mail.

Report contents

The following tables list the possible report items, one table per report section. The items in each section describe the results of the actions taken to undo the changes made by the respective deprovisioning policy. Report items also inform about success or failure of each action. In the event of a failure, the report item includes an error description.

Not all the listed items must necessarily be present in a report. An actual report only includes the report items related to the deprovisioning policies that were in effect when the user or group was deprovisioned.

Report section: Undo User Account Deprovisioning

Table 41: Undo User Account Deprovisioning

Report item (Success)	Report item (Failure)
The user account is enabled.	Failed to enable the user account.
The user password is reset to a known value with the following password options: <List of options>	Failed to reset the user password.
The current user password is left unchanged.	N/A
The user name is restored. Old name: name Restored name: name	Failed to restore the user name. Current name: name Failed to set this name: name
User properties are restored. List: • User properties, new property values	Failed to restore user properties. List: • User properties, error description

Report section: Undo Group Membership Removal

Table 42: Undo Group Membership Removal

Report item (Success)	Report item (Failure)
In accord with policy, the user's membership in security groups is not restored.	N/A
In accord with policy, the user's membership in distribution groups or mail-	N/A

Report item (Success)	Report item (Failure)
enabled security groups is not restored.	
The user's membership in security groups is restored. Details: • Security groups to which the user is added	Failed to restore the user's membership in some security groups. Details: • Security groups to which the user is added • Security groups to which the user is not added due to an error
The user's membership in distribution groups and mail-enabled security groups is restored. Details: • Distribution and mail-enabled security groups to which the user is added	Failed to restore the user's membership in some distribution groups or mail-enabled security groups. Details: • Distribution and mail-enabled security groups to which the user is added • Distribution or mail-enabled security groups to which the user is not added due to an error

Report section: Undo Exchange Mailbox Deprovisioning

Table 43: Undo Exchange Mailbox Deprovisioning

Report item (Success)	Report item (Failure)
Restoration of the user mailbox is skipped because the user did not have an Exchange mailbox at the time of deprovisioning.	N/A
The original state of the user mailbox is restored in the Global Address List (GAL).	Failed to restore the original state of the user mailbox in the Global Address List (GAL).
The original settings for non-delivery reports sending are restored on the user mailbox.	Failed to restore the original settings for non-delivery reports sending on the user mailbox.
The original configuration of the e-mail forwarding is restored on the user mailbox.	Failed to restore the original configuration of the e-mail forwarding on the user mailbox.
The original security settings are restored on the user mailbox.	Failed to restore the original security settings on the user mailbox.
Automatic replies turned off.	Failed to turn off automatic replies.

Report section: Undo Home Folder Deprovisioning

Table 44: Undo Home Folder Deprovisioning

Report item (Success)	Report item (Failure)
Restoration of the home folder is skipped because the user did not have a home folder at the time of deprovisioning.	N/A
The original security settings are restored on the user home folder.	Failed to restore the original security settings on the user home folder.

Report section: Undo User Account Relocation

Table 45: Undo User Account Relocation

Report item (Success)	Report item (Failure)
No changes to undo.	N/A
The user account is moved to its original location. Former location: name of container Restored original location: name of container	Failed to move the user account to its original location. Current location: name of container Failed to move to this location: name of container

Report section: Undo User Account Permanent Deletion

Table 46: Undo User Account Permanent Deletion

Report Item (Success)	Report Item (Failure)
No changes to undo.	*Not applicable*
Scheduled deletion of the user account is canceled.	Failed to cancel scheduled deletion of the user account. The account is going to be deleted on this date: *date*

Report section: Undo Group Object Deprovisioning

Table 47: Undo Group Object Deprovisioning

Report Item (Success)	Report Item (Failure)
The group is changed back to the Security group type.	Failed to change the group back to the Security group type.

Report Item (Success)	Report Item (Failure)
The group is restored in the Global Address List (GAL).	Failed to restore the group in the Global Address List (GAL).
The group name is restored. Old name: *name* Restored name: *name*	Failed to restore the group name. Current name: *name* Failed to set this name: *name*
The membership list of the group is restored. Details: List of the members added to the group	Failed to restore the membership list of the group. Details: List of the members added to the group List of the members that are not added to the group due to an error
Group properties are restored. List: • *Group properties, new property values*	Failed to restore group properties. List: • *Group properties, error description*

Report section: Undo Group Object Relocation

Table 48: Undo Group Object Relocation

Report Item (Success)	Report Item (Failure)
No changes to undo.	*Not applicable*
The group is moved to its original location. Former location: *name of container* Restored original location: *name of container*	Failed to move the group to its original location. Current location: *name of container* Failed to move to this location: *name of container*

Report section: Undo Group Object Permanent Deletion

Table 49: Undo Group Object Permanent Deletion

Report Item (Success)	Report Item (Failure)
No changes to undo.	*Not applicable*
Scheduled deletion of the group is canceled.	Failed to cancel scheduled deletion of the group. The group is going to be deleted on this date: *date*

Container Deletion Prevention policy

A bulk deletion may occur in a situation where an administrator selects and deletes a container object, such as an Organizational Unit, that has subordinate objects. Although bulk deletions are rare, they are disruptive events you can guard against by leveraging a new policy—Container Deletion Prevention.

One of the most common bulk deletions is a container deletion, which occurs when Active Roles is used to delete a container object that holds other (subordinate) objects. By default, a container deletion has the following characteristics:

- First, Active Roles builds a list of all the objects found in the container (subordinate objects), and then starts deleting the listed objects one by one.
- Then, for every object in the list, Active Roles performs an access check to determine if the user or process that requested the deletion has sufficient rights to delete the object. If the access check allows the deletion, then the object is deleted; otherwise, Active Roles does not delete the object, and proceeds to deletion of a subsequent object in the list.
- Finally, once all the subordinate objects are deleted, Active Roles deletes the container itself. If any of the subordinate objects are not deleted, the container is not deleted as well.

As a result of this behavior, an administrator who has full control over an organizational unit in Active Roles can accidentally delete the entire organizational unit, with all its contents, within a single operation. To prevent this, Active Roles provides for a certain policy to deny deletion of non-empty containers.

The Container Deletion Prevention policy defines a configurable list of names of object types as specified by the Active Directory schema (for example, the Organizational Unit object type). When an Active Roles client requests the deletion of a particular container, the Administration Service evaluates the request in order to determine whether the type of the container is in the list defined by the policy. If the container type is in the list and the container holds any objects, the Administration Service denies the request, preventing the deletion of the container. In this case, the client prompts to delete all objects held in the container before attempting to delete the container itself.

To configure a Container Deletion Prevention policy

1. In the console tree, select **Configuration | Policies | Administration | Builtin.**
2. In the details pane, double-click **Built-in Policy - Container Deletion Prevention**.
3. On the **Policies** tab, select the policy from the list and then click **View/Edit**.
4. On the **Types of Containers** tab, click **Add** and use the **Select Object Type** dialog box to select the type (or types) of container you want to protect, and then click **OK**.

   For example, you can select the Organizational Unit object type in order to prevent deletion of non-empty organizational units.
5. Click **OK** to close the dialog boxes you opened.

The built-in Policy Object you have configured using the above instructions prevents deletion of non-empty containers in any managed domain.

You may not want Active Roles to prevent deletion of non-empty containers that are outside a certain scope (such as a certain domain, organizational unit, or Managed Unit), whereas deletion should be prohibited on the non-empty containers that fall within that particular scope. In this scenario, you need to create and configure a copy of the built-in Policy Object and apply that copy to the scope in question. Then, block the effect of the built-in Policy Object by selecting the **Disable all policies included in this Policy Object** check box on the **Policies** tab in the dialog box for managing properties of the Policy Object.

If you only need to allow deletion of non-empty containers within a certain scope, then you can simply block the effect of the built-in Policy Object on the object representing the scope in question. Thus, if you want to allow deletion of organizational units that fall within a certain Managed Unit, you can use the **Enforce Policy** command on that Managed Unit to display the dialog box for managing policy settings and then select the **Blocked check** box next to the name of the built-in Policy Object.

Protecting objects from accidental deletion

Another option to guard organizational units against accidental deletion is by using an Active Roles feature that allows you to deny deletion of particular objects. When creating an organizational unit by using Active Roles, you have the option to protect the newly created organizational unit from deletion. You can also use Active Roles to enable this protection on any existing organizational units or other objects in the managed Active Directory domains and Active Directory Lightweight Directory Services (AD LDS) partitions.

On the pages for creating an organizational unit in the Active Roles console or Web Interface, you can select the **Protect container from accidental deletion** check box. This option removes the Delete and Delete Subtree permissions on the organizational unit and the "Delete All Child Objects" permission on the parent container of the organizational unit. An organizational unit created with this option cannot be deleted, whether using Active Roles or other tools for Active Directory administration, as the deletion-related permissions are removed by applying the appropriate Access Templates in Active Roles and replicating the resulting permission entries to Active Directory.

The option to protect existing organizational units or other objects from deletion is available on the **Object** tab of the **Properties** page for an object in the Active Roles console or Web Interface. If you select the **Protect object from accidental deletion** check box on that tab, Active Roles configures the permission entries on the object in the same way as with the **Protect container from accidental deletion** option for an organizational unit. When somebody attempts to delete a protected object, the operation returns an error indicating that the object is protected or access is denied.

The option to protect an object from deletion adds the following Access Template links:

- On the object to protect, adds a link to the Objects - Deny Deletion Access Template for the Everyone group.

- On the parent container of the object, adds a link to the Objects - Deny Deletion of Child Objects Access Template for the Everyone group. (Active Roles does not add this link if it detects that a link of the same configuration already exists.)

The links are configured to apply the Access Template permission entries not only in Active Roles but also in Active Directory. This adds the following access control entries (ACEs) in Active Directory:

- On the object to protect, adds explicit Deny ACEs for the Delete and Delete Subtree permissions for the Everyone group.
- On the parent container of the object, adds an explicit Deny ACE for the "Delete All Child Objects" permission for the "Everyone" group. (Active Roles does not add this ACE if it detects that an ACE of the same configuration already exists.)

If you clear the **Protect object from accidental deletion** check box for a given object, Active Roles the updates the object to remove the link to the "Objects - Deny Deletion" Access Template in Active Roles along with the explicit Deny ACEs for the "Delete" and "Delete Subtree" permissions for the "Everyone" group in Active Directory. As a result, the object is no longer guarded against deletion. Note that clearing the check box for a particular object removes the Access Template links and ACEs from only that object, leaving the Access Template links and ACEs on the parent container intact. This is because the parent container may hold other objects that are protected from deletion. If the container does not hold any protected objects, you could remove the link to the "Objects - Deny Deletion of Child Objects" Access Template by using the **Delegate Control** command on that container in the Active Roles console, which will also delete the corresponding ACE in Active Directory.

It is possible to configure Active Roles so that the **Protect container from accidental deletion** check box will be selected by default on the pages for creating organizational units in the Active Roles console or Web Interface. To enable this behavior within a domain or container, apply the "Built-in Policy - Set Option to Protect OU from Deletion" Policy Object to that domain or container. This Policy Object ensures that organizational units created by Active Roles are protected from deletion regardless of the method used to create them. Thus, organizational units created using Active Roles script interfaces will also be protected by default.

Picture management rules

You can use the Active Roles console or Web Interface to add a picture for a user, group, or contact object. An advantage of using pictures, such as the photographs or logos, is that a picture makes it easier to recognize the user, group, or contact in e-mail clients and web applications that can retrieve the picture from Active Directory. When you supply a picture for a user, group or contact via Active Roles, the picture is saved in the `thumbnailPhoto` attribute of that user, contact, or group in Active Directory.

Active Roles provides a policy to enforce the picture size limits, including maximum and minimum dimensions and the option to resize the picture automatically. When you add a picture to the user, group, or contact, Active Roles checks the dimensions of the picture, and does not apply the picture in case of policy violation. If automatic picture resizing is

enabled, Active Roles reduces the dimensions of the picture as needed by resampling down the original picture.

You can use the following policy options to configure the picture management rules:

- **Controlled property and object type**. Specifies the object class and the attribute intended to store the picture. The policy fires upon a request to save a picture in the specified attribute of an object of the specified object class. By default, the policy controls the `thumbnailPhoto` attribute of the user, contact, or group object class. You can choose a different attribute for each object class separately. For instance, you can configure the policy to control the `thumbnailLogo` or `jpegPhoto` user attribute while retaining control of the `thumbnailPhoto` attribute of groups and contacts.
- **Maximum allowed size, in pixels**. Specifies the maximum allowed dimensions of the picture. If the width or height of a given picture is greater than specified by this option, then the policy prevents the picture from being applied. The policy has the option to resample pictures of large size. You can configure the policy so that Active Roles automatically reduces the size of the original picture to meet the policy requirements and then applies the resulting picture.
- **Minimum allowed size, in pixels**. Specifies the minimum allowed dimensions of the picture. If the width or height of a given picture is less than specified by this option, then the policy prevents the picture from being applied.
- **Enable automatic picture resizing**. Causes Active Roles to resample the pictures whose dimensions exceed the maximum allowed size. If you select this option, Active Roles reduces the dimensions of the picture as appropriate and then applies the resulting picture; otherwise, Active Roles merely rejects the pictures that are too big.

To view or modify the policy options

1. Open the Active Roles console.
2. In the console tree, select **Configuration | Policies | Administration | Builtin**.
3. In the details pane, double-click **Built-in Policy - Picture Management Rules**.
4. On the **Policies** tab in the **Properties** dialog box that appears, click the policy in the list, and then click **View/Edit**.
5. In the **Properties** dialog box that appears, do the following:
    - On the **Controlled Property** tab, view or change the object class and attribute to which the policy applies.
    - On the **Picture Sizing** tab, view or change the policy settings that restrict the size of the picture stored by the controlled property.

By default, the built-on Policy Object is applied to the Active Directory node in the Active Roles namespace, so the policy options affect all users, groups and contacts in the managed domains. If you need different policy options for different domains or containers, create a copy of the built-in Policy Object, and then configure and apply the copy as appropriate.

Policy extensions

In Active Roles, administrators can configure policies of the pre-defined types that are installed with Active Roles. By default, the list of policy types in the Active Roles console contains only the pre-defined types, such as **Home Folder AutoProvisioning** or **User Account Deprovisioning**. It is possible to extend the list by adding new types of policy.

Each policy type determines a certain policy action (for example, creating a home folder for a user account) together with a collection of policy parameters to configure the policy action (for example, parameters that specify the network location where to create home folders). Active Roles provides the ability to implement and deploy custom types of policy. It enables custom policy types to be created as necessary, and listed along with the pre-defined policy types, allowing administrators to configure policies that perform custom actions determined by those new types of policy.

Active Roles allows the creation of custom policies based on the Script Execution built-in policy type. However, creating and configuring a script policy from scratch can be time-consuming. Custom policy types provide a way to mitigate this overhead. Once a custom policy type is deployed that points to a particular script, administrators can easily configure and apply policies of that type, having those policies perform the actions determined by the script. The policy script also defines the policy parameters specific to the policy type.

Custom policy types provide an extensible mechanism for deploying custom policies. This capability is implemented by using the Policy Type object class. Policy Type objects can be created by using the Active Roles console, with each object representing a certain type of custom policy.

Design elements

The policy extensibility feature is designed around two interactions: policy type deployment and policy type usage.

Policy type deployment

The deployment process involves: the development of a script that implements the policy action and declares the policy parameters; the creation of a Script Module containing that script; and the creation of a Policy Type object referring to that Script Module. To deploy a policy type to a different environment, an administrator can export the policy type to an export file in the source environment and then import the file in the destination environment. Using export files makes it easy to distribute custom policy types.

Policy type usage

This is the process of configuring policies. It occurs when an administrator creates a new Policy Object or adds policies to an existing Policy Object. For example, the wizard for creating a Policy Object includes a page that prompts to select a policy. The page lists the policy types defined in Active Roles, including the custom policy types. If a custom policy type is selected, the wizard provides a page for configuring the policy parameters specific to that policy type. Once the wizard is completed, the Policy Object contains a fully functional policy of the selected custom type.

Active Roles provides a graphical user interface, complete with a programming interface, for creating and managing custom policy types. Using those interfaces, Active Roles policies can be extended to meet the needs of a particular environment. Active Roles also has a deployment mechanism by which administrators put new types of policy into operation.

Since policy extension involves two interactions, Active Roles provides solutions in both areas. The Administration Service maintains policy type definitions, exposing policy types to its clients such as the Active Roles console or ADSI Provider. The console can be used to:

- Create a new custom policy type, either from scratch or by importing a policy type that was exported from another environment.
- Make changes to the definition of an existing custom policy type.
- Add a policy of a particular custom type to a Policy Object, making the necessary changes to the policy parameters provided for by the policy type definition.

Normally, an Active Roles expert develops a custom policy type in a separate environment, and then exports the policy type to an export file. An Active Roles administrator deploys the policy type in the production environment by importing the export file. After that, the Active Roles console can be used to configure and apply policies of the new type.

Policy Type objects

The policy extensibility feature is built upon Policy Type objects, each of which represents a single type of policy. Policy Type objects are used within both the policy type deployment and policy type usage processes. The process of deploying a new policy type involves the creation of a Policy Type object. During the process of adding a policy of a custom type, the policy type definition is retrieved from the respective Policy Type object.

Each Policy Type object holds the following data to define a single policy type:

- **Display name**. Identifies the policy type represented by the Policy Type object. This name is displayed on the wizard page where you select a policy to configure when creating a new Policy Object or adding a policy to an existing Policy Object.
- **Description**. A text describing the policy type. This text is displayed when you select the policy type in the wizard for creating a new Policy Object or in the wizard for adding a policy to an existing Policy Object.

- **Reference to Script Module**. Identifies the script to run upon the execution of a policy of this type. When adding a policy of a custom policy type, you effectively create a policy that runs the script from the Script Module specified by the respective Policy Type object.
- **Policy Type category**. Identifies the category of Policy Object to which a policy of this type can be added. A policy type may have the category option set to either Provisioning or Deprovisioning, allowing policies of that type to be added to either provisioning or deprovisioning Policy Objects respectively.
- **Function to declare parameters**. Identifies the name of the script function that declares the configurable parameters for the administration policy that is based on this policy type. The function must exist in the Script Module selected for the policy type. By default, it is assumed that the parameters are declared by the function named onInit.
- **Policy Type icon** The image that appears next to the display name of the policy type on the wizard page where you select a policy to configure, to help identify and visually distinguish this policy type from the other policy types.

To create a custom policy type, you first need to create a Script Module that holds the policy script. Then, you can create a Policy Type object referring to that Script Module. When you import a policy type, Active Roles automatically creates both the Script Module and the Policy Type object for that policy type. After the Policy Type object has been created, you can add a policy of the new type to a Policy Object.

Creating and managing custom policy types

In Active Roles, Policy Type objects provide the ability to store the definition of a custom policy type in a single object. Policy Type objects can be exported and imported, which makes it easy to distribute custom policies to other environments.

When creating a new Policy Object or adding a policy to an existing Policy Object, an administrator is presented with a list of policy types derived from the Policy Type objects. Selecting a custom policy type from the list causes Active Roles to create a policy based on the settings found in the respective Policy Type object.

This section covers the following tasks specific to custom policy types:

- Creating a Policy Type object
- Changing an existing Policy Type object
- Using Policy Type containers
- Exporting policy types
- Importing policy types
- Configuring a policy of a custom type
- Deleting a Policy Type object

For more information about Policy Type objects, including instructions on scripting for Policy Type objects, refer to the Active Roles SDK.

Creating a Policy Type object

Active Roles stores Policy Type objects in the **Policy Types** container. You can access that container in the Active Roles console by expanding the **Configuration/Server Configuration** branch of the console tree.

To create a new Policy Type object

1. In the console tree, under **Configuration/Server Configuration/Policy Types**, right-click the Policy Type container in which you want to create a new object, and select **New | Policy Type**.

   For example, if you want to create a new object in the root container, right-click **Policy Types**.

2. In the New Object - Policy Type wizard, type a name, a display name and, optionally, a description for the new object.

   The display name and description are displayed on the page for selecting a policy, in the wizards that are used to configure Policy Objects.

3. Click **Next**.
4. Click **Browse** and select the Script Module containing the script that will be run by the policies of this policy type.

   The Script Module must exist under the Configuration/Script Modules container and hold a policy script. For information about policy scripts, see the Active Roles SDK documentation.

5. In the **Policy Type category** area, do one of the following:

   a. Click **Provisioning** if policies of this type are intended for Policy Objects of the provisioning category.

   b. Click **Deprovisioning** if policies of this type are intended for Policy Objects of the deprovisioning category.

   The policy types that have the **Provisioning** option selected appear on the page for selecting a policy in the wizard that is used to create a provisioning Policy Object or to add policies to an existing provisioning Policy Object. The policy types that have the **Deprovisioning** option selected appear in the wizard for creating a deprovisioning Policy Object or adding policies to such a Policy Object.

6. From the **Function to declare parameters** list, select the name of the script function that defines the parameters specific to this type of administration policy.

   The list contains the names of all the functions found in the script you selected in Step 4. Every policy of this type will have the parameters that are specified by the function you select from the **Function to declare parameters** list. Normally, this is a function named onInit (see the Active Roles SDK documentation).

7. Click **Policy Type Icon** to verify the image that denotes this type of policy. To choose a different image, click **Change** and open an icon file containing the image you want.

This image appears next to the display name of the policy type on the wizard page for selecting a policy to configure, to help identify and visually distinguish this policy type from the other policy types.

The image is stored in the Policy Type object. In the dialog box that appears when you click **Policy Type Icon**, you can view the image that is currently used. To revert to the default image, click **Use Default Icon**. If the button is unavailable, then the default image is currently used.

8. Click **Next** and follow the steps in the wizard to complete the creation of the new Policy Type object.

Changing an existing Policy Type object

You can change an existing Policy Type object by changing the general properties, script, category, or icon. The general properties include the name, display name, and description. The Policy Type objects are located under **Configuration/Server Configuration/Policy Types** in the Active Roles console.

The following table summarizes the changes you can make to an existing Policy Type object, assuming that you have found the object in the Active Roles console.

Table 50: Changing an existing Policy Type object

To change	Do this	Commentary
Name of the object	Right-click the object and click **Rename**.	The name is used to identify the object, and must be unique among the objects held in the same Policy Type container.
Display name or description	Right-click the object, click **Properties** and make the necessary changes on the **General** tab.	Changing the display name or description also changes the policy name or description on the page for selecting a policy in the Policy Object management wizards.
Script Module	Right-click the object, click **Properties**, click the **Script** tab, click **Browse**, and then select the Script Module you want.	You can change the script in the Script Module that is currently associated with the Policy Type object instead of selecting a different Script Module. To view or change the script, find and select the Script Module in the Active Roles console tree, under **Configuration/Script Modules**. Changing the script affects all the existing policies of this policy type. If you add a policy to a Policy Object and then change the script for the Policy Type object based on which the policy was created, the policy will run the changed script.

To change	Do this	Commentary
Policy Type category	Right-click the object, click **Properties**, click the **Script** tab, and then click either **Provisioning** or **Deprovisioning**.	Changing this option changes the appearance of the respective policy type in the Policy Object management wizards. For example, once the option has been changed from **Provisioning** to **Deprovisioning**, the policy type is no longer displayed in the wizard for configuring a provisioning Policy Object; instead, it appears in the wizard for configuring a deprovisioning Policy Object. However, changing the Policy Type category does not affect the existing policies of this policy type. For example, once a policy is added to a provisioning Policy Object, the policy is retained in that Policy Object after changing the Policy Type category from **Provisioning** to **Deprovisioning** in the respective Policy Type object.
Function to declare parameters	Right-click the object, click **Properties**, click the **Script** tab, and then choose the appropriate function from the **Function to declare parameters** list.	Changing this setting changes the list of the policy parameters specific to this policy type. The changes do not affect the parameters of the existing policies of this type. When you add a new policy based on this policy type, the list of the policy parameters is built using the new function to declare parameters.
Policy Type icon	Right-click the object, click **Properties**, click the **Script** tab, click **Policy Type Icon**, and then do one of the following: • Click **Change** and open an icon file containing the image you want. • Click **Use Default Icon** to revert to the default image.	Changing this setting changes the image that appears next to the display name of the policy type in the Policy Object management wizards, on the page that prompts you to select a policy to configure.

Using Policy Type containers

You can use a Policy Type container to store related Policy Type objects and other Policy Type containers.

Containers give you an additional way to categorize custom policy types, making it easier to locate and select the policy to configure in the wizards for managing Policy Objects. Thus, when you create a Policy Object, the wizard page that prompts you to select a policy displays the custom policy types along with the containers that hold the respective Policy Type objects.

To create a new Policy Type container

1. In the console tree, under **Configuration/Server Configuration/Policy Types**, right-click the Policy Type container in which you want to create a new container, and select **New | Policy Type Container**.

   For example, if you want to create a new container in the root container, right-click **Policy Types**.

2. In the New Object - Policy Type Container wizard, type a name and, optionally, a description for the new container.

   The name and description are displayed on the page for selecting a policy, in the wizards that are used to configure Policy Objects.

3. Click **Next** and follow the steps in the wizard to complete the creation of the new container.

Exporting policy types

You can export Policy Type objects so that the definition of the policy types is stored in an XML file that can be imported in a different Active Roles environment. Exporting and then importing Policy Type objects make it easy to distribute custom policies to other environments.

To export a Policy Type object or container

- Right-click the Policy Type object or container, click **Export** and specify a file to hold the export data.

You can select multiple Policy Objects to export, or you can select a container to export all Policy Type objects and containers held in that container. In either case, the Export operation creates a single XML file that can later be imported to any container under the **Policy Types** node.

Exporting Policy Type objects creates an XML file representing both the objects and the Script Modules containing the policy scripts for each policy type being exported. During an import, Active Roles creates the Policy Type objects and the Script Modules based on the data found in the XML file. As a result of the import, the policy types are replicated to the new environment and can be used the same way as in the environment from which they were exported.

Importing policy types

You can import the exported Policy Type objects and containers, which will add them to a Policy Type container and allow you to configure and use policies defined by those Policy Type objects. All the data required to deploy the policy types is represented in an XML file. To see an example of the XML document that represents a policy type, export a Policy Type object and view the saved XML file.

To import the exported Policy Type objects and containers

1. In the Active Roles console tree, under **Configuration/Server Configuration/Policy Types**, right-click the Policy Type container in which you want to import the Policy Type objects and containers.
2. Click **Import Policy Types**, and then open the export data file you want to import.

This will create new Policy Type objects and containers in the selected container. In addition, new Script Modules will be created in the **Configuration/Script Modules** container and associated with the newly created Policy Type objects.

Configuring a policy of a custom type

Once a custom policy type has been deployed, an Active Roles administrator can add a policy of that type to a Policy Object. This is accomplished by selecting the policy type in the wizard that creates a new Policy Object or in the wizard that adds a policy to an existing Policy Object.

Which wizards to use, depends upon the policy type category:

- For a policy type of the Provisioning category, a policy of that type can be added only to a Provisioning Policy Object.
- For a policy type of the Deprovisioning category, a policy of that type can be added only to a Deprovisioning Policy Object.

To configure a policy of a custom policy type

1. Follow the steps in the wizard for creating a new Policy Object or in the wizard for adding a policy to an existing Policy Object.

   For example, if the policy type is of the Provisioning category, you could use the New Provisioning Policy Object wizard opened by the **New | Provisioning Policy** command on a container under **Configuration/Policies/Administration** in the Active Roles console.

2. On the **Policy to Configure** page in the wizard, click the type of the policy you want.

   The **Policy to Configure** page lists the custom policy types together with the pre-defined Active Roles policy types. Each custom policy type is identified by the display name of the respective Policy Type object.

The custom policy types are organized in a tree-like structure that reflects the existing hierarchy of the Policy Type containers. For example, if a Policy Type container is created to hold a particular Policy Type object, the container also appears on the wizard page, so you may need to expand the container to view or select the policy type.

3. On the **Policy Parameters** page, set parameter values for the policy: Click the name of a parameter in the list, and then click **Edit**.

   Parameters control the behavior of the policy. When Active Roles executes the policy, it passes the parameter values to the policy script. The actions performed by the script, and the results of those actions, depend upon the parameter values.

   Clicking **Edit** displays a page where you can add, remove or select a value or values for the selected parameter. For each parameter, the policy script defines the name of the parameter and other characteristics, such as a description, a list of acceptable values, the default value, and whether a value is required. If a list of acceptable values is defined, then you can only select values from that list.

4. Follow the wizard pages to complete the wizard.

Deleting a Policy Type object

You can delete a Policy Type object when you no longer need to add policies of the type represented by that object.

Before you delete a Policy Type object, consider the following:

- You can delete a Policy Type object only if no policies of the respective policy type exist in any Policy Object. Examine each Policy Object and remove the policies of that type, if any, from the Policy Object before deleting the Policy Type object.
- Deleting a Policy Type object permanently deletes it from the Active Roles database. If you want to use this policy type again, you should export the Policy Type object to an XML file before deleting the object.
- Deleting a Policy Type object does not delete the Script Module associated with that object. This is because the Script Module may be used by other policies. If the Script Module is no longer needed, it can be deleted separately.

To delete a Policy Type object

- Right-click the Policy Type object in the Active Roles console and click **Delete**.

7

Workflows

- Understanding workflow
- Workflow activities overview
- Configuring a workflow
- Example: Approval workflow
- Email based approval
- Automation workflow
- Activity extensions

Understanding workflow

Active Roles provides a rich workflow system for directory data management automation and integration. Based on Microsoft's Windows Workflow Foundation technology, this workflow system enables IT to define, automate, and enforce management rules quickly and easily. Workflows extend the capabilities of Active Roles by delivering a framework that combines versatile management rules such as provisioning and de-provisioning of identity information in the directory, enforcement of policy rules on changes to identity data, routing data changes for approval, email notifications of particular events and conditions, as well as the ability to implement custom actions using script technologies such as Microsoft Windows PowerShell or VBScript.

Suppose you need to provision user accounts based on data from external systems. The data is retrieved and then conveyed to the directory by using feed services that work in conjunction with Active Roles. A workflow can be created to coordinate the operations in account provisioning. For example, different rules can be applied for creating or updating accounts held in different containers.

Workflows may also include approval rules that require certain changes to be authorized by designated persons (approvers). When designing an approval workflow, the administrator specifies which kind of operation causes the workflow to start, and adds approval rules to the workflow. The approval rules determine who is authorized to approve the operation, the required sequence of approvals, and who needs to be notified of approval tasks or decisions.

By delivering email notifications, workflows extend the reach of management process automation throughout the enterprise. Notification activities in a workflow notify people via email about events, conditions, or tasks awaiting their attention. For example, approval rules can notify of change requests pending approval, or separate notification rules can be applied to inform about data changes in the directory. Notification messages include all necessary supporting information, and provide hyperlinks allowing message recipients to take actions using a standard Web browser.

Key features and definitions

This section summarizes some important concepts that apply to designing and implementing workflows in Active Roles.

Workflow

A workflow is a model describing a process that consists of steps or activities. Workflows describe the order of execution and relationships between activities required to perform particular operations. In Active Roles, workflows provide a way to customize operations of provisioning and overall administration of directory data. Thus, workflows can be used to add approvals to user provisioning processes or integrate user provisioning processes with external systems.

Workflow definition

Workflow definition is a representation of the workflow structure. The definition of a workflow is stored as a single object in the Active Roles configuration data store, and can be structured as an XML document defining the workflow start conditions, the activities, the parameters for the activities, and the order in which the activities should run.

Workflow start conditions

The workflow settings that determine which operations cause the workflow to start are referred to as the workflow start conditions. For example, a workflow can be configured so that any request to create a user account in a specific container starts the workflow.

Workflow instance

Starting a workflow creates a workflow instance based on the settings found in the workflow definition. Each workflow instance stores the runtime data indicating the current state of a single workflow that is in progress.

Workflow activity

A workflow activity is a logically isolated unit that implements a particular operational step of a workflow. The logic incorporated in an activity takes effect both at design time, when you add the activity to a workflow definition, and at runtime, when a workflow instance is executed. When all the activities in a given flow path are finished running, the workflow instance is completed.

Workflow Designer

The Workflow Designer is a graphical tool provided by Active Roles for constructing workflows. The tool represents the workflow definition as a process diagram, with icons denoting workflow activities and directional arrows denoting transitions between activities. Users drag activities from the activities panel onto the process diagram and configure them using the pages provided by the designer interface. Separate pages are provided for configuring workflow start conditions.

Workflow engine

Active Roles leverages Microsoft's Windows Workflow Foundation runtime engine for creating and maintaining workflow instances. The engine can support multiple workflow instances running concurrently. When a workflow is started, the engine monitors the state of the workflow instance, coordinates the routing of activities in the workflow instance, determines which activities are eligible to run, and runs activities. The workflow engine is hosted in-process with the Administration Service, which enables workflows to communicate with Active Roles at run time.

E-mail Notifications

Users are notified via e-mail about specific situations that manifest within a workflow. A notification message is generated and sent to the designated recipients to inform them that a certain event has occurred, such as a new approval task has been submitted to the approvers or the operation has been completed. A notification configuration involves such elements as the event to notify of, the list of the notification recipients, and the notification message template.

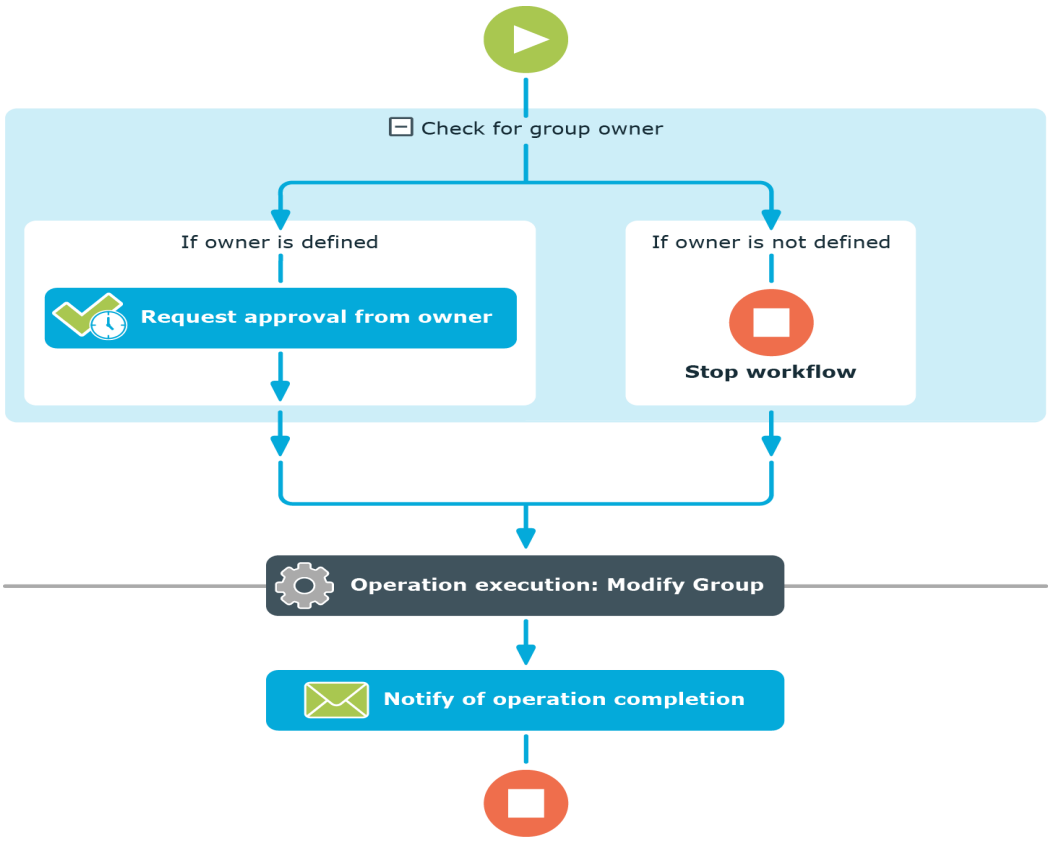
About workflow processes

The logic of an automated management process can be implemented by using administrative policies in Active Roles. Yet creating and maintaining complex, multi-step processes in that way can be challenging. Workflows provide a different approach, allowing

IT administrators to define a management process graphically. This can be faster than building the process by applying individual policies, and it also makes the process easier to understand, explain, and change.

The figure below shows a workflow process created in the Active Roles console. In this simple example, upon a request to add a user to a certain group, the workflow first checks to see if the group has an owner. If the group has no owner, the requested changes are denied and the workflow is complete; otherwise, the changes are submitted to the group owner for approval. When approval is received, Active Roles applies the changes, adding the user to the group. On the process diagram, this step is referred to as **Operation execution**. If the owner rejects the changes, the workflow finishes on the previous (approval) step so that the changes are not applied. After the changes are made, the workflow sends an e-mail notification to the person who requested the changes, and then finishes.

Figure 90: Workflow process in Active Roles



In the above example, the workflow manages the process of adding a user to a group according to the rules defined at design time. The rules constitute the workflow definition, and include the activities that occur within the process and the relationships between activities. An activity in a process definition can be a pre-defined function available out of the box, such as a request for approval or a notification of conditions that require user interaction, or it can be a custom function created using script technologies.

A workflow process starts when the requested changes meet the conditions specified in the workflow definition. In the above example, the conditions may set up so that the workflow

starts whenever an Active Roles user makes changes to the membership list of a certain group. Once the conditions are fulfilled, the workflow process starts to drive the changes through the workflow definition, performing automated steps and, if necessary, requesting human interaction such as approval.

Workflow processing overview

In Active Roles, directory objects such as users, groups, or computers are managed by the Administration Service. These objects can be created, changed, or deleted through requests made to the Administration Service. Every request initiates an operation to make the requested changes to directory data. For example, a request to create a user or group initiates the Create operation with the target object type set to User or Group, respectively; a request to add users to a group initiates the Modify operation on that group.

Once an operation has been initiated, the Administration Service starts processing the operation. Each operation is represented by a single object, usually referred to as the Request object, which contains all information necessary to perform the operation. Therefore, operation processing takes the form of passing the Request object through a number of phases within the Administration Service.

The operation processing model in Active Roles is composed of four main phases: access check, pre-execution, execution, and post-execution. The Request object passes through these phases in the following order:

- **Access check**. In this phase, the Administration Service checks to see whether the user or system that issued the request has sufficient rights to make the requested changes. If there are insufficient rights, the operation is denied.
- **Pre-execution**. During this phase, the Administration Service first runs the pre-execution workflow activities. These are the activities located in the upper part of the workflow process diagram, above the **Operation execution** line. A typical example includes Approval activities: It is at this point that approvers can permit or reject the operation.

  Then, after the pre-execution activities are completed so that the operation is not rejected, the Administration Service runs the pre-execution policies. Typical examples of such policies include property generation and validation rules and the functions implementing so-called pre-event handlers in script policies.
- **Execution** In this phase, the Administration Service performs the operation, making the requested changes to directory data. For example, when the creation of a user is requested, the user is actually created during this phase.
- **Post-execution**. During this phase, the Administration Service first runs the post-execution policies. For example, upon creation of a user, the provisioning of a home folder or group memberships for that user occurs at this point. The functions that implement post-event handlers in script policies are also run in this step.

  Finally, after the post-execution polices finish running, the Administration Service runs the post-execution workflow activities. These are the activities located in the lower part of the workflow process diagram, beneath the **Operation execution** line.

A typical example is Notification activities that send out e-mails informing of the operation completion.

The Administration Service runs the workflow activities one by one, in sequential order as shown on the workflow process diagram, until the last activity finishes. If-Else activities can be used to achieve conditional branching in workflows, which makes it possible to switch the sequence of activities depending on the data involved in the request.

At the beginning of the pre-execution phase, the Administration Service determines the workflows to start. The request is compared to all the existing workflow definitions. In order for a workflow to start, the requested operation needs to satisfy the start conditions defined for that workflow. If the start conditions are satisfied, the workflow is matched to the request.

For a workflow that is matched to the request, the Administration Service runs the activities found in that workflow during the corresponding phases of the operation processing. One workflow or multiple workflows can be matched to a single request. In case of multiple workflows, the Administration Service starts each of them one by one, and first runs all the pre-execution activities included in those workflows. Then, during the post-execution phase, the Administration Service runs all the post-execution activities included in those workflows.

If multiple workflows are matched to a single request, then Active Roles uses the `edsaWorkflowPriority` attribute of the workflow definition object to determine the order in which to execute the workflows. The activities of the workflow with a lower value of that attribute are executed prior to the activities of the workflow with a higher value of that attribute. The workflows with the same priority value are executed in ascending order of workflow names. The `edsaWorkflowPriority` attribute is set to 500 by default. If the `edsaWorkflowPriority` attribute is not set, Active Roles assumes that the workflow has the priority value of 500. You can change the value of the `edsaWorkflowPriority` attribute to ensure that a given workflow takes precedence over other workflows. A lower value of that attribute indicates a higher priority whereas a higher value indicates a lower priority. To view or change the `edsaWorkflowPriority` attribute, use the **Advanced Properties** command on the workflow definition object in the Active Roles console.

About start conditions

To deploy a workflow in Active Roles, you create a workflow definition, configure the start conditions for that workflow, and add and configure workflow activities. When configuring workflow start conditions, you specify:

- A type of operation, such as Create, Rename, Modify or Delete; the workflow is matched to the request only if an operation of that type is requested.
- A type of object, such as User, Group or Computer; the workflow is matched to the request only if the operation requests changes to an object of that type.
- For the Modify operation type, a list of object properties; the workflow is matched to the request only if the operation requests changes to any of those properties of an object.

- The identity of an operation requestor (initiator), such as a user, group, or service; the workflow is matched to the request only if the operation is requested on behalf of that identity.
- A container, such as an Organizational Unit or Managed Unit; the workflow is matched to the request only if the operation requests changes to, or creation of, an object in that container.
- (Optional) A filter that defines any additional conditions on entities involved in an operation; the workflow is matched to the request only if the operation satisfies those conditions. If no filter is set, then no additional conditions are in effect.

Upon a request for any operation that meets all the start conditions specified on a workflow, the Administration Service matches the workflow to the request and runs the activities found in the workflow.

Workflow activities overview

Activities are units of work, each of which contributes to the accomplishment of a workflow process. Active Roles offers a default set of activities that provide pre-defined functionality for approval, notification, control flow, and conditions. Scripting can be used to have an activity perform custom functions.

Activities are the primary building blocks for workflows. A workflow is basically a set of activities organized in a process diagram. When you construct a workflow using the Workflows Designer, you drag activities from the activities panel onto the process diagram and then configure them there. The configurable settings common to every activity are:

- **Name**. The name is used to identify the activity on the workflow diagram.
- **Description**. This optional text can be helpful to distinguish the activity. The description is displayed when you point with the mouse to the activity on the process diagram.

The following sections elaborate on the types of activity that are included with Active Roles, and provide information about the configurable settings specific to each activity type.

Approval activity

An Approval activity, also referred to as an approval rule, represents a decision point in a workflow that is used to obtain authorization from a person before continuing the workflow. Workflow start conditions determine which operations start the workflow and the approval rules added to the workflow determine who is designated to approve the operation, the

required sequence of approvals, and who needs to be notified of approval tasks or decisions.

Active Roles creates an approval task as part of the processing of an approval rule, and assigns the task to the approvers. The approver is expected to complete the task by making a decision to allow or deny the operation. Until the task is completed, the operation remains in a pending state.

The following topics cover the configurable settings specific to an Approval activity.

Approvers and escalation

Approvers are the users or groups of users designated to perform approval tasks. When processing an approval rule, Active Roles creates an approval task and assigns it to the approvers defined by the rule. The state of the task governs the workflow transition: The task must receive the Approve resolution for the operation to pass the approval rule. If the task has received the Reject resolution, the operation is denied and the workflow instance is completed.

Approvers may be selected by browsing the available users and groups, or particular role holders may be designated as approvers. For example, an approval rule can be configured so as to require approval by the manager of the operation requestor or by the manager of the group or container that is affected by the operation.

An approval rule may define two or more approver levels, with each level containing a separate list of approvers. Active Roles uses approver levels when escalating time-limited approval tasks. For each approver level the approval rule can specify a certain time period. If an approver of a given level does not complete the approval task within the specified time period, then Active Roles can assign the task to the approvers of the next level. This process is referred to as *escalation*.

Each approver level has the following configuration options:

- **List of approvers**. Specifies the users or groups of users that are designated as approvers for the approver level in question.

  A valid approval rule must, at a minimum, specify a list of approvers for the initial approver level. Active Roles first assigns the approval task to the approvers of that level. To enable escalation, a separate list of approvers must be specified for one or more escalation levels.

- **Approval task has no time limit**. When this option is selected, the approval rule does not require that the approvers of the given level complete the approval task within a certain time period.

- **Approval task has a time limit of** <number> **days** <number> **hours**. When this option is selected, the approval rule requires that the approvers of the given level complete the approval task within the specified time period.

  If the approval task is not completed within the specified time period, then, depending upon the selected configuration option, the approval rule can either cancel the operation waiting for approval or escalate the approval task. The latter option requires a list of approvers to be specified for the subsequent escalation level.

- **Allow approver to delegate approval task**. When this option is selected, the approver of the given level is allowed to assign the approval task to other persons. On the pages for performing the approval task, the approver can use the **Delegate** button to select the persons to assign the task to.
- **Allow approver to escalate approval task**. When this option is selected, the approver of the given level is allowed to escalate the approval task. On the pages for performing the approval task, the approver can use the **Escalate** button to assign the task to the approvers of the subsequent escalation level. This option requires a list of approvers to be specified for the subsequent escalation level.

Request for information

You can configure the Approval activity so that the approver will be requested to supply certain properties of the object when performing the approval task. Suppose the creation of a user is submitted for approval. The approver may be requested to supply certain properties of the user in addition to the properties specified in the creation request. Thus, you may configure the Approval activity to prompt the approver to specify the mailbox database for the mailbox of the user to be created.

It is also possible to configure the Approval activity so that the approver will be requested to review the object properties submitted for approval. One more option is to allow the approver to make changes to those properties.

The pages for configuring an Approval activity in the Active Roles console include the following options related to request for information:

- **Show this instruction to the approver**. When performing the approval task, the approver will see this instruction on the page intended to review, supply, or change the properties that are subject to the approval task. You can supply an instruction on how to perform the task.
- **Request the approver to supply or change these properties**. When performing the approval task, the approver will be prompted to supply or change the properties specified in this option.
- **Show the original request to the approver**. This option adds a separate section on the pages for performing the approval task that lists the properties submitted for approval.
- **Allow the approver to modify the original request**. Unless this option is selected, the approver is only allowed to view the properties submitted for approval. You could select this check box to allow the approver to change those properties.

Customization

You can configure the Approval activity to specify how the approval tasks created by that activity are to be identified in the Approval section of the Web Interface. The Approval section contains a list of approval tasks, with each task identified by a header that provides basic information about the task, including the title of the task and information about the

target object of the operation that is subject to approval. The title of the task is located in the middle of the task's header. The properties that identify the operation target object are displayed above the title of the task.

The pages for configuring an Approval activity in the Active Roles console provide the following customization options related to the header of the approval task:

- **Display this title to identify the approval task**. When performing the approval task, the approver will see this instruction on the page intended to review, supply or change the properties that are subject to the approval task. You can supply an instruction on how to perform the task.
- **Display these properties of the object submitted for approval**. These properties will be displayed in the task's header area on the pages for performing the approval task. You can add properties to help the approver identify the target object of the operation submitted for approval.
- **Display the operation summary in the task header area**. This option extends the approval task's header area to provide summary information about the changes that are subject to approval, including the type of the changes and the reason for the changes.

You can configure the Approval activity to specify the actions the approver can take on the approval task. On the pages for performing the approval task, in the Approval section of the Web Interface, the task header contains the action buttons that are intended to apply the appropriate resolution to the task, such as **Approve** or **Reject**. The action buttons are located at the bottom of the header area. Which buttons are displayed depends upon configuration of the Approval activity.

The pages for configuring an Approval activity in the Active Roles console provide the following customization options related to the action buttons:

- **Customize action buttons**. Action buttons appear on the pages for performing the approval task. Each button applies a certain action to the task. Normally, two built-in buttons, titled **Approve** and **Reject** by default, are displayed for each approval task. Other buttons may be displayed depending on the configuration of the approval activity. You can add buttons to create custom actions. Depending on the button's action type, clicking a custom action button causes the workflow to allow (Complete action type) or deny (Reject action type) the operation that is subject to approval. If-Else activities can refer to a custom action button by the button's title and elect the appropriate branch of the workflow when the approver clicks that custom action button.
- **Show this instruction for action buttons**. You can use this option to supply an instruction on how to use action buttons. The approver will see this instruction above the action buttons on the pages for performing the approval task.
- **Suppress the confirmation dialog upon completion of approval task**. If this option is not selected, Active Roles requests the approver to fill in a confirmation dialog box every time the approver performs an approval task. You can select this option to prevent the confirmation dialog box from appearing so that the approver can complete the task without having to supply a reason for the completion of the task.

Notification

Notification is used to subscribe recipients to the notifications of approval-related events, configure notification e-mails, and set up e-mail transport. Approval rules provide e-mail notifications to workflow users in association with various events, such as the creation of approval tasks upon operation requests. Thus, approvers can be notified of the requests awaiting their approval via e-mails that include hypertext links to the approval-related section in the Web Interface. An Approval activity has the following notification settings.

Notification recipients

Notification recipients are the users or groups to which the activity sends e-mails. A recipient can be any mailbox-enabled user or mail-enabled group. There are also a number of options allowing you to select recipients based on their role, such as operation requestor, approver, manager of operation requestor, or manager of object affected by the operation.

Notification delivery

The delivery options determine whether notifications are to be sent immediately or on a scheduled basis. The option of immediate delivery causes the activity to generate a separate message upon every occurrence of the event to notify of. The option of scheduled delivery can be used for aggregating notifications. If you select the scheduled delivery option, all notifications about the event occurrences within a time period of your choice are grouped and sent as a single message.

Notification message

Notification messages are based on a message template that determines the format and contents of an e-mail notification message, including the message subject and body. A template is an HTML-formatted document that you can view or change as required to customize notification messages. The template text may include dynamic content that is generated at run time by retrieving information from the running instance of the workflow process. Notification messages are created, and normally sent, in HTML format. You can optionally configure the activity to format and send notification messages as plain text.

Web Interface address

The Web Interface address setting specifies the address (URL) of the Active Roles Web Interface. The activity uses this setting to construct hyperlinks in the notification messages.

E-mail server

The e-mail server setting determines the name and other parameters of the e-mail server that is used for delivery of notification messages.

Notification activity

A Notification activity in a workflow is used to subscribe recipients to the notifications of the following events:

- **Executing this activity**. This event occurs upon execution of the notification activity. When configured to notify of this event, the activity creates and instantly sends an e-mail message informing about the fact of executing the notification activity. Notification of this event is normally intended to inform that the workflow execution process has reached the notification activity.
- **Workflow completed successfully**. When configured to notify of this event, the activity creates a message to be sent upon workflow completion. When the workflow is completed, Active Roles will send that message if no considerable errors occurred during execution of the workflow.
- **Workflow encountered an error**. When configured to notify of this event, the activity creates a message to be sent upon workflow completion. When the workflow is completed, Active Roles will send that message if some errors occurred during execution of the workflow.
- **Operation performed**. When configured to notify of this event, the activity creates a message to be sent upon workflow completion. When the workflow is completed, Active Roles will send that message if the operation that started the workflow is successfully performed.

The configuration of a Notification activity specifies the event to notify of, and notification recipients. When executed by the workflow, the Notification activity prepares a notification message appropriate to the specified event. Active Roles retains the message prepared by the activity, and sends the message to the specified recipients upon occurrence of that event. The configurable settings of a Notification activity are similar to the notification settings of an Approval activity, and include the following.

Notification recipients

Notification recipients are the users or groups to which the activity sends e-mails. A recipient can be any mailbox-enabled user or mail-enabled group. There are also a number of options allowing you to select recipients based on their role, such as operation requestor, approver, manager of operation requestor, or manager of object affected by the operation.

Notification message

Notification messages are based on a message template that determines the format and contents of an e-mail notification message, including the message subject and body. A template is an HTML-formatted document that you can view or change as required to customize notification messages. The template text may include dynamic content that is generated at runtime by retrieving information from the running instance of the workflow process. You have the option to attach a report on the workflow execution results to the notification message. Notification messages are created, and normally sent, in HTML format. You can optionally configure the activity to format and send notification messages as plain text.

Web Interface address

The Web Interface address setting specifies the address (URL) of the Active Roles Web Interface. The activity uses this setting to construct hyperlinks in the notification messages.

E-mail server

The e-mail server setting determines the name and other parameters of the e-mail server that is used for delivery of notification messages.

Script activity

Script activities are typically used to perform automated steps in a workflow process. A Script activity is defined by a Script module created in Active Roles. Each Script module contains script code implementing certain functions. New Script modules can freely be added and the Script contained in a Script module can be developed and revised as necessary. This provides a mechanism for creating custom functions, enabling the extensibility of actions performed by a workflow.

Script activity has the following basic configuration settings:

- **Script to use**. Identifies the Script module to be used by the activity. Normally, the script held in the Script Module implements at least two functions: the function that will be run by the activity and the function that defines the activity parameters.
- **Function to run**. Identifies the script function that will be run by the activity.
- **Function to declare parameters**. Identifies the Script function that defines the activity parameters. For each parameter, this function defines the name of the parameter and other characteristics, such as a description, a list of possible values,

the default value, and whether a value is required. Normally, the parameters are declared by a function named onInit.

- **Parameter values**. When Active Roles executes a Script activity, it passes the parameter values to the script function being run by that activity. The actions performed by the activity, and the results of those actions, depend upon the parameter values.

More information and instructions that apply to designing, implementing and using scripts, script modules, and script activities can be found in the Active Roles SDK documentation.

Notification

You can configure a Script activity to subscribe recipients to the notifications of the following events:

- **Activity completed successfully**. When configured to notify of this event, the activity causes Active Roles to send a notification e-mail if no significant errors occurred during execution of this activity.
- **Activity encountered an error**. When configured to notify of this event, the activity causes Active Roles to send a notification e-mail if any significant errors occurred during execution of this activity.

The notification settings specify the event to notify of, and notification recipients. When executed by the workflow, the activity prepares a notification message appropriate to the specified event. Active Roles retains the message prepared by the activity, and sends the message to the specified recipients upon occurrence of that event. The notification settings are similar to the notification settings of a Notification activity (see Notification activity earlier in this document).

Error handling

When configuring a Script activity, you can choose whether to suppress errors encountered by that activity. The following option is available: **Continue workflow even if this activity encounters an error**. If this option is not selected (default setting), then an error condition encountered by the activity causes Active Roles to terminate the workflow. If you select this option, the workflow continues regardless of whether or not the activity encounters an error condition.

If-Else activity

An If-Else activity is used to conditionally run one of two or more alternative branches depending on the conditions defined on the branches. It contains an ordered set of branches and runs the first branch whose condition evaluates to TRUE. You can add as

many branches as you want to an If-Else activity, and you can add as many activities as you want to every branch.

Each branch of an If-Else activity may have an individual condition set on it. When an If-Else activity starts, if evaluates the condition on its first (leftmost) branch. If the condition if fulfilled, the activities that are contained in the branch are executed; otherwise, the condition on the next branch (from left to right) is evaluated, and so on.

When configuring If-Else branch conditions, consider that:

- Only the first branch whose condition evaluates to TRUE is executed.
- An If-Else activity can finish without having run any of its branches, if the condition on each of the branches evaluates to FALSE.

The situation where no condition is defined on a branch is treated as if the branch had a constant TRUE condition. Therefore, the final (rightmost) branch should normally have no condition, which means it always evaluates to TRUE. In this way, the final branch acts as the Else branch that runs if the conditions on the other branches are not fulfilled. It is advisable to define a condition on each branch in an If-Else activity except the last branch, to ensure that the activity always executes a certain branch.

If-Else branch conditions

An If-Else activity is intended to select exactly one branch of the activity from a given set of branches. For each branch, the activity checks the branch conditions and executes the first of the branches whose condition evaluates to TRUE.

When you configure an If-Else branch, you need to add at least one condition, but you are not limited in the number of conditions that you can add for a given branch. You can add, delete, and group conditions using various operators. It is possible to nest condition groups within other condition groups to achieve the results that you want.

A condition group contains one or more conditions connected by the same logical operator. By grouping conditions, you specify that those conditions should be evaluated as a single unit. The effect is the same as if you put parentheses around an expression in a mathematical equation or logic statement.

By default, a single, implied condition group is created when you add a branch condition. You can create additional condition groups to group a set of conditions and nest grouped conditions within other condition groups.

In a condition group, conditions are connected using the AND, OR, NOT AND, or NOT OR logical operator:

- AND group evaluates to TRUE if all conditions in the group are TRUE.
- OR group evaluates to TRUE if any condition in the group is TRUE.
- NOT AND group evaluates to TRUE if any condition in the group evaluates to FALSE.
- NOT OR group evaluates to TRUE if all conditions in the group evaluate to FALSE.

By default, AND is the logical operator between the conditions in a condition group. It is possible to change the logical operator by converting the condition group to a different group type.

When you add a condition, the workflow designer first prompts you to specify what you want the condition to evaluate. The following options are available:

- **Property of workflow initiator**. This option is intended to evaluate the value of a certain property of the user whose request started the workflow. You can select the desired property when you configure a condition.
- **Activity execution status**. This option is intended to evaluate whether or not Active Roles encountered an error when executing a certain activity. You can select the desired activity when you configure a condition. Note that this option requires the activity configuration to allow the workflow to continue even if the activity encounters an error (see Error handling for create, read, update, and delete activities).
- **Workflow parameter value**. This option is intended to evaluate the value of a certain parameter of the workflow. You can select the desired parameter when you configure a condition.
- **Property of object from workflow data context**. This option is intended to evaluate the value of a certain property of the object that will be selected by the If-Else activity on the basis of the data found in the workflow environment at the time of executing the workflow. When you configure a branch condition, you can choose the desired property and specify which object you want the activity to select upon evaluating the condition at workflow run time.
- **Value generated by rule expression**. This option is intended to evaluate the string value of a certain rule expression. By using a rule expression, you can compose a string value based on properties of various objects found in the workflow environment at the time of executing the workflow. Active Roles calculates the value of your rule expression upon evaluating the condition at workflow run time.

Within a change workflow, the following options are available in addition to the options listed above:

- **Property of workflow target object**. This option is intended to evaluate the value of a certain property of the target object of the request that started the workflow. You can select the desired property when you configure a condition.
- **Changed value of workflow target object property**. This option is intended to evaluate the value that is requested to be assigned to a certain property of the workflow target object, which represents the requested change to the property of the target object of the request that started the workflow. You can select the desired property when you configure a condition.
- **Approver action choice**. This option is intended to evaluate the name of the action button applied by the approver to complete the approval task created by a certain Approval activity. Use this option to determine which action button the approver applied to allow the operation that was subject to approval. You can select the desired Approval activity when you configure a condition.

Once you have specified the entity or field that you want the condition to evaluate, you can choose a comparison operator and specify a comparison value. The list of options that are available to specify a comparison value depends upon the entity or field you have configured the condition to evaluate. The following table summarizes the comparison value options.

Table 51: Comparison value options

Condition to evaluate	Comparison value options
Property of workflow target object - OR - Property of workflow initiator - OR - Changed value of workflow target object property - OR - Workflow parameter value - OR - Property of object from workflow data context - OR - Value generated by rule expression	• Text string • Property of workflow target object • Property of workflow initiator • Changed value of workflow target object property • Workflow parameter value • Property of object from workflow data context • Value generated by rule expression
Activity execution status	• Not executed • Completed successfully • Encountered an error
Approver action choice	• The name of an action button • Value generated by script

For a brief description of comparison operators and comparison value options, see Search filter.

Error handling

When configuring an If-Else activity, you can choose whether to suppress errors encountered by that activity. The following option is available: **Continue workflow even if this activity encounters an error**. If this option is not selected (default setting), then an error condition encountered by the activity causes Active Roles to terminate the workflow. If you select this option, the workflow continues regardless of whether or not the If-Else activity or any activity within the If-Else activity encounters an error condition.

Stop/Break activity

A Stop/Break activity is used to immediately end all activities of a running workflow instance. You can use it within a branch of an If-Else activity, so as to terminate the workflow once a certain condition occurs.

An example is a requirement for the validation of the requested data changes to deny certain operations because applying such operations would result in unacceptable data being written to the directory. To address this requirement, you can use a workflow with an If-Else branch that runs upon detection of unacceptable data in the requested operation, and add a Stop/Break activity to that branch. In this way, your workflow will block the unwanted operations, safeguarding the directory data.

The Stop/Break activity logs a message when terminating the workflow instance. You can specify a message text as an activity setting to provide the reason for the workflow instance termination. The activity includes that message in the event that is recorded to the Active Roles event log on the computer running the Active Roles Administration Service.

Add Report Section activity

You can use the Add Report Section activity to add custom information to the change history report (in case of workflow started by an operation request) or run history report (in case of automation workflow). The activity adds a separate section to the **Workflow activities and policy actions** area of the report. The section consists of a header and a body. The activity provides the following options for configuring the text to be displayed in the header and the body of the report section:

- You can specify whether the report section is intended to display information about successful operation or error condition. In the latter case, the text of the header and the body of the report section is displayed in red.
- You can compose the header text of data entries that will be calculated during execution of the activity. The activity offers various data entry types, allowing the header text to include properties of objects involved in the workflow and related objects, date and time of activity execution, and workflow parameters.
- You can configure the body text to include multiple strings, with each string composed by using the same options that are available for the header text string. Thus, in addition to literal text strings and formatting characters, the body text may include information about object properties and other string values the activity will calculate in workflow run time.

You can also add the Add Report Section activity to a certain If-Else branch to have the report indicate that the workflow executed that branch of activities.

Search activity

A Search activity allows you to perform searches against directory data to find objects, such as users or groups, that match the criteria you specify based on object properties, object location, and other information available in the execution environment of the workflow, and to pass these objects to other activities so that the workflow can perform the appropriate actions on them. You can insert activities into a Search activity and have those activities process the objects found by the Search activity.

The following topics cover the configurable settings of a Search activity:

- Search scenario
- Object type
- Search scope
- Search options
- Search for inactive accounts
- Search filter
- Notification
- Error handling
- "Run as" options
- Additional settings
- Stop Search activity

Search scenario

You can configure a Search activity to:

- **Search in the Organizational Unit or container**. Search a certain OU or container for objects that match your search criteria.
- **Search for resources managed or owned by the user or group**. Search for the managed objects of a particular user or group that match your search criteria. Managed objects of a user or group are those for which the user or group is the primary owner (manager) or a secondary owner.
- **Search the group for its members**. Search for the members of a certain group that match your search criteria.
- **Search for direct reports of the user**. Search for the direct reports of a particular user that match your search criteria. Direct reports of a given user are the users for which that user is the manager.
- **Search within the object's attribute (ASQ search)**. Search for the objects listed in a certain attribute of a particular object that match your search criteria.

Object type

You can specify the type of the objects you want the activity to search for. The list from which to select the object type varies depending on the search scenario you have selected.

Table 52: Search activity: Object type

Search scenario	Object types to search for
Search in the Organizational Unit or container.	• Users

Search scenario	Object types to search for
	• Contacts • Groups • Computers • Printers • Organizational Units • Shared Folders • Exchange Recipients • Inactive Accounts • All Objects
Search for resources managed or owned by the user or group. - OR - Search within the object's attribute (ASQ search).	• Users • Contacts • Groups • Computers • Printers • Organizational Units • Shared Folders • Exchange Recipients • All Objects
Search the group for its members.	• Users • Contacts • Groups • Computers • Exchange Recipients • All Objects
Search for direct reports of the user.	• Users • All Objects

Search scope

The search scope determines where to search for the objects of the specified type. The search scope settings depend upon the search scenario, and are as follows.

Table 53: Search activity: Search scope

Search scenario	Search scope settings available
Search in the Organizational Unit or container.	• **Fixed container in directory**. Search in the given OU or container. You can select the desired OU or container in Active Directory when you configure a Search activity. • **Parent OU of workflow target object**. Search in the OU that holds the target object of the request that started the workflow. • **Object identified by workflow parameter**. Search in the OU or container specified by the value of a certain parameter of the workflow. You can choose the desired parameter when you configure a Search activity. • **Object from workflow data context**. Search in the OU or container that will be selected by the Search activity on the basis of the data found in the workflow environment at the time of executing the workflow. When configuring a Search activity, you can specify which OU or container you want the activity to select at workflow run time. • **Object identified by DN-value rule expression**. Search in the OU or container whose Distinguished Name (DN) is specified by the string value of a certain rule expression. By using a rule expression you can compose a string value based on properties of various objects found in the workflow environment at the time of executing the workflow. You can create the desired rule expression when you configure a Search activity.
Search for resources managed or owned by the user or group.	• **Workflow target object**. Search for resources managed or owned by the target object of the request that started the workflow. • **Object identified by workflow parameter**. Search for resources managed or owned by the object specified by the value of a certain parameter of the workflow. You can choose the desired parameter when you configure a Search activity. • **Object from workflow data context**. Search for resources managed or owned by the object that will be selected by the Search activity on the basis of the data found in the workflow environment at the time of executing the workflow. When configuring a Search activity, you can specify which object you want the activity to select at workflow run time. • **Object identified by DN-value rule expression**. Search for resources managed or owned by the object

Search scenario	Search scope settings available
	whose Distinguished Name (DN) is specified by the string value of a certain rule expression. By using a rule expression, you can compose a string value based on properties of various objects found in the workflow environment at the time of executing the workflow. You can create the desired rule expression when you configure a Search activity.
Search the group for its members.	• **Workflow target object**. Search for members of the group that is the target object of the request that started the workflow. • **Object identified by workflow parameter**. Search the group specified by the value of a certain parameter of the workflow. You can choose the desired parameter when you configure a Search activity. • **Object from workflow data context**. Search for members of the group object that will be selected by the Search activity on the basis of the data found in the workflow environment at the time of executing the workflow. When configuring a Search activity, you can specify which group object you want the activity to select at workflow run time. • **Object identified by DN-value rule expression**. Search the group whose Distinguished Name (DN) is specified by the string value of a certain rule expression. By using a rule expression, you can compose a string value based on properties of various objects found in the workflow environment at the time of executing the workflow. You can create the desired rule expression when you configure a Search activity.
Search for direct reports of the user.	• **Workflow target object**. Search for direct reports of the target object of the request that started the workflow. • **Object identified by workflow parameter**. Search for direct reports of the object specified by the value of a certain parameter of the workflow. You can choose the desired parameter when you configure a Search activity. • **Object from workflow data context**. Search for direct reports of the object that will be selected by the Search activity on the basis of the data found in the workflow environment at the time of executing the workflow. When configuring a Search activity, you can specify which object you want the activity to select at workflow run time.

Search scenario	Search scope settings available
	• **Object identified by DN-value rule expression**. Search for direct reports of the object whose Distinguished Name (DN) is specified by the string value of a certain rule expression. By using a rule expression, you can compose a string value based on properties of various objects found in the workflow environment at the time of executing the workflow. You can create the desired rule expression when you configure a Search activity.
Search within the object's attribute (ASQ search).	• **Fixed object in directory**. Search in a certain attribute of the given object. You can select the desired object in Active Directory when you configure a Search activity. • **Workflow target object**. Search in a certain attribute of the target object of the request that started the workflow. • **Object from workflow data context**. Search in a certain attribute of the object that will be selected by the Search activity on the basis of the data found in the workflow environment at the time of executing the workflow. When configuring a Search activity, you can specify which object you want the activity to select at workflow run time.

Search options

The activity provides various options allowing you to refine your search. Which options are available depends upon the search scenario and the object type to search for, as shown in the tables that follow.

The following table summarizes the search scenario-specific search options.

Table 54: Search activity: Search options

Search scenario	Search options available
Search in the Organizational Unit or container.	• **Retrieve only immediate child objects of the Organizational Unit or container**. Use this option to restrict the search to objects for which the given OU or container is the immediate parent in Active Directory. • **Retrieve any objects held in the Organizational Unit or container**. Use this option to search in the entire directory tree rooted in the given OU or container.
Search for resources managed or owned by the user or group.	• **Retrieve objects managed by the user or group (primary owner)**. Use this option to search for objects that have the given user or group specified in the

Search scenario	Search options available
	Managed By property. • **Retrieve objects for which the user or group is a secondary owner**. Use this option to search for objects that have the given user or group specified in the Secondary Owners property. • **Retrieve objects managed or owned due to membership in groups (indirect ownership)**. Use this option to search for objects for which the given user or group is a direct or indirect member of the group specified in the Managed By or Secondary Owners property.
Search the group for its members.	• **Also retrieve indirect members**. Use this option for your search results to include indirect members of the given group. With this option, the activity searches not only for objects that are directly added to the group (direct members) but also for indirect members-objects that belong to the group because of their membership in other groups which are direct or indirect members of the given group. • **Also retrieve pending members**. Use this option for your search results to include objects that are scheduled to be added to the group by using the Temporal Group Memberships capability of Active Roles.
Search within the object's attribute (ASQ search).	• **Search within this attribute**. Specifies the attribute for the ASQ search. This must be an attribute that stores Distinguished Names, such as the Member Of or Managed By attribute. The search is performed against the objects that are identified by the Distinguished Names found in that attribute. For example, a search within the Member Of attribute of a user account looks for groups in which the user is a member.

The following table lists the search options that are specific to the object type. The search results contain only the objects that match the options you selected.

Table 55: earch activity: Object type

Objects to search for	Search options available
Users	• **Users with Exchange mailbox**. Search for Microsoft Exchange mailbox-enabled users. • **Users with external e-mail addresses**. Search for Microsoft Exchange mail-enabled users. • **Inactive user accounts**. Search for user accounts

Objects to search for	Search options available
	that haven't been used to log on for more than a certain number of days, have the password age of more that a certain number of days, or are expired for more than a certain number of days. • **Expiring user accounts**. Search for user accounts that will expire within a certain number of days.
Contacts	• **Contacts with external e-mail addresses**. Search for Microsoft Exchange mail-enabled contacts.
Groups	• **Mail-enabled groups** .ge Search for Microsoft Exchange mail-enabled groups (distribution lists). • **Security**. Search for security groups. • **Distribution**. Search for distribution groups. • **Domain local**. Search for domain local groups. • **Global**. Search for global groups. • **Universal**. Search for universal groups. • **Dynamic Group**. Search for groups that are configured as Dynamic Groups in Active Roles. • **Group Family**. Search for groups that store Group Family configurations for Active Roles (Group Family configuration storage groups). • **Controlled by Group Family**. Search for groups controlled by Group Family rules in Active Roles. • **Empty**. Search for groups that have no members. • **Deprovisioned** Search for groups that are deprovisioned by using Active Roles.
Computers	• **Computer role**. Search for computers in a certain role. You can restrict the search to workstations and servers or to domain controllers. • **Inactive computer accounts**. Search for computer accounts that haven't been used to log on for more than a certain number of days, have the password age of more that a certain number of days, or are expired for more than a certain number of days.
Printers	• **Printer features**. Search for printers with particular features, such as the printer model, paper size, print resolution, print speed, and other capabilities including the ability to print double-sided, the ability to print multiple colors, and the ability to staple.

Objects to search for	Search options available
Exchange Recipients	• **Users with Exchange mailbox**. Search for Microsoft Exchange mailbox-enabled users. • **Users with external e-mail addresses**. Search for Microsoft Exchange mail-enabled users. • **Mail-enabled groups** . Search for Microsoft Exchange mail-enabled groups (distribution lists). • **Contacts with external e-mail addresses**. Search for Microsoft Exchange mail-enabled contacts. • **Mail-enabled Public Folders**. Search for Microsoft Exchange mail-enabled public folders. • **Query-based Distribution Groups**. Search for Microsoft Exchange query-based distribution groups. • **Room mailboxes**. Search for user accounts representing Microsoft Exchange room mailboxes. • **Equipment mailboxes**. Search for user accounts representing Microsoft Exchange equipment mailboxes. • **Linked mailboxes**. Search for user accounts representing Microsoft Exchange linked mailboxes. • **Shared mailboxes**. Search for user accounts representing Microsoft Exchange shared mailboxes. • **Mailboxes on this server**. Search for user accounts representing Microsoft Exchange mailboxes hosted on a certain Mailbox server. You can select the desired Mailbox server. • **Mailboxes in this mailbox store or database**. Search for user accounts representing Microsoft Exchange mailboxes held in a certain mailbox store or database. You can select the desired mailbox store or database.
Inactive Accounts	• **Account type**. Search for user accounts only, computer accounts only, or both user and computer accounts. • **Criteria of inactivity**. Search for accounts that haven't logged on in the past number of days, accounts whose password has not changed in the past number of days, or accounts that expired more than a certain number of days before the current date.

Search for inactive accounts

If you choose the **Search in the Organizational Unit or container** option, then you can configure the activity to search for inactive user or computer accounts. The **Inactive Accounts** object type provides for the following search options:

- **Account type to search for**. You can choose to search for user accounts only, search for computer accounts only, or search for both user and computer accounts.
- **Search for accounts that haven't logged on in the past number of days**. This option allows you to specify the period, in days, that an account is not used to log on, after which the account is considered inactive. The search retrieves a given account if no successful logons to that account have occurred for more days than specified by this option.

  The search activity uses the `lastLogonTimeStamp` attribute to determine the last time that a given user or computer successfully logged on. Active Directory updates that attribute only periodically, rather than every time that a user or computer logs on. Normally, the period of update is 14 days. This means that the `lastLogonTimeStamp` value could be off by as much as 14 days, so the true last logon time is later than `lastLogonTimeStamp`. Hence, it is advisable to choose the logon inactivity period of more than 14 days.
- **Search for accounts whose password has not changed in the past number of days**. This option allows you to specify the password age, in days, after which an account is considered inactive. The search retrieves a given account if the password of the account remains unchanged for more days than specified by this option.
- **Search for accounts that expired more than a certain number of days before the current date**. This option allows you to specify the number of days after which an expired account is considered inactive. The search retrieves a given account if the account remains in the expired state for more days than specified by this option.

The option to search for inactive accounts is also available when you configure the activity to search for the **Users** or **Computers** object type. You can restrict the search to inactive accounts by choosing the appropriate options to determine what accounts are considered inactive. These options are the same as with the **Inactive Accounts** object type.

Search filter

The search filter option allows you to refine your search in order to locate directory objects based on the properties (attributes) of the objects. For example, you may want to find all the team members in a certain department that report to the manager named John Smith or you may be interested in computer accounts that were not used for a certain time period. In either case, you can use a search filter to look for specific values in the object properties, thereby ensuring that the search results contain only the objects with the desired properties.

A search filter is composed of conditions combined using And or Or logic. Each condition is a certain statement that specifies the criteria the activity should use to determine whether

a given object is to be included in the search results. To create a filter, you need to add at least one condition, but you are not limited in the number of conditions you can add. By using multiple conditions, you can create very complex filters. You can add, delete, and group filter conditions using different operators. You can even nest condition groups within other condition groups to achieve the results that you want. When the activity runs, the filter is evaluated to determine if the objects found by the search meet the criteria you specified in the filter. If a given object meets the criteria, the object is added to the search results; otherwise, the object is filtered out. If you don't create a filter, then all objects found by the search are included in the search results.

A filter condition is composed of three parts: the name of a certain property, the comparison operator, and the value to compare the property with (comparison value). Some operators do not require a comparison value. When creating a condition, you first choose a certain property. Then, you select the desired comparison operator and, if necessary, specify the comparison value you want. The list from which to select a comparison operator depends on the type of the property you are creating the condition for. Whether you have to specify a comparison value depends on the comparison operator. The following tables summarize the comparison operators and comparison values that are available.

The comparison operators from which you can choose when configuring a filter condition are as follows.

Table 56: Comparision operators

Comparison operator	Indicates that
equals	The property value of the object matches the comparison value.
does not equal	The property value of the object does not match the comparison value.
greater or equal	The property value of the object is greater than or equal to the comparison value.
less or equal	The property value of the object is less than or equal to the comparison value.
contains	The property value of the object contains the text specified by the comparison value.
does not contain	The property value of the object does not contain the text specified by the comparison value.
starts with	The text specified by the comparison value occurs at the beginning of the object's property value.
does not start with	The text specified by the comparison value does not occur at the beginning of the object's property value.
ends with	The text specified by the comparison value occurs at the end of the object's property value.

Comparison operator	Indicates that
does not end with	The text specified by the comparison value does not occur at the end of the object's property value.
is empty	The property is not specified on the object.
is not empty	The property of the object has a non-null value.
bitwise and	Each bit of the object's property value matches the corresponding bit of the comparison value.
bitwise or	Any bit of the object's property value matches the corresponding bit of the comparison value.
matches regular expression	The object's property value matches a certain regular expression. This requires the comparison value to be a text string representing the desired regular expression.

The comparison values from which you can choose when configuring a filter condition are as follows.

Table 57: Comparison values

Comparison value	Description
Text string	A literal string of characters. You can type the desired string when you configure a filter condition.
Property of workflow target object	The value of a certain property of the target object of the request that started the workflow. You can select the desired property when you configure a filter condition. Normally, this should be a string-value property.
Property of workflow initiator	The value of a certain property of the user whose request started the workflow. You can select the desired property when you configure a filter condition. Normally, this should be a string-value property.
Changed value of workflow target object property	The value that is requested to be assigned to a certain property of the target object of the request that started the workflow, which represents the requested change to the property of the target object. You can select the desired property when you configure a filter condition. Normally, this should be a string-value property.
Property of object from workflow data context	The value of a certain property of the object that will be selected by the Search activity on the basis of the data found in the workflow environment at the time of executing the workflow. When you configure a filter condition in a Search activity, you can choose the desired property and specify which object you want the activity to select upon evaluating the condition at workflow run time.

Comparison value	Description
Value generated by rule expression	The string value of a certain rule expression. By using a rule expression you can compose a string value based on properties of various objects found in the workflow environment at the time of executing the workflow.
Fixed object in directory	A certain object, such as a user, group, or computer. You can select the desired object in Active Directory when you configure a filter condition. This comparison value is applicable to filter conditions for DN-value properties.
Object from workflow data context	The object that will be selected by the Search activity on the basis of the data found in the workflow environment at the time of executing the workflow. When you configure a filter condition in a Search activity, you can specify which object you want the activity to select upon evaluating the condition at workflow run time. This comparison value is applicable to filter conditions for DN-value properties.
Object identified by DN-value rule expression	The object whose Distinguished Name (DN) is specified by the string value of a certain rule expression. By using a rule expression, you can compose a string value based on properties of various objects found in the workflow environment at the time of executing the workflow. You can create the desired rule expression when you configure a filter condition. This comparison value is applicable to filter conditions for DN-value properties.
Object identified by workflow parameter	The object specified by the value of a certain parameter. You can choose the desired parameter when you configure a filter condition. This comparison value is applicable to filter conditions for DN-value properties.
Workflow initiator object	The user account of the user whose request started the workflow. This comparison value is applicable to filter conditions for DN-value properties.
Workflow target object	The target object of the request that started the workflow. This comparison value is applicable to filter conditions for DN-value properties.
Fixed date and time	A literal date and time value. You can choose the desired date and time when you configure a filter condition. This comparison value is applicable to filter conditions for Date/Time-value properties.
Workflow date and time	A certain point in time relative to the date and time of the Search activity run. You have the option to specify a date that occurs a particular number of days before or after the Search activity run. This comparison value is applicable to filter

Comparison value	Description
	conditions for Date/Time-value properties.
True	The literal Boolean value of True.
False	The literal Boolean value of False.
Value generated by script	The value returned by a certain script function. You can choose the desired script function when you configure a filter condition. The Search activity will execute that script function upon evaluating the condition at workflow run time.
Workflow parameter value	The value of a certain workflow parameter. You can choose the desired parameter when you configure a filter condition.

Notification

You can configure a Search activity to subscribe recipients to the notifications of the following events:

- **Activity completed successfully**. When configured to notify of this event, the activity causes Active Roles to send a notification e-mail if no significant errors occurred during execution of this activity.
- **Activity encountered an error**. When configured to notify of this event, the activity causes Active Roles to send a notification e-mail if any significant errors occurred during execution of this activity.

The notification settings specify the event to notify of, and notification recipients. When executed by the workflow, the activity prepares a notification message appropriate to the specified event. Active Roles retains the message prepared by the activity, and sends the message to the specified recipients upon occurrence of that event. The notification settings are similar to the notification settings of a Notification activity (see Notification activity earlier in this document).

Error handling

When configuring a Search activity, you can choose whether to suppress errors encountered by that activity. The following option is available: **Continue workflow even if this activity encounters an error**. If this option is not selected (default setting), then an error condition encountered by the activity causes Active Roles to terminate the workflow. If you select this option, the workflow continues regardless of whether or not the Search activity or any activity within the Search activity encounters an error condition.

“Run as” options

By default, the Search activity is executed under the user account specified by the “run as” setting in the workflow options and start conditions. This could be the service account of the Active Roles Administration Service or the account of the user who caused the workflow to start. You can configure the activity to override the default “run as” setting by choosing to run the activity under the service account or the account of the user who caused the workflow to start. The account under which the activity is running determines the access rights of the activity in the directory.

Additional settings

A Search activity has the following additional configuration options:

- **Terminate the search activity if the search returns more than <number> objects**. Use this option to specify the maximum number of objects the activity is allowed to return when performing a search. If you want to receive all the objects that match the search conditions, you can disable this option.
- **Exclude or include request controls from the activity operation request**. Request controls are certain pieces of data in an operation request that can be used to pass additional information to Active Roles on how to process the request. Request controls are optional. If no request controls are added to a request, then Active Roles determines how to process the request based solely on the type of the request. You can configure the activity to add certain controls to its operation requests (include request controls) or to ensure that certain controls never occur in the activity operation requests (exclude request controls). For information about Active Roles request controls, see the Active Roles SDK documentation.

Stop Search activity

You can use a Stop Search activity within a Search activity to stop the search being performed by the Search activity. Basically, a Stop Search activity is intended to be used within an If-Else activity nested into a Search activity, in order to stop the search if certain conditions occur. In this scenario, the If-Else activity analyzes data returned by the search, and executes the If-Else branch containing the Stop Search activity if the data returned by the search meets the conditions of that If-Else branch.

CRUD activities

Active Roles offers a number of workflow activities, collectively referred to as CRUD activities, intended to create new objects, and modify or delete existing objects in Active Directory. The CRUD abbreviation designates the key operations that can be performed by

using these activities: Create, Read, Update, Delete. The following CRUD activities are available in the Active Roles workflow designer:

- "Create" activity Creates an object, such as a user, group, or computer, in Active Directory.
- "Update" activity Changes properties of an object, such as a user, group, or computer, in Active Directory.
- "Add to group" activity Adds an object, such as a user, group, or computer, to specified groups in Active Directory.
- "Remove from group" activity Removes an object, such as a user, group, or computer, from specified groups in Active Directory.
- "Move" activity Moves an object, such as a user, group, or computer, to a specified container in Active Directory.
- "Deprovision" activity Deprovisions a user or group, by applying the Active Roles deprovisioning policy.
- "Undo deprovision" activity Restores a user or group that was deprovisioned by using Active Roles.
- "Delete" activity Deletes an object, such as a user, group, or computer, in Active Directory.

The following topics in this section provide an overview of the configuration settings that are common to CRUD activities:

- Notification Active Roles can notify via e-mail about whether or not the activity has encountered an error condition at run time.
- Error handling Determines whether or not the workflow is allowed to continue if the activity has encountered an error condition at run time.
- "Run as" options Determines the user account under which to run the activity.
- Additional settings Some advanced configuration options that allow you to adjust the processing of the operation requested by the activity.

"Create" activity

"Create" activity is intended to create an object, such as a user, computer, or group in Active Directory. The activity allows you to configure the following characteristics of the object to be created:

- **Container**. You can specify the Organizational Unit (OU) or container in which you want the activity to create an object. The following options are available:
  - **Fixed container in directory**. Create an object in the given OU or container. You can select the desired OU or container in Active Directory when you configure a Create activity.

- **Parent OU of workflow target object**. In case of a change workflow, create an object in the OU that holds the target object of the request that started the workflow.
- **Activity target object**. Create an object in the OU or container created or otherwise processed by another CRUD activity at the time of executing the workflow. You can select the desired CRUD activity from the workflow definition when you configure a Create activity.
- **Object identified by workflow parameter**. Create an object in the OU or container specified by the value of a certain parameter of the workflow. You can choose the desired parameter from the workflow definition when you configure a Create activity.
- **Object from workflow data context**. Create an object in the OU or container that will be selected by the Create activity on the basis of the data found in the workflow environment at the time of executing the workflow. When configuring a Create activity, you can specify which OU or container you want the activity to select at workflow run time.
- **Object identified by DN-value rule expression**. Create an object in the OU or container whose Distinguished Name (DN) is specified by the string value of a certain rule expression. By using a rule expression you can compose a string value based on properties of various objects found in the workflow environment at the time of executing the workflow. You can create the desired rule expression when you configure a Create activity.

- **Object type and name**. You can specify the type and the name of the object to be created by the activity. When you configure a Create activity, you can choose the appropriate object type and define how the activity will generate the object name when creating an object. The following options are available:
  - **Text string**. Use the given string of characters as the name of the object. You can specify the desired string when you configure a Create activity.
  - **Name of workflow target object**. In case of a change workflow, use the name of the target object of the request that started the workflow.
  - **Name of workflow target object, followed by text string**. In case of a change workflow, use a certain text string prefixed with the name of the target object of the request that started the workflow. You can specify the desired text string when you configure a Create activity.
  - **Workflow parameter value**. The name of the object is specified by the string value of a certain parameter of the workflow. You can choose the desired parameter from the workflow definition when you configure a Create activity.
  - **Property of object from workflow data context**. The name of the object is specified by the value of a certain property of the object that will be selected by the Create activity on the basis of the data found in the workflow run-time environment. When you configure a Create activity, you can choose the desired property and specify which object you want the activity to select at workflow run time.

    - **Value generated by rule expression**. The name of the object is identified by the string value of a certain rule expression. By using a rule expression you can compose a string value based on properties of various objects found in the workflow environment at the time of executing the workflow. You can create the desired rule expression when you configure a Create activity.
- **Object properties**. You can define how you want the activity to populate the properties of the new object. When you configure a Create activity, you can choose the properties you want the activity to populate and, for each property, specify the value to be assigned to that property. The following options are available:
    - **Text string**. Use the given string of characters as the value of the property. You can specify the desired string when you configure a Create activity.
    - **Property of workflow target object**. In case of a change workflow, the value of a certain property of the target object of the request that started the workflow. You can select the desired property when you configure a Create activity.
    - **Property of workflow initiator**. Use the value of a certain property of the user whose request started the workflow. You can select the desired property when you configure a Create activity.
    - **Changed value of workflow target object property**. In case of a change workflow, use the value that is requested to be assigned to a certain property of the workflow target object. You can select the desired property when you configure a Create activity.
    - **Workflow parameter value**. Use the value of a certain parameter of the workflow. You can choose the desired parameter from the workflow definition when you configure a Create activity.
    - **Property of object from workflow data context**. Use the value of a certain property of the object that will be selected by the Create activity on the basis of the data found in the workflow environment at the time of executing the workflow. When you configure a Create activity, you can choose the desired property and specify which object you want the activity to select at workflow run time.
    - **Value generated by rule expression**. Use the string value of a certain rule expression. By using a rule expression you can compose a string value based on properties of various objects found in the workflow environment at the time of executing the workflow. You can create the desired rule expression when you configure a Create activity.

"Create" activity also has a number of configuration settings that are common to CRUD activities:

- Notification Active Roles can notify via e-mail about whether or not the activity has encountered an error condition at run time.
- Error handling Determines whether or not the workflow is allowed to continue if the activity has encountered an error condition at run time.
- "Run as" options Determines the user account under which to run the activity.

- Additional settings Some advanced configuration options that allow you to adjust the processing of the operation requested by the activity.

"Update" activity

"Update" activity is intended to make changes to particular properties of a certain object. This activity has the following configuration options:

- **Activity target**. This option lets you specify the object whose properties you want the activity to change. You can select the object when you configure the activity, or you can configure the activity to select the appropriate object at workflow run time. See Activity target for further details.
- **Target properties**. You can define how you want the activity to change the properties of the object. When you configure an Update activity, you can choose the properties you want the activity to change and, for each property, specify the new value to be assigned to that property. For a multi-value property, you can choose to add or remove the value from that property. The following options are available:
  - **Text string**. Use the given string of characters as the value of the property. You can specify the desired string when you configure an Update activity.
  - **Property of workflow target object**. In case of a change workflow, use the value of a certain property of the target object of the request that started the workflow. You can select the desired property when you configure an Update activity.
  - **Property of workflow initiator**. Use the value of a certain property of the user whose request started the workflow. You can select the desired property when you configure an Update activity.
  - **Changed value of workflow target object property**. In case of a change workflow, use the value that is requested to be assigned to a certain property of the workflow target object. You can select the desired property when you configure an Update activity.
  - **Workflow parameter value**. Use the value of a certain parameter of the workflow. You can choose the desired parameter from the workflow definition when you configure an Update activity.
  - **Property of object from workflow data context**. Use the value of a certain property of the object that will be selected by the Update activity on the basis of the data found in the workflow environment at the time of executing the workflow. When you configure an Update activity, you can choose the desired property and specify which object you want the activity to select at workflow run time.
  - **Value generated by rule expression**. Use the string value of a certain rule expression. By using a rule expression you can compose a string value based on properties of various objects found in the workflow environment at the time of executing the workflow. You can create the desired rule expression when you configure an Update activity.

"Update" activity also has a number of configuration settings that are common to CRUD activities:

- Notification Active Roles can notify via e-mail about whether or not the activity has encountered an error condition at run time.
- Error handling. Determines whether or not the workflow is allowed to continue if the activity has encountered an error condition at run time.
- "Run as" options. Determines the user account under which to run the activity.
- Additional settings Some advanced configuration options that allow you to adjust the processing of the operation requested by the activity.

"Add to group" activity

"Add to group" activity is intended to add a certain object, such as a user, computer, or group, to particular groups in Active Directory. This activity has the following configuration options:

- **Activity target**. This option lets you specify the object you want the activity to add to groups. You can select the object when you configure the activity, or you can configure the activity to select the appropriate object at workflow run time. See Activity target for further details.
- **Groups**. This option lets you define the groups to which you want the activity to add the object. When you configure an "Add to group" activity, you can specify a list of groups. For each of the groups in the list, the activity will add the object to that group. To add a group to the list, you can choose from the following options:
    - **Fixed group in directory**. You can select the desired group in Active Directory when you configure an "Add to group" activity. A unique identifier of the group is saved in the configuration of the activity. The activity will use that identifier to select the group when calculating the list of groups at workflow execution time.
    - **Object from workflow data context**. The group will be selected by the activity on the basis of the data found in the workflow environment at the time of executing the workflow. When configuring an "Add to group" activity, you can specify which group you want the activity to select at workflow execution time.
    - **Object identified by DN-value rule expression**. The Distinguished Name (DN) of the group is specified by the string value of a certain rule expression. By using a rule expression you can compose a string value based on properties of various objects found in the workflow environment at the time of executing the workflow. You can create the desired rule expression when you configure an "Add to group" activity.

"Add to group" activity also has a number of configuration settings that are common to CRUD activities:

- Notification Active Roles can notify via e-mail about whether or not the activity has encountered an error condition at run time.
- Error handling. Determines whether or not the workflow is allowed to continue if the activity has encountered an error condition at run time.
- "Run as" options Determines the user account under which to run the activity.
- Additional settings Some advanced configuration options that allow you to adjust the processing of the operation requested by the activity.

"Remove from group" activity

"Remove from group" activity is intended to remove a certain object, such as a user, computer or group, from particular groups in Active Directory. This activity has the following configuration options:

- **Activity target**. This option lets you specify the object you want the activity to remove from groups. You can select the object when you configure the activity, or you can configure the activity to select the appropriate object at workflow run time. See Activity target for further details.
- **Groups - Remove the object from all groups**. This options configures the activity to remove the object from all groups in Active Directory. Note that an object cannot be removed from the object's primary group, so the activity will remove the object from all groups except the object's primary group.
- **Groups - Remove the object from these groups**. This option lets you list the groups from which you want the activity to remove the object. You can specify a list of groups when you configure a "Remove from group" activity. For each of the groups in the list (with the exception of the object's primary group), the activity will remove the object from that group. To add a group to the list, you can choose from the following options:
    - **Fixed group in directory**. You can select the desired group in Active Directory when you configure a "Remove from group" activity. A unique identifier of the group is saved in the configuration of the activity. The activity will use that identifier to select the group when calculating the list of groups at workflow execution time.
    - **Object from workflow data context**. The group will be selected by the activity on the basis of the data found in the workflow environment at the time of executing the workflow. When configuring a "Remove from group" activity, you can specify which group you want the activity to select at workflow execution time.
    - **Object identified by DN-value rule expression**. The Distinguished Name (DN) of the group is specified by the string value of a certain rule expression. By using a rule expression you can compose a string value based on properties of various objects found in the workflow environment at the time of executing the workflow. You can create the desired rule expression when you configure a "Remove from group" activity.

"Remove from group" activity also has a number of configuration settings that are common to CRUD activities:

- Notification Active Roles can notify via e-mail about whether or not the activity has encountered an error condition at run time.
- Error handling Determines whether or not the workflow is allowed to continue if the activity has encountered an error condition at run time.
- "Run as" options Determines the user account under which to run the activity.
- Additional settings Some advanced configuration options that allow you to adjust the processing of the operation requested by the activity.

"Move" activity

"Move" activity is intended to move a certain object to a particular container in Active Directory. The activity has the following configuration options:

- **Activity target**. This option lets you specify the object you want the activity to move. You can select the object when you configure the activity, or you can configure the activity to select the appropriate object at workflow run time. See Activity target for further details.
- **Destination container**. You can specify the Organizational Unit (OU) or container to which you want the activity to move the object. The following options are available:
    - **Fixed container in directory**. Move the object to the given OU or container. You can select the desired OU or container in Active Directory when you configure a Move activity.
    - **Parent OU of workflow target object**. In case of a change workflow, move the object to the OU that holds the target object of the request that started the workflow.
    - **Activity target object**. Move the object to the OU or container created or otherwise processed by another CRUD activity at the time of executing the workflow. You can select the desired CRUD activity from the workflow definition when you configure a Move activity.
    - **Object identified by workflow parameter**. Move the object to the OU or container specified by the value of a certain parameter of the workflow. You can choose the desired parameter from the workflow definition when you configure a Move activity.
    - **Object from workflow data context**. Move the object to the OU or container that will be selected by the Move activity on the basis of the data found in the workflow environment at the time of executing the workflow. When configuring a Move activity, you can specify which OU or container you want the activity to select at workflow run time.
    - **Object identified by DN-value rule expression**. Move the object to the OU or container whose Distinguished Name (DN) is specified by the string value of a certain rule expression. By using a rule expression you can compose

a string value based on properties of various objects found in the workflow environment at the time of executing the workflow. You can create the desired rule expression when you configure a Move activity.

"Move" activity also has a number of configuration settings that are common to CRUD activities:

- Notification. Active Roles can notify via e-mail about whether or not the activity has encountered an error condition at run time.
- Error handling. Determines whether or not the workflow is allowed to continue if the activity has encountered an error condition at run time.
- "Run as" options Determines the user account under which to run the activity.
- Additional settings Some advanced configuration options that allow you to adjust the processing of the operation requested by the activity.

"Deprovision" activity

"Deprovision" activity is intended to apply the Active Roles deprovisioning policies to a particular user or group. This activity causes Active Roles to perform all the tasks prescribed by the deprovisioning policies, thereby deprovisioning the user or group.

The activity allows you to specify the user or group object you want the activity to deprovision. You can select the object when you configure the activity, or you can configure the activity to select the appropriate object at workflow run time. See Activity target for further details.

"Deprovision" activity also has a number of configuration settings that are common to CRUD activities:

- Notification Active Roles can notify via e-mail about whether or not the activity has encountered an error condition at run time.
- Error handling Determines whether or not the workflow is allowed to continue if the activity has encountered an error condition at run time.
- "Run as" options Determines the user account under which to run the activity.
- Additional settings Some advanced configuration options that allow you to adjust the processing of the operation requested by the activity.

"Undo deprovision" activity

"Undo deprovision" activity is intended to restore a particular user or group that was deprovisioned by using Active Roles. The activity causes Active Roles to roll back the changes that were made to the user or group object by applying the Active Roles deprovisioning policies. As a result, the object reverts to the state it was in before the deprovisioning-related changes where made.

The activity allows you to specify the user or group object you want the activity to restore. You can select the object when you configure the activity, or you can configure the activity

to select the appropriate object at workflow run time. See Activity target for further details.

"Undo deprovision" activity also has a number of configuration settings that are common to CRUD activities:

- Notification. Active Roles can notify via email about whether or not the activity has encountered an error condition at run time.
- Error handling Determines whether or not the workflow is allowed to continue if the activity has encountered an error condition at run time.
- "Run as" options Determines the user account under which to run the activity.
- Additional settings Some advanced configuration options that allow you to adjust the processing of the operation requested by the activity.

"Delete" activity

"Delete" activity is intended to delete a particular object in Active Directory. The activity allows you to specify the object you want the activity to delete. You can select the object when you configure the activity, or you can configure the activity to select the appropriate object at workflow run time. See Activity target for further details.

"Delete" activity also has a number of configuration settings that are common to CRUD activities:

- Notification Active Roles can notify via e-mail about whether or not the activity has encountered an error condition at run time.
- Error handling Determines whether or not the workflow is allowed to continue if the activity has encountered an error condition at run time.
- "Run as" options Determines the user account under which to run the activity.
- Additional settings Some advanced configuration options that allow you to adjust the processing of the operation requested by the activity.

Activity target

The execution of a CRUD activity results in a request to perform a certain operation on a certain object. For example, an "Update" activity requests Active Roles to make changes to the properties of a certain object, an "Add to group" activity requests Active Roles to add a certain object to particular groups, and so forth. The object on which the operation is requested by a CRUD activity is referred to as the target object of that activity, or simply *activity target*.

When you configure a CRUD activity, you can use the following options to specify the activity target for that activity:

- **Fixed object in directory**. The activity target is the given object. You can select the desired object in Active Directory when you configure a CRUD activity.

- **Object identified by workflow parameter**. The activity target is the object specified by the value of a certain parameter of the workflow. You can choose the desired parameter from the workflow definition when you configure a CRUD activity.
- **Object from workflow data context**. The activity target will be selected by the activity on the basis of the data found in the workflow environment at the time of executing the workflow. When configuring a CRUD activity, you can specify which object you want the activity to select at workflow run time.
- **Object identified by DN-value rule expression**. The Distinguished Name (DN) of the activity target is specified by the string value of a certain rule expression. By using a rule expression, you can compose a string value based on properties of various objects found in the workflow environment at the time of executing the workflow. You can create the desired rule expression when you configure a CRUD activity.

The following table helps distinguish CRUD activity targets.

Table 58: CRUD activity targets

Activity	Activity target
Update	The object whose properties are to be changed. An Update activity requests Active Roles to change certain properties of a particular object. That object is referred to as the activity target of the Update activity.
Add to group	The object to be added to the groups. An "Add to group" activity requests Active Roles to add a certain object to particular groups. That object is referred to as the activity target of the "Add to group" activity.
Remove from group	The object to be removed from the groups. A "Remove from group" activity requests Active Roles to remove a certain object from particular groups. That object is referred to as the activity target of the "Remove from group" activity.
Move	The object to be moved. A Move activity requests Active Roles to move a certain object to a particular container in Active Directory. That object is referred to as the activity target of the Move activity.
Deprovision	The object to be deprovisioned. A Deprovision activity requests Active Roles to deprovision a certain object. That object is referred to as the activity target of the Deprovision activity.
Undo deprovision	The object to be restored. An "Undo deprovision" activity requests Active Roles to restore a certain object that was deprovisioned. That object is referred to as the activity target of the "Undo deprovision" activity.
Delete	The object to be deleted. A Delete activity requests Active Roles to delete a certain object. That object is referred to as the activity target of the Delete activity.

Notification

You can configure a CRUD activity to subscribe recipients to the notifications of the following events:

- **Activity completed successfully**. When configured to notify of this event, the activity causes Active Roles to send a notification e-mail if no significant errors occurred during execution of this activity.
- **Activity encountered an error**. When configured to notify of this event, the activity causes Active Roles to send a notification e-mail if any significant errors occurred during execution of this activity.

The notification settings specify the event to notify of, and notification recipients. When executed by the workflow, the activity prepares a notification message appropriate to the specified event. Active Roles retains the message prepared by the activity, and sends the message to the specified recipients upon occurrence of that event. The notification settings are similar to the notification settings of a Notification activity (see Notification activity earlier in this document).

Error handling

When configuring a CRUD activity, you can choose whether to suppress errors encountered by that activity. The following option is available: **Continue workflow even if this activity encounters an error**. If this option is not selected (default setting), an error encountered by the activity causes Active Roles to terminate the workflow.

If you configure a CRUD activity so that the workflow is allowed to continue in case of an error encountered by that activity, then you can have the workflow take an appropriate compensation action. This could be accomplished by using an If-Else activity with a branch condition that evaluates the "Encountered an error" execution status of the CRUD activity. Add an If-Else activity following the CRUD activity and configure a condition on an If-Else branch to detect the "Encountered an error" execution status of that CRUD activity. Then, configure that If-Else branch to contain the activities you want to perform the compensation action. As a result, once the CRUD activity has encountered an error, the "Encountered an error" branch condition evaluates to TRUE, causing the workflow to execute the activities intended to perform the compensation action.

"Run as" options

By default, any CRUD activity is executed under the user account specified by the "run as" setting in the workflow options and start conditions. This could be the service account of the Active Roles Administration Service or the account of the user who caused the workflow to start. You can configure the activity to override the default "run as" setting by choosing to run the activity under the service account or the account of the user who caused the workflow to start. The account under which the activity is running determines the access rights of the activity in the directory.

One more option determines whether to apply approval rules to the operation requested by the activity if the activity is executed under a privileged account, such as the Active Roles service account, an Active Roles Admin account, or the account of the user who is designated as an approver. By default, the activity uses the option setting specified in the workflow options and start conditions. However, the workflow-wide option setting can be overridden on a per-activity basis.

When you configure a CRUD activity, you can enable or disable the **Enforce approval** option for that activity. When enabled, this option causes the approval rules to be applied, submitting the operation for approval regardless of the account under which the activity is executed. Otherwise, the operation requested by the activity bypasses approval rules if the activity is executed under the Active Roles service account, an Active Roles Admin account, or the account of the user who is designated as an approver, so the operation is not submitted for approval.

Additional settings

A CRUD activity has the following additional configuration options:

- **Use this text instead of the original operation reason text**. If the operation requested by the CRUD activity is subject to approval, you can specify the operation reason text to be shown to the approver instead of the reason text specified in the operation request that started the workflow. The **Use only if the operation reason is not originally specified** sub-option configures the activity to replace the reason text only if the operation request that started the workflow does not have any reason text specified.
- **Allow the request created by this activity to start a new instance of the workflow containing this activity**. This option is normally disabled to prevent recurrent execution of the CRUD activity in the situation where the operation requested by that activity within a given workflow matches the start conditions of that same workflow. Enabling this option could result in a loop of workflow instances executing the same activity again and again, and eventually would cause an overflow condition.
- **Exclude or include request controls from the activity operation request**. Request controls are certain pieces of data in an operation request that can be used to pass additional information to Active Roles on how to process the request. Request controls are optional. If no request controls are added to a request, then Active Roles determines how to process the request based solely on the type of the request. You can configure the activity to add certain controls to its operation requests (include request controls) or to ensure that certain controls never occur in the activity operation requests (exclude request controls). For information about Active Roles request controls, see the Active Roles SDK documentation.

Save Object Properties activity

Save Object Properties activity is intended to save properties of a particular object at workflow execution time. The properties are saved in the workflow data context, and can be retrieved by other activities before or after the object has changed. This capability is instrumental in situations that require knowing not only the changed object state or properties but also the previous or old values of certain properties. Old values may be required to determine the previous state of an object in order to make some decision or perform a certain action based on those values. For example, to notify of object deletions, you can create a workflow that starts when deletion of an object is requested, saves the object's name, and then, after the object is deleted, sends a notification message that includes the saved name of the deleted object.

This activity has the following configuration options:

- **Activity target**. This option lets you specify the object whose properties you want the activity to save. You can choose to specify:
  - **Workflow target object**. In a change workflow, the target object of the request that started the workflow. For example, in a workflow that starts upon a deletion request, this choice causes the activity to save the properties of the object whose deletion is requested.
  - **Fixed object in directory**. A particular object you select from Active Directory.
  - **Object identified by workflow parameter**. The object specified by the value of a certain parameter of the workflow. You can choose the desired parameter from the workflow definition.
  - **Object from workflow data context**. The object will be selected by the activity on the basis of the data found in the workflow environment at the time of executing the workflow. You can specify which object you want the activity to select at workflow execution time.
  - **Object identified by DN-value rule expression**. The object whose Distinguished Name (DN) is specified by the string value of a certain rule expression. By using a rule expression, you can compose a string value based on properties of various objects found in the workflow environment at the time of executing the workflow. You can create the desired rule expression when you configure the activity.
- **Target properties**. This option lets you specify the object properties you want the activity to save. The workflow designer proposes the default list of properties, and allows you to change the list as needed. By default, the activity saves all single-value non-constructed attributes found in the directory schema for the target object, including custom virtual attributes added to the directory schema by Active Roles.
- **Notification**. You can configure the activity to subscribe recipients to the notifications of the following events:
  - **Activity completed successfully**. When configured to notify of this event, the activity causes Active Roles to send a notification e-mail if no significant errors occurred during execution of this activity.

- **Activity encountered an error**. When configured to notify of this event, the activity causes Active Roles to send a notification e-mail if any significant errors occurred during execution of this activity.

The notification settings specify the event to notify of, and notification recipients. When executed by the workflow, the activity prepares a notification message appropriate to the specified event. Active Roles retains the message prepared by the activity, and sends the message to the specified recipients upon occurrence of that event. The notification settings are similar to the notification settings of a Notification activity (see Notification activity earlier in this document).

- **Error handling**. You can choose whether to suppress errors encountered by the activity. The following option is available: **Continue workflow even if this activity encounters an error**. If this option is not selected (default setting), then an error condition encountered by the activity causes Active Roles to terminate the workflow. If you select this option, the workflow continues regardless of whether or not the encounters an error condition.

Retrieving saved properties

In a workflow that includes an activity of the Save Object Properties type, you can configure other activities to retrieve object properties saved by that activity:

- By using the following expression in a Script activity:
  `$workflow.SavedObjectProperties("activityName").get("attributeName")`

  In this expression, `activityName` stands for the name of the Save Object Properties activity and `attributeName` is the LDAP display name of the attribute representing the property you want the script to retrieve. You should specify an attribute listed in the **Target properties** setting of the Save Object Properties activity; otherwise, this expression returns no property value at workflow execution time.

- By adding the **Workflow - Saved Object Properties** token to the notification message template (see Events, recipients and messages).

- To add the token:

  1. In the **Insert Token** dialog box, click **Workflow - Saved Object Properties** in the list of tokens, and then click **OK**.
  2. In the dialog box that appears, select the name of the Save Object Properties activity and the saved property you want the token to retrieve.

     You should select a property listed in the **Target properties** setting of the Save Object Properties activity; otherwise, the token you have configured returns no property value at workflow execution time.

- By choosing the **Property of object from workflow data context** configuration option, available in If-Else branch conditions, Search filter, "Create" activity, "Update" activity, and Add Report Section activity (see also Configuring an Add Report Section activity).

- If you choose this option, then you need to perform the following configuration steps:

  In the **Object Property** dialog box, click the link in the **Target object** field, and then click **More choices**.

  1. In the dialog box that appears, click **Saved Object Properties** in the left pane, select the name of the Save Object Properties activity from the **Activity** list, and then click **OK**.
  2. In the **Object Property** dialog box, click the link in the **Target property** field, and select the property you want.

     You should select a property listed in the **Target properties** setting of the Save Object Properties activity; otherwise, the entry you have configured returns no property value at workflow execution time.

Modify Requested Changes activity

Modify Requested Change activity is intended to update the change request that started the workflow, allowing you to add or remove changes to the properties of the workflow target object at workflow execution time. For example, in a workflow that starts when the creation of an object is requested, you can use this activity to modify the properties that are going to be assigned to the new object, or change the container in which to create the object. In a workflow that starts upon a request to change an object, you can use this activity to modify the requested changes to the properties of that object.

This activity has the following configuration options:

- **Target changes**. You can define the property changes to add or remove from the change request. When you configure this activity, you can choose the properties you want the activity to change and, for each property, choose to remove the property from the request, clear the property value in the request, or specify the new value to be assigned to that property. For a multi-value property, you can choose to add or remove a value from that property. The following options are available:
  - **Text string**. Use the given string of characters as the value of the property. You can type the desired string.
  - **Property of workflow target object**. Use the value of a certain property of the target object of the request that started the workflow. You can select the desired property from a list of object properties.
  - **Property of workflow initiator**. Use the value of a certain property of the user whose request started the workflow. You can select the desired property from a list of object properties.
  - **Changed value of workflow target object property**. Use the value that is requested to be assigned to a certain property of the workflow target object. You can select the desired property from a list of object properties.
  - **Workflow parameter value**. Use the value of a certain parameter of the workflow. You can choose the desired parameter from a list of the workflow parameters.

- **Property of object from workflow data context**. Use the value of a certain property of the object that will be selected by the activity on the basis of the data found in the workflow run-time environment. You can choose the desired property and specify which object you want the activity to select at workflow run time.
- **Value generated by rule expression**. Use the string value of a certain rule expression. By using a rule expression, you can compose a string value based on properties of various objects found in the workflow run-time environment. You can create the desired rule expression when you configure the activity.

- **Notification**. You can configure the activity to subscribe recipients to the notifications of the following events:
  - **Activity completed successfully**. When configured to notify of this event, the activity causes Active Roles to send a notification e-mail if no significant errors occurred during execution of this activity.
  - **Activity encountered an error**. When configured to notify of this event, the activity causes Active Roles to send a notification e-mail if any significant errors occurred during execution of this activity.

  The notification settings specify the event to notify of, and notification recipients. When executed by the workflow, the activity prepares a notification message appropriate to the specified event. Active Roles retains the message prepared by the activity, and sends the message to the specified recipients upon occurrence of that event. The notification settings are similar to the notification settings of a Notification activity (see Notification activity earlier in this document).

- **Error handling**. You can choose whether to suppress errors encountered by the activity. The following option is available: **Continue workflow even if this activity encounters an error**. If this option is not selected (default setting), then an error condition encountered by the activity causes Active Roles to terminate the workflow. If you select this option, the workflow continues regardless of whether or not the encounters an error condition.
- **Additional settings**. You can configure the activity to:
  - Change the container where to create new objects while ensuring that the policies and workflows are applied from the container where the object will actually be created rather than from the container that was originally specified in the object creation request.
  - Add or remove Active Roles controls from the request.

  Controls are certain pieces of data that can be used to provide additional information to Active Roles on how to process the request. If no controls are added to a request, then Active Roles determines how to process the request based solely on the type of the request. You can configure the activity to add certain controls to the request (include controls) or to ensure that certain controls never occur in the request (exclude controls). For information about Active Roles controls, see Active Roles SDK.

NOTE: The Modify Requested Changes activity type is unavailable in case of an automation workflow. You can add activities of this type to a change workflow only.

Configuring a workflow

Workflows provide a powerful and convenient way to add new logic to directory data management and provisioning processes in Active Roles. To configure a workflow, you create a workflow definition and then use the Workflow Designer to add and configure workflow activities.

This section covers the following tasks:

- Creating a workflow definition
- Configuring workflow start conditions
- Configuring workflow parameters
- Adding activities to a workflow
- Configuring an Approval activity
- Configuring a Notification activity
- Configuring a Script activity
- Configuring an If-Else activity
- Configuring a Stop/Break activity
- Configuring an Add Report Section activity
- Configuring a Search activity
- Configuring CRUD activities
- Configuring a Save Object Properties activity
- Configuring a Modify Requested Changes activity
- Enabling or disabling an activity
- Enabling or disabling a workflow
- Using the initialization script

Creating a workflow definition

The Active Roles console provides the Workflow Designer for creating and configuring workflows. First, you create a workflow definition. Then, you use the Workflow Designer to construct a workflow, saving the workflow configuration data in the workflow definition.

To create a workflow definition

1. In the Active Roles console tree, expand **Configuration | Policies**, right-click **Workflow**, and select **New | Workflow**.
2. Follow the steps in the wizard for creating the workflow definition.
3. On the **Workflow Type** page, accept the default setting.

By default, the wizard creates a change workflow that starts upon a request to change data in the directory. Another option is to create an automation workflow that can be run on a scheduled basis or on user demand. See Automation workflow for further details.

Once you have created a workflow definition, you can open it in the Workflow Designer to add workflow activities and specify workflow start conditions.

You can create containers to store related workflows and other containers. To create a workflow container, right-click **Workflow** in the console tree and select **New | Container**. To create a workflow definition in a given container, right-click the container in the console tree, and select **New | Workflow**.

You can delete a workflow definition as follows: In the console tree under **Configuration | Policies | Workflow**, right-click the object representing the workflow definition, and click **Delete**.

Configuring workflow start conditions

The workflow start conditions determine which operations cause the workflow to start. For example, an approval workflow can be configured so that any request to create a user in a specific container starts the workflow, thereby requiring approval for the request. You can specify the start conditions for a workflow by editing its definition in the Workflow Designer.

To view or change the start conditions for a workflow

1. In the Active Roles console tree, expand **Configuration | Policies | Workflow**, and select the workflow you want to configure.

   This opens the Workflow Designer window in the details pane, representing the workflow definition as a process diagram.

2. In the details pane, click the **Workflow options and start conditions** button to expand the area above the process diagram, and then click the **Configure** button.
3. Click the **Conditions** tab in the **Change Workflow Options and Start Conditions** dialog box.

This displays a page where you can view or change:

- Operation conditions
- Initiator conditions
- Filtering conditions
- "Run as" options

Operation conditions

The operation conditions specify:

- An object type, such as User, Group or Computer; the workflow starts only if an operation requests changes to an object of that type.
- An operation type, such as Create, Rename, Modify or Delete; the workflow starts only if an operation of that type is requested.
- For the Modify operation type, a list of object properties; the workflow starts only if an operation requests changes to any of those properties of an object.

To view or change the operation conditions

1. In the **Change Workflow Options and Start Conditions** dialog box, go to the **Conditions** tab, and click **Select operation** in the **Operation Conditions** area.

   This opens the page where you can view or change the object type and operation type settings.

2. To change the object type settings, select a type of object from the drop-down list.

   To select an object type that is not included in the drop-down list, click the button next to the drop-down list.

3. To change the operation type setting, click the appropriate option.
4. If the Modify operation type (the **Modify properties** option) is selected, click **Next** to view or change the selection of properties.
5. Click **Finish**.

Initiator conditions

The initiator conditions specify:

- The identity of an operation requestor (initiator), such as a user or group; the workflow starts only if an operation is requested by that identity.
- A container, such as an organizational unit or Managed Unit; the workflow starts only if an operation requests changes to, or creation of, an object in that container.

To view or change the initiator conditions

1. In the **Change Workflow Options and Start Conditions** dialog box, go to the **Conditions** tab, and observe the list in the **Initiator Conditions** area.

   Each entry in the list represents a single initiator condition, with the first field identifying the operation requestor and the second field identifying the container. If the list is missing, no initiator conditions are defined.

2. To define an initiator condition:
   a. Click **Add** in the **Initiator Conditions** area.
   b. Populate the list of operation requestors.
   c. Select the container.
3. To delete an initiator condition, select the corresponding entry from the **Initiator Conditions** list, and click **Remove**.

If multiple initiator conditions are defined, the workflow starts if any one of them is fulfilled.

If multiple operation requestors are defined within a single initiator condition, the condition is considered fulfilled if the operation is requested by any one of those identities.

Filtering conditions

A filter can be used to define any additional conditions on objects involved in an operation. The workflow starts only if the operation satisfies those conditions. If no filter is set, then no additional conditions are in effect.

When you configure a filter, you need to add at least one condition, but you are not limited in the number of conditions that you can add. You can add, delete, and group conditions using various operators. It is possible to nest condition groups within other condition groups to achieve the results that you want.

A condition group contains one or more conditions connected by the same logical operator. By grouping conditions, you specify that those conditions should be evaluated as a single unit. The effect is the same as if you put parentheses around an expression in a mathematical equation or logic statement.

By default, a single condition group is created when you add a condition. You can create additional condition groups to group a set of conditions and nest grouped conditions within other condition groups.

In a condition group, conditions are connected using the AND, OR, NOT AND, or NOT OR logical operator:

- AND group evaluates to TRUE if all conditions in the group are TRUE.
- OR group evaluates to TRUE if any condition in the group is TRUE.
- NOT AND group evaluates to TRUE if any condition in the group evaluates to FALSE.
- NOT OR group evaluates to TRUE if all conditions in the group evaluate to FALSE.

By default, AND is the logical operator between the conditions in a condition group. It is possible to change the logical operator by converting the condition group to a different group type.

When you add a condition, the workflow designer first prompts you to specify what you want the condition to evaluate. The following options are available:

- **Property of workflow target object**. This option is intended to evaluate the value of a certain property of the target object of the request that started the workflow. You can select the desired property when you configure a condition.
- **Property of workflow initiator**. This option is intended to evaluate the value of a certain property of the user whose request started the workflow. You can select the desired property when you configure a condition.
- **Changed value of workflow target object property**. This option is intended to evaluate the value that is requested to be assigned to a certain property of the workflow target object, which represents the requested change to the property of the

target object of the request that started the workflow. You can select the desired property when you configure a condition.

- **Workflow parameter value**. This option is intended to evaluate the value of a certain parameter of the workflow. You can select the desired parameter from the workflow definition when you configure a condition.
- **Property of object from workflow data context**. This option is intended to evaluate the value of a certain property of the object that will be selected on the basis of the data found in the workflow environment at the time of evaluating the workflow start conditions. When you configure a condition, you can choose the desired property and specify which object you want the workflow engine to select upon evaluating the condition at workflow start time.
- **Value generated by rule expression**. This option is intended to evaluate the string value of a certain rule expression. By using a rule expression, you can compose a string value based on properties of various objects found in the workflow environment at the time of evaluating the workflow start conditions. The workflow engine calculates the value of your rule expression upon evaluating the condition at workflow start time.

Once you have specified the entity or field that you want the condition to evaluate, you can choose a comparison operator and specify a comparison value. The comparison operator determines the operation of comparing the entity or field to evaluate with the comparison value you specified, and causes the condition to evaluate to TRUE or FALSE depending on the outcome of that operation.

You can choose from the following options to specify a comparison value:

- **Text string**. Performs comparison with a literal string of characters. You can type the desired string when you configure a condition.
- **Property of workflow target object**. Performs comparison with the value of a certain property of the target object of the request that started the workflow. You can select the desired property when you configure a condition.
- **Property of workflow initiator**. Performs comparison with the value of a certain property of the user whose request started the workflow. You can select the desired property when you configure a condition.
- **Changed value of workflow target object property**. Performs comparison with the value that is requested to be assigned to a certain property of the workflow target object, which represents the requested change to the property of the target object of the request that started the workflow. You can select the desired property when you configure a condition.
- **Workflow parameter value**. Performs comparison with the value of a certain parameter of the workflow. You can select the desired parameter from the workflow definition when you configure a condition.
- **Property of object from workflow data context**. Performs comparison with the value of a certain property of the object that will be selected on the basis of the data found in the workflow environment at the time of evaluating the workflow start conditions. When you configure a condition, you can choose the desired property and

specify which object you want the workflow engine to select upon evaluating the condition at workflow start time.

- **Value generated by rule expression**. Performs comparison with the string value of a certain rule expression. By using a rule expression you can compose a string value based on properties of various objects found in the workflow environment at the time of evaluating the workflow start conditions. The workflow engine calculates the value of your rule expression upon evaluating the condition at workflow start time.

Steps to configure filtering conditions

The **Change Workflow Options and Start Conditions** dialog box provides a condition builder for configuring a filter specific to workflow start conditions, located in the **Filtering Conditions** area on the **Conditions** tab. You can access the condition builder in the box under the **Workflow starts only if these conditions are fulfilled** heading.

When you configure a filter, you need to add at least one condition. Initially, you add a condition to the default condition group. You can create additional condition groups to group a set of conditions and nest grouped conditions within other condition groups.

A condition group contains one or more conditions connected by the same logical operator. By grouping conditions, you specify that those conditions should be evaluated as a single unit. The effect is the same as if you put parentheses around an expression in a mathematical equation or logic statement.

To add a condition to a condition group

- In the condition builder, click the name of the condition group and then click **Insert condition**.

  Click the plus sign (+) next to the name of the condition group.

You can remove a condition, if needed, by clicking the **Delete condition** button labeled **X** on the right side of the list item representing the condition in the condition builder.

To add a condition group into another condition group

- Click the name of the condition group, point to **Insert condition group**, and then click an option to specify the logical operator:
  - **AND group**. The condition group evaluates to TRUE if all conditions in the group are TRUE.
  - **OR group**. The condition group evaluates to TRUE if any condition in the group is TRUE.
  - **NOT AND group**. The condition group evaluates to TRUE if any condition in the group evaluates to FALSE.
  - **NOT OR group**. The condition group evaluates to TRUE if all conditions in the group evaluate to FALSE.

By default, AND is the logical operator between the conditions in a condition group. It is possible to change the logical operator by converting the condition group to a different

group type: Click the name of the group, point to **Convert condition group to**, and then click the option appropriate to the desired logical operator.

You can remove an entire condition group, if needed, by clicking the name of the group and then clicking **Delete condition group**.

Once you have added a condition to a condition group, you can use the following steps to configure the condition.

To configure a condition

1. Click **Configure condition to evaluate**, and then choose from the following options to specify the entity or field you want the condition to evaluate:
   - Click **Property of workflow target object** to evaluate a certain property of the workflow target object. Then, click to choose the target property.
   - Click **Property of workflow initiator** to evaluate a certain property of the workflow initiator. Then, click to choose the target property.
   - Click **Changed value of workflow target object property** to evaluate requested changes to a certain property of the workflow target object. Then, click to choose the target property.
   - Click **Workflow parameter value** to evaluate the value of a certain parameter of the workflow. Then, click to choose the desired parameter.
   - Click **Property of object from workflow data context** to evaluate a certain property of a certain object involved in the workflow. Then, click to choose the target object and the target property.
   - Click **Value generated by rule expression** to evaluate the string value generated by a certain rule expression. Then, click to add entries to the rule expression.
2. Click the current comparison operator, if needed, and then click the operator you want the condition to use.

   By default, a condition is configured to use the **equals** operator.
3. Click **Define value to compare to**, and then choose from the following options to specify the desired comparison value:
   - Click **Text string** to performs comparison with a literal string of characters. Then, type the desired string.
   - Click **Property of workflow target object** to perform comparison with the value of a certain property of the workflow target object. Then, click to choose the target property.
   - Click **Property of workflow initiator** to perform comparison with the value of a certain property of the workflow initiator. Then, click to choose the target property.
   - Click **Changed value of workflow target object property** to perform comparison with the value that is requested to be assigned to a certain property of the workflow target object. Then, click to choose the target property.

- Click **Workflow parameter value** to perform comparison with the value of a certain parameter of the workflow. Then, click to choose the desired parameter.
- Click **Property of object from workflow data context** to perform comparison with the value of a certain property of a certain object involved in the workflow. Then, click to choose the target object and the target property.
- Click **Value generated by rule expression** to perform comparison with the string value generated by a certain rule expression. Then, click to add entries to the rule expression.

Configuring script-based conditions

To configure a script-based condition, you need to create and apply a script module containing a function that analyzes the requested operation to determine whether to start the workflow. The function may use the Active Roles ADSI Provider to access properties of objects involved in the operation, analyze the properties, and return TRUE or FALSE depending on the result of the analysis. The workflow starts if the function returns TRUE.

To apply a script-based condition

1. In the condition builder, click the name of the condition group, and then click **Insert condition**.
2. Click **Configure condition to evaluate**, and then click **Value generated by rule expression**.
3. In the **Configure Rule Expression** dialog box, click **Add entry** and then click **Value generated by script**.
4. Use the **Configure Entry** dialog box to select the appropriate script module and script function.
5. Click **OK** to close the **Configure Entry** dialog box.
6. Click **OK** to close the **Configure Rule Expression** dialog box.
7. In the condition builder, verify that comparison operator **equals** is selected.
8. Click **Define value to compare to**, and then click **Text string**.
9. In the **Configure Entry** dialog box, under **Text string**, type TRUE.
10. Click **OK** to close the **Configure Entry** dialog box.
11. Click **OK** to close the **Change Workflow Options and Start Conditions** dialog box.
12. Save your changes to the workflow definition.

As a result of these steps, the workflow will start if the function specified in Step 4 returns TRUE upon evaluating the condition at workflow start time.

For more information and instructions, see “Developing Script Condition Functions” in the Active Roles SDK documentation.

“Run as” options

The “run as” options determine the user account that the workflow runs under. Click the **“Run as” options** link on the **Workflow Options and Start Conditions** page to view or change the account setting. You can choose from the following options:

- **The service account of Active Roles**. The workflow runs under the service account of the Administration Service that executes the workflow.
- **The account of the user who started the workflow**. The workflow runs under the Windows account of the user who requested the operation that started the workflow.

All activities within the workflow normally run under the account identified by the “run as” options for the workflow. However, each activity can be configured to use individual “run as” options. The property page for the activity contains the **“Run as” options** link allowing you to override the workflow “run as” setting on a per-activity basis.

When running under the account of the Administration Service, the workflow activities have the same rights and permissions as the Administration Service itself and thus can perform any tasks allowed for the Administration Service.

When running under the account of the user who started the workflow, the activities can perform only the tasks that Active Roles allows for that user account. The Administration Service processes the activity operation requests as if they were submitted by that user via an Active Roles user interface, so the activities have the rights and permissions the user account is given in Active Roles.

Enforce approval

The **Enforce approval** option determines whether to apply approval rules to the changes requested by the workflow running under a privileged account. When selected, this option causes the approval-pending changes requested by the workflow activities to be submitted for approval regardless of the account under which the workflow is running. Otherwise, the changes are applied without waiting for approval if the workflow is running under the service account of Active Roles, under the account of the approver, or under the account of an Active Roles administrator. This option setting can be overridden on a per-activity basis.

Configuring workflow parameters

Workflow parameters are intended for the purpose of passing their value to workflow activities at run time. You can specify parameter values when you configure a workflow. In this case, Active Roles stores the parameter values as part of the workflow definition, and retrieves them as needed when running the workflow. Another option is to use a script for generating the value of a workflow parameter at run time.

You can use parameters to increase the reusability of a workflow; for example, if a value is specified in the configuration of a workflow activity, then you need to reconfigure that activity if you want to change the value. With workflow parameters, you can reuse the

existing configuration of the activity by passing the appropriate value through a parameter. Here are some examples of workflow parameter usage:

- **Workflow start conditions**. When configuring workflow start conditions, you can create a filter that causes the workflow to start if the properties of the operation request match the value of a certain parameter.
- **If-Else branch conditions**. When configuring conditions for an If-Else branch, you can set up a condition that causes the workflow to choose that branch if a certain parameter has a particular value.
- **Search container**. When configuring a Search activity, you can choose the option that causes the activity to search in the Organizational Unit or container identified by the value of a certain parameter.
- **Search filter**. When configuring a Search activity, you can set up a search filter condition that causes the activity to search for objects whose properties match the value of a certain parameter.
- **Creation container**. You can configure a Create activity with the option to create objects in the Organizational Unit or container identified by the value of a certain parameter.
- **Setting object properties**. You can configure a Create activity or Update activity with the option to set or change the properties of the object based on the value of a certain parameter.
- **Selecting target object**. You can configure an activity to make changes to the object identified by the value of a certain parameter. This applies to activities intended to make changes to objects in Active Directory, such as Update activity, "Add to group" activity, Move activity, and so on.
- **Destination container**. You can configure a Move activity to move the object to the Organizational Unit or container identified by the value of a certain parameter.

Each parameter has a number of properties that define the parameter, including:

- **Name**. Each parameter must have a unique name in the workflow definition.
- **Description**. You can use this property to describe the purpose of the parameter.
- **Display name**. This property specifies the user-friendly name of the parameter.
- **Syntax**. This property determines the data type of the parameter value.
    - **String**. This syntax indicates that the parameter value is a string of characters. You can type the string when you set the value of the parameter.
    - **DateTime**. This syntax indicates that the parameter stores a date and time value. You can use the date and time picker to supply the parameter value.
    - **DN**. This syntax indicates that the parameter value is the Distinguished Name of a certain object. You can use the object picker to supply the parameter value.
    - **ObjectGUID**. This syntax indicates that the parameter value is the Globally Unique Identifier (GUID) of a certain object. You can use the object picker to supply the parameter value.

- **SID**. This syntax indicates that the parameter value is the Security Identifier (SID) of a certain object. You can use the object picker to supply the parameter value.
- **SecureString**. This syntax indicates that the workflow definition stores the parameter value in encrypted form using an encryption key provided by the Active Roles service. You can use this syntax to handle sensitive data such as passwords.
- **AttributeName**. This syntax indicates that the parameter value is the name of a certain attribute from the directory schema. You can use the attribute picker to supply the parameter value.

- **Number of values**. By default, a parameter can store a single value. You can configure a parameter to store a collection of multiple values.
- **Value is required**. By default, a parameter may have no value. You can configure a parameter so that the workflow designer does not allow the workflow definition to be saved if no value is assigned to that parameter.
- **List of acceptable values**. This property specifies a list of values that are allowed to be assigned to the attribute. If a given parameter has this property, then the workflow designer requires a value for that parameter to be selected from the list when you supply the parameter value. When you configure a parameter, you can specify a list explicitly, or you can configure the parameter to use a script that will generate a list of acceptable values or a single value for that parameter at workflow run time.

To add a parameter to a workflow definition

1. In the Active Roles console tree, expand **Configuration | Policies | Workflow**, and select the workflow you want to configure.

   This opens the Workflow Designer window in the details pane, representing the workflow definition as a process diagram.

2. In the details pane, click the **Workflow options and start conditions** button to expand the area above the process diagram, and then click the **Configure** button.
3. Click the **Parameters** tab in the dialog box that opens.
4. On the **Parameters** page, click the **Add** button to open the **Parameter Definition** dialog box.
5. In the **Parameter Definition** dialog box, complete the following fields:
   - **Name**. In this box, type the name you want to assign to the parameter. The name must be unique in the workflow definition.
   - **Description**. Use this box to type a description of the parameter. This field is optional.
   - **Display name**. In this box, type the user-friendly name you want to assign to the parameter.
   - **Syntax**. From this list, select the syntax you want to the parameter to have. See a list of syntax options earlier in this topic.

If you select the **AttributeName** syntax option, you are prompted to configure the attribute picker for this parameter. Select the object class whose attributes you want the attribute picker to list by default, and specify whether you want the attribute picker to allow selecting a different object class. You can also specify whether you want the attribute picker to allow selecting a single attribute or multiple attributes.

6. If you want the parameter to store a collection of multiple values, select the **This parameter is multivalued** check box.
7. If you want the workflow designer to require that a value be assigned to the parameter, select the **This parameter must have a value** check box.
8. If you want to specify a list of acceptable values for the parameter, do one of the following:
   - Configure an explicit list of values by using the **Add**, **Change**, and **Remove** buttons below the **Acceptable values** box.
   - Click **Use script to determine parameter values** below the **Acceptable values** box if you want a list of acceptable values to be generated by a script at workflow run time. Then, click the button next to the **Script name** box to select the script module containing the desired script. The script module must be created beforehand. After you have selected a script module, in the **Function to define a list of acceptable values** list, click the name of the script function. You can choose from the script functions that exist in the script module. The function must be designed to return a collection of values that match the syntax of the parameter.
9. If you want to use a script to assign a value to the parameter at workflow run time, click **Use script to determine parameter values** below the **Acceptable values** box. Then, click the button next to the **Script name** box to select the script module containing the desired script. The script module must be created beforehand. After you have selected a script module, in the **Function to assign a value to this parameter** list, click the name of the script function. You can choose from the script functions that exist in the script module. The function must be designed to return a value that matches the syntax of the parameter.

Parameters are used to specify certain data when configuring or starting the workflow and then pass that data to workflow activities when the workflow is running. The data is represented as parameter values. To assign a value to a given parameter, select the parameter from the list on the **Parameters** tab, and then click the **View or change parameter value** button.

Adding activities to a workflow

The Active Roles console provides the Workflow Designer for creating and configuring workflows. First, you create a workflow definition. Then, you use the Workflow Designer to construct the workflow by adding and configuring workflow activities.

To add an activity to a workflow

1. In the Active Roles console tree, expand **Configuration** | **Policies** | **Workflow**, and select the workflow to which you want to add an activity.

   This opens the Workflow Designer window in the details pane, representing the workflow definition as a process diagram.

2. In the details pane, drag the activity from the left panel onto the process diagram.
3. Right-click the name of the activity in the process diagram and click **Properties**.
4. Use the **Properties** dialog box to configure the activity. See instructions later in this chapter.

If you add an activity to the upper part of the diagram (above the **Operation execution** line), the activity will be run in the pre-execution phase of operation processing (see Workflow processing overview earlier in this chapter). If you add an activity to the lower part of the diagram (beneath the **Operation execution** line), the activity will be run in the post-execution phase of operation processing. Certain activities, such as an Approval activity, which are intended to run in the pre-execution phase, cannot be added to the lower part of the diagram.

In the **Properties** dialog box, you can change the name and description of the activity. These settings are common to all activities. The name identifies the activity in the process diagram. The description appears as a tooltip when you point to the activity in the process diagram. To remove an activity from the process diagram, right-click the name of the activity and click **Delete**.

Configuring an Approval activity

The task of configuring an Approval activity includes the following steps:

- **Choose approvers and configure escalation**. You have to specify, at a minimum, a list of approvers for the initial approver level. Active Roles first assigns approval tasks to the approvers of that level. You can configure additional approver levels to enable escalation of approval tasks.
- **Choose properties for the approver to review, supply or change**. You can list the object properties that the approver must supply when performing the approval tasks (request for additional information), and choose whether the approver is allowed to view or change the object properties that are submitted for approval (review request).
- **Customize the pages for performing the approval task**. You can customize the header of the approval task page by choosing the task title and object properties to be included in the header, and configure custom action buttons in addition to the default action buttons **Approve** and **Reject**.
- **Configure notification**. You can choose the workflow events to notify of, specify the notification recipients and delivery options, and customize the notification message.

This section provides instructions on how to:

- Configure approvers
- Configure escalation
- Configure request for additional information
- Configure request for review
- Customize the header of the approval task page
- Customize approval action buttons

For instructions on how to configure notification settings, see Configuring a Notification activity later in this document.

Configure approvers

A valid approval rule must, at a minimum, specify a list of approvers for the initial approver level. Active Roles first assigns the approval task to the approvers of that level. You can configure additional approver levels to enable escalation of approval tasks.

To specify approvers for the initial approver level

1. In the Active Roles console tree, expand **Configuration | Policies | Workflow**, and select the workflow containing the Approval activity you want to configure.

   This opens the Workflow Designer window in the details pane, representing the workflow definition as a process diagram.

2. In the process diagram, right-click the name of the Approval activity and click **Properties**.
3. In the **Properties** dialog box, click the **Approvers** tab.
4. Verify that the **Initial approver - level 0** item is selected in the **Select approver level to configure** box.
5. Click the **Designate approvers** button.
6. On the **Approvers Selection** page, select check boxes to specify approvers.
7. If you have selected **These users or groups**, use the **Add** and **Remove** buttons to configure the list of approvers.

If you enable escalation on the initial approver level (see Configure escalation), then you have to specify approvers for escalation level 1 (the escalation level subsequent to the initial approver level). Active Roles allows up to 10 escalation levels, each containing a separate list of approvers. If you enable escalation on a given escalation level, then you have to specify approvers for the subsequent escalation level.

To specify approvers for a certain escalation level

1. In the **Select approver level to configure** list, click the escalation level you want to configure.

   To configure a particular escalation level, you must first specify approvers and enable escalation on the preceding approver level.

2. Click the **Designate approvers** button.
3. On the **Approvers Selection** page, select check boxes to specify approvers.
4. If you have selected **These users or groups**, use the **Add** and **Remove** buttons to configure the list of approvers.

The selection of approvers can be based on the Manager or Managed By property:

- By selecting the **Manager of person who requested operation** check box, you configure the Approval activity so that the operations requested by a given user require approval from the manager of that user. With this option, the operation initiated by the user submits the approval task to the person specified as the manager of the user in the directory.
- By selecting the **Manager of operation target object** or **Manager of organizational unit where operation target object is located** check box, you configure the Approval activity so that the changes to a given object require approval from the manager of that object or from the manager of the OU containing that object, respectively. With these options, the operation requesting changes to a given object submits the approval task to the person specified as the manager of the object or OU in the directory.
- By selecting the **Secondary owners of operation target object** check box, you configure the Approval activity so that the changes to the operation target object require approval from any person who is designated as a secondary owner of that object. Secondary owners may be assigned to an object, in addition to the manager (primary owner), to load balance the management of the object.
- By selecting the **Manager of person being added or removed from target group** check box, you configure the Approval activity so that the addition or removal of an object from the operation target group requires approval from the manager of that object. For example, given a request to add a user to the operation target group, this option causes the Approval activity to submit the approval task to the person specified as the manager of the user in the directory.

When you specify approvers for an escalation level, additional options are available:

- **Manager of approver of preceding level**. Use this option to escalate the approval task to the manager of the user or group that is designated as an approver on the preceding approver level. Suppose a given user is an initial approver, and escalation is enabled on the initial approver level. When escalation occurs, the approval task will be assigned to the manager of that user.
- **Secondary owner of approver of preceding level**. Use this option to escalate the approval task to the secondary owner of the user or group that is designated as an approver on the preceding approver level. Suppose a given group is an initial approver, and escalation is enabled on the initial approver level. When escalation occurs, the approval task will be assigned to the secondary owner of that group.

The selection of approvers may also be based on a script function that chooses the approver when the Approval activity is being executed. The function may access properties of objects involved in the operation, analyze the properties, and return an identifier of the user or group to be selected as an approver. For more information and instructions, refer to the "Developing Functions for Designating Approvers" topic in the Active Roles SDK documentation.

Configure escalation

An Approval activity may define multiple approver levels, each containing a separate list of approvers. Active Roles uses approver levels when escalating time-limited approval tasks. For each approver, level the Approval activity can specify a certain time period. If an approver of a given level does not complete the approval task within the specified time period, then Active Roles assigns the task to the approvers of the next level. This process is referred to as escalation.

A valid Approval activity must specify a list of approvers for the initial approver level. Active Roles first assigns the approval task to the approvers of that level. To enable escalation, a separate list of approvers must be specified for the subsequent escalation level.

To configure escalation on the initial approver level

1. Specify approvers for the initial approver level (for instructions, see Configure approvers earlier in this document).
2. Verify that the **Initial approver - level 0** item is selected in the **Select approver level to configure** box.
3. Select one or both of these options:
   - **Approval task has a time limit of** <number> **days** <number> **hours**. Specify the time period within which the initial approver has to complete the approval task.
   - **Allow approver to escalate approval task**. When selected, allows the approvers of the initial level to reassign their approval tasks to the approvers of escalation level 1.
4. If you have selected only the first option (a time limit for the task), then select the **Escalate approval task to Escalation level 1** option. Otherwise, escalation is not enabled.
5. In the **Select approver level to configure** box, click **Escalation level 1**.
6. Specify approvers for escalation level 1 (for instructions, see Configure approvers earlier in this document).

Active Roles allows up to 10 escalation levels, each containing a separate list of approvers. You can configure escalation levels one after another to create an escalation chain. Thus, after you have configured escalation on the initial approver level, you can configure escalation on escalation level 1, then you can configure escalation on escalation level 2, and so on. As a result, you could achieve the following sequence of events. If the initial approvers do not complete the approval task on time, then the task is assigned to the approvers of escalation level 1. If the approvers of escalation level 1 do not complete the approval task within their time frame, the task is assigned to the approvers of escalation level 2 with the new time limit. This escalation chain may contain up to 10 escalation levels.

To configure escalation on a certain escalation level

1. In the **Select approver level to configure** list, click the escalation level you want to configure.

   To configure a particular escalation level, you must first specify approvers and enable escalation on the preceding approver level.

2. Select one or both of these options:

   - **Approval task has a time limit of** <number> **days** <number> **hours** Specify the time period within which the initial approver has to complete the approval task.
   - **Allow approver to escalate approval task** When selected, allows the approvers of the current level to reassign their approval tasks to the approvers of the next level.

3. If you have selected only the first option (a time limit for the task), then select the **Escalate approval task to Escalation level** <number> option. Otherwise, escalation is not enabled.

4. In the **Select approver level to configure** box, click the item representing the subsequent escalation level.

   For example, if you are configuring escalation level 1, click the **Escalation level 2** list item.

5. Specify approvers for the subsequent escalation level (for instructions, see Configure approvers earlier in this document).

Note that each approver level has an individual configuration, so the escalation options of a given level apply only to that level. Thus, each approver level has a separate time limit, the option that determines whether to escalate the approval task after the time limit has expired, and whether the approvers of the given level are allowed to escalate the approval task manually.

Configure request for additional information

You can configure the Approval activity so that the approver is requested to supply certain properties of the object when performing the approval task. Suppose the creation of a user is submitted for approval. The approver may be requested to supply certain properties of the user in addition to the the properties specified in the creation request. Thus, you may configure the Approval activity to prompt the approver to specify the mailbox database for the mailbox of the user to be created.

To configure request for additional information

1. Go to the **Request for information** tab in the **Properties** dialog box for the Approval activity.
2. Add the desired properties to the **Request the approver to supply or change these properties** list.

When performing the approval task, the approver will be prompted to supply or change the properties presented in that list. The approver can provide the requested information in the Approval section of the Web Interface, under the **Supply or change the following properties** heading on the **Object Properties** tab of the **Approval Task** page. The tab also displays an instruction specified by the Approval activity. You can view or change the instruction text on the **Request for information** tab in the **Properties** dialog box for the Approval activity, under the **Show this instruction to the approver** heading.

Configure request for review

You can configure the Approval activity so that the approver will be requested to review the object properties submitted for approval. One more option is to allow the approver to make changes to those properties.

To configure request for review

1. Go to the **Request for information** tab in the **Properties** dialog box for the Approval activity.
2. Select the **Show the original request to the approver** check box to enable the approver to review the properties submitted for approval.
3. Optionally, select the **Allow the approver to modify the original request** check box to allow the approver to make changes to the properties submitted for approval.

When the **Show the original request to the approver** check box is selected, the **Object Properties** tab of the **Approval Task** page in the Approval section of the Web Interface displays the object properties submitted for approval. The property values are shown read-only in the area under the **Review the properties submitted for approval** heading. You can configure the Approval activity to allow the approver to change those property values by selecting the **Allow the approver to modify the original request** check box. If you do not want the approver to view the properties submitted for approval, clear the **Show the original request to the approver** check box.

Customize the header of the approval task page

You can configure the Approval activity to specify how the approval tasks created by that activity are identified in the Approval section of the Web Interface. The Approval section contains a list of approval tasks, with each task identified by a header that provides basic information about the task, including the title of the task and information about the target object of the operation that is subject to approval. The title of the task is located in the middle of the task's header. The properties that identify the operation target object are displayed above the title of the task.

To change the title of the approval task

1. Go to the **Customization** tab in the **Properties** dialog box for the Approval activity.
2. Click **Customize the task header area**.

3. Type the appropriate title in the **Display this title to identify the approval task** box.

   By default, the title is **Approve operation**.

To change the properties that identify the operation target object

1. Under **Customize the task header area**, verify that the **Display these properties of the object submitted for approval** check box is selected.
2. Use the **Add** and **Remove** buttons to configure the list of properties.

   By default, the list contains the **Friendly Name** property, which causes Active Roles to use the display name of the object. If the object does not have a display name, then Active Roles uses the name of the object.

By default, the approval task's header provides summary information about the changes that are subject to approval, including the type of the changes and the reason for the changes. You can configure the header not to display that information by clearing the **Display the operation summary in the task header area** check box.

Changes to the configuration of the task's header have an effect on the tasks created by the Approval activity after the changes were made, and don't affect the tasks created earlier.

Customize approval action buttons

You can configure the Approval activity to specify the actions the approver can take on the approval task. On the pages for performing the approval task, in the Approval section of the Web Interface, the task header contains the action buttons that are intended to apply the appropriate resolution to the task, such as **Approve** or **Reject**. The action buttons are located at the bottom of the header area. Which buttons are displayed depends upon configuration of the Approval activity.

To rename or hide an action button

1. Go to the **Customization** tab in the **Properties** dialog box for the Approval activity.
2. Click **Customize action buttons**.
3. Click the title of the button in the list, and then click **Edit**.
4. In the **Action Button Properties** dialog box, perform the following tasks:
   - To rename the button, type the appropriate name in the **Button title** box.
   - The new name will appear on the action button in the Web Interface.
   - To hide the button, clear the **Is visible on the pages for performing the approval task** check box.
   - As a result, the Web Interface will not display the action button.

You can restore the action button in the Web Interface by selecting the **Is visible on the pages for performing the approval task** check box. Note that this option is unavailable for the Escalate or Delegate action type. The Web Interface displays the Escalate or

Delegate button if the Approval activity allows the approver to escalate or reassign (delegate) the approval task, respectively.

Action buttons appear on the pages for performing the approval task. Each button applies a certain action to the task. You can add buttons to create custom actions. Clicking a custom action button allows (**Complete** action type) or denies (**Reject** action type) the operation that is subject to approval. If-Else activities can refer to a custom action button by title and elect the appropriate branch of the workflow when the approver clicks that button.

To add a custom action button

1. Go to the **Customization** tab in the **Properties** dialog box for the Approval activity.
2. Click **Customize action buttons**.
3. Click **Add**.
4. In the **Action Button Properties** dialog box, do the following:
   a. In the **Button title** box, type the appropriate name of the button.

      This name will appear on the action button in the Web Interface

   b. From the **Action type** list, select the appropriate type of the action button.

      When applied to an approval task, the **Complete** action type, causes the workflow to continue, allowing the operation that is subject to approval; the **Reject** action type button denies the operation.

   c. Select the **Is visible on the pages for performing the approval task** check box.

When you add a custom action button, you may want to include an instruction explaining the meaning and purpose of the custom action. You can type the text of the instruction in the **Show this instruction for action buttons** box in the **Customize action buttons** area on the **Customization** tab in the **Properties** dialog box for the Approval activity. The approver will see that text above the action buttons on the pages for performing the approval task in the Web Interface.

To complete an approval task, the approver normally has to fill in a confirmation dialog box. You can configure the Approval activity to prevent the confirmation dialog box from appearing: Select the **Suppress the confirmation dialog upon completion of approval task** check box in the **Customize action buttons** area on the **Customization** tab in the **Properties** dialog box for the Approval activity.

Configuring a Notification activity

When configuring a Notification activity, you can specify notification settings such as workflow events to notify of, notification recipients, and notification message template. The same settings apply to the Notification section of other activities such as an Approval activity, a Search activity, and CRUD activities.

To view or change notification settings

1. In the Active Roles console tree, expand **Configuration | Policies | Workflow**, and select the workflow containing the activity you want to configure.

   This opens the Workflow Designer window in the details pane, representing the workflow definition as a process diagram.

2. In the process diagram, right-click the name of the activity and click **Properties**.
3. Click the **Notification** tab in the **Properties** dialog box.

The page for configuring notifications includes three areas:

- Events, recipients and messages In this area you can add, view, change, or remove notifications, each of which determines an event to notify of, the recipients of the notification message, the message delivery options, and the message template.
- Active Roles Web Interface This area is used to specify the address (URL) of the Active Roles Web Interface, for constructing hyperlinks in the notification messages.
- E-mail server settings In this area you can view or change the name and other settings of the e-mail server that is used for delivery of notification messages.

Events, recipients and messages

To add a notification

1. In the **Events, recipients and messages** area, click **Add**.
2. In the **Notification Settings** dialog box, in the **Select an event**, list click the event to notify of.
3. On the **Notification Recipients** tab, select check boxes to specify the notification recipients.

   The e-mail addresses of the recipients you select on the **Notification Recipients** tab appear in the **To** field of the notification e-mail messages. To add recipient addresses to the **Cc** or **Bcc** field, click the **Cc Recipients** or **Bcc Recipients** button, respectively. This opens a page that is similar to the **Notification Recipients** tab, allowing you to view or change which recipient addresses will appear in the **Cc** or **Bcc** field.

4. On the **Notification Delivery** tab, select the delivery options you want:
   - Select the **Immediate** option for the notification message to be sent immediately, on every occurrence of the event.
   - Select the **Scheduled** option for the notification messages within a certain time period to be grouped and sent as a single message; then, specify the desired period. This option is available only for the **Task created** event in an Approval activity.
5. On the **Notification Message** tab, click **Modify** to view or change the message template, including the subject and the body of the notification message.

6. In case of a Notification activity, choose additional options on the **Notification Message** tab as needed:
    - If you want the notification message to include the Change History report (in case of a change workflow) or Run History report (in case of an automation workflow), select the **Attach a report of workflow execution to notification message** check box.
    - For the activity to send plain-text notification messages, select the **Format notification message as plain text** check box. Otherwise, the activity sends notification messages in HTML format.
7. Click **OK** to close the **Notification Settings** dialog box.

For the **Task created** event in an Approval activity, notification can be configured so that notification messages are grouped together and sent out on a scheduled basis. If you select the **Scheduled** option on the **Notification Delivery** tab, the messages within a certain, scheduled period are accumulated in a temporary storage instead of being sent out immediately upon event occurrences. Upon the expiration of that period, all the collected messages are sent out as a single message. You can configure the activity to deliver notification on a daily or hourly schedule.

Clicking **Modify** on the **Notification Message** tab opens a window where you can view and modify e-mail notification templates. For each event type, the notification configuration defines a default template based on which Active Roles composes e-mail notification messages. Each template includes XHTML markup along with the text and tokens representing information about the event.

To make notification messages more meaningful to the recipients, notification templates provide the option for the messages to include tokens representing additional information about the event. Click the **Insert Token** button to view a list of the available tokens. The list provides a brief description for each token.

You can edit templates in order to customize the contents and format of notification e-mails. The changes to templates are notification-specific and event-specific: When you modify the template for a certain event within the configuration of a certain notification, your changes have no effect on the other notifications or events. This allows different notifications and events to have different, custom notification templates.

To view or change a notification

- Click an entry in the **Events, recipients and messages** list, click **Edit**, and use the **Notification Settings** dialog box as described earlier in this topic.

To delete a notification

- Click an entry in the **Events, recipients, and messages** list, and then click **Remove**.

Active Roles Web Interface

The address (URL) specified in this area is used to construct hyperlinks in the notification messages so that notification recipients can easily access the Web Interface pages for

performing workflow tasks.

To specify the address of the Active Roles Web Interface

1. In the edit box under **Active Roles Web Interface**, type the address (URL) of the Active Roles Web Interface site (for example, http://<server>/ARServerAdmin).
2. Click **Test** to verify the address. If the address is correct, this opens the Web Interface site in your Web browser.

E-mail server settings

The settings specified in this area determine the server to use for notification delivery.

To configure e-mail server settings

1. In the **E-mail server settings** area, click **Properties**.
2. Use the **Properties** dialog box to view or change the e-mail server settings.

To select a different e-mail server configuration

- Click the name of the desired configuration in the **Configuration of the outgoing mail server** list.

To create an e-mail server configuration

- In the Active Roles console tree, expand **Configuration | Server Configuration**, right-click **Mail Configuration**, and select **New | Mail Configuration**.

Configuring a Script activity

When configuring a Script activity, you select the Script module that contains the script to be used by the activity, and then, from the functions held in that script, you choose the function to be run by the activity and, optionally, the function that declares the activity parameters. If any parameters are declared, then you need to supply parameter values. For information and instructions on how to create a script for a Script activity, refer to Active Roles SDK documentation.

To configure a Script activity

1. In the Active Roles console tree, expand **Configuration | Policies | Workflow**, and select the workflow containing the Script activity you want to configure.

   This opens the Workflow Designer window in the details pane, representing the workflow definition as a process diagram.
2. In the process diagram, right-click the name of the activity and click **Properties**.
3. Go to the **General** tab in the **Script Activity Properties** dialog box.

4. Do one of the following:
   - If the activity has no Script Module selected (for example, the activity has just been added to the process diagram), click **Browse** and select the Script Module containing the script you want the activity to use.
   - If the activity already has a Script Module selected and you want to use a different Script Module, click the **Browse** button to select the Script Module you want.
5. In the **Function to run** box, view the name of the script function that will be run by this activity. You can choose the appropriate function from the **Function to run** list.

   The list contains the names of all script functions found in the selected Script Module. The activity runs the function specified in the **Function to run** box.
6. In **Function to declare parameters** box, view the name of the function that defines the activity parameters. Click **Specify Parameters**, and then do the following:
   - If necessary, from the **Function to declare parameters** list, choose the function that defines the parameters specific to this activity.

     The list contains the names of all script functions found in the selected Script Module. The activity has the parameters that are defined by the function specified in the **Function to declare parameters** box. Normally, this is a function named onInit.
   - Under **Parameter values**, view or change the values of the activity parameters. To change the value of a parameter, select the name of the parameter and click **Edit**.

     Clicking **Edit** displays a page where you can add, remove, or select a value or values for the selected parameter. For each parameter, the function that is used to declare parameters defines the name of the parameter and other characteristics, such as a description, a list of possible values, the default value, and whether a value is required. If a list of possible values is defined, then you can only select values from that list.
7. (Optional) Go to the **Notification** tab in the **Script Activity Properties** dialog box, and use the steps for Configuring a Notification activity to subscribe recipients to the notifications of the following events:
   - **Activity completed successfully**. When configured to notify of this event, the activity causes Active Roles to send a notification e-mail if no significant errors occurred during execution of this activity.
   - **Activity encountered an error**. When configured to notify of this event, the activity causes Active Roles to send a notification e-mail if any significant errors occurred during execution of this activity.
8. (Optional) Go to the **Error handling** tab in the **Script Activity Properties** dialog box, and select or clear the **Continue workflow even if this activity encounters an error** check box to specify whether you want Active Roles to suppress errors encountered by this Script activity.

   If this check box is not selected (default setting), then an error condition encountered by the activity causes Active Roles to terminate the workflow. If you select this check

box, the workflow continues regardless of whether or not the activity encounters an error condition.

Configuring an If-Else activity

An If-Else activity is a composite activity. It is composed of several branches, each of which has individual conditions specified. An If-Else branch may contain any number of other activities. Every operation that satisfies the conditions specified on a given branch causes Active Roles to run the activities included in that branch. Only one branch of a single If-Else activity can be run even though an operation may satisfy the conditions on more than one branch.

Typically, an If-Else activity has two branches, with certain conditions specified on the first (leftmost) branch. The second branch has no conditions specified on it, so as to act as the Else branch. If an operation satisfies the conditions, the activities included in the first branch are run; otherwise, the operation flows through the activities found in the second branch.

Configuring an If-Else activity involves the following tasks:

- Adding a branch
- Adding activities to a branch
- Configuring branch conditions (see Configuring conditions for an If-Else branch)
- Configuring error handling (see Steps to configure error handling)

To add a branch to an If-Else activity

1. In the Active Roles console tree, expand **Configuration | Policies | Workflow**, and select the workflow containing the If-Else activity you want to configure.

   This opens the Workflow Designer window in the details pane, representing the workflow definition as a process diagram.

2. In the process diagram, right-click the name of the If-Else activity and click **Add Branch**.

This adds a branch with the default name of **If-Else** Branch. Right-click the name of the branch and click **Properties** to change the name as necessary. You can delete a branch by clicking the name of the branch and then clicking **Delete**.

To add an activity to an If-Else branch

- Drag the activity from the left panel onto the branch.

If you add an activity to the upper part of the diagram (above the **Operation execution** line), the activity will be run in the pre-execution phase of operation processing (see Workflow processing overview earlier in this chapter).

If you add an activity to the lower part of the diagram (beneath the **Operation execution** line), the activity will be run in the post-execution phase of operation processing. Certain

activities, such as an Approval activity, which are intended to run in the pre-execution phase, cannot be added to the lower part of the diagram.

You can delete an activity from a branch by clicking the name of the activity and then clicking **Delete**.

The following topic provides instructions on how to configure conditions for an If-Else branch: Configuring conditions for an If-Else branch.

Steps to configure error handling

When configuring an If-Else activity, you can configure error handling to suppress errors encountered by that If-Else activity and all activities included in that If-Else activity.

To configure error handling for an If-Else activity

1. In the process diagram, right-click the name of the If-Else activity and click **Properties**.
2. Go to the **Error handling** tab in the **If-Else Activity Properties** dialog box, and select or clear the **Continue workflow even if this activity encounters an error** check box on that tab.

If the **Continue workflow even if this activity encounters an error** check box is not selected (default setting), then an error condition encountered by the activity causes Active Roles to terminate the workflow. If you select this check box, the workflow continues regardless of whether or not the If-Else activity or any activity within the If-Else activity encounters an error condition.

Configuring conditions for an If-Else branch

An If-Else activity is intended to select exactly one branch of the activity from a given set of branches. For each branch, the activity checks the branch conditions and executes the first of the branches whose condition evaluates to TRUE.

The workflow designer provides a condition builder for configuring branch conditions, located in the **If-Else Branch Activity Properties** dialog box.

To access the condition builder for an If-Else branch

1. Right-click the name of the branch and click **Properties**.
2. Go to the **Conditions** box in the **If-Else Branch Activity Properties** dialog box that opens.

When you configure an If-Else branch, you need to add at least one condition. By default, a single, implied condition group is created when you add a branch condition. You can create additional condition groups to group a set of conditions and nest grouped conditions within other condition groups.

A condition group contains one or more conditions connected by the same logical operator. By grouping conditions, you specify that those conditions should be evaluated as a single

unit. The effect is the same as if you put parentheses around an expression in a mathematical equation or logic statement.

To add a condition to a condition group

- In the **Conditions** box, click the name of the condition group and then click **Insert condition**.

  Click the plus sign (+) next to the name of the condition group.

You can remove a condition, if needed, by clicking the **Delete condition** button labeled **X** on the right side of the list item representing the condition in the **Conditions** box.

To add a condition group into another condition group

- In the **Conditions** box, click the name of the condition group, point to **Insert condition group**, and then click an option to specify the logical operator:
  - **AND group**. The condition group evaluates to TRUE if all conditions in the group are TRUE.
  - **OR group**. The condition group evaluates to TRUE if any condition in the group is TRUE.
  - **NOT AND group**. The condition group evaluates to TRUE if any condition in the group evaluates to FALSE.
  - **NOT OR group**. The condition group evaluates to TRUE if all conditions in the group evaluate to FALSE.

By default, AND is the logical operator between the conditions in a condition group. It is possible to change the logical operator by converting the condition group to a different group type: Click the name of the group, point to **Convert condition group to**, and then click the option appropriate to the desired logical operator.

You can remove an entire condition group, if needed, by clicking the name of the group and then clicking **Delete condition group**.

Once you have added a condition to a condition group, you can use the following steps to configure the condition.

To configure a condition

1. Click **Configure condition to evaluate**, and then choose from the following options to specify the entity or field you want the condition to evaluate:
   - **Property of workflow target object**. Evaluate the value of a certain property of the target object of the request that started the workflow. The condition builder prompts you to choose the desired property. This option is unavailable in case of automation workflow.
   - **Property of workflow initiator**. Evaluate the value of a certain property of the user whose request started the workflow. The condition builder prompts you to choose the desired property.
   - **Changed value of workflow target object property**. Evaluate the value that is requested to be assigned to a certain property of the workflow target

object, which represents the requested change to the property of the target object of the request that started the workflow. The condition builder prompts you to choose the desired property. This option is unavailable in case of automation workflow.

- **Activity execution status**. Evaluate whether or not Active Roles encountered an error when executing a certain activity. The condition builder prompts you to select the desired activity. Note that this option requires the activity configuration to allow the workflow to continue even if the activity encounters an error.
- **Approver action choice**. Evaluate the name of the action button applied by the approver to complete the approval task created by a certain Approval activity. Use this option to determine which action button the approver applied to allow the operation that was subject to approval. The condition builder prompts you to select the desired Approval activity. This option is unavailable in case of automation workflow.
- **Workflow parameter value**. Evaluate the value of a certain parameter of the workflow. The condition builder prompts you to select the desired parameter from the workflow definition.
- **Property of object from workflow data context**. Evaluate the value of a certain property of the object that will be selected by the If-Else activity on the basis of the data found in the workflow environment at the time of executing the workflow. The condition builder prompts you to choose the desired property and specify which object you want the activity to select upon evaluating the condition at workflow run time.
- **Value generated by rule expression**. Evaluate the string value of a certain rule expression. The condition builder prompts you to configure a rule expression. By using a rule expression you can compose a string value based on properties of various objects found in the workflow environment at the time of executing the workflow. Active Roles calculates the value of your rule expression upon evaluating the condition at workflow run time.

2. Click the current comparison operator, if needed, and then click the operator you want the condition to use.
3. Click **Define value to compare to**, and then choose an option to specify the desired comparison value.

The list of options that are available to specify a comparison value depends upon the entity or field you have configured the condition to evaluate. The following table summarizes the comparison value options.

Table 59: Comparision value options

Condition to evaluate	Comparison value options
Property of workflow target object - OR - Property of workflow initiator	• Text string • Property of workflow target object • Property of workflow initiator

Condition to evaluate	Comparison value options
- OR - Changed value of workflow target object property - OR - Workflow parameter value - OR - Property of object from workflow data context - OR - Value generated by rule expression	• Changed value of workflow target object property • Workflow parameter value • Property of object from workflow data context • Value generated by rule expression
Activity execution status	• Not executed • Completed successfully • Encountered an error
Approver action choice	• The name of an action button • Value generated by script

For a brief description of comparison operators and comparison value options, see Search filter.

Configuring a script-based condition

To configure a script-based condition, you need to create and apply a script module containing a function that analyzes the requested operation to determine whether to run the branch. The function could use the Active Roles ADSI Provider to access properties of objects involved in the operation, analyze the properties, and return TRUE or FALSE depending on the result of the analysis. The branch runs if the function returns TRUE.

To apply a script-based condition

1. Right-click the name of the branch and click **Properties**.
2. In the **If-Else Branch Activity Properties** dialog box, under **Conditions**, do the following:
   a. Click the title of the condition group and then click **Insert condition**.
   b. Click **Configure condition to evaluate** and then click **Value generated by rule expression**.
3. In the **Configure Rule Expression** dialog box, click **Add entry** and then click **Value generated by script**.
4. Use the **Configure Entry** dialog box to select the appropriate script module and script function.

5. Click **OK** to close the **Configure Entry** dialog box.
6. Click **OK** to close the **Configure Rule Expression** dialog box.
7. In the **If-Else Branch Activity Properties** dialog box, under **Conditions**, do the following:
    a. Verify that comparison operator **equals** is selected.
    b. click **Define value to compare to**, and then click **Text string**.
8. In the **Configure Entry** dialog box, under **Text string**, type TRUE.
9. Click **OK** to close the **Configure Entry** dialog box.
10. Click **OK** to close the **If-Else Branch Activity Properties** dialog box.
11. Save your changes to the workflow definition.

As a result of these steps, the If-Else branch you have configured will be selected if the function specified in Step 4 returns TRUE at workflow run time. For more information and instructions, see "Developing Script Condition Functions" in the Active Roles SDK documentation.

Configuring a Stop/Break activity

When configuring a Stop/Break activity, you can specify the text of an information message. The activity terminates the workflow instance and reports the corresponding event to the Active Roles event log. The message is included in the event description. If possible, the activity also displays that message in the client user interface (such as the Active Roles console or Web Interface) that was used to request the operation that started the workflow.

To configure a Stop/Break activity

1. In the Active Roles console tree, expand **Configuration | Policies | Workflow**, and select the workflow containing the Stop/Break activity you want to configure.

    This opens the Workflow Designer window in the details pane, representing the workflow definition as a process diagram.
2. In the process diagram, right-click the name of the activity and click **Properties**.
3. View and, if necessary, change the message text in the **Information message** box.

Configuring an Add Report Section activity

You can use an Add Report Section activity to add custom information to the change history report (in case of workflow started by an operation request) or run history report (in case of automation workflow). The activity adds a separate section to the Workflow activities and policy actions area of the report. The section consists of a header and a body. When

you configure an Add Report Section activity, you specify what information you want the header and the body to contain.

To configure an Add Report Section Activity

1. In the Active Roles console tree, expand **Configuration | Policies | Workflow**, and select the workflow containing the Add Report Section activity you want to configure.

   This opens the Workflow Designer window in the details pane, representing the workflow definition as a process diagram.

2. In the process diagram, right-click the name of the activity and click **Properties**.
3. Under **This report section is intended to display information about**, select the **Error condition** option if you want the report to display the text of the header and the body of the report section in red. Otherwise, select the **Successful operation** option.
4. Under **Header of the report section**, click **Define** to compose the text of the header. The following options are available:
   - **Text string**. Specify a literal string of characters to be displayed as the header of the report section. The Workflow Designer prompts you to type the desired string.
   - **Value generated by rule expression**. Compose the header text of data entries to be calculated during execution of the activity. The Workflow Designer prompts you to configure a string of entries, and offers various entry types allowing the header text to include properties of objects involved in the workflow and related objects, date and time of activity execution, and workflow parameters.
5. Under **Body of the report section**, click **Add text** and choose from the following options to configure the body text of the report section:
   - **Text string**. Add a literal string of characters. The Workflow Designer prompts you to type the desired string.
   - **Workflow date and time**. Add a date/ time string representing the date and time that the activity is started at workflow run time (referred to as the current date and time in the Workflow Designer). You can change the format of the date/time string and specify a time offset, in days, if needed.
   - **Workflow parameter value**. Add a text string specified by a particular parameter of the workflow. The Workflow Designer prompts you to select the desired parameter.
   - **Workflow parameter value**. Add a text string specified by a particular parameter of the workflow. The Workflow Designer prompts you to select the desired parameter.
   - **Newline character (CR/LF)**. Add the end-of-line code to start a new string.
   - **Tab character**. Add a tab character to the string.

- **Bullet character**. Add a bullet point to the string. You can use a bullet point followed by a tab character at the beginning of a string to format the string as a bulleted list item.
- **Property of object from workflow data context**. Add the value of a certain property of the object that will be selected by the activity on the basis of the data found in the workflow environment at the time of executing the workflow. The Workflow Designer prompts you to choose the desired property and specify which object you want the activity to select upon creating the report section at workflow run time.

In the **Body of the report section** box, you can modify, reorder, or remove text entries. To modify a text entry, click the text and then click **Edit**. To reorder or remove text entries, use the buttons on the right side of the list items representing the text entries in the **Body of the report section** box. Thus, to remove an entry, click the **X** button on the right side of the list item representing that entry in the **Body of the report section** box.

Configuring a Search activity

You can use a Search activity to perform a search against directory data to find objects, such as users or groups, that match the criteria you specify based on object properties, object location and other information available in the execution environment of the workflow, and to pass these objects to other activities so that the workflow can perform the appropriate actions on them. You can insert activities into a Search activity and have those activities process the objects found by the Search activity.

To add an activity to a Search activity

1. In the Active Roles console tree, expand **Configuration | Policies | Workflow**, and select the workflow containing the Search activity.

   This opens the Workflow Designer window in the details pane, representing the workflow definition as a process diagram.

2. In the details pane, drag the activity from the left panel onto Search activity in the process diagram.

To configure a Search activity, right-click the name of that activity in the process diagram and click **Properties**. Then, perform the following tasks in the **"Search" Activity Properties** dialog box:

- Configure scope and filter
- Configure notification
- Configure error handling
- Configure "run as" options
- Configure additional settings

Configure scope and filter

Use the scope and filter settings to define where you want the activity to search and what you want the activity to search for. These settings are configured on the **Search and scope** tab in the **"Search" Activity Properties** dialog box.

To configure scope and filter settings

1. From the **Use this activity to** list, choose the option appropriate to your search scenario:
   - Choose **Search in the Organizational Unit or container** to search a certain OU or container for objects that match your search criteria.
   - Choose **Search for resources managed or owned by the user or group** to search for the managed objects of a particular user or group that match your search criteria. Managed objects of a user or group are those for which the user or group is the primary owner (manager) or a secondary owner.
   - Choose **Search the group for its members** to search for the members of a certain group that match your search criteria.
   - Choose **Search for direct reports of the user** to search for the direct reports of a particular user that match your search criteria. Direct reports of a given user are the users for which that user is the manager.
   - Choose **Search within the object's attribute (ASQ search)** to search for the objects listed in a certain attribute of a particular object that match your search criteria.
2. From the **Find** list, choose the type of object to search for.

   Depending on the search scenario option, you can choose from the following object types:
   - **Users** Search for user accounts.
   - **Contact**s Search for contact objects.
   - **Groups** Search for groups.
   - **Computers** Search for computer accounts
   - **Printers** Search for printer objects.
   - **Organizational Units** Search for Organizational Units.
   - **Shared Folders** Search for shared folder objects.
   - **Exchange Recipient**s Search for mailboxes or mail-enabled users, groups, or contacts.
   - **Inactive Accounts** Search for users computers that haven't logged on for more than a certain number of days, have the password age of more that a certain number of days, or are expired for more than a certain number of days.
   - **All Objects** Search for objects of any type.

   Some of these object types are unavailable for certain search scenario options. For example, with the option to search for direct reports, the only available object types

are **Users** and **All Objects**. Consult the Object type topic to see what object types are available for a given search scenario option.

3. Click in the **In** box to specify where you want the activity to search.

   The role of the object you configure in the **In** box depends upon your search scenario option:

   - With the **Search in the Organizational Unit or container** option, the activity will search the OU or container specified in the **In** box.
   - With the **Search for resources managed or owned by the user or group** option, the activity will search for the managed objects of the user or group specified in the **In** box.
   - With the **Search the group for its members** option, the activity will search for members of the group specified in the **In** box.
   - With the **Search for direct reports of the user** option, the activity will search for direct reports of the user specified in the **In** box.
   - With the **Search within the object's attribute (ASQ search)** option, the activity will search for objects listed in a certain attribute of the object specified in the **In** box. You can choose the attribute to search.
   - When you click in the **In** box, the workflow designer offers a number of options for you to specify the desired object. Depending on your search scenario, you can choose from the following options:

Table 60: Configure and scope filter settings

Search scenario	"Find-in" options available
Search in the Organizational Unit or container.	• **Fixed container in directory**. Search in the given OU or container. You can select the desired OU or container in Active Directory when you configure a Search activity. • **Parent OU of workflow target object**. Search in the OU that holds the target object of the request that started the workflow. • **Object identified by workflow parameter**. Search in the OU or container specified by the value of a certain parameter of the workflow. You can choose the desired parameter when you configure a Search activity. • **Object from workflow data context**. Search in the OU or container that will be selected by the Search activity on the basis of the data found in the workflow environment at the time of executing the workflow. When configuring a Search activity, you can specify which OU or container you want the activity to select at workflow run time. • **Object identified by DN-value rule expression**. Search in the OU or container whose Distinguished Name

Search scenario	"Find-in" options available
	(DN) is specified by the string value of a certain rule expression. By using a rule expression you can compose a string value based on properties of various objects found in the workflow environment at the time of executing the workflow. You can create the desired rule expression when you configure a Search activity.
Search for resources managed or owned by the user or group.	• **Workflow target object**. Search for resources managed or owned by the target object of the request that started the workflow. • **Object identified by workflow parameter**. Search for resources managed or owned by the object specified by the value of a certain parameter of the workflow. You can choose the desired parameter when you configure a Search activity. • **Object from workflow data context**. Search for resources managed or owned by the object that will be selected by the Search activity on the basis of the data found in the workflow environment at the time of executing the workflow. When configuring a Search activity, you can specify which object you want the activity to select at workflow run time. • **Object identified by DN-value rule expression**. Search for resources managed or owned by the object whose Distinguished Name (DN) is specified by the string value of a certain rule expression. By using a rule expression you can compose a string value based on properties of various objects found in the workflow environment at the time of executing the workflow. You can create the desired rule expression when you configure a Search activity.
Search the group for its members.	• **Workflow target object**. Search for members of the group that is the target object of the request that started the workflow. • **Object identified by workflow parameter**. Search the group specified by the value of a certain parameter of the workflow. You can choose the desired parameter when you configure a Search activity. • **Object from workflow data context**. Search for members of the group object that will be selected by the Search activity on the basis of the data found in the workflow environment at the time of executing the workflow. When configuring a Search activity, you can

Search scenario	"Find-in" options available
	specify which group object you want the activity to select at workflow run time. • **Object identified by DN-value rule expression**. Search the group whose Distinguished Name (DN) is specified by the string value of a certain rule expression. By using a rule expression, you can compose a string value based on properties of various objects found in the workflow environment at the time of executing the workflow. You can create the desired rule expression when you configure a Search activity.
Search for direct reports of the user.	• **Workflow target object**. Search for direct reports of the target object of the request that started the workflow. • **Object identified by workflow parameter**. Search for direct reports of the object specified by the value of a certain parameter of the workflow. You can choose the desired parameter when you configure a Search activity. • **Object from workflow data context**. Search for direct reports of the object that will be selected by the Search activity on the basis of the data found in the workflow environment at the time of executing the workflow. When configuring a Search activity, you can specify which object you want the activity to select at workflow run time. • **Object identified by DN-value rule expression**. Search for direct reports of the object whose Distinguished Name (DN) is specified by the string value of a certain rule expression. By using a rule expression you can compose a string value based on properties of various objects found in the workflow environment at the time of executing the workflow. You can create the desired rule expression when you configure a Search activity.
Search within the object's attribute (ASQ search).	• **Fixed object in directory**. Search in a certain attribute of the given object. You can select the desired object in Active Directory when you configure a Search activity. • **Workflow target object**. Search in a certain attribute of the target object of the request that started the workflow. • **Object from workflow data context**. Search in a certain attribute of the object that will be selected by the

Search scenario	"Find-in" options available
	Search activity on the basis of the data found in the workflow environment at the time of executing the workflow. When configuring a Search activity, you can specify which object you want the activity to select at workflow run time.

4. Select the appropriate option to further define your search scenario.

   If you chose to search in an Organizational Unit or container, then, under **When searching the Organizational Unit or container**, select one these options:

- **Retrieve only immediate child objects of the Organizational Unit or container**. Restricts the search to objects for which the given OU or container is the immediate parent in Active Directory.
- **Retrieve any objects held in the Organizational Unit or container**. Search in the entire directory tree rooted in the given OU or container.

  If you chose to search for resources managed or owned by a given user or group, then, under **When searching for managed resources**, select any combination of these options:

- **Retrieve objects managed by the user or group (primary owner)**. Search for objects that have the given user or group specified in the "Managed By" property.
- **Retrieve objects for which the user or group is a secondary owner**. Search for objects that have the given user or group specified in the "Secondary Owners" property.
- **Retrieve objects managed or owned due to membership in groups (indirect ownership)**. Search for objects for which the given user or group is a direct or indirect member of the group specified in the "Managed By" or "Secondary Owners" property.

  If you chose to search for members of a given group, then, under **When searching the group for its members**, select any combination of these options:

- **Also retrieve indirect members**. Have your search results include indirect members of the given group. With this option, the activity searches not only for objects that are directly added to the group (direct members) but also for indirect members-objects that belong to the group because of their membership in other groups which are direct or indirect members of the given group.
- **Also retrieve pending members**. Have your search results include objects that are scheduled to be added to the group by using the "Temporal Group Memberships" capability of Active Roles.

  If you chose to perform an ASQ search, then click in the **Search within this attribute** box to select the attribute for the ASQ search. This must be an attribute that stores Distinguished Names, such as the "Member Of" or "Managed By" attribute. The search is performed against the objects that are identified by the Distinguished Names found in that attribute. For example, a search within the

"Member Of" attribute of a user account looks for groups in which the user is a member.

5. Click in the **Search options** box to restrict your search to objects with particular characteristics. The available search options are specific to the object type you chose to search for.

   If you chose to search for users:

- Click the hyperlink under **Retrieve only these Exchange recipients**. to restrict your search to Microsoft Exchange mailbox-enabled users or Microsoft Exchange mail-enabled users.
- Click the hyperlink under **Retrieve only inactive user accounts**. to restrict your search to user accounts that meet certain inactivity conditions. In the dialog box that opens, you can choose the inactivity conditions as appropriate.
- Click the hyperlink under **Retrieve only expiring user accounts** to restrict your search to user accounts that will expire within a certain number of days. In the dialog box that opens, you can set the number of days you want.

   If you chose to search for contacts:

- Click the hyperlink under **Retrieve only these Exchange recipients**. to restrict your search to Microsoft Exchange mail-enabled contacts.

   If you chose to search for groups:

- Click the hyperlink under **Retrieve only these Exchange recipients**. to restrict your search to Microsoft Exchange mail-enabled groups.
- Click the hyperlink under **Retrieve only these group types**. to restrict your search to groups that meet certain conditions, such as groups of certain type and scope, empty groups, deprovisioned groups, or groups controlled by Active Roles. In the dialog box that opens, you can choose the conditions for groups as appropriate.

   If you chose to search for computers:

- Click the hyperlink under **Retrieve computers in this role**. to restrict your search to workstations or servers, or domain controllers.
- Click the hyperlink under **Retrieve only inactive computer accounts**. to restrict your search to computer accounts that meet certain inactivity conditions. In the dialog box that opens, you can choose the inactivity conditions as appropriate.

   If you chose to search for printers:

- Click hyperlinks under **Retrieve only printers with these features**. to restrict your search to printers with certain features, such as the printer model, paper size, print resolution, print speed, and other capabilities including the ability to print double-sided, the ability to print multiple colors, and the ability to staple. In the dialog box that opens, you can choose the printer features as appropriate.

   If you chose to search for Exchange recipients:

- Click the hyperlink under **Retrieve only these Exchange recipients**. to restrict your search to recipients that meet certain conditions, such as users with Exchange

mailbox, users with external e-mail addresses, mail-enabled groups, contacts with external e-mail addresses, mail-enabled Public Folders, Query-based Distribution Groups, room mailboxes, equipment mailboxes, linked mailboxes, or shared mailboxes. In the dialog box that opens, you can choose the conditions for Exchange recipients as appropriate.

- Click the hyperlink under **Retrieve mailboxes matching this storage filter**. to restrict your search to mailbox hosted on a certain mailbox server or held in a certain mailbox database. In the dialog box that opens, you can choose the desired server or database.

  If chose to search for inactive accounts, click a hyperlink under **Retrieve these account types** or **Retrieve accounts that meet any of these conditions**, and then, in the dialog box that opens, view or change the following search options specific to inactive accounts:

- Under **Retrieve these account types**, select the appropriate option depending on whether you want to search for inactive user accounts only, inactive computer accounts only, or both user and computer accounts that are inactive.
- Under **Retrieve accounts that meet any of the selected conditions**, choose and configure the account inactivity conditions. Accounts that meet any of the conditions you choose will be considered inactive. The following condition options are available:
  - **Account has not logged on in the past** <*number*> **days** This option allows you to specify the period, in days, that an account is not used to log on, after which the account is considered inactive. The search retrieves a given account if no successful logons to that account have occurred for more days than specified by this option.

    The search activity uses the lastLogonTimeStamp attribute to determine the last time that a given user or computer successfully logged on. Active Directory updates that attribute only periodically, rather than every time that a user or computer logs on. Normally, the period of update is 14 days. This means that the lastLogonTimeStamp value could be off by as much as 14 days, so the true last logon time is later than lastLogonTimeStamp. Hence, it is advisable to choose the logon inactivity period of more than 14 days.
  - **Account's password has not changed in the past** <*number*> **days** This option allows you to specify the password age, in days, after which an account is considered inactive. The search retrieves a given account if the password of the account remains unchanged for more days than specified by this option.
  - **Account expired more than** <*number*> **days before the current date** This option allows you to specify the number of days after which an expired account is considered inactive. The search retrieves a given account if the account remains in the expired state for more days than specified by this option.

6. Optionally, configure a filter to further refine your search. See instructions that follow.

Configuring a search filter

The search filter option allows you to refine your search in order to locate directory objects based on the properties (attributes) of the objects. You can use a search filter to look for specific values in the object properties, thereby ensuring that the search results contain only the objects with the desired properties.

A search filter is composed of conditions combined using "And" or "Or" logic. Each condition is a certain statement that specifies the criteria the activity should use to determine whether a given object is to be included in the search results. The workflow designer provides a condition builder for configuring filter conditions, located in the **Search options** box on the **Scope and filter** tab in the **"Search" Activity Properties** dialog box.

When you configure a search filter, you need to add at least one condition. By default, a single, implied condition group is created when you add a filter condition. You can create additional condition groups to group a set of conditions and nest grouped conditions within other condition groups.

A condition group contains one or more conditions connected by the same logical operator. By grouping conditions, you specify that those conditions should be evaluated as a single unit. The effect is the same as if you put parentheses around an expression in a mathematical equation or logic statement.

To add a condition to a condition group

- In the **Search options** box, under **Filter**, click the name of the condition group and then click **Insert condition**.

  Click the plus sign (+) next to the name of the condition group.

You can remove a condition, if needed, by clicking the **Delete condition** button labeled **X** on the right side of the list item representing the condition in the **Conditions** box.

To add a condition group into another condition group

- In the **Search options** box, under **Filter**, click the name of the condition group, point to **Insert condition group**, and then click an option to specify the logical operator:
  - **AND group**. The condition group evaluates to TRUE if all conditions in the group are TRUE.
  - **OR group**. The condition group evaluates to TRUE if any condition in the group is TRUE.
  - **NOT AND group**. The condition group evaluates to TRUE if any condition in the group evaluates to FALSE.
  - **NOT OR group**. The condition group evaluates to TRUE if all conditions in the group evaluate to FALSE.

By default, AND is the logical operator between the conditions in a condition group. It is possible to change the logical operator by converting the condition group to a different group type: Click the name of the group, point to **Convert condition group to**, and then click the option appropriate to the desired logical operator.

You can remove an entire condition group, if needed, by clicking the name of the group and then clicking **Delete condition group**.

Once you have added a condition to a condition group, you can use the following steps to configure the condition.

To configure a condition

1. Click **Configure condition to evaluate**, and then choose the property you want the condition to evaluate.
2. Click the current comparison operator, if needed, and then click the operator you want the condition to use.

   By default, a condition is configured to use the **equals** operator. The list of operators that are available depends upon the property you select in Step 1.
3. Click **Define value to compare to**, and then choose an option to specify the desired comparison value. The following options are available:

Table 61: Search filter options

Option	Description
Text string	A literal string of characters. You can type the desired string when you configure a filter condition.
Property of workflow target object	The value of a certain property of the target object of the request that started the workflow. You can select the desired property when you configure a filter condition. Normally, this should be a string-value property.
Property of workflow initiator	The value of a certain property of the user whose request started the workflow. You can select the desired property when you configure a filter condition. Normally, this should be a string-value property.
Changed value of workflow target object property	The value that is requested to be assigned to a certain property of the target object of the request that started the workflow, which represents the requested change to the property of the target object. You can select the desired property when you configure a filter condition. Normally, this should be a string-value property.
Property of object from workflow data context	The value of a certain property of the object that will be selected by the Search activity on the basis of the data found in the workflow environment at the time of executing the workflow. When you configure a filter condition in a Search activity, you can choose the desired property and specify which object you want the activity to select upon evaluating the condition at workflow run time.
Value generated by rule expression	The string value of a certain rule expression. By using a rule expression you can compose a string value based on

Option	Description
	properties of various objects found in the workflow environment at the time of executing the workflow.
Fixed object in directory	A certain object, such as a user, group, or computer. You can select the desired object in Active Directory when you configure a filter condition. This comparison value is applicable to filter conditions for DN-value properties.
Object from workflow data context	The object that will be selected by the Search activity on the basis of the data found in the workflow environment at the time of executing the workflow. When you configure a filter condition in a Search activity, you can specify which object you want the activity to select upon evaluating the condition at workflow run time. This comparison value is applicable to filter conditions for DN-value properties.
Object identified by DN-value rule expression	The object whose Distinguished Name (DN) is specified by the string value of a certain rule expression. By using a rule expression, you can compose a string value based on properties of various objects found in the workflow environment at the time of executing the workflow. You can create the desired rule expression when you configure a filter condition. This comparison value is applicable to filter conditions for DN-value properties.
Object identified by workflow parameter	The object specified by the value of a certain parameter. You can choose the desired parameter when you configure a filter condition. This comparison value is applicable to filter conditions for DN-value properties.
Workflow initiator object	The user account of the user whose request started the workflow. This comparison value is applicable to filter conditions for DN-value properties.
Workflow target object	The target object of the request that started the workflow. This comparison value is applicable to filter conditions for DN-value properties.
Fixed date and time	A literal date and time value. You can choose the desired date and time when you configure a filter condition. This comparison value is applicable to filter conditions for Date/Time-value properties.
Workflow date and time	A certain point in time relative to the date and time of the Search activity run. You have the option to specify a date that occurs a particular number of days before or after the Search activity run. This comparison value is applicable to filter conditions for Date/Time-value properties.

Option	Description
True	The literal Boolean value of True.
False	The literal Boolean value of False.
Value generated by script	The value returned by a certain script function. You can choose the desired script function when you configure a filter condition. The Search activity will execute that script function upon evaluating the condition at workflow run time.
Workflow parameter value	The value of a certain workflow parameter. You can choose the desired parameter when you configure a filter condition.

Configure notification

You can configure a Search activity to subscribe recipients to the notifications of the following events:

- **Activity completed successfully**. When configured to notify of this event, the activity causes Active Roles to send a notification e-mail if no significant errors occurred during execution of this activity.
- **Activity encountered an error**. When configured to notify of this event, the activity causes Active Roles to send a notification e-mail if any significant errors occurred during execution of this activity.

To configure notification for a Search activity

1. In the process diagram, right-click the name of the Search activity and click **Properties**.
2. Go to the **Notification** tab in the **"Search" Activity Properties** dialog box, and use the steps for Configuring a Notification activity to configure the notification settings.

The notification settings specify the event to notify of, and notification recipients. When executed by the workflow, the activity prepares a notification message appropriate to the specified event. Active Roles retains the message prepared by the activity, and sends the message to the specified recipients upon occurrence of that event.

Configure error handling

When configuring a Search activity, you can configure error handling to suppress errors encountered by that Search activity and all activities included in that Search activity.

To configure error handling for a Search activity

1. In the process diagram, right-click the name of the Search activity and click **Properties**.
2. Go to the **Error handling** tab in the **"Search" Activity Properties** dialog box, and select or clear the **Continue workflow even if this activity encounters an error** check box on that tab.

If the **Continue workflow even if this activity encounters an error** check box is not selected (default setting), then an error condition encountered by the activity causes Active Roles to terminate the workflow. If you select this check box, the workflow continues regardless of whether or not the Search activity or any activity within the Search activity encounters an error condition.

Configure "run as" options

By default, the Search activity is executed under the user account specified by the "run as" setting in the workflow options and start conditions. This could be the service account of the Active Roles Administration Service or the account of the user who caused the workflow to start. You can configure the activity to override the default "run as" setting.

To configure "run as" options for a Search activity

1. In the process diagram, right-click the name of the Search activity and click **Properties**.
2. Click the **"Run as" options** hyperlink at the bottom of the **"Search" Activity Properties** dialog box.
3. To override the default "run as" setting for this activity, select the **Run this activity under** check box, and then choose the account under which you want the activity to run:
   - Click **The service account of Active Roles** if you want this activity to run under the service account of the Active Roles Administration Service.
   - Click **The account of the user who started the workflow** if you want this activity to run under the account of the user who caused the workflow to start. Depending on the type of the workflow, this is either the user who requested the operation that started the workflow or the user who started the workflow on demand.

The account under which the activity is running determines the access rights of the activity in the directory.

Configure additional settings

By using additional settings, you can configure a Search activity to stop the search if the number of the objects that meet the search conditions exceeds a certain threshold. It is also possible to modify behavior of a Search activity using so-called request controls to

pass additional information to Active Roles on how to process operation requests created by that activity.

To configure additional settings for a Search activity

1. In the process diagram, right-click the name of the Search activity and click **Properties**.
2. Click the **Additional settings** hyperlink at the bottom of the **"Search" Activity Properties** dialog box.
3. To have the Search activity stop the search if the number of the objects found by the search exceeds a certain threshold, select the **Terminate the search activity if the search returns more than check** box, and specify the maximum number of objects the activity is allowed to return when performing a search.
4. Add, change, or remove request controls in the **Include or exclude these controls from the activity operation requests** list.

   To add or change a control, click **Add** or **Change**, and then, in the dialog box that opens, specify the name and, if applicable, the value of the control. If you want the activity to add the control to the requests, click **Include this control in the activity operation requests**. If you want to ensure that the control never occurs in the requests created by this activity, click **Exclude this control from the activity operation requests**.

Request controls are certain pieces of data in an operation request that can be used to pass additional information to Active Roles on how to process the request. Request controls are optional. If no request controls are added to a request, then Active Roles determines how to process the request based solely on the type of the request. For information about request controls, see the Active Roles SDK documentation.

Configuring CRUD activities

Active Roles offers a number of workflow activities, collectively referred to as CRUD activities, intended to create new objects, and modify or delete existing objects in Active Directory. The CRUD abbreviation designates the key operations that can be performed by using these activities: Create, Read, Update, Delete. The topics in this section provide instructions on how to configure the following CDUD activities:

- "Create" activity Creates an object, such as a user, group, or computer, in Active Directory.
- "Update" activity Changes properties of an object, such as a user, group, or computer, in Active Directory.
- "Add to group" activity Adds an object, such as a user, group or computer, to specified groups in Active Directory.
- "Remove from group" activity Removes an object, such as a user, group, or computer, from specified groups in Active Directory.

- "Move" activity Moves an object, such as a user, group or computer, to a specified container in Active Directory.
- "Deprovision" activity Deprovisions a user or group by applying the Active Roles deprovisioning policy.
- "Undo deprovision" activity Restores a user or group that was deprovisioned by using Active Roles.
- "Delete" activity Deletes an object, such as a user, group, or computer, in Active Directory.

The following topics in this section provide the steps for configuring the settings that are common to CRUD activities:

- Configuring notification Active Roles can notify via e-mail about whether or not the activity has encountered an error condition at run time.
- Configuring error handling. Determines whether or not the workflow is allowed to continue if the activity has encountered an error condition at run time.
- Configuring "run-as" options Determines the user account under which to run the activity.
- Configuring additional settings Some advanced configuration options that allow you to adjust the processing of the operation requested by the activity.

"Create" activity

When you configure a Create activity, you can specify the Organizational Unit or container where you want the activity to create objects, choose the object type and name, and specify how you want the activity to populate the properties of the newly created objects. Additional options are available such as notification, error handling, and "run as" options.

To configure a Create activity

1. In the Active Roles console tree, expand **Configuration | Policies | Workflow**, and select the workflow containing the Create activity you want to configure.

   This opens the Workflow Designer window in the details pane, representing the workflow definition as a process diagram.

2. In the process diagram, right-click the name of the activity and click **Properties**.
3. Go to the **Container** tab in the **"Create" Activity Properties** dialog box.
4. Click in the **Activity creates the object in this container** box to specify the Organizational Unit (OU) or container in which you want the activity to create an object. The following options are available:
   - **Fixed container in directory**. With this option, the activity creates an object in the given OU or container. You can select the desired OU or container in Active Directory when you configure the activity.

- **Parent OU of workflow target object**. With this option, the activity creates an object in the OU that holds the target object of the request that started the workflow. This option is unavailable in case of an automation workflow.
- **Activity target object**. With this option, the activity creates an object in the OU or container created or otherwise processed by another CRUD activity at the time of executing the workflow. You can select the desired CRUD activity from the workflow definition when you configure the activity.
- **Object identified by workflow parameter**. With this option, the activity creates an object in the OU or container specified by the value of a certain parameter of the workflow. You can choose the desired parameter from the workflow definition when you configure the activity.
- **Object from workflow data context**. With this option, the activity creates an object in the OU or container that is selected by the Create activity on the basis of the data found in the workflow environment at the time of executing the workflow. When configuring a Create activity, you can specify which OU or container you want the activity to select at workflow run time.
- **Object identified by DN-value rule expression**. With this option, the activity creates an object in the OU or container whose Distinguished Name (DN) is specified by the string value of a certain rule expression. By using a rule expression you can compose a string value based on properties of various objects found in the workflow environment at the time of executing the workflow. You can create the desired rule expression when you configure the activity.

5. Go to the **Object name** tab in the **"Create" Activity Properties** dialog box.
6. Click in the **Object type** box, and select the type of the objects you want the activity to create, such as User, Group, Computer, and so forth.
7. Click in the **Object name** box to specify how you want the activity to generate the object name when creating an object. The following options are available:
   - **Text string**. With this option, the activity uses the given string of characters as the name of the object. You can specify the desired string when you configure the activity.
   - **Name of workflow target object**. With this option, the activity uses the name of the target object of the request that started the workflow. This option is unavailable in case of an automation workflow.
   - **Name of workflow target object, followed by text string**. With this option, the activity uses a certain text string prefixed with the name of the target object of the request that started the workflow. You can specify the desired text string when you configure the activity. This option is unavailable in case of an automation workflow.
   - **Workflow parameter value**. With this option, the activity uses the name specified by the string value of a certain parameter of the workflow. You can choose the desired parameter from the workflow definition when you configure the activity.

- **Property of object from workflow data context**. With this option, the activity uses the name identified by the value of a certain property of the object that will be selected by the Create activity on the basis of the data found in the workflow environment at the time of executing the workflow. When you configure a Create activity, you can choose the desired property and specify which object you want the activity to select at workflow run time.
- **Value generated by rule expression**. With this option, the activity uses the name identified by the string value of a certain rule expression. By using a rule expression you can compose a string value based on properties of various objects found in the workflow environment at the time of executing the workflow. You can create the desired rule expression when you configure the activity.

8. Go to the **Object properties** tab in the **"Create" Activity Properties** dialog box.
9. Configure the list of the properties you want the activity to populate:
   - To add a property to the list, click **Add property**, and then select the name of the desired property.
   - To remove a property from the list, click the **Delete** button labeled **X** on the right side of the list item representing that property.
10. After you have added a property to the list, click in the **Value** filed to specify the value you want the activity to assign to that property of the newly created object. The following options are available:
    - **Text string**. With this option, the activity assigns the given string of characters to the property. You can specify the desired string when you configure the activity.
    - **Property of workflow target object**. With this option, the activity assigns the value of a certain property of the target object of the request that started the workflow. You can select the desired property when you configure the activity. This option is unavailable in case of an automation activity.
    - **Property of workflow initiator**. With this option, the activity assigns the value of a certain property of the user whose request started the workflow. You can select the desired property when you configure the activity.
    - **Changed value of workflow target object property**. With this option, the activity assigns the value that is requested to be assigned to a certain property of the workflow target object. You can select the desired property when you configure the activity. This option is unavailable in case of an automation activity.
    - **Workflow parameter value**. This option causes the activity to populate the property with the value of a certain parameter of the workflow. You can choose the desired parameter from the workflow definition when you configure the activity.
    - **Property of object from workflow data context**. This option causes the activity to populate the property with the value of a certain property of the object that will be selected by the Create activity on the basis of the data found

in the workflow environment at the time of executing the workflow. When you configure a Create activity, you can choose the desired property and specify which object you want the activity to select at workflow run time.

- **Value generated by rule expression**. This option causes the activity to populate the property with the string value of a certain rule expression. By using a rule expression you can compose a string value based on properties of various objects found in the workflow environment at the time of executing the workflow. You can create the desired rule expression when you configure the activity.

11. View or change notification settings. For instructions, see Configuring notification.
12. View or change error handling settings. For instructions, see Configuring error handling.
13. View or change "run as" options. For instructions, see Configuring "run-as" options.
14. View or change advanced settings. For instructions, see Configuring additional settings.

"Update" activity

When you configure an Update activity, you can specify the rules for selecting the object whose properties you want the activity to change, and define how you want the activity to change the properties of the object. Additional options are available such as notification, error handling, and "run as" options.

To configure an Update activity

1. In the Active Roles console tree, expand **Configuration | Policies | Workflow**, and select the workflow containing the Update activity you want to configure.

   This opens the Workflow Designer window in the details pane, representing the workflow definition as a process diagram.

2. In the process diagram, right-click the name of the activity and click **Properties**.
3. Go to the **Activity target** tab in the **"Update" Activity Properties** dialog box.
4. Click in the **Activity performs the operation on this object** box to specify the object whose properties you want the activity to change. This object is referred to as activity target. You can choose from the following options to specify the activity target:
   - **Fixed object in directory**. The activity target is the given object. You can select the desired object in Active Directory when you configure the activity.
   - **Object identified by workflow parameter**. The activity target is the object specified by the value of a certain parameter of the workflow. You can choose the desired parameter from the workflow definition when you configure the activity.

- **Object from workflow data context**. The activity target is selected by the activity on the basis of the data found in the workflow environment at the time of executing the workflow. When configuring an Update activity, you can specify which object you want the activity to select at workflow run time.
- **Object identified by DN-value rule expression**. The Distinguished Name (DN) of the activity target is specified by the string value of a certain rule expression. By using a rule expression you can compose a string value based on properties of various objects found in the workflow environment at the time of executing the workflow. You can create the desired rule expression when you configure the activity.

5. Go to the **Target properties** tab in the **"Update" Activity Properties** dialog box.
6. Configure the list of the properties you want the activity to modify:
   - To add a property to the list, click **Add property**, and then select the name of the desired property.
   - To remove a property from the list, click the **Delete** button labeled **X** on the right side of the list item representing that property.
7. After you have added a property, click in the **Action** field to specify the type of the changes you want the activity to make to that property:
   - Click **Set** to have the activity assign a new value to the property.
   - Click **Clear** to have the activity remove the property from the object.
   - In case of a multi-value property, click **Add value** or **Remove value** for the activity to add or remove the value of the property.
8. If an action other than **Clear** is selected in the **Action** field, click in the **Value** filed to specify the property value you want the activity to set, add or remove. The following options are available:
   - **Text string**. Use the given string of characters as the value of the property. You can specify the desired string when you configure the activity.
   - **Property of workflow target object**. Use the value of a certain property of the target object of the request that started the workflow. You can select the desired property when you configure an Update activity. This option is unavailable in case of an automation workflow.
   - **Property of workflow initiator**. Use the value of a certain property of the user whose request started the workflow. You can select the desired property when you configure the activity.
   - **Changed value of workflow target object property**. Use the value that is requested to be assigned to a certain property of the workflow target object. You can select the desired property when you configure the activity. This option is unavailable in case of an automation workflow.
   - **Workflow parameter value**. Use the value of a certain parameter of the workflow. You can choose the desired parameter from the workflow definition when you configure the activity.

- **Property of object from workflow data context**. Use the value of a certain property of the object that will be selected by the Update activity on the basis of the data found in the workflow environment at the time of executing the workflow. When you configure the Update activity, you can choose the desired property and specify which object you want the activity to select at workflow run time.
- **Value generated by rule expression**. Use the string value of a certain rule expression. By using a rule expression you can compose a string value based on properties of various objects found in the workflow environment at the time of executing the workflow. You can create the desired rule expression when you configure the activity.

9. View or change notification settings. For instructions, see Configuring notification.
10. View or change error handling settings. For instructions, see Configuring error handling.
11. View or change "run as" options. For instructions, see Configuring "run-as" options.
12. View or change advanced settings. For instructions, see Configuring additional settings.

"Add to group" activity

When you configure an "Add to group" activity, you can specify the rules for selecting the object you want the activity to add to groups, and define the groups to which you want the activity to add the object. Additional options are available such as notification, error handling, and "run as" options.

To configure an "Add to group" activity

1. In the Active Roles console tree, expand **Configuration** | **Policies** | **Workflow**, and select the workflow containing the "Add to group" activity you want to configure.

   This opens the Workflow Designer window in the details pane, representing the workflow definition as a process diagram.

2. In the process diagram, right-click the name of the activity and click **Properties**.
3. Go to the **Activity target** tab in the **"Add to Group" Activity Properties** dialog box.
4. Click in the **Activity performs the operation on this object** box to specify the object you want the activity to add to groups. This object is referred to as activity target. You can choose from the following options to specify the activity target:
   - **Fixed object in directory**. The activity target is the given object. You can select the desired object in Active Directory when you configure the "Add to group" activity.
   - **Object identified by workflow parameter**. The activity target is the object specified by the value of a certain parameter of the workflow. You can choose the desired parameter from the workflow definition when you configure

the "Add to group" activity.

- **Object from workflow data context**. The activity target is selected by the activity on the basis of the data found in the workflow environment at the time of executing the workflow. When configuring the "Add to group" activity, you can specify which object you want the activity to select at workflow run time.
- **Object identified by DN-value rule expression**. The Distinguished Name (DN) of the activity target is specified by the string value of a certain rule expression. By using a rule expression you can compose a string value based on properties of various objects found in the workflow environment at the time of executing the workflow. You can create the desired rule expression when, you configure the "Add to group" activity.

5. Go to the **Groups** tab in the **"Add to Group" Activity Properties** dialog box.
6. Configure the list of groups to which you want the activity to add the target object.

   To add a group to the list, click **Add group**, and then choose from the following options:

   - **Fixed group in directory**. You can select the desired group in Active Directory when you configure the "Add to group" activity. A unique identifier of the group is saved in the configuration of the activity. The activity uses that identifier to select the group when calculating the list of groups at workflow execution time.
   - **Object from workflow data context**. The group is selected by the activity on the basis of the data found in the workflow environment at the time of executing the workflow. When configuring the "Add to group" activity, you can specify which group you want the activity to select at workflow execution time.
   - **Object identified by DN-value rule expression**. The Distinguished Name (DN) of the group is specified by the string value of a certain rule expression. By using a rule expression, you can compose a string value based on properties of various objects found in the workflow environment at the time of executing the workflow. You can create the desired rule expression when you configure the "Add to group" activity.

   To remove a group from the list, click the **Delete** button labeled **X** on the right side of the list item representing that group.

7. View or change notification settings. For instructions, see Configuring notification.
8. View or change error handling settings. For instructions, see Configuring error handling.
9. View or change "run as" options. For instructions, see Configuring "run-as" options.
10. View or change advanced settings. For instructions, see Configuring additional settings.

“Remove from group” activity

When you configure a “Remove from group” activity, you can specify the rules for selecting the object you want the activity to remove from groups, and define the groups from which you want the activity to remove the object. Additional options are available, such as notification, error handling, and “run as” options.

To configure a “Remove from group” activity

1. In the Active Roles console tree, expand **Configuration | Policies | Workflow**, and select the workflow containing the “Remove from group” activity you want to configure.

   This opens the Workflow Designer window in the details pane, representing the workflow definition as a process diagram.

2. In the process diagram, right-click the name of the activity and click **Properties**.
3. Go to the **Activity target** tab in the **“Remove from Group” Activity Properties** dialog box.
4. Click in the **Activity performs the operation on this object** box to specify the object you want the activity to remove from groups. This object is referred to as the activity target. You can choose from the following options to specify the activity target:
   - **Fixed object in directory**. The activity target is the given object. You can select the desired object in Active Directory when you configure the “Remove from group” activity.
   - **Object identified by workflow parameter**. The activity target is the object specified by the value of a certain parameter of the workflow. You can choose the desired parameter from the workflow definition when you configure the “Remove from group” activity.
   - **Object from workflow data context**. The activity target is selected by the activity on the basis of the data found in the workflow environment at the time of executing the workflow. When configuring the “Remove from group” activity, you can specify which object you want the activity to select at workflow run time.
   - **Object identified by DN-value rule expression**. The Distinguished Name (DN) of the activity target is specified by the string value of a certain rule expression. By using a rule expression, you can compose a string value based on properties of various objects found in the workflow environment at the time of executing the workflow. You can create the desired rule expression when you configure the “Remove from group” activity.
5. Go to the **Groups** tab in the **“Remove from Group” Activity Properties** dialog box.
6. Choose from these options:
   - **Remove the object from all groups**. This option configures the activity to remove the object from all groups in Active Directory. Note that an object

cannot be removed from its primary group, so the activity will remove the object from all groups except the object's primary group.

- **Remove the object from these groups**. This option lets you list the groups from which you want the activity to remove the object. For each of the groups in the list (with the exception of the object's primary group), the activity will remove the object from that group.

7. If you chose the option **Remove the object from these groups**, configure the list of groups from which you want the activity to remove the target object. To add a group to the list, click **Add group**, and then choose from the following options:
   - **Fixed group in directory**. You can select the desired group in Active Directory when you configure the "Remove from group" activity. A unique identifier of the group is saved in the configuration of the activity. The activity uses that identifier to select the group when calculating the list of groups at workflow execution time.
   - **Object from workflow data context**. The group is selected by the activity on the basis of the data found in the workflow environment at the time of executing the workflow. When configuring the "Remove from group" activity, you can specify which group you want the activity to select at workflow execution time.
   - **Object identified by DN-value rule expression**. The Distinguished Name (DN) of the group is specified by the string value of a certain rule expression. By using a rule expression, you can compose a string value based on properties of various objects found in the workflow environment at the time of executing the workflow. You can create the desired rule expression when you configure the "Remove from group" activity.

   To remove a group from the list, click the **Delete** button labeled **X** on the right side of the list item representing that group.
8. View or change notification settings. For instructions, see Configuring notification.
9. View or change error handling settings. For instructions, see Configuring error handling.
10. View or change "run as" options. For instructions, see Configuring "run-as" options.
11. View or change advanced settings. For instructions, see Configuring additional settings.

"Move" activity

When you configure a Move activity, you can specify the rules for selecting the object you want the activity to move, and specify the container to move the object to (destination container). Additional options are available such as notification, error handling, and "run as" options.

To configure a Move activity

1. In the Active Roles console tree, expand **Configuration | Policies | Workflow**, and select the workflow containing the Move activity you want to configure.

   This opens the Workflow Designer window in the details pane, representing the workflow definition as a process diagram.

2. In the process diagram, right-click the name of the activity and click **Properties**.
3. Go to the **Activity target** tab in the **"Move" Activity Properties** dialog box.
4. Click in the **Activity performs the operation on this object** box to specify the object you want the activity to move. This object is referred to as activity target. You can choose from the following options to specify the activity target:
   - **Fixed object in directory**. The activity target is the given object. You can select the desired object in Active Directory when you configure the Move activity.
   - **Object identified by workflow parameter**. The activity target is the object specified by the value of a certain parameter of the workflow. You can choose the desired parameter from the workflow definition when you configure the Move activity.
   - **Object from workflow data context**. The activity target will be selected by the activity on the basis of the data found in the workflow environment at the time of executing the workflow. You can specify which object you want the activity to select at workflow run time.
   - **Object identified by DN-value rule expression**. The Distinguished Name (DN) of the activity target is specified by the string value of a certain rule expression. By using a rule expression you can compose a string value based on properties of various objects found in the workflow environment at the time of executing the workflow. You can create the desired rule expression when you configure the Move activity.
5. Go to the **Destination container** tab in the **"Move" Activity Properties** dialog box.
6. Click in the **Activity moves the object to this container** box to specify the container to which you want the activity to move the target object. You can choose from the following options:
   - **Fixed container in directory**. With this option, the activity moves the object to the given OU or container. You can select the desired OU or container in Active Directory when you configure the Move activity.
   - **Parent OU of workflow target object**. With this option, the activity moves the object to the OU that holds the target object of the request that started the workflow. This option is unavailable in case of an automation workflow.
   - **Activity target object**. With this option, the activity moves the object to the OU or container created or otherwise processed by another CRUD activity at the time of executing the workflow. You can select the desired CRUD activity from the workflow definition when you configure the Move activity.

- **Object identified by workflow parameter**. With this option, the activity moves the object to the OU or container specified by the value of a certain parameter of the workflow. You can choose the desired parameter from the workflow definition when you configure the Move activity.
- **Object from workflow data context**. With this option, the activity moves the object to the OU or container that is selected by the Move activity on the basis of the data found in the workflow environment at the time of executing the workflow. When configuring a Move activity, you can specify which OU or container you want the activity to select at workflow run time.
- **Object identified by DN-value rule expression**. With this option, the activity moves the object to the OU or container whose Distinguished Name (DN) is specified by the string value of a certain rule expression. By using a rule expression, you can compose a string value based on properties of various objects found in the workflow environment at the time of executing the workflow. You can create the desired rule expression when you configure the Move activity.

7. View or change notification settings. For instructions, see Configuring notification.
8. View or change error handling settings. For instructions, see Configuring error handling.
9. View or change "run as" options. For instructions, see Configuring "run-as" options.
10. View or change advanced settings. For instructions, see Configuring additional settings.

"Deprovision" activity

A **Deprovision** activity is intended to apply the Active Roles deprovisioning policies to a particular user or group. This activity causes Active Roles to perform all the tasks prescribed by the deprovisioning policies, thereby deprovisioning the user or group.

When you configure a **Deprovision** activity, you can specify the rules for selecting the user or group you want the activity to deprovision. Additional options are available such as notification, error handling, and "run as" options.

To configure a Deprovision activity

1. In the Active Roles console tree, expand **Configuration | Policies | Workflow**, and select the workflow containing the **Deprovision** activity you want to configure.

   This opens the Workflow Designer window in the details pane, representing the workflow definition as a process diagram.

2. In the process diagram, right-click the name of the activity and click **Properties**.
3. Go to the **Activity target** tab in the **"Deprovision" Activity Properties** dialog box.
4. Click in the **Activity performs the operation on this object** box to specify the user or group you want the activity to deprovision. This object is referred to as

activity target. You can choose from the following options to specify the activity target:

- **Fixed object in directory**. The activity target is the given object. You can select the desired object in Active Directory when you configure the **Deprovision** activity.
- **Object identified by workflow parameter**. The activity target is the object specified by the value of a certain parameter of the workflow. You can choose the desired parameter from the workflow definition when you configure the **Deprovision** activity.
- **Object from workflow data context**. The activity target is selected by the activity on the basis of the data found in the workflow environment at the time of executing the workflow. When configuring the **Deprovision** activity, you can specify which object you want the activity to select at workflow run time.
- **Object identified by DN-value rule expression**. The Distinguished Name (DN) of the activity target is specified by the string value of a certain rule expression. By using a rule expression, you can compose a string value based on properties of various objects found in the workflow environment at the time of executing the workflow. You can create the desired rule expression when you configure the **Deprovision** activity.

5. View or change notification settings. For instructions, see Configuring notification.
6. View or change error handling settings. For instructions, see Configuring error handling.
7. View or change "run as" options. For instructions, see Configuring “run-as” options.
8. View or change advanced settings. For instructions, see Configuring additional settings.

“Undo deprovision” activity

An “Undo deprovision” activity is intended to restore a particular user or group that was deprovisioned by using Active Roles. The activity causes Active Roles to roll back the changes that were made to the user or group object by applying the Active Roles deprovisioning policies. As a result, the object reverts to the state it was in before the deprovisioning-related changes where made.

When you configure an “Undo deprovision” activity, you can specify the rules for selecting the user or group you want the activity to restore. Additional options are available such as notification, error handling, and “run as” options.

To configure an “Undo deprovision” activity

1. In the Active Roles console tree, expand **Configuration** | **Policies** | **Workflow**, and select the workflow containing the “Undo deprovision” activity you want to configure.

   This opens the Workflow Designer window in the details pane, representing the workflow definition as a process diagram.

2. In the process diagram, right-click the name of the activity and click **Properties**.
3. Go to the **Activity target** tab in the **"Undo Deprovision" Activity Properties** dialog box.
4. Click in the **Activity performs the operation on this object** box to specify the user or group you want the activity to restore. This object is referred to as activity target. You can choose from the following options to specify the activity target:
   - **Fixed object in directory**. The activity target is the given object. You can select the desired object in Active Directory when you configure the "Undo deprovision" activity.
   - **Object identified by workflow parameter**. The activity target is the object specified by the value of a certain parameter of the workflow. You can choose the desired parameter from the workflow definition when you configure the "Undo deprovision" activity.
   - **Object from workflow data context**. The activity target is selected by the activity on the basis of the data found in the workflow environment at the time of executing the workflow. When configuring the "Undo deprovision" activity, you can specify which object you want the activity to select at workflow run time.
   - **Object identified by DN-value rule expression**. The Distinguished Name (DN) of the activity target is specified by the string value of a certain rule expression. By using a rule expression, you can compose a string value based on properties of various objects found in the workflow environment at the time of executing the workflow. You can create the desired rule expression when you configure the "Undo deprovision" activity.
5. View or change notification settings. For instructions, see Configuring notification.
6. View or change error handling settings. For instructions, see Configuring error handling.
7. View or change "run as" options. For instructions, see Configuring "run-as" options.
8. View or change advanced settings. For instructions, see Configuring additional settings.

"Delete" activity

When you configure a Delete activity, you can specify the rules for selecting the object you want the activity to delete in Active Directory. Additional options are available such as notification, error handling, and "run as" options.

To configure a Delete activity

1. In the Active Roles console tree, expand **Configuration | Policies | Workflow**, and select the workflow containing the Delete activity you want to configure.

   This opens the Workflow Designer window in the details pane, representing the workflow definition as a process diagram.

2. In the process diagram, right-click the name of the activity and click **Properties**.
3. Go to the **Activity target** tab in the **"Delete" Activity Properties** dialog box.
4. Click in the **Activity performs the operation on this object** box to specify the object you want the activity to delete. This object is referred to as activity target. You can choose from the following options to specify the activity target:
   - **Fixed object in directory**. The activity target is the given object. You can select the desired object in Active Directory when you configure the Delete activity.
   - **Object identified by workflow parameter**. The activity target is the object specified by the value of a certain parameter of the workflow. You can choose the desired parameter from the workflow definition when you configure the Delete activity.
   - **Object from workflow data context**. The activity target is selected by the activity on the basis of the data found in the workflow environment at the time of executing the workflow. When configuring the Delete activity, you can specify which object you want the activity to select at workflow run time.
   - **Object identified by DN-value rule expression**. The Distinguished Name (DN) of the activity target is specified by the string value of a certain rule expression. By using a rule expression, you can compose a string value based on properties of various objects found in the workflow environment at the time of executing the workflow. You can create the desired rule expression when you configure the Delete activity.
5. View or change notification settings. For instructions, see Configuring notification.
6. View or change error handling settings. For instructions, see Configuring error handling.
7. View or change "run as" options. For instructions, see Configuring "run-as" options.
8. View or change advanced settings. For instructions, see Configuring additional settings.

Configuring notification

You can configure a CRUD activity to subscribe recipients to the notifications of the following events:

- **Activity completed successfully** When configured to notify of this event, the activity causes Active Roles to send a notification e-mail if no significant errors occurred during execution of this activity.
- **Activity encountered an error** When configured to notify of this event, the activity causes Active Roles to send a notification e-mail if any significant errors occurred during execution of this activity.

To configure notification for a CRUD activity

1. In the process diagram, right-click the name of the activity and click **Properties**.
2. Go to the **Notification** tab in the **Properties** dialog box, and use the steps for Configuring a Notification activity to configure the notification settings.

The notification settings specify the event to notify of, and notification recipients. When executed by the workflow, the activity prepares a notification message appropriate to the specified event. Active Roles retains the message prepared by the activity, and sends the message to the specified recipients upon occurrence of that event.

Configuring error handling

When configuring a CRUD activity, you can use error handling to suppress errors encountered by that activity.

To configure error handling for a CRUD activity

1. In the process diagram, right-click the name of the activity and click **Properties**.
2. Go to the **Error handling** tab in the **Properties** dialog box, and select or clear the **Continue workflow even if this activity encounters an error** check box on that tab.

If the **Continue workflow even if this activity encounters an error** check box is not selected (default setting), then an error condition encountered by the activity causes Active Roles to terminate the workflow. If you select this check box, the workflow continues regardless of whether or not the activity encounters an error condition.

Configuring "run-as" options

By default, CRUD activities are executed under the user account specified by the "run as" setting in the workflow options and start conditions. This could be the service account of the Active Roles Administration Service or the account of the user who caused the workflow to start. You can configure the activity to override the default "run as" setting.

To configure "run as" options for a CRUD activity

1. In the process diagram, right-click the name of the activity and click **Properties**.
2. Click the **"Run as" options** hyperlink at the bottom of the **Properties** dialog box.
3. To override the default "run as" setting for this activity, select the **Run this activity under** check box, and then choose the account under which you want the activity to run:
    - Click **The service account of Active Roles** if you want this activity to run under the service account of the Active Roles Administration Service.

- Click **The account of the user who started the workflow** if you want this activity to run under the account of the user who caused the workflow to start. Depending on the type of the workflow, this is either the user who requested the operation that started the workflow or the user who started the workflow on demand.

The account under which the activity is running determines the access rights of the activity in the directory.

4. View or change the settings under the **Approval enforcement option** heading.

The **Approval enforcement option** settings determine whether to apply approval rules to the operation requested by the activity if the activity is executed under a privileged account, such as the Active Roles service account, an Active Roles Admin account, or the account of the user who is designated as an approver. The following settings are available:

- **Inherit from the workflow options and start conditions** Select this option if you want the activity to use the approval enforcement option selected in the workflow options and start conditions.
- **Use the following option for this activity** Click this option and then select or clear the **Enforce approval** check box if you want this activity to override the approval enforcement option selected in the workflow options and start conditions.

  When selected, the **Enforce approval** check box causes the approval rules to be applied, submitting the operation for approval regardless of the account under which the activity is executed. Otherwise, the operation requested by the activity bypasses approval rules if the activity is executed under the Active Roles service account, an Active Roles Admin account, or the account of the user who is designated as an approver, so the operation is not submitted for approval.

Configuring additional settings

By using additional settings, you can override the default operation reason text, and add so-called request controls to modify behavior of the activity.

To configure additional settings for a CRUD activity

1. In the process diagram, right-click the name of the activity and click **Properties**.
2. Click the **Additional settings** link at the bottom of the **Properties** dialog box.
3. In the **Additional Settings** dialog box, view or change the following options:
   - **Use this text instead of the original operation reason text**. If the operation requested by the CRUD activity is subject to approval, you can specify the operation reason text to be shown to the approver instead of the reason text specified in the operation request that started the workflow.
   - Select **Use this text instead of the original operation reason text**. check box and type the appropriate reason text to replace the original reason text. Select the **Use only if the operation reason is not originally specified** check box if you want the activity to use your reason text only if the

operation request that started the workflow does not have any reason text specified.

- **Allow the request created by this activity to start a new instance of the workflow containing this activity**. This check box should normally be cleared to prevent recurrent execution of the activity in the situation where the operation requested by that activity within a given workflow matches the start conditions of that same workflow. Selecting this check box may result in a loop of workflow instances executing the same activity again and again, and eventually would cause an overflow condition.
- **Exclude or include request controls from the activity operation request**. Request controls are certain pieces of data in an operation request that can be used to pass additional information to Active Roles on how to process the request. Request controls are optional. For information about request controls, see Active Roles SDK.

To add or change a control, click **Add** or **Change**, and then, in the dialog box that opens, specify the name and, if applicable, the value of the control. If you want the activity to add the control to the requests, click **Include this control in the activity operation requests**. If you want to ensure that the control never occurs in the requests created by this activity, click **Exclude this control from the activity operation requests**.

Configuring a Save Object Properties activity

When you configure a Save Object Properties activity, you can specify the rules for selecting the object whose properties you want the activity to save, and list the properties for the activity to save. Additional options are available, such as notification and error handling.

To configure a Save Object Properties activity

1. In the Active Roles console tree, expand **Configuration | Policies | Workflow**, and select the workflow containing the Save Object Properties activity you want to configure.

   This opens the Workflow Designer window in the details pane, representing the workflow definition as a process diagram.

2. In the process diagram, right-click the name of the activity and click **Properties**.
3. Go to the **Activity target** tab in the **"Save Object Properties" Activity Properties** dialog box.
4. Click in the **Activity saves properties of this object** box to specify the object whose properties you want the activity to save. This object is referred to as activity target. You can choose from the following options to specify the activity target:
   - **Workflow target object**. In a change workflow, the activity target is the target object of the request that started the workflow. For example, in a workflow that starts upon a deletion request, this choice causes the activity to

save the properties of the object whose deletion is requested.

- **Fixed object in directory**. The activity target is a particular object you select from Active Directory.
- **Object identified by workflow parameter**. The activity target is the object specified by the value of a certain parameter of the workflow. You can choose the desired parameter from the workflow definition.
- **Object from workflow data context**. The activity target will be selected by the activity on the basis of the data found in the workflow environment at the time of executing the workflow. You can specify which object you want the activity to select at workflow execution time.
- **Object identified by DN-value rule expression**. The Distinguished Name (DN) of the activity target is specified by the string value of a certain rule expression. By using a rule expression, you can compose a string value based on properties of various objects found in the workflow environment at the time of executing the workflow. You can create the desired rule expression when you configure the activity.

5. Go to the **Target properties** tab in the **"Save Object Properties" Activity Properties** dialog box.
6. Configure the list of the properties you want the activity to save:
   - To add a property to the list, click **Add**, and then select the name of the desired property.
   - To remove a property from the list, click the name of the property in the list, and then click **Remove**.

   The Workflow Designer provides a default list of properties. You can remove all properties from the list by clicking **Clear list** or revert to the default list by clicking **Restore default**.
7. Go to the **Notification** tab in the **"Save Object Properties" Activity Properties** dialog box to view or change notification settings. For instructions, see Configuring notification.
8. Go to the **Error handling** tab in the **"Save Object Properties" Activity Properties** dialog box to view or change error handling settings. For instructions, see Configuring error handling.

Configuring a Modify Requested Changes activity

When you configure a Modify Requested Changes activity, you can define the property changes to add or remove from the change request. You can choose the properties you want the activity to change and, for each property, choose to remove the property from the request, clear the property value in the request, or specify the new value to be assigned to that property. For a multi-value property, you can choose to add or remove a value from that property. Additional options are available such as notification, error handling,

changing the container where to create new objects, and adding or removing Active Roles controls from change requests.

To configure a Modify Requested Changes activity

1. In the Active Roles console tree, expand **Configuration | Policies | Workflow**, and select the change workflow containing the Modify Requested Changes activity you want to configure.

   This opens the Workflow Designer window in the details pane, representing the workflow definition as a process diagram.

2. In the process diagram, right-click the name of the activity and click **Properties**.
3. Go to the **Target changes** tab in the **"Modify Requested Changes" Activity Properties** dialog box.
4. Configure the list of the properties you want the activity to modify:
   - To add a property to the list, click **Add property**, and then select the desired property.
   - To remove a property from the list, click the **Delete** button labeled **X** on the right side of the list item representing that property.
5. After you have added a property, click in the **Action** field to specify the type of the changes you want the activity to make to that property:
   - Click **Set** to have the activity assign a new value to the property.
   - Click **Clear** to have the activity remove the property from the object.
   - In case of a multi-value property, click **Add value** or **Remove value** for the activity to add or remove the value of the property.
   - Click **Remove from request** if you want the workflow not to apply the changes to the property that were specified in the request that started the workflow.
6. If an action other than **Clear** or **Remove from request** is selected, click in the **Value** filed to specify the property value you want the activity to set, add or remove. The following options are available:
   - **Text string**. Use the given string of characters as the value of the property. You can type the desired string.
   - **Property of workflow target object**. Use the value of a certain property of the target object of the request that started the workflow. You can select the desired property from a list of object properties.
   - **Property of workflow initiator**. Use the value of a certain property of the user whose request started the workflow. You can select the desired property from a list of object properties.
   - **Changed value of workflow target object property**. Use the value that is requested to be assigned to a certain property of the workflow target object. You can select the desired property from a list of object properties.

- **Workflow parameter value**. Use the value of a certain parameter of the workflow. You can choose the desired parameter from a list of the workflow parameters.
- **Property of object from workflow data context**. Use the value of a certain property of the object that will be selected by the activity on the basis of the data found in the workflow run-time environment. You can choose the desired property and specify which object you want the activity to select at workflow run time.
- **Value generated by rule expression**. Use the string value of a certain rule expression. You can configure a rule expression to compose a string value based on properties of various objects found in the workflow run-time environment.

7. Go to the **Notification** tab in the **"Modify Requested Changes" Activity Properties** dialog box to view or change notification settings. For instructions, see Configuring notification.
8. Go to the **Error handling** tab in the **"Modify Requested Changes" Activity Properties** dialog box to view or change error handling settings. For instructions, see Configuring error handling.
9. Click the **Additional settings** link at the bottom of the **"Modify Requested Changes" Activity Properties** dialog box.
10. In the **Additional Settings** dialog box that appears, you can configure the activity to:
    - Change the container where to create new objects. Click in the **Modify object creation requests so as to create objects in this container** box, and then choose from the following options:
        - **Fixed container in directory**. With this option, objects will be created in the given OU or container. You can select the desired OU or container in Active Directory when you configure the activity.
        - **Parent OU of workflow target object**. With this option, objects are created in the OU that holds the target object of the request that started the workflow.
        - **Activity target object**. With this option, objects are created in the OU or container created or otherwise processed by a particular CRUD activity at the time of executing the workflow. You can select the desired CRUD activity from the workflow definition when you configure the activity.
        - **Object identified by workflow parameter**. With this option, objects are created in the OU or container specified by the value of a certain parameter of the workflow. You can choose the desired parameter from the workflow definition when you configure the activity.
        - **Object from workflow data context**. With this option, objects are created in the OU or container that will be selected by the

activity on the basis of the data found in the workflow environment at the time of executing the workflow. You can specify which OU or container you want the activity to select.

- **Object identified by DN-value rule expression**. With this option, objects are created in the OU or container whose Distinguished Name (DN) is specified by the string value of a certain rule expression. By using a rule expression, you can compose a string value based on properties of various objects found in the workflow environment at the time of executing the workflow. You can create the desired rule expression when you configure the activity.

- Add or remove Active Roles controls from the request. To add or change a control, click **Add** or **Change**, and then, in the dialog box that appears, specify the name and, if applicable, the value of the control. If you want the activity to add the control to the request, click **Include this control in the change request**. If you want to ensure that the control never occurs in the request, click **Exclude this control from the change request**.

Controls can be used to pass additional information to Active Roles on how to process the request. See Active Roles SDK for information about controls.

Enabling or disabling an activity

Temporarily disabling an activity may be useful when the workflow is under construction so the workflow definition is not finalized and the activity should not run until a certain time.

To disable an activity or enable a disabled activity

1. Select the workflow definition in the console tree to display the workflow as a process diagram.
2. In the process diagram, right-click the activity and click **Disable** or **Enable**, respectively.

While an activity is disabled in a given workflow, Active Roles skips that activity when running that workflow. When you enable a disabled activity in a given workflow, you allow Active Roles to execute that activity when running that workflow.

Enabling or disabling a workflow

Temporarily disabling a workflow may be useful when the workflow is under construction so the workflow definition is not finalized and the activities included in the workflow should not run until a certain time.

To disable a workflow or enable a disabled workflow

- Right-click the workflow definition in the console tree and click **Disable Workflow** or **Enable Workflow**, respectively.

While a workflow is disabled, Active Roles does not run any activities included in that workflow regardless of the workflow start conditions. When you enable a disabled workflow, you allow Active Roles to run the activities included in that workflow.

Using the initialization script

When executing a workflow instance, Active Roles uses a single PowerShell operating environment, referred to as a runspace, for all script activities held in that workflow. The workflow runtime engine creates a runspace once the workflow instance has been started, and maintains the runspace during the execution of the workflow instance.

When you configure a workflow, you can specify PowerShell commands you want the workflow runtime engine to execute immediately after the runspace creation. These commands constitute the initialization script that the workflow engine runs prior to performing script activities.

With an initialization script, you can define runspace configuration data separately from the logic of the script activities and use it to initialize the environment for executing script activities. Specifically, you can:

- **Load PowerShell modules and snap-ins**. All activity scripts can use the modules and snap-ins loaded in the initialization script, without having to load the prerequisite modules or snap-ins on a per-activity basis.

  The modules and snap-ins loaded in the initialization script are available to all script activities at workflow runtime. For example, the `Import-Module 'SmbShare'` command added to the initialization script makes the Server Message Block (SMB) Share-specific commandlets available to all script activities within the workflow.

- **Initialize environment-specific variables, referred to as global variables**. All activity script can retrieve and update global variables, which makes it possible to exchange data between different activity scripts.

  The global variables are visible to all script activities at workflow run time. For example, the `$rGuid = [Guid]::NewGuid()` command added to the initialization script makes the `$rGuid` variable available to all script activities within the workflow. To reference a variable that is defined in the initialization script, the activity script must use the `$global:` qualifier, such as `$global:rGuid`.

  When execution of the workflow instance is suspended (for example, waiting for approval), and then resumed (for example, after receiving an approval decision), the runspace is reinitialized so the global variables may change. If you need to preserve the value of a global variable, add the `[Persist()]` attribute to the variable's name in the initialization script, such as `[Persist()]$rGuid = [Guid]::NewGuid()`. The global variables defined in this way are saved to a persistent storage upon suspending the workflow instance and restored from the storage when the workflow instance is resumed. To save a variable, Active Roles creates and stores an XML-based

representation of the object signified by that variable, similarly to the `Export-Clixml` command in Windows PowerShell. When restoring the variable, Active Roles retrieves the XML data that represents the object, and creates the object based on that data, similarly to the `Import-Clixml` command.

To view or change the initialization script

1. In the Active Roles console tree, expand **Configuration | Policies | Workflow**, and select the workflow you want to configure.

   This opens the Workflow Designer window in the details pane, representing the workflow definition as a process diagram.

2. In the details pane, click the **Workflow options and start conditions** button to expand the area above the process diagram, and then click the **Configure** button.
3. Click the **Initialization script** tab in the dialog box that opens.

The **Initialization script** tab displays the current script. You can add or modify the script by typing in the edit box on that tab.

Example: Approval workflow

Approval workflow complements automated policies, to make provisioning and deprovisioning decisions based on human input. While automated policies require no manual intervention, approval-based fulfillment of administrative operations adds to process automation the ability to manually accept or deny operation requests, and to monitor the execution of request-processing tasks to ensure they are responded in a timely manner.

Approval workflow can service a range of requests, which are user actions intended to perform administrative operations. Examples of such operations include the creation, modification, and deprovisioning of user accounts.

When a requested operation requires permission from certain individuals in an organization, a workflow can be started to coordinate the approval process. The system only performs the requested operation after approval is given by an authorized person.

Active Roles administrators can create and configure approval workflows by using the Workflow Designer-a graphical tool provided in the Active Roles console for constructing workflows. When designing an approval workflow, the administrator specifies which kind of operation causes the workflow to start, and adds approval rules to the workflow. The approval rules determine who is authorized to approve the operation, the required sequence of approvals, and who needs to be notified of approval tasks or decisions.

The approval workflow solution provided by Active Roles includes:

- The Workflow Designer for constructing workflows, available from the Active Roles console. You use the Workflow Designer to configure an approval workflow by adding approval activities to the workflow definition.

- The directory management interfaces, such as the Web Interface or Active Roles Console for submitting operation requests for approval. For example, approval workflow could be configured so that the creation of a user account via Active Roles starts the approval workflow instead of immediately executing the user creation operation.
- The approval-related section of the Web Interface to manage operation requests. This section provides a "to-do" list of the approval tasks a designated user has to carry out, allowing the user to perform tasks such as approving or rejecting operation requests.

Definition of terms

This section summarizes some important definitions that apply to approval workflow.

Approval

A decision point in a workflow that is used to obtain authorization from a person before continuing the workflow.

Approval rule (Approval activity)

Workflow activities of the Approval category are referred to as approval rules. Workflow start conditions determine which operations start the workflow and the approval rules added to the workflow determine who is authorized to approve the operation, the required sequence of approvals, and who needs to be notified of approval tasks or decisions.

Approval task

A task created as part of the processing of an approval rule and assigned to an approver. The approver is expected to complete the task by making a decision to allow or deny the operation.

Approver

The person designated to perform an approval task. The setting that determines the approvers is a configuration element of an approval rule. When processing an approval rule, Active Roles creates an approval task and assigns it to the approvers defined by the rule. The state of the task governs the workflow transition: the task must receive the "Approve" resolution for the operation to pass through the approval

rule. If the task has received the “Reject” resolution, the operation is denied and the workflow instance is completed.

Initiator (requestor)

The identity of the user or service that has requested an operation in Active Roles. For example, when the Active Roles console is used to change or create an object, the console user is identified as the initiator of the respective operation. The initiator of an operation is also referred to as the operation requestor.

Notification

The means used to notify a user or group of users about a specific predefined situation that could manifest within a workflow. A notification message is generated and sent to the designated recipients via e-mail to inform them that a certain event has occurred, such as a new approval task has been submitted to the approvers or the operation has been completed. A notification configuration, stored as part of an approval rule, involves such elements as the event to notify of, the list of the notification recipients and the notification message template. Active Roles also provides a separate category of workflow activity for the purpose of notification, in addition to approval rules.

Operation

A request for certain changes to be made to directory data, such as creating users or adding users to groups. An operation can start an approval workflow, in which case the requested changes are made only after they are approved.

Operation target object

The object to be changed or created by the operation. For example, if creation of a user account is requested, that account is referred to as the operation target object. With a request to add a user to a group, the group is referred to as the operation target object.

How it works

Approval workflow is governed by workflow start conditions and approval rules. Workflow start conditions determine which kind of operation causes the workflow to start, and the

approval rules added to the workflow determine the persons who are authorized to approve the operation (approvers).

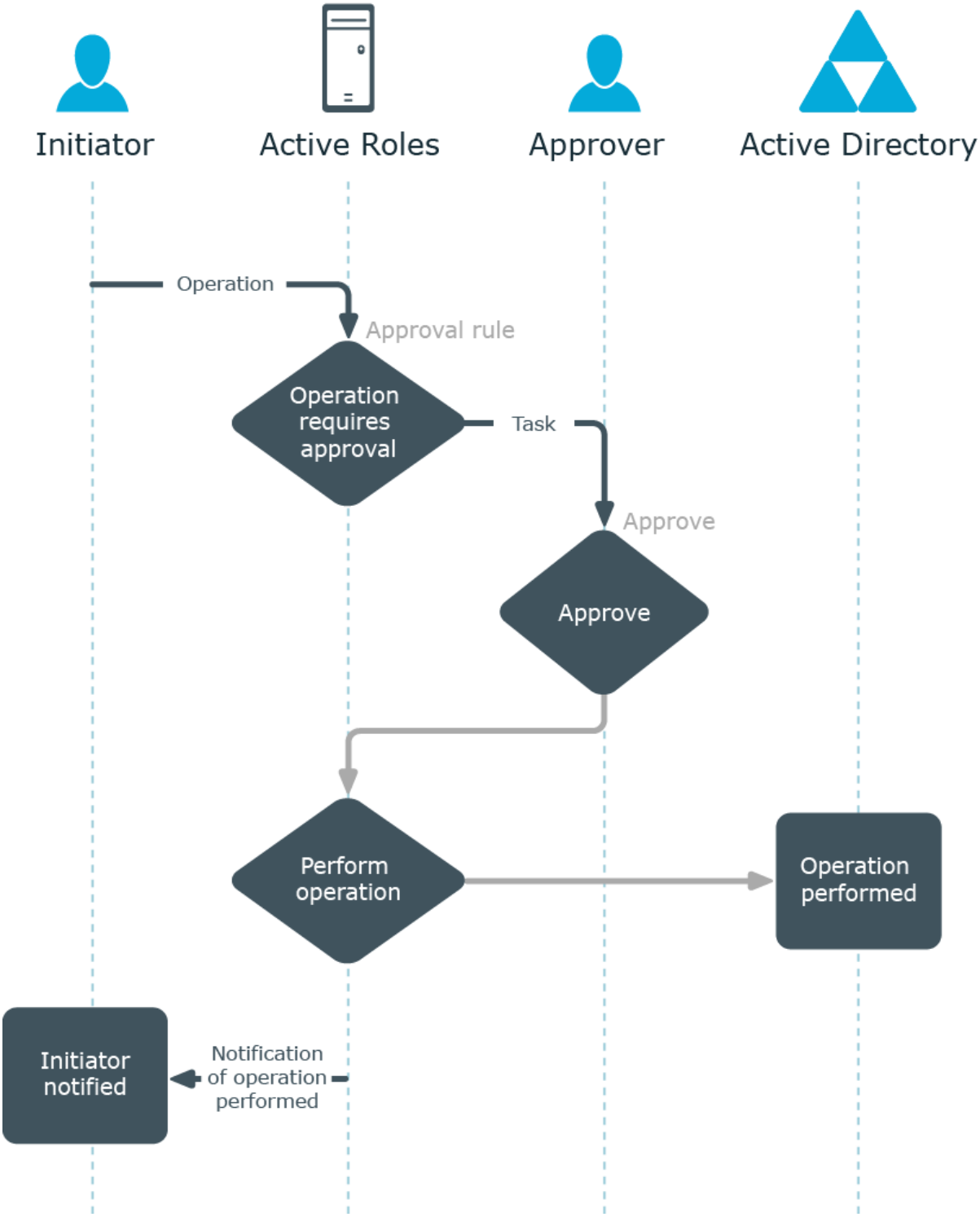
When an Active Roles user requests an operation, Active Roles checks to see whether the operation meets the start conditions of any workflow, and starts the workflow whose conditions are met. An approval rule included in the workflow then generates an approval task and assigns the task to the approvers defined by the rule.

An approver completes an approval task by applying the Approve or Reject action to the task. This changes the status of the task from “Pending” to “Approved”, or “Rejected” respectively.

Action: Approve

If the approver applies the Approve action to the task, Active Roles allows the operation to be performed.

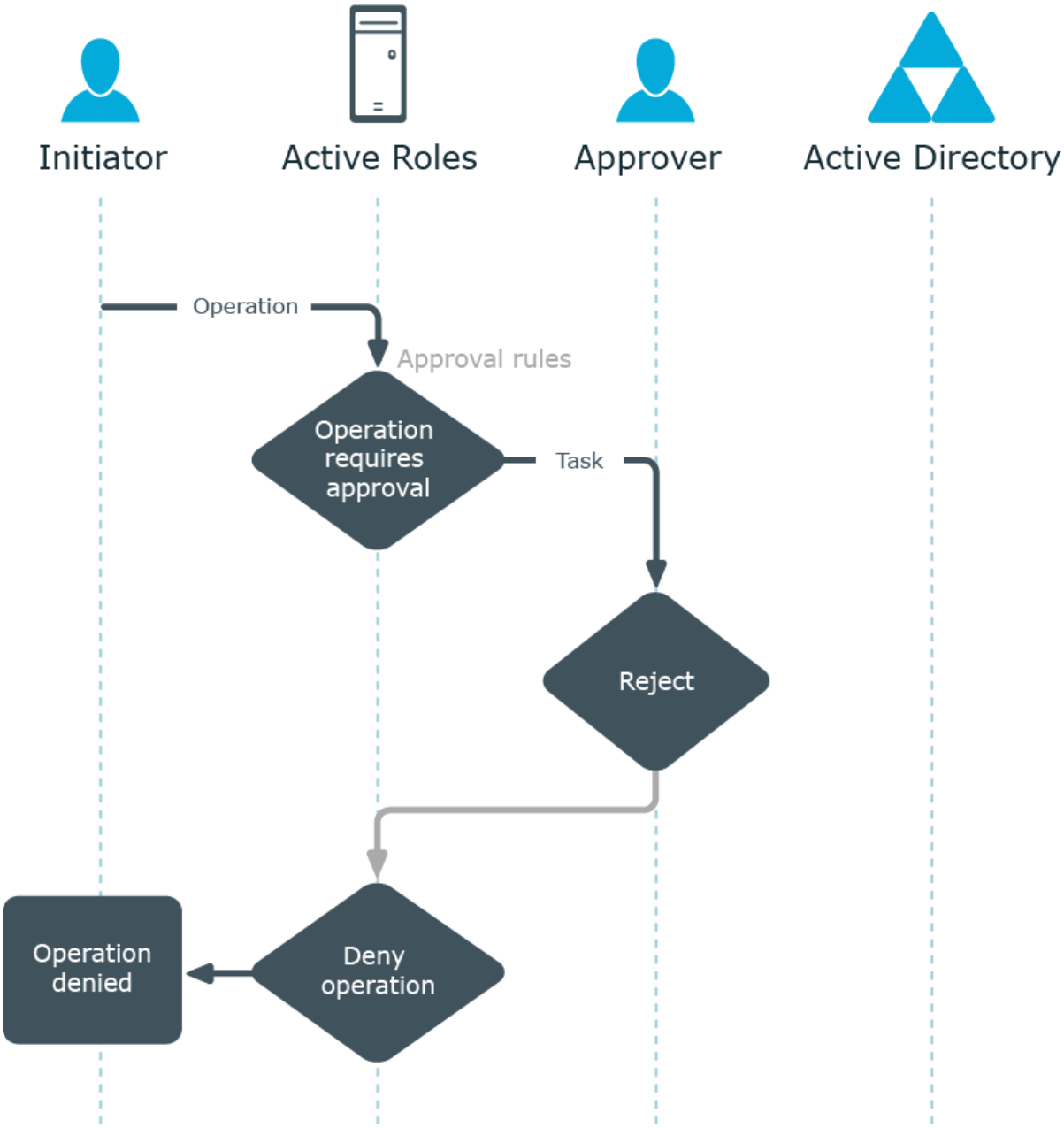
Figure 91: Approve action



Action: Reject

If the approver applies the Reject action to the task, Active Roles cancels the operation.

Figure 92: Reject action

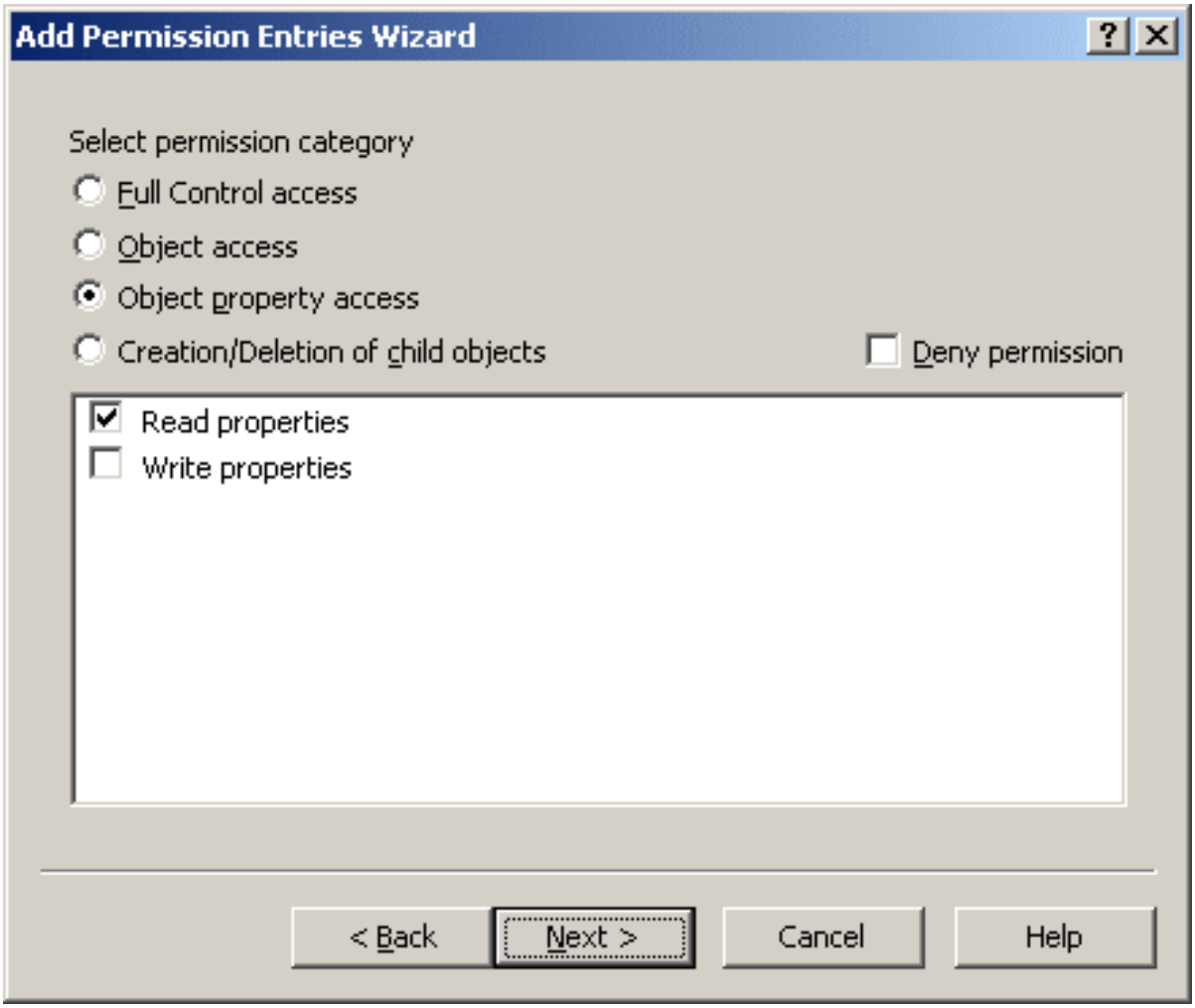


Multiple approvers

An approval rule may be configured so that a single task is assigned to multiple approvers. For example, a group can be designated as an approver, which causes the task to be assigned to every member of the group. If this is the case, the first of the approvers to apply the Approve or Reject action to the task, completes the task.

If the task receives the Approve action, Active Roles allows the operation to be performed. If the Reject action is applied to the task, Active Roles cancels the operation.

Figure 93: Multiple approvers



Multiple tasks

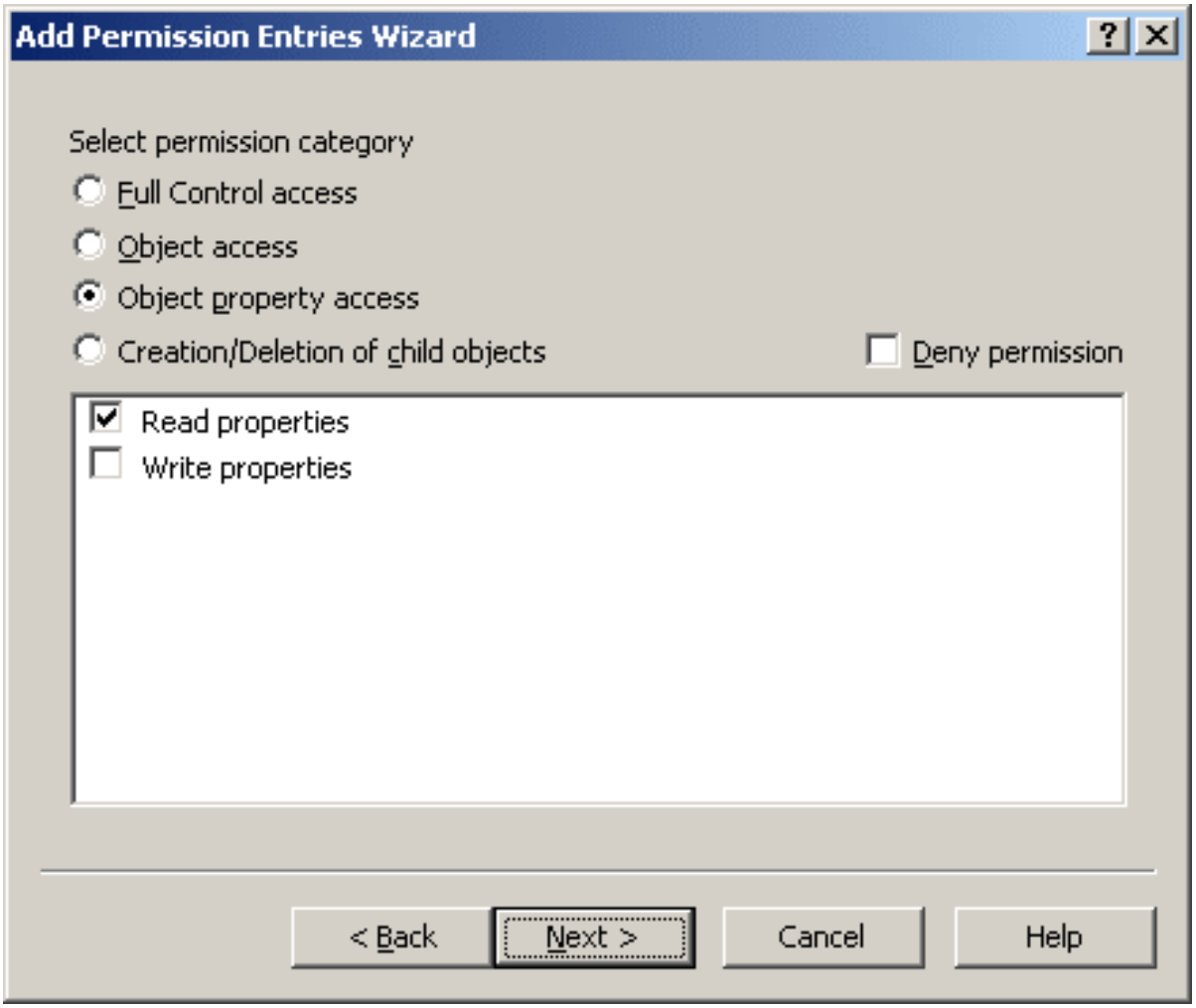
The number of approval tasks generated by a single workflow instance depends on how many approval rules are included in the workflow (one task per each rule). Therefore, if a

workflows has multiple approval rules, multiple tasks will be created and assigned to the respective approvers.

Within a single workflow, approval rules are applied in a sequential manner. This means that a subsequent rule is applied only after the requested operation has passes the previous rule.

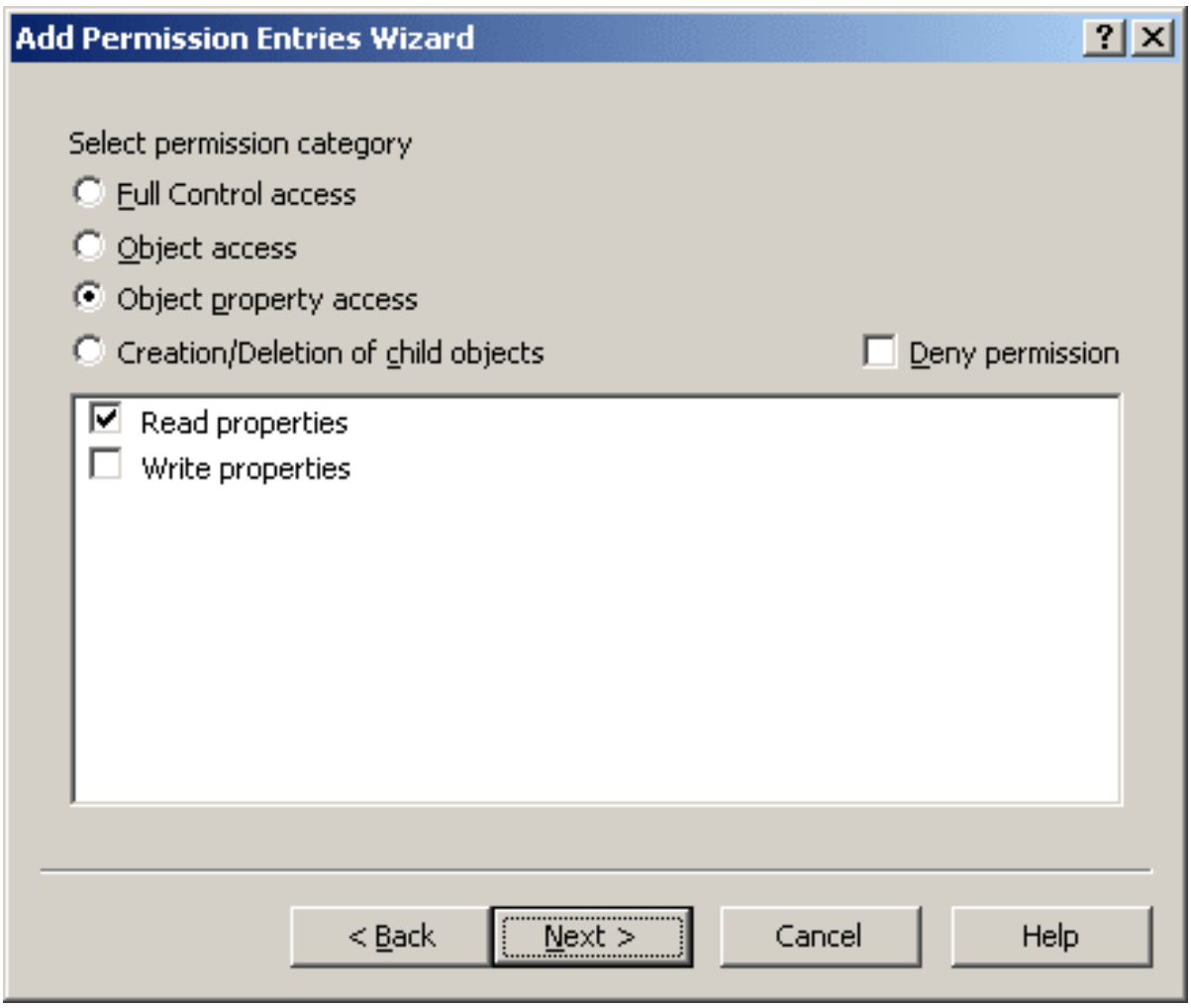
If each of the tasks receives the Approve action, Active Roles allows the operation to be performed.

Figure 94: Multiple tasks



If at least one of the tasks receives the Reject action, Active Roles cancels the operation.

Figure 95: Cancellation of task



Creating and configuring an approval workflow

To implement an approval scenario where certain operations require approval in Active Roles, you create a workflow definition, configure the workflow start conditions, and add and configure approval activities (approval rules) as appropriate. All these tasks are performed using the Workflow Designer-a graphical tool included in the Active Roles console.

When configuring workflow start conditions, you specify:

- A type of operation, such as **Create**, **Rename**, **Modify**, or **Delete**. The workflow starts only if an operation of that type is requested.

- A type of object, such as **User**, **Group** or **Computer**. The workflow starts only if the operation requests changes to an object of that type.
- For the Modify operation type, a **list of object** properties. The workflow starts only if the operation requests changes to any of those properties of an object.
- The identity of an operation requestor (initiator), such as a **user**, **group**, or **service**. The workflow starts only if the operation is requested on behalf of that identity.
- A container, such as an **Organizational Unit** or **Managed Unit**. The workflow starts only if the operation requests changes to, or creation of, an object in that container.
- (Optional) A filter that defines any additional conditions on entities involved in an operation. The workflow starts only if the operation satisfies those conditions. If no filter is set, then no additional conditions are in effect.

Any operation that meets all the start conditions specified on a workflow causes the workflow to start.

When configuring an approval rule within a workflow, you specify:

- **A list of approvers, such as users or groups**. This setting identifies the persons who are authorized to allow or deny operations that start the workflow.
- **Notification settings**. This includes workflow events to notify of, notification recipients, delivery options, and notification message template.

Creating a workflow definition

The Active Roles console provides the Workflow Designer for creating and configuring workflows. First, you create a workflow definition. Then, you use the Workflow Designer to construct the workflow by adding workflow activities and making other changes to the workflow definition.

For step-by-step instructions, see Creating a workflow definition earlier in this chapter.

Specifying workflow start conditions

You can specify the start conditions for a workflow by editing its definition in the Workflow Designer. The start conditions determine which operations cause the workflow to start.

For step-by-step instructions, see Configuring workflow start conditions earlier in this chapter.

Example Suppose you want the creation of user accounts in a certain organizational unit to require approval. You can implement this scenario by configuring the workflow start conditions as follows:

- Set type of operation to **'Create'**.
- Set type of object to **'User'**.

- Set initiator to **'Any User'**.
- Set container by selecting the organizational unit you want.

As a result of these conditions, the workflow will start whenever Active Roles is used to create a user account in that organizational unit.

Specifying approvers

When constructing an approval workflow, you add one or more approval activities to the workflow definition, thereby creating approval rules, and then configure those activities to define approvers for each rule. The entities that can be designated as approvers include manager of operation requestor, manager of operation target object, and manager of container that holds operation target object. It is also possible to select any particular user or group of users for the role of approver.

Example: Extending the previous example, suppose you want the creation of user accounts to be approved by the manager of the organizational unit in which the accounts are going to be created. You can implement this scenario by adding an approval activity to the workflow and then using the **Properties** command on that activity to select the corresponding option on the **Approvers Selection** page.

For step-by-step instructions, see Configuring an Approval activity earlier in this chapter.

Configuring notification

You can configure approval rules to notify approvers or other interested parties of specific events that may occur in the approval process. For example, an approval rule can be configured so that the approvers defined by the rule receive a notification e-mail whenever an operation is requested that requires their approval. Other events to notify of include the completion of an approval task indicating that an approver has either allowed or denied the requested changes, the completion of the operation indicating that the requested changes have been applied, and the operation failure because of an error condition.

Notification recipients

When configuring notification settings in an approval rule, you choose an event, and specify who you want to receive email notification of that event-notification recipients. A recipient can be any mailbox-enabled user or mail-enabled group. There are also a number of options allowing you to select recipients based on their role, such as operation requestor, approver, manager of operation requestor, or manager of operation target object. A single rule can be configured to notify of one or more events, with an individual list of recipients being defined for each event.

Notification delivery

Along with an event to notify of and notification recipients, you can select delivery options. In addition to immediate delivery (which causes every occurrence of the event to generate a separate notification message), there is the scheduled delivery option for aggregating notifications. If you select the scheduled delivery option, all notifications about the event occurrences within a time period of your choice are grouped and sent as a single message. In this case, the message body is composed of the aggregated notifications about every single occurrence of the event.

Notification messages are routed for delivery by an SMTP service, such as that provided by Microsoft Exchange or Internet Information Services. The address and other parameters of the outgoing e-mail server are specified as part of the notification settings on each approval rule.

Notification message template

Notification messages are based on a message template that determines the format and contents of an e-mail notification message, including the message subject and body. You can access the template from the page where you select an event together with notification recipients. When you change the template, your changes only take effect on the messages specific to the notification you are configuring.

Example: In the previous example, you could configure the approval activity so that the approver would receive an e-mail notification whenever a user creation operation is requested that requires their approval. Open the **Properties** page for that activity and go to the **Notification** step. Then, click **Add**, verify that the **Task created** event is highlighted, and select the appropriate check box under **Notification recipients**.

For step-by-step instructions, see Configuring a Notification activity earlier in this chapter.

Email based approval

In addition to the Web Interface pages for performing approval tasks, Active Roles provides the facility to approve or reject a pending request by replying to a notification message that informs of the request. An approval workflow can be configured to behave as follows:

- Upon the receipt of a change request that requires approval, Active Roles sends a notification message to the designated approvers, with the message body containing the option to approve or reject the request.
- The approver replies to the notification message, choosing the desired option—approve or reject. In the reply message the approver is expected to provide a comment explaining the reason for that choice.
- Active Roles receives the reply massage from the approver, checks to see if the approver elected to approve or reject the request, and then allows or denies the requested changes accordingly.

This way, the capabilities to work with approval requests are integrated into the e-mail client. The approvers do not need a web browser to view, and respond to, their approval requests. This, for instance, enables Microsoft Office Outlook users to manage approvals even when they are offline. One more opportunity is to manage approvals using an e-mail client on a mobile device.

> IMPORTANT: To manage approval requests by replying to notification e-mails, you must be logged on to the approver's mailbox as the owner of the mailbox or as an identity that has full access to the mailbox (including the Send As permission). The Send on Behalf permission will not suffice. Active Roles detects the situation where the reply is sent on behalf of the mailbox owner, and disregards the reply message in that case.

Integration with Microsoft Outlook

For organizations that have deployed Microsoft Exchange Server 2013 or later, and use Microsoft Office Outlook 2010 or later as their standard e-mail client, Active Roles provides an approvals management facility integrated in Outlook. This allows Microsoft Office end-users to manage approvals in Active Roles through the e-mail application they use on a day-to-day basis.

The Add-in for Outlook component that is included with Active Roles offers the basic functionality for processing and submitting approvals. Active Roles Add-in for Outlook allows Microsoft Outlook users to approve or reject requests that are sent to them for approval. Requests are delivered through notification e-mail messages, and can be approved or rejected directly from the notification e-mail message, without having to use Active Roles' Web Interface pages. In every e-mail message from Active Roles that notifies of an approval request, Active Roles Add-in for Outlook adds the **Approve** and **Reject** buttons along with **Approve** and **Reject** menu commands allowing the approver to respond by selecting the appropriate button or command.

Software and configuration requirements

Integration with Microsoft Office Outlook requires the following software and configuration prerequisites:

- **Microsoft Exchange Server 2013 or later**. Integration with Outlook requires at least one server running Exchange 2013 or later that holds the Client Access server role and Mailbox server role, to be deployed in your Exchange organization.
- **Microsoft Office Outlook 2010 or later**. The approvers use Outlook 2010 or later as their e-mail client application.
- **Active Roles Add-in for Outlook (32-bit)**. The Add-in for Outlook component of Active Roles is installed on the computer running Microsoft Office Outlook. The software requirements specific to Active Roles Add-in for Outlook are listed in the

Active Roles Quick Start Guide.

> NOTE: The Active Roles Add-in for Outlook does not support the 64-bit version of Microsoft Office Outlook.

- **Approvers' mailboxes**. The mailboxes where approval and rejection takes place are on a Mailbox server running Exchange 2013 or later. Although not mandatory, this condition is highly advisable.
- **Active Roles' mailbox**. A mailbox reserved for the exclusive use of Active Roles. This mailbox should be on a Mailbox server running Exchange 2013 or later.
- **Exchange Web Services**. The approval workflow has the approval rule notification settings configured so that Active Roles uses Exchange Web Services to communicate with Exchange. These settings include the address (URL) of the Exchange Web Services endpoint on an Exchange server that holds the Client Access server role, along with the credentials that identify Active Roles' mailbox.

Integration with non-Outlook e-mail clients

For organizations that have deployed Microsoft Exchange Server 2013 or later, but use an e-mail client application other than Outlook 2010 or later, Active Roles offers the ability to approve or reject change requests by simply replying to notification messages that inform approvers of approval tasks. In this case, the notification message contains selectable options that, when clicked or tapped, cause the e-mail application to create a new message in reply to the notification message. The reply message contains indication of the approval decision (approve or reject) and prompts the approver to supply a comment on the approval decision (approval or rejection reason). Then, the approver sends the reply message, thereby completing the approval task.

Software and configuration requirements

The ability to manage approvals from non-Outlook e-mail clients calls for the same software and configuration prerequisites as Outlook integration (see Integration with Microsoft Outlook), with the following exceptions and additions:

- The e-mail client applications that can be used to manage approvals are not restricted to Microsoft Office Outlook 2010 or later. It is possible to use, for instance, earlier Outlook versions or e-mail applications on mobile devices.
- Active Roles Add-in for Outlook does not need to be installed on the computer running the e-mail client application.
- The approval rule notification settings are configured so that the notification messages originated by Active Roles have integration with the Web Interface turned off. Ensure that the **Send approval response by e-mail** option is selected in the properties of the e-mail configuration that is used by the approval rule (this is the default setting).

E-mail transport via Exchange Web Services

Active Roles can use Exchange Web Services (rather than SMTP server) to communicate with Exchange Server when sending notification messages and getting response to notification messages. This enables notification recipients to perform approval tasks by replying to notification messages from their regular e-mail clients, instead of using the Web Interface pages to approve or reject the requests. With the use of Exchange Web Services, Active Roles makes it possible for an approval workflow to behave as follows:

- A change request that requires approval causes Active Roles to send a notification message to the designated approver, with the message body containing the option to approve or reject the request.
- The approver replies to the notification message by choosing the desired option (either approve or reject) and typing in a text to explain the reason for that choice.
- Active Roles receives the reply message from the approver, checks to see if the approver elected to approve or reject the request, and then allows or denies the requested changes accordingly.

The use of Exchange Web Services calls for the following prerequisites:

- Exchange Server 2013 or later. Exchange Web Services is deployed with the Client Access server role.
- Dedicated mailbox hosted on Exchange Server 2013 or later. The mailbox should be reserved for the exclusive use of Active Roles.

Configuration settings

The following configuration settings are available with the Exchange Web Services option for e-mail transport.

Exchange Web Services address

This setting identifies the URL of the Exchange Web Services endpoint, which locates the `exchange.asmx` file on the Exchange server running the Client Access server role. For example, https://CAServer.domain.com/EWS/exchange.asmx

Active Roles' mailbox credentials

This setting specifies the user name and password of the mailbox through which Active Roles will send and receive e-mail. The mailbox must be located on Exchange Server 2013 or later, and must be reserved for the exclusive use of Active Roles.

It is important that no applications other than Active Roles access this mailbox. Processing e-mail messages in Active Roles' mailbox by other applications, such as Office Outlook, can cause an adverse effect on the functionality of Active Roles.

Options for the Approve and Reject links

This setting controls the behavior of the Approve and Reject links in the notification messages delivered using this e-mail configuration. Two options are available:

- **Send approval response by e-mail**
- **Approve or reject via Web Interface**

If **Send approval response by e-mail** is selected, notification recipients can perform approval tasks from within their e-mail application. When an approver chooses one of the links provided in a notification message to approve or reject a request, the e-mail application replies with an e-mail message containing information about the approval decision. Active Roles receives the reply message, checks it to see if the approver elected to approve or reject the request, and then allows or denies the requested changes accordingly.

If **Approve or reject via Web Interface** is selected, choosing the Approve or Reject link in a notification message directs the e-mail application to open a Web Interface page for performing the approval task. The page may not open as expected if the e-mail application does not support HTML format or an appropriate Web browser does not exist on the device running the e-mail application.

Steps to configure the use of Exchange Web Services

Perform the following steps in the Active Roles console to configure the default mail settings with the option to use Exchange Web Services:

1. In the console tree, select **Configuration | Server Configuration | Mail Configuration**.
2. In the details pane, double-click **Default Mail Settings**.
3. In the **Default Mail Settings Properties** dialog box, configure the settings on the **Mail Setup** tab:
    a. From the **Settings for** list, select **Exchange Web Services**.
    b. In the **Exchange Web Services address** box:
        i. For on-premises Exchange mailbox, supply the URL of the Exchange Web Services endpoint. This URL locates the `exchange.asmx` file on the Exchange server that is running the Client Access server role. For example, https://CAServer.domain.com/EWS/exchange.asmx.
        ii. For the Exchange mailbox on the cloud, use https://outlook.office365.com/EWS/Exchange.asmx.
    c. Under **Mailbox credentials**:
        i. For on-premises Exchange mailbox, supply the user name and password of the mailbox through which Active Roles will send and receive e-mail.

ii. For the Exchange mailbox on the cloud, supply the Azure user credentials of the Azure mailbox.

This mailbox must be created on a server running Exchange 2013 or later and reserved for the exclusive use of Active Roles.

d. Verify the settings you have configured. Click **Verify Settings**, supply a valid e-mail address, and then click **Send**.

This causes Active Roles to send a diagnostic e-mail message to the address you supplied. The message is attempted to be delivered from Active Roles' mailbox by using Exchange Web Services. You can check the mailbox with the address you supplied to see if the diagnostic message has been received.

4. Verify that the **Send approval response by e-mail** option is selected on the **Mail Setup** tab.
5. Select **Approve or reject via Web Interface** to manage emails through the Web Interface.
6. When finished, click **OK** to close the **Default Mail Settings Properties** dialog box.

Automation workflow

Workflow refers to a sequence of actions that leads to the completion of a certain task. The sequence is carried out according to a set of rules or policies. A workflow can be configured to start upon a change request that satisfies the start conditions of the workflow. An example is a workflow that coordinates the process of approving certain changes to directory data such as creation of new users or population of security groups. In Active Roles, this kind of workflow is referred to as a change workflow.

A workflow can perform some routine administrative tasks on a scheduled basis or on user demand. In this case, workflow is not attached to any change request. With Active Roles, you can configure a workflow to perform certain actions at a specific time. You can also allow users to run a workflow at any time on demand. This workflow category is referred to as an automation workflow.

Automation workflow can automate the completion of complex administrative tasks to help you manage large task volumes. It also allows you to build checks or restrictions in directory administration processes to ensure consistency and compliance with your company policies and legal requirements. By using automation workflow, you can ensure that directory administration tasks are performed in a consistent and efficient manner.

Automation workflow options and start conditions

The start conditions of an automation workflow determine the trigger that causes the workflow to start. You can use a time-based trigger or an event-based trigger to start an automation workflow. It is also possible to allow an automation workflow to be started on user demand.

With a time-based trigger, you can configure an automation workflow to start at a specific time of a day or you can schedule an automation workflow to start multiple times on a daily, weekly, or monthly basis. An event-based trigger allows you to start an automation workflow upon startup of the Active Roles Administration Service. Each automation workflow can have only one trigger.

To enable a time-based trigger, an automation workflow must be configured with the option to run the workflow on a schedule. This option is available on the **Workflow Options and Start Conditions** page in the Workflow Designer provided by the Active Roles console. The page contains a number of options discussed in the sections that follow.

Run the workflow on a schedule

If you select the **Run the workflow on a schedule** option, then you can choose from the following options to run the workflow:

- **One time**. Lets you choose the date and time to run the workflow.
- **Hourly**. Lets you choose the date and time to run the workflow for the first time, and the recurrence interval (in hours and minutes) for the workflow. Thus, an interval of one hour causes the workflow to run every hour and an interval of two hours causes the workflow to run every other hour.
- **Daily**. Lets you choose the date to run the workflow for the first time, the time of the day to run the workflow, and the recurrence interval (in days) for the workflow. Thus, an interval of one causes the workflow to run every day and an interval of two causes the workflow to run every other day. The workflow will start at the specified time each day.
- **Weekly**. Lets you choose the date to run the workflow for the first time, the time of the day to run the workflow, the days of the week on which to run the workflow, and the recurrence interval (in weeks) for the workflow. Thus, an interval of one causes the workflow to run every week and an interval of two causes the workflow to run every other week. The workflow will start at the specified time on each of the specified days.
- **Monthly**. Lets you choose the date to run the workflow for the first time, the time of the day to run the workflow, the months in which to run the workflow, and the day of the month on which to run the workflow. You can choose either the number of the day, or the first, second, third, fourth, or last occurrence of a certain day of the week day during the month. The desired day of the week can be selected from a list.

- **When the Administration Service starts**. Causes the workflow to start immediately after the Active Roles Administration Service has started up. This option applies to the Administration Service identified by the **Run the workflow on** setting.

Server to run the workflow

When started by a schedule, the workflow runs on a certain instance of the Active Roles Administration Service. The instance is identified by the **Run the workflow on** setting. This setting indicates the name of the computer running the Administration Service. You can choose the desired computer from the **Run the workflow on** list.

Allow the workflow to be run on demand

If you select the **Allow the workflow to be run on demand** option, users can run the workflow manually, regardless of a schedule. This option allows a user to run the workflow at any time if necessary. A workflow can be started on demand from the Active Roles console or Web Interface, by choosing the **Run** command on the workflow definition object. For details, see Running an automation workflow on demand later in this document.

Active Roles normally allows only one instance of the workflow to run at a time. However, you can change this behavior for the case of running the workflow on demand. The following options are available:

- If the workflow is already running, then do not start a new instance.
- If the workflow is already running and a new instance is started on demand, then run the new instance in parallel.

The second option allows a new instance of the workflow to be started on demand even though the workflow is already running. This option applies only to the case of running the workflow on demand. In the case of a scheduled run Active Roles allows only one instance of the workflow to run at a time.

"Run as" options

The "run as" options determine the user account that the workflow runs under. Click the **"Run as" options** link on the **Workflow Options and Start Conditions** page to view or change the account setting. You can choose from the following options:

- **The service account of Active Roles**. The workflow runs under the service account of the Administration Service that executes the workflow.
- **The account of the user who started the workflow**. The workflow runs under the Windows account of the user who requested the operation that started the workflow.

All activities within the workflow normally run under the account identified by the "run as" options for the workflow. However, each activity can be configured to use individual "run

as" options. The property page for the activity contains the **"Run as" options** link allowing you to override the workflow "run as" setting on a per-activity basis.

When running under the account of the Administration Service, the workflow activities have the same rights and permissions as the Administration Service itself and thus can perform any tasks allowed for the Administration Service.

When running under the account of the user who started the workflow, the activities can perform only the tasks that Active Roles allows for that user account. The Administration Service processes the activity operation requests as if they were submitted by that user via an Active Roles user interface, so the activities have the rights and permissions the user account is given in Active Roles.

Enforce approval

The **Enforce approval** option determines whether to apply approval rules to the changes requested by the workflow running under a privileged account. When selected, this option causes the approval-pending changes requested by the workflow activities to be submitted for approval regardless of the account under which the workflow is running. Otherwise, the changes are applied without waiting for approval if the workflow is running under the service account of Active Roles, under the account of the approver, or under the account of an Active Roles administrator. This option setting can be overridden on a per-activity basis.

Additional settings

The additional settings specify whether to terminate the workflow if it runs longer that a certain time period. Click the **Additional settings** link on the **Workflow Options and Start Conditions** page to view or change the following setting:

- **Terminate the workflow if it runs longer than:** <*time period*>

This setting allows you to limit the amount of time the workflow is allowed to run. Use this setting to limit the automation workflow that might take a long period of time to execute, causing an inconvenience to the user.

Parameters

When you configure workflow options and start conditions for an automation workflow, you can set up workflow parameters and assign values to workflow parameters. Parameter values are used by the workflow activities when the workflow is running. An activity may retrieve the value of the desired parameter and perform the action depending upon the parameter value.

By default, the workflow does not have any parameters defined. You can add, modify (edit) or remove parameter definitions on the **Parameters** page. Once the definition of a parameter has been added to the workflow, you can:

- Assign a value to the parameter. To do this, select the parameter from the list on the **Parameters** page and click the **View or change parameter value** button. The value assigned to the parameter is stored in the workflow definition. The workflow activities can retrieve the parameter value from the workflow definition when the workflow is running.
- Configure the parameter so that the user can set the parameter value when starting the workflow on demand. To do this, select the parameter from the list on the **Parameters** page, click the **Edit** button, and then clear the **Don't show this parameter when starting the workflow on demand** check box. Active Roles will prompt the user to set the parameter value when the user starts the workflow on demand. The parameter value supplied by the user will only be used during the current run of the workflow.
- View or change various properties of the parameter. To do this, select the parameter from the list in the **Parameters** page, click the **Edit** button, and then use the options in the **Parameter Definition** dialog box.

Each parameter has a number of properties that define it, including the parameter name, parameter description, syntax of parameter values, a list of acceptable parameter values, whether the parameter accepts a single value or multiple values, and whether the parameter must have a value. The acceptable values can be determined either by a static list of values or by using a script. In the latter case, the script calculates the list of the acceptable values each time the workflow is started. A script can also be used to assign a value to the parameter. The script calculates the value each time the workflow is started.

For further information about workflow parameters, see Configuring workflow parameters earlier in this document.

Initialization script

When you configure an automation workflow, you can specify PowerShell commands you want the workflow run-time engine to execute immediately after creation of the PowerShell operating environment for the script activities held in that workflow. These commands constitute the initialization script that the workflow engine runs prior to performing script activities.

With the initialization script, you can:

- Load PowerShell modules and snap-ins. All activity scripts can use the modules and snap-ins loaded in the initialization script, without having to load the prerequisite modules or snap-ins on a per-activity basis.
- Initialize environment-specific variables, referred to as global variables. All activity script can retrieve and update global variables, which makes it possible to exchange data between different activity scripts.

For further information, see Using the initialization script earlier in this document.

Using automation workflow

This section contains information and step-by-step instructions that explain how to use the Active Roles user interface to manage automation workflows. The following topics are covered:

- Creating an automation workflow definition
- Configuring start conditions for an automation workflow
- Adding activities to an automation workflow
- Running an automation workflow on demand
- Viewing run history of an automation workflow
- Terminating a running automation workflow
- Disabling an automation workflow from running
- Re-enabling an automation workflow to run
- Delegating automation workflow tasks

Creating an automation workflow definition

The Active Roles console provides the Workflow Designer for creating and configuring automation workflows. First, you create an automation workflow definition. Then, you use the Workflow Designer to construct an automation workflow, saving the configuration data in the workflow definition.

To create an automation workflow definition

1. In the Active Roles console tree, expand **Configuration | Policies**, right-click **Workflow**, and select **New | Workflow**.
2. Follow the steps in the New Workflow wizard:
    a. On the **Name and Description** page, type in a name and, optionally, a description for the new workflow.
    b. On the **Workflow Type** page, under **This workflow is intended to start**, click **On user demand or on a scheduled basis (automation workflow)**.
    c. On the Completion page, click **Finish**.

Once you have created a workflow definition, you can open it in the Workflow Designer to add workflow activities and specify workflow start conditions.

You can create containers to store related workflows and other containers. To create a workflow container, right-click **Workflow** in the console tree and select **New | Container**. To create an automation workflow definition in a given container, right-click the container in the console tree, and select **New | Workflow**.

You can delete an automation workflow definition as follows: In the console tree under **Configuration | Policies | Workflow**, right-click the object representing the workflow definition, and click **Delete**.

Configuring start conditions for an automation workflow

The start conditions of an automation workflow determine the trigger that causes the workflow to start. You can use a time-based trigger or an event-based trigger to start an automation workflow. It is also possible to allow a workflow to be started on demand. Use the Workflow Designer to view or change the start conditions for an automation workflow.

To view or change the start conditions for an automation workflow

1. In the Active Roles console tree, expand **Configuration | Policies | Workflow**, and select the automation workflow you want to configure.

   This opens the Workflow Designer window in the details pane, representing the automation workflow definition as a process diagram.

2. In the details pane, click the **Workflow options and start conditions** button to expand the area above the process diagram, and then click the **Configure** button.

This opens the **Workflow Options and Start Conditions** page where you can view or change the following:

- The schedule settings that determine the frequency with which to run the workflow. To enable these settings, select the **Run the workflow on a schedule** check box. This causes the workflow to run according to a schedule, and the options below the check box allow you to set the schedule. For details, see Run the workflow on a schedule earlier in this document.
- The workflow can be run on demand. By selecting the **Allow the workflow to be run on demand** check box, you specify that users can manually run the workflow at any time regardless of the schedule. For more information, see Allow the workflow to be run on demand earlier in this document.
- The "run as" options determine the account under which to run the workflow. Click the **"Run as" options** link to view or change the account setting. For details, see "Run as" options earlier in this document.
- Choose whether to terminate the workflow if it runs longer that a certain time period. Click the **Additional settings** link to view or change that setting. For details, see Additional settings earlier in this document.
- Specify parameters to specify certain data when configuring or starting the workflow and then pass that data to workflow activities when the workflow is running. The data is represented as parameter values. To assign a value to a given parameter, click the **Parameters** tab, select the parameter from the list, and then click the **View or change parameter value** button. For more information, see Parameters earlier in this document.

When finished, click **OK** to close the **Workflow Options and Start Conditions** page, and then click **Save Changes** in the Workflow Designer.

Adding activities to an automation workflow

The Active Roles console provides the Workflow Designer for creating and configuring workflows. First, you create a workflow definition. Then, you use the Workflow Designer to construct the workflow by adding and configuring workflow activities.

To add an activity to an automation workflow

1. In the Active Roles console tree, expand **Configuration** | **Policies** | **Workflow**, and select the automation workflow to which you want to add an activity.

   This opens the Workflow Designer window in the details pane, representing the automation workflow definition as a process diagram.

2. In the details pane, drag the activity from the left panel onto the process diagram.
3. Right-click the name of the activity on the process diagram and click **Properties**.
4. Use the **Properties** dialog box to configure the activity.

The steps for configuring an activity depend upon the type of the activity. See topics in the Configuring a workflow section earlier in this document for instructions on how to configure each activity type.

In the **Properties** dialog box, you can change the name and description of the activity. These settings are common to all activities. The name identifies the activity on the process diagram. The description appears as a tooltip when you point to the activity on the process diagram.

You can remove activity from the workflow: Right-click the name of the activity in the process diagram and then clicking **Delete**. This deletes all the configuration settings of the activity from the workflow. It is possible to disable an activity, preserving the activity's configuration settings: Right-click the activity name and click **Disable**. Active Roles does not execute the disabled activities when running the workflow. The ability to disable rather than remove an activity is useful if you plan to temporarily turn off the activity within the workflow. Later, you could easily re-enable a disabled activity by right-clicking its name and then clicking **Enabled**.

Running an automation workflow on demand

An automation workflow can be configured so that users can run it manually regardless of the schedule. This allows them to start the workflow on demand. One can only run an automation workflow on demand if the workflow is enabled and the **Allow the workflow to be run on demand** setting is selected in the workflow start conditions. For information about enabling a workflow, see Re-enabling an automation workflow to run. For instructions on how to view or change workflow start conditions, see Configuring start conditions for an automation workflow.

You can run an automation workflow on demand from the Active Roles console or Web Interface.

To run an automation workflow on demand from the Active Roles console

1. In the Active Roles console tree, under **Configuration | Policies | Workflow**, right-click the desired automation workflow and click **Run**.
2. If prompted, examine or change the values of the workflow parameters.
3. Click **OK** in the confirmation message box.

To run an automation workflow on demand from the Web Interface

1. On the home page in the Web Interface, click **Directory Management**.
2. In the TREE pane, expand the **Workflow** branch and click the container that holds the desired workflow.
3. In the list of the workflow names, to the right of the TREE pane, click the name of the desired workflow.
4. Choose the **Run** command from the menu.
5. If prompted, examine or change the values of the workflow parameters.
6. Click **OK** in the confirmation message box.

Active Roles prompts you for parameter values if the workflow has any parameters that need to be supplied by the user running the workflow on demand. If the workflow has no parameters that require user input, then Active Roles will start the workflow without prompting you for parameter values.

Once you have started an automation workflow, Active Roles opens a run history report, allowing you to examine the progress of workflow execution. The report displays the workflow execution status along with information about the activities performed during workflow run. For a workflow that is in progress you have the option to cancel execution of the workflow by clicking the **Terminate** button.

Viewing run history of an automation workflow

You can use the run history report to examine the running or completed instances of the automation workflow. The report displays the workflow execution status (success or failure) along with the activities that were performed during each workflow run.

After the workflow is completed, the report retains history information about the workflow run. For each completed run of the workflow, the report allows you to identify when and by whom the workflow was started, when the workflow was completed, and what parameter values were used.

The report also lists the workflow activities that were executed during the workflow run. For each activity, you can determine whether the activity was completed successfully or returned an error. In case of error, the report provides an error description. For activities requesting changes to directory data (for example, activities that create new objects or modify existing objects), you can examine the requested changes in detail by clicking the

Operation ID number in the run history report. The report sections have the same contents as with Change History reports (see Workflow activity report sections in the Active Roles Administration Guide).

To view run history of an automation workflow from the Active Roles console

- In the Active Roles console tree, under **Configuration | Policies | Workflow**, right-click the desired automation workflow and click **Run History**.

To view run history of an automation workflow from the Web Interface

1. On the Home page in the Web Interface, click **Directory Management**.
2. In the TREE pane, expand the **Workflow** branch and click the container that holds the desired workflow.
3. In the list of the workflow names, to the right of the TREE pane, click the name of the desired workflow.
4. Choose the **Run History** command from the menu.

Terminating a running automation workflow

You can terminate a running automation workflow to stop the workflow from completing its actions.

To terminate a running automation workflow

- Click the **Terminate** button on the page that displays the automation workflow's run history.

  For instructions on how to access run history, see Viewing run history of an automation workflow.

The **Run History** page displays both running and completed instances of the automation workflow. The **Terminate** button is available on each instance that is currently running. After you click the button to terminate a running instance of an automation workflow, you may experience a delay (up to several minutes) before the workflow shuts down.

Terminating a running automation workflow does not roll back or cancel the workflow activities that have already been performed; this only stops the workflow from running the activities that are in progress or not yet started.

Disabling an automation workflow from running

If you want to prevent an automation workflow from running for a certain period of time, you can disable the workflow. The workflow can be enabled at a later time so that it is allowed to run. For more information, see Re-enabling an automation workflow to run.

To disable an automation workflow from running

- In the Active Roles console tree, under **Configuration | Policies | Workflow**, right-click the desired automation workflow and click **Disable Workflow**.

Re-enabling an automation workflow to run

When an automation workflow is disabled, which prevents the workflow from running, you can re-enable the workflow so that it can be run on demand or when it is scheduled to run.

To re-enable an automation workflow to run

- In the Active Roles console tree, under **Configuration | Policies | Workflow**, right-click the desired automation workflow and click **Enable Workflow**.

Delegating automation workflow tasks

Active Roles provides a number of Access Templates that allow the administrator to delegate the following tasks related to automation workflow:

- **Configure automation workflow**. To perform this task, the delegated administrator needs full control of automation workflow definitions, including the rights to add, configure, and remove workflow activities, view and change the workflow start conditions, add and remove workflow parameters, and assign values to workflow parameters.
- **Run automation workflow**. To perform this task, the delegated administrator needs the rights to view the definition of an automation workflow, run the automation workflow on demand, and view run history of the automation workflow.
- **View run history**. To perform this task, the delegated administrator needs the rights to view the definition of an automation workflow, and view run history reports on running and completed instances of the automation workflow.

This section provides instructions on how to delegate these tasks to regular users or groups that do not have administrator rights in Active Roles.

Allowing access to workflow containers

Automation workflow tasks require access to containers that hold workflow definition objects. By default, Active Roles allows any authenticated user to view the **Configuration/Policies/Workflow** container itself. You can enable appropriate users or groups to view containers held in the **Workflow** container by applying the **Workflow - View Workflow Containers** Access Template to that container.

To enable users or groups to view workflow containers

1. In the console tree, expand **Configuration | Policies**, right-click the **Workflow container**, and then click **Delegate Control**.
2. In the **Active Roles Security** dialog box, click **Add** to start the Delegation of Control Wizard.
3. On the **Users or Groups** page in the wizard, click **Add**, and then use the **Select Objects** dialog box to select the desired users or groups.
4. On the **Access Templates** page in the wizard, under **Access Templates | Configuration**, select the **Workflow - View Workflow Containers** check box.
5. Follow the instructions in the wizard and accept the default settings.
6. Click **OK** in the **Active Roles Security** dialog box.

Delegating full control of automation workflows

By giving full control of an automation workflow to a user or group, you authorize the user or group to perform the following tasks:

- View the workflow definition.
- Make any changes to the workflow.
- Run the workflow.
- View the workflow run history reports.

You can delegate full control of all automation workflows held in a certain container by applying the **Automation Workflow - Full Control** Access Template to that container.

To delegate full control of all automation workflows held in a certain container

1. In the console tree, right-click the desired container under **Configuration | Policies | Workflow**, and then click **Delegate Control.**
2. In the **Active Roles Security** dialog box, click **Add** to start the Delegation of Control Wizard.
3. On the **Users or Groups** page in the wizard, click **Add**, and then use the **Select Objects** dialog box to select the desired users or groups.
4. On the **Access Templates** page in the wizard, under **Access Templates | Configuration**, select the **Automation Workflow - Full Control** check box.
5. Follow the instructions in the wizard and accept the default settings.
6. Click **OK** in the **Active Roles Security** dialog box.

It is also possible to delegate full control of a single automation workflow by applying the Access Template to the workflow definition object.

To delegate full control of a single automation workflow

1. On the **View** menu, select **Advanced Details Pane**.
2. In the console tree, under **Configuration | Policies | Workflow**, select the container that holds the desired workflow definition object.
3. In the upper part of the details pane, select the workflow definition object.
4. In the lower part of the details pane, on the **Active Roles Security** tab, right-click a blank area and click **Add** to start the Delegation of Control Wizard.
5. On the **Users or Groups** page in the wizard, click **Add**, and then use the **Select Objects** dialog box to select the desired users or groups.
6. On the **Access Templates** page in the wizard, under **Access Templates | Configuration**, select the **Automation Workflow - Full Control** check box.
7. Follow the instructions in the wizard and accept the default settings.

Delegating the task of running automation workflow

You can authorize users or groups to run all automation workflows held in a certain container by applying the **Automation Workflow - View and Run** Access Template to that container. This allows the users or groups to run the automation workflow without giving them the right to make any changes to the workflow.

To delegate the task of running all automation workflows held in a certain container

1. In the console tree, right-click the desired container under **Configuration | Policies | Workflow**, and then click **Delegate Control.**
2. In the **Active Roles Security** dialog box, click **Add** to start the Delegation of Control Wizard.
3. On the **Users or Groups** page in the wizard, click **Add**, and then use the **Select Objects** dialog box to select the desired users or groups.
4. On the **Access Templates** page in the wizard, under **Access Templates | Configuration**, select the **Automation Workflow - View and Run** check box.
5. Follow the instructions in the wizard and accept the default settings.
6. Click **OK** in the **Active Roles Security** dialog box.

It is also possible to authorize users or groups to run a single automation workflow by applying the Access Template to the workflow definition object.

To delegate the task of running a single automation workflow

1. On the **View** menu, select **Advanced Details Pane**.
2. In the console tree, under **Configuration | Policies | Workflow**, select the container that holds the desired workflow definition object.
3. In the upper part of the details pane, select the workflow definition object.

4. In the lower part of the details pane, on the **Active Roles Security** tab, right-click a blank area and click **Add** to start the Delegation of Control Wizard.
5. On the **Users or Groups** page in the wizard, click **Add**, and then use the **Select Objects** dialog box to select the desired users or groups.
6. On the **Access Templates** page in the wizard, under **Access Templates | Configuration**, select the **Automation Workflow - View and Run** check box.
7. Follow the instructions in the wizard and accept the default settings.

Delegating the task of viewing run history of automation workflow

You can authorize users or groups to view run history of all automation workflows held in a certain container by applying the **Automation Workflow - View** Access Template to that container. This enables the users or groups to view run history of the automation workflow without giving them the right to modify or run the workflow.

To delegate the task of viewing run history of all automation workflows held in a certain container

1. In the console tree, right-click the desired container under **Configuration | Policies | Workflow**, and then click **Delegate Control.**
2. In the **Active Roles Security** dialog box, click **Add** to start the Delegation of Control Wizard.
3. On the **Users or Groups** page in the wizard, click **Add**, and then use the **Select Objects** dialog box to select the desired users or groups.
4. On the **Access Templates** page in the wizard, under **Access Templates | Configuration**, select the **Automation Workflow - View** check box.
5. Follow the instructions in the wizard and accept the default settings.
6. Click **OK** in the **Active Roles Security** dialog box.

It is also possible to authorize users or groups to view run history of a single automation workflow by applying the Access Template to the workflow definition object.

To delegate the task of viewing run history of a single automation workflow

1. On the **View** menu, select **Advanced Details Pane**.
2. In the console tree, under **Configuration | Policies | Workflow**, select the container that holds the desired workflow definition object.
3. In the upper part of the details pane, select the workflow definition object.
4. In the lower part of the details pane, on the **Active Roles Security** tab, right-click a blank area and click **Add** to start the Delegation of Control Wizard.
5. On the **Users or Groups** page in the wizard, click **Add**, and then use the **Select Objects** dialog box to select the desired users or groups.

6. On the **Access Templates** page in the wizard, under **Access Templates | Configuration**, select the **Automation Workflow - View** check box.
7. Follow the instructions in the wizard and accept the default settings.

Sample Azure Hybrid Migration

To create a Remote Mailbox for an existing user, you can use the Office 365 workflow and modify the sample script Sample Azure Hybrid Migration.ps1 available in `Configuration\Script Modules\Builtin\ Sample Azure Hybrid Migration location`. The workflow for remote mailbox is available in `Configuration\Policies\Workflow\Builtin\ Sample Azure Hybrid Migration` location.

o determine the sequence of actions to create a remote mailbox, the state of the user's mailbox (migrated, non-migrated, mail-enabled user's, and so on) must be considered. Depending upon the environment in which the remote mailbox is intended to work, select either of these two options:

- **EnableRemoteMailBox** function to enable remote mailboxes for the users in the workflow scope. Select **EnterExchangeCreds_params** as the function to declare parameters in the script and provide the Exchange username and password for running **EnableRemoteMailBox** function in workflow.
- **DisableRemoteMailBox** function to disable remote mailboxes for the users in the workflow scope. Select **EnterExchangeCreds_params** as the function to declare parameters in the script and provide the Exchange username , password and Exchange Recipient Type Details for running **DisableRemoteMailBox** function in workflow.

For more information on declaring script parameters, see Script activity. In the script, specify the exchange server FQDN and modify the required code blocks.

After the script is modified, enable or copy the default Sample Azure Hybrid Migration workflow and run.

By default, a remote mailbox is created for users with valid exchange online license and no exchange mailbox on-premise presence. For more information on creating a Remote Mailbox for new users, see Create a new Hybrid user using web interface.

NOTE:

- The Exchange management tools of the on-premise Exchange 2013 or later must be present to enable remote mailbox.
- The 'Remote mailbox migration (RemoteMailbox.ps1)' script has been provided as a sample script only, to illustrate the steps required, and should not be used as-is in a production situation without modification and enhancement. The use of security credentials within a script in clear text should never be considered appropriate or secure. In testing this script, care and consideration should be given to the authentication and use of credentials, and clear text credentials should not be left in the script once testing is complete.
- ARS service account must be a part of the **Recipient Management group** to run exchange hybrid commands.

For more details refer the KB article: https://support.oneidentity.com/kb/310525 .

Managing Remote Mailbox

After creating the Remote Mailbox, you can manage it through the console and the Web Interface. The supported operations are mentioned below:

- **Exchange General**
  - View or change the alias
  - View or change the option to use MAPI rich text format
  - Hide the user or contact from Exchange address lists
  - View or change custom attributes
- **Exchange Advanced**
  - View or change the simple display name
  - Downgrade high priority mail bound for X.400.
  - View or change the Internet Locator Service (ILS) settings
- **Email Address**
  - View, add, edit or remove e-mail addresses
  - View or change the default reply address for each address type
  - View or change the external e-mail address
  - Set the option to update e-mail addresses based on e-mail address policy
- **Mail flow Settings**
  - View or change message size restrictions and message delivery restrictions

For more information on Exchange Online Properties, see *View or modify the Exchange Online properties* on the *Active Roles Administration Guide*.

Office 365 automation workflow

Automation workflow enhancements allow you to add Azure or Office 365 modules in PowerShell and run the Office 365 services such as Skype for Business, Azure AD, and Azure PowerShell Az PowerShell scripts within existing Active Roles workflows.

As part of creating automation workflow, a new **O365ScriptExecutionConfiguration** activity is added to fetch the already configured tenant details. Select a tenant from the **Select a Tenant to configure O365 Services** drop-down list that displays all the tenants configured on the Active Roles Web Interface.

Selected tenant is displayed in **This tenant will be used to configure O365 Services if the override tenant details is not provided** textbox.

However, if required, you can override Office 365 script execution. Double-click on **O365ScriptExecutionConfiguration** and edit the O365 User and O365 Password workflow parameters.

NOTE: By default, the tenant is not selected.

If no tenant is selected and no override tenant details are provided, the workflow fails and the following error message appears:

```
Please select a configured Tenant from the Select a Tenant to configure O365
Services drop-down or provide valid Tenant Id, Login username and password to
override details from workflow.
```

If the tenant is selected in the drop-down and the override details are also provided, the O365 services will be configured using the override details.

Prerequisites to run the Office 365 workflow scripts

The following PowerShell modules must be installed to the system running Active Roles to run the Office 365 PowerShell script execution activity.

- Azure Active Directory (Azure AD)
- Azure PowerShell Az
- Microsoft Teams (includes Skype for Business)

NOTE: To run a Sample Azure Hybrid Migration script, on-premises Exchange must be available.

As Exchange Online is connected to Exchange Online PoweShell, make sure that https://outlook.office365.com/powershell-liveid/ is not be blocked and network connectivity is available.

Creating an Office 365 automation workflow definition

To create an Office 365 automation workflow definition

1. In the Active Roles console tree, expand **Configuration** > **Policies**, right-click **Workflow**, and select **New** > **Workflow**.
2. Follow the steps in the New Workflow wizard:
   a. On the **Name and Description** page, type in a name and, optionally, a description for the new workflow.
   b. On the **Workflow Type** page, under **This workflow is intended to start**, click **On user demand** or on a scheduled basis (automation workflow).
   c. On the **Completion** page, click **Finish**.
3. From the tool box within the workflow, select **O365ScriptExecutionConfiguration** and configure the Office 365 automation workflow.
4. From the workflow, select the Script activity and configure the script activity to point to the PowerShell script and function to execute.

   NOTE: The string or command entered in the API O365ExecuteScriptCmd must be well formed and complete for the workflow to run successfully.

Sample Office 365 workflow scripts

$context.O365ImportModules(array of modules)

This script enables the multiple modules specified in the array to be loaded and creates a connection to the modules with the login details retrieved through **O365ScriptExecutionConfiguration** activity. The **O365 User** and **O365 Password** parameters that are configured earlier within workflow are used.

Example:

```
$array = @("MSOnline", "Lync")
       $context.O365ImportModules($array)
       $context.O365RemoveAllModulesSessions()
```

$context.O365ImportModule (module)

This script enables the single module specified in the script to be loaded.

Example:

```
$context.O365ImportModule("AzureRm) #latest version azureRm is imported
        $context.O365ImportModule("AzureRm, 4) #latest version azureRm version 4 is
imported
        $context.O365RemoveAllModulesSessions()
```

$context.O365ExecuteScriptCmd(string/cmd)

Passing any string or command specified in the script will execute and return the results as PSObject.

$context.O365RemoveAllModulesSessions()

This script disconnects the PSSessions and removes the modules from the PowerShell pool, allowing to import new modules again.

Example:

```
#Get a list of disabled users and Directory Roles available
        $_usersinroles= @()
        $_default_log = "C:\temp\Roles.csv"
        $context.O365ImportModule("AzureAD")
        $context.O365ExecuteScriptCmd("get-azureaduser -filter 'accountEnabled eq
false'" +" | Export-Csv " +"c:\temp\DisabledUsers.csv" +" -NoTypeInformation")
        $context.O365ExecuteScriptCmd("Get-AzureADDirectoryRole | Export-csv "+$_
default_log )
        $context.O365RemoveAllModulesSessions()
```

Creating Office 365 shared mailbox

To create a new Office 365 shared mailbox, use **Create Office 365 Shared Mailboxes** builtin workflow. This workflow uses the builtin activity **O365 script execution configuration** and the builtin script **Create Office 365 Shared Mailboxes**.

The workflow is disabled by default and can be used as an initial template to create custom workflows with required values in the script, such as, mailbox name, mailbox display name, alias, and recipients to grant the **Send As** permission.

The workflow is located in **Configuration/Policies/Workflow/Builtin** container in the Active Roles console, and has the name **Create Office 365 shared mailbox**. The corresponding script is available in **Configuration/Policies/Script Modules/Builtin** container, with the name **Create Office 365 shared mailbox**.

Enabling Azure Roles

To enable an existing directory role in Azure Active Directory, use **Enabling Azure Roles** builtin workflow. This workflow uses the builtin activity **O365 script execution configuration** and the builtin script **Enabling Azure Roles**. The workflow is disabled by default and can be used as an initial template to create custom workflows with required values in the script, such as, the directory role display name.

The workflow is located in **Configuration/Policies/Workflow/Builtin** container in the Active Roles console, and has the name **Enabling Azure Roles**. The corresponding script

is available in **Configuration/Policies/Script Modules/Builtin** container, with the name **Enabling Azure Roles**.

Activity extensions

In Active Roles, administrators can configure workflow activities of the pre-defined types that are installed with Active Roles. By default, the list of activities in the Workflow Designer contains only the pre-defined activity types, such as **Approval Activity** or **Notification Activity**. It is possible to extend the list by adding new types of activity.

Each activity type determines a certain workflow action (for example, originating an approval task or notification) together with a collection of activity parameters to configure the workflow action (for example, parameters that specify the approvers or notification recipients). Active Roles builds upon this concept, providing the ability to implement and deploy custom types of workflow activity. It enables custom activity types to be created as necessary, and listed in the Workflow Designer along with the pre-defined activity types, allowing administrators to configure workflow activities that perform custom actions determined by those new types of workflow activity.

Active Roles allows the creation of custom activities based on the **Script Activity** built-in activity type. However, creating and configuring a script activity from scratch can be time-consuming. Custom activity types provide a way to mitigate this overhead. Once a custom activity type is deployed that points to a particular script, administrators can easily configure and apply workflow activities of that type, having those activities perform the actions determined by the script. The activity script also defines the activity parameters specific to the activity type.

Custom activity types provide an extensible mechanism for deploying custom workflow activities. This capability is implemented by using the Policy Type object class. Policy Type objects can be created by using the Active Roles console, with each object representing a certain type of custom workflow activity.

Design elements

The extensibility of workflow activity types is designed around two interactions: activity type deployment and activity type usage.

Activity type deployment

The deployment process involves the development of a script that implements the workflow action and declares the activity parameters the creation of a Script Module containing that script and the creation of a Policy Type object referring to that Script Module. To deploy an activity type to a different environment, you can export the activity type to an export file in the source environment and then import the file in

the destination environment. The use of export files makes it easy to distribute custom activity types.

Activity type usage

This is the process of configuring workflow activities. It occurs whenever you add an activity to a workflow in the Workflow Designer. To add an activity to a workflow, you drag the desired activity type from the toolbox onto the workflow process diagram. The toolbox, located on the left of the diagram, lists all the activity types defined in Active Roles, including the custom activity types. For each activity of a custom type the Workflow Designer provides a page for configuring the activity parameters specific to that activity type. Once the activity parameters have been configured, the workflow contains a fully functional activity of the selected custom type.

Active Roles provides a graphical user interface, complete with a programming interface, for creating and managing custom activity types. Using those interfaces, Active Roles workflows can be extended to meet the needs of a particular environment. Active Roles also has a deployment mechanism by which administrators put new types of workflow activity into operation.

Since workflow activity extension involves two interactions, Active Roles provides solutions in both areas. The Administration Service maintains activity type definitions, exposing activity types to its clients such as the Active Roles console or ADSI Provider. The console can be used to:

- Create a new custom activity type, either from scratch or by importing an activity type that was exported from another environment.
- Make changes to the definition of an existing custom activity type.
- Add an activity of a particular custom type to a workflow, making the necessary changes to the activity parameters provided for by the activity type definition.

Normally, an Active Roles expert develops a custom activity type in a separate environment, and then exports the activity type to an export file. An Active Roles administrator deploys the activity type in the production environment by importing the export file. After that, the Workflow Designer can be used to configure and apply activities of the new type.

Policy Type objects

The extensibility of workflow activity types builds upon Policy Type objects of the workflow activity category, each of which represent a single type of workflow activity. Policy Type objects are used within both the activity type deployment and activity type usage processes. The process of deploying a new activity type involves the creation of a Policy Type object. During the process of adding an activity of a custom type to a workflow, the activity type definition is retrieved from the respective Policy Type object.

Each Policy Type object of the workflow activity category holds the following data to define a single activity type:

- **Display name**. Identifies the activity type in the Workflow Designer. This name is displayed in the activities toolbox located on the left of the workflow process diagram.
- **Description**. A text describing the activity type. This text is used as a default description for every activity that is based on this Policy Type object.
- **Reference to Script Module**. Identifies the Script Module that will be used by the workflow activities of this type. When adding an activity of a custom type to a workflow, you effectively create an activity that runs a certain script function from the Script Module specified by the respective Policy Type object.
- **Policy Type category**. The Policy Type objects that define custom workflow activities fall in a separate policy type category named "workflow activity."
- **Workflow category**. Determines whether the custom activity can be used in change workflows only, automation workflows only, or both change and automation workflows.
- **Function to run**. Identifies the script function that is run by the workflow activities of this type. The function must exist in the Script Module selected for the policy type.
- **Function to declare parameters**. Identifies the script function that declares the parameters for the workflow activities of this type. The function must exist in the Script Module selected for the policy type. By default, it is assumed that the parameters are declared by the function named onInit.
- **Policy Type icon**. The image that appears next to the display name of the activity type in the Workflow Designer, to help identify and visually distinguish this activity type from the other types of workflow activity.

To create a custom activity type, first create a Script Module that holds the script function that will be run by the workflow activities of that type. Then, you can create a Policy Type object referring to that Script Module. When you import an activity type, Active Roles automatically creates both the Script Module and the Policy Type object for that activity type. After the Policy Type object has been created, you can add an activity of the new type to a workflow.

Creating and managing custom activity types

In Active Roles, Policy Type objects provide the ability to store the definition of a custom activity type in a single object. Policy Type objects can be exported and imported, which makes it easy to distribute custom workflow activities to other environments.

In the Workflow Designer, an administrator is presented with a list of activity types derived from the Policy Type objects. Selecting a custom activity type from the list causes Active Roles to create a workflow activity based on the settings found in the respective Policy Type object.

This section covers the following tasks specific to custom activity types:

- Creating a Policy Type object
- Changing an existing Policy Type object
- Using Policy Type containers
- Exporting activity types
- Importing activity types
- Configuring an activity of a custom type
- Deleting a Policy Type object

For more information about Policy Type objects, including instructions on scripting for Policy Type objects, refer to the Active Roles SDK documentation.

Creating a Policy Type object

Active Roles stores Policy Type objects in the **Policy Types** container. You can access that container in the Active Roles console by expanding the **Configuration/Server Configuration** branch of the console tree.

To create a new Policy Type object

1. In the console tree, under **Configuration/Server Configuration/Policy Types**, right-click the Policy Type container in which you want to create a new object, and select **New | Policy Type**.

   For example, if you want to create a new object in the root container, right-click **Policy Types**.

2. In the **New Object - Policy Type** wizard, type a name, a display name and, optionally, a description for the new object.

   The display name identifies the activity type in the Workflow Designer. The description text is used as a default description for every activity that is based on this Policy Type object.

3. Click **Next**.
4. Click **Browse** and select the Script Module containing the script that will be used by the workflow activities of this type.

   The Script Module must exist under the **Configuration/Script Modules** container.

5. In the **Policy Type category** area, select the **Workflow activity** option.
6. From the **Function to run** list, select the name of the script function that will be run by the workflow activities of this type.

   The list contains the names of all the functions found in the script you selected in Step 4. Every activity of this type will run the function you select from the **Function to run** list.

7. From the **Use in** list, select the appropriate option to indicate the category of the workflow in which the activity of this type can be used:

- **Change workflow**. The activity can be used only in change workflows, that is, workflows intended to run upon operation requests that meet certain conditions.
- **Automation workflow**. The activity can be used only in automation workflows, that is, workflows intended to run on a scheduled basis or on user demand.
- **Any workflow**. The activity can be used in both change and automation workflows.

8. From the **Function to declare parameters** list, select the name of the script function that defines the parameters specific to this type of workflow activity.

   The list contains the names of all the functions found in the script you selected in Step 4. Every activity of this type will have the parameters that are specified by the function you select from the **Function to declare parameters** list. Normally, this is a function named onInit (see Active Roles SDK for details).

9. Click **Policy Type Icon** to verify the image that denotes this type of activity. To choose a different image, click **Change** and open an icon file containing the image you want.

   This image appears next to the display name of the activity type in the Workflow Designer, to help identify and visually distinguish this activity type from the other activity types.

   The image is stored in the Policy Type object. In the dialog box that appears when you click **Policy Type Icon**, you can view the image that is currently used. To revert to the default image, click **Use Default Icon**. If the button is unavailable, then the default image is currently used.

10. Click **Next** and follow the steps in the wizard to complete the creation of the new Policy Type object.

Changing an existing Policy Type object

You can change an existing Policy Type object by changing the general properties, script, or icon. The general properties include the name, display name, and description. The Policy Type objects are located under **Configuration/Server Configuration/Policy Types** in the Active Roles console.

The following table summarizes the changes you can make to an existing Policy Type object, assuming that you have found the object in the Active Roles console.

Table 62: Policy Type object changes

To change	Do this	Commentary
Name of the object	Right-click the object and click **Rename**.	The name is used to identify the object, and must be unique among the objects held in the same Policy Type container.

To change	Do this	Commentary
Display name or description	Right-click the object, click **Properties** and make the necessary changes on the **General** tab.	Changing the display name also changes the name of the activity type in the Workflow Designer. You may need to refresh the view in the Workflow Designer for the new name to be displayed.
Script Module	Right-click the object, click **Properties**, click the **Script** tab, click **Browse**, and then select the Script Module you want.	You can change the script in the Script Module that is currently associated with the Policy Type object instead of selecting a different Script Module. To view or change the script, find and select the Script Module in the Active Roles console tree, under **Configuration/Script Modules**. Changing the script affects all the existing workflow activities of this type. If you add an activity to a workflow and then change the script for the Policy Type object based on which the activity was created, the activity will run the changed script.
Function to run	Right-click the object, click **Properties**, click the **Script** tab, and then choose the appropriate function from the **Function to run** list.	Changing this setting causes the activities of this type to run function you have selected. Changing the function does not affect the existing activities of this type. If you add a new activity of this type, the activity will run the new function.
Workflow category	Right-click the object, click **Properties**, click the **Script** tab, and then choose the appropriate option from the **Use in** list.	This setting determines the workflow category (change workflow, automation workflow, or any workflow) in which the activity of this type is allowed. After you have changed this setting, an activity of this type can only be added to the corresponding workflow category. Thus, if you select the **Change workflow** option, the activity of this type cannot be added to an automation workflow.
Function to declare parameters	Right-click the object, click **Properties**, click the **Script** tab, and then choose the appropriate function from the **Function to declare parameters** list.	Changing this setting changes the list of the activity parameters specific to this activity type. The changes do not affect the parameters of the existing activities of this type. When you add a new activity of this type, the list of the activity

To change	Do this	Commentary
		parameters is built using the new function to declare parameters.
Policy Type icon	Right-click the object, click **Properties**, click the **Script** tab, click **Policy Type Icon**, and then do one of the following: • Click **Change** and open an icon file containing the image you want. • Click **Use Default Icon** to revert to the default image.	Changing this setting changes the image that appears next to the display name of the activity type in the Workflow Designer, on the pane located next to the workflow process diagram.

Using Policy Type containers

You can use a Policy Type container to store related Policy Type objects and other Policy Type containers.

Containers provide a means for additional categorization of custom activity types, making it easier to locate and select an activity type in the Workflow Designer. The activities toolbox next to the workflow process diagram lists the custom activity types along with the containers that hold the respective Policy Type objects. To prevent containers from cluttering the activities toolbox, the Workflow Designer displays only the containers that are direct descendants of the **Policy Types** container, and disregards the lower-level containers. To clarify this behavior, let us consider a path to a Policy Type object such as Policy Types/Container A/Container B/Object C. In this case, the Workflow Designer only displays Container A and the activity type C under Container A, disregarding Container B.

To create a new Policy Type container

1. In the console tree, under **Configuration/Server Configuration/Policy Types**, right-click the Policy Type container in which you want to create a new container, and select **New | Policy Type Container**.

   For example, if you want to create a new container in the root container, right-click **Policy Types**.

2. In the **New Object - Policy Type Container** wizard, type a name and, optionally, a description for the new container.

   The name of the container will be displayed in the Workflow Designer if the container is located directly in the **Policy Types** container.

3. Click **Next** and follow the steps in the wizard to complete the creation of the new container.

Exporting activity types

You can export Policy Type objects so that the definition of the activity types is stored in an XML file which can be imported in a different Active Roles environment. Exporting and then importing Policy Type objects make it easy to distribute custom activity types to other environments.

To export a Policy Type object or container

- Right-click the Policy Type object or container in the Active Roles console, click **Export** and then specify an XML file to hold the export data.

You can select multiple Policy Objects to export, or you can select a container to export all Policy Type objects and containers held in that container. In either case, the Export operation creates a single XML file that can later be imported to any container under the **Policy Types** node.

Export of Policy Type objects creates an XML file representing both the objects and the Script Modules containing the scripts for each activity type being exported. During an import, Active Roles creates the Policy Type objects and the Script Modules based on the data found in the XML file. As a result of the import, the activity types are replicated to the new environment and can be used the same way as in the environment from which they were exported.

Importing activity types

You can import the exported Policy Type objects and containers, which will add them to a Policy Type container and allow you to configure and use custom activities defined by those Policy Type objects. All the data required to deploy the activity types is represented in an XML file. To see an example of the XML document that represents an activity type, export a Policy Type object and view the saved XML file.

To import the exported Policy Type objects and containers

1. In the Active Roles console tree, under **Configuration/Server Configuration/Policy Types**, right-click the Policy Type container in which you want to import the exported Policy Type objects and containers.
2. Click **Import Policy Types**, and then open the XML file you want to import.

This will create new Policy Type objects and containers in the selected container. In addition, new Script Modules will be created in the **Configuration/Script Modules** container and associated with the newly created Policy Type objects.

Configuring an activity of a custom type

Once a custom activity type has been deployed, an Active Roles administrator can add an activity of that type to a workflow. This is accomplished by dragging the activity type onto the workflow process diagram in the Workflow Designer.

To configure a workflow activity of a custom type

1. In the Active Roles console tree, expand **Configuration | Policies | Workflow**, and select the workflow to which you want to add an activity.

   This opens the Workflow Designer window in the details pane, representing the workflow definition as a process diagram.

2. In the details pane, drag the activity type from the left panel onto the process diagram.

   The panel on the left of the workflow process diagram lists all the activity types defined in your Active Roles environment. The built-in activity types are listed in the **Basic** area, along with the custom activity types whose Policy Type objects are located directly in the **Policy Types** container. The other custom activity types are listed below the names of the containers that hold the corresponding Policy Type objects. The list includes only those containers that are located directly in the **Policy Types** container. The names of the intermediate containers are not shown.

3. Right-click the name of the activity you have added on the process diagram, and then click **Properties**.

4. On the **Properties** page, set parameter values for the activity: Click the name of a parameter in the list, and then click **Edit**.

   Parameters control the behavior of the activity. When Active Roles executes the activity, it passes the parameter values to the script function. The actions performed by the script function, and the results of those actions, depend upon the parameter values.

   Clicking **Edit** displays a page where you can add, remove, or select a value or values for the selected parameter. For each parameter, the script being used by the activity defines the name of the parameter and other characteristics, such as a description, a list of possible values, the default value, and whether a value is required. If a list of possible values is defined, then you can only select values from that list.

5. Click **OK** to close the **Properties** dialog box, and then click **Save Changes** in the Workflow Designer.

Deleting a Policy Type object

You can delete a Policy Type object when you no longer need to add activities of the type defined by that object.

Before you delete a Policy Type object, consider the following:

- You can delete a Policy Type object only if no activities of the respective type exist in any workflow. Examine each workflow definition and remove the activities of that type, if any, from the workflow before deleting the Policy Type object.
- Deleting a Policy Type object permanently deletes it from the Active Roles database. If you want to use this activity type again, you should export the Policy Type object to an XML file before deleting the object.
- Deleting a Policy Type object does not delete the Script Module associated with that object. This is because the Script Module may be used by other activities. If the Script Module is no longer needed, it can be deleted separately.

To delete a Policy Type object

- Right-click the Policy Type object in the Active Roles console and click **Delete**.

8

Temporal Group Memberships

- Understanding temporal group memberships
- Using temporal group memberships

Understanding temporal group memberships

By using temporal group memberships, Active Roles provides the ability to automate the tasks of adding or removing group members that only need group membership for a specific time period. When adding objects, such as users, computers or groups, to a particular group, an administrator can specify that the objects should be added to the group at the time of choice, as well as indicate when those objects should be removed from the group.

The temporal group membership functionality offered by Active Roles can aid organizations in efficiently assigning users and other objects to groups for a required period of time. Although in many cases objects that are added to a group remain the members of the group for an indefinite period of time, many organizations have requirements of temporarily assigning objects to particular groups. Typical scenarios include allowing access to specific resources for the duration of a certain project, or temporarily allowing an individual to act as a server administrator.

Management of temporal group assignments represents significant challenges for administrators since a high degree of administrative oversight is required to ensure that the group assignments are truly temporary and do not become permanent because of poor control over group memberships. Active Roles addresses these requirements by enabling addition or removal of group members to occur automatically on a scheduled basis.

The temporal group membership functionality expands the benefits of Active Roles in the following areas:

- **Security** By providing tight control over changes to group memberships, including policy-based rules and constraints, change approval, and change auditing, Active Roles reduces security risks for systems, applications and services that use Active Directory groups for access authorization. Adding and removing group members in a

timely manner ensure that users have access to systems and resources for only the required amount of time, thereby restricting the possibility and scope of access.

- **Availability** By automatically populating groups based on configurable policy rules, Active Roles makes appropriate network resources available to appropriate users at the time that they need access to those resources. The ability to set a schedule for adding and removing group members is helpful in situations where temporary access is required for a relatively short time period or when numerous requests to change group memberships arise on short notice.
- **Manageability** Active Roles streamlines the management of assigning users to groups as well as removal of members from groups. Consistent and reliable control of these provisioning and de-provisioning activities reduces overhead for those managing Active Directory groups. Unattended, schedule-based handling of temporal group memberships helps assure compliance with change and access policies while simplifying the management of group membership change requests.
- **Compliance** Active Roles lowers regulatory compliance risks by ensuring that proper and effective controls are in place for group memberships. Since Active Directory groups are used to authorize access to systems, applications and data, controlling the assignment of users to groups on a temporal basis helps organizations comply with separation of duties and data privacy requirements.

Active Roles provides the temporal group membership functionality for both Active Directory Domain Services (AD DS) and Active Directory Lightweight Directory Services (AD LDS).

The temporal group membership functionality automates the tasks of adding and removing users from groups in the situations where users need group memberships for only a specific time period. By applying temporal membership settings, administrators can schedule selected objects to be assigned to a particular group and specify when the objects are to be removed from the group.

The key capabilities provided by Active Roles for managing temporal group memberships are as follows:

- **Add temporal group members** The user interface for selecting objects, in both the Active Roles console and Web Interface, provides a number of options to specify when the selected objects should be added to the selected group and when the selected objects should be removed from the group. It is possible to add the objects to the group immediately as well as to indicate that the objects should not be removed from the group.
- **View temporal members of a group** The list of group members (the **Members** page) displayed by the Active Roles console or Web Interface makes it possible to distinguish between regular group members and temporal group members. In addition, it is possible to hide or display the temporal members that are scheduled to be added to the group in the future but are not actual members of the group so far.
- **View temporal memberships of an object** The list of group memberships for a particular object (the **Member Of** page) makes it possible to distinguish between the groups in which the object is a regular member and the groups in which the object is a temporal member. It is also possible to hide or display the groups to which the object is scheduled to be added in the future.

- **Reschedule temporal group memberships** Both the **Members** and **Member Of** pages provide the ability to view or modify the temporal membership settings. On the **Members** page for a particular group, you can select a member, and view or modify the date and time when the member should be added or removed from the group. On the **Member Of** page for a particular object, you can select a group, and view or modify the date and time when the object should be added or removed from the group.
- **Make a temporal member permanent** The temporal membership settings provide the option to indicate that the object should not be removed from the group, thus making a temporal member permanent. If temporal membership settings on a particular object are configured to add the object to a certain group immediately and never remove it from the group, then the object becomes a regular member of that group. Similarly, specifying any other temporal membership settings on a regular member converts it to a temporal member.
- **Remove temporal group members** Both the **Members** and **Member Of** pages provide the Remove function for group memberships, whether temporal or regular. When you use the Remove function on temporal members of a group, the members are removed along with all the temporal membership settings that were in effect on those members. The same is true when you use the Remove function on groups in which a particular object is a temporal member.

With the temporal group membership functionality, Active Roles assures that users have group memberships for only the time they actually need to, enforcing the temporal nature of group memberships when required and eliminating the risk of retaining group memberships for longer than needed.

Using temporal group memberships

By using temporal group memberships, you can manage group memberships of objects such as user or computer accounts that need to be members of particular groups for only a certain time period. This feature of Active Roles gives you flexibility in deciding and tracking what objects need group memberships and for how long.

This section guides you through the tasks of managing temporal group memberships in the Active Roles console. If you are authorized to view and modify group membership lists, then you can add, view and remove temporal group members as well as view and modify temporal membership settings on group members.

Adding temporal members

A temporal member of a group is an object, such as a user, computer or group, scheduled to be added or removed from the group. You can add and configure temporal members using the Active Roles console.

To add temporal members of a group

1. In the Active Roles console, right-click the group and click **Properties**.
2. On the **Members** tab in the **Properties** dialog box, click **Add**.
3. In the **Select Objects** dialog box, click **Temporal Membership Settings**.
4. In the **Temporal Membership Settings** dialog box, choose the appropriate options, and then click **OK**:
   a. To have the temporal members added to the group on a certain date in the future, select **On this date** under **Add to the group**, and choose the date and time you want.
   b. To have the temporal members added to the group at once, select **Now** under **Add to the group**.
   c. To have the temporal members removed from the group on a certain date, select **On this date** under **Remove from the group**, and choose the date and time you want.
   d. To retain the temporal members in the group for indefinite time, select **Never** under **Remove from the group**.
5. In the **Select Objects** dialog box, type or select the names of the objects you want to make temporal members of the group, and click **OK**.
6. Click **Apply** in the **Properties** dialog box for the group.

NOTE:

- To add temporal members of a group, you must be delegated the authority to add or remove members from the group. The appropriate authority can be delegated by applying the **Groups - Add/Remove Members** Access Template.
- You can make an object a temporal member of particular groups by managing properties of the object rather than properties of the groups. Open the **Properties** dialog box for that object, and then, on the **Member Of** tab, click **Add**. In the **Select Objects** dialog box, specify the temporal membership settings and supply the names of the groups as appropriate for your situation.

Viewing temporal members

The list of group members displayed by the Active Roles console makes it possible to distinguish between regular group members and temporal group members. It is also possible to hide or display so-called *pending members*, the temporal members that are scheduled to be added to the group in the future but are not actual members of the group so far.

To view temporal members of a group

1. In the Active Roles console, right-click the group and click **Properties**.
2. Examine the list on the **Members** tab in the **Properties** dialog box:
    - An icon of a small clock overlays the icon for the temporal members.
    - If the **Show pending members** check box is selected, the list also includes the temporal members that are not yet added to the group. The icons identifying such members are shown in orange.

The list of group memberships for a particular object makes it possible to distinguish between the groups in which the object is a regular member and the groups in which the object is a temporal member. It is also possible to hide or display so-called *pending group memberships*, the groups to which the object is scheduled to be added in the future.

To view groups in which an object is a temporal member

1. In the Active Roles console, right-click the object and click **Properties**.
2. Examine the list on the **Member Of** tab in the **Properties** dialog box:
    - An icon of a small clock overlays the icon for the groups in which the object is a temporal member.
    - If the **Show pending group memberships** check box is selected, the list also includes the groups to which the object is scheduled to be added in the future. The icons identifying such groups are shown in orange.

Rescheduling temporal group memberships

The temporal membership settings on a group member include the *start time* and *end time* settings.

The start time setting specifies when the object is to be actually added to the group. This can be specific date and time or an indication that the object should be added to the group right away.

The end time setting specifies when the object is to be removed from the group. This can be specific date and time or an indication that the object should not be removed from the group.

You can view or modify both the start time and end time settings using the Active Roles console.

To view or modify the start or end time setting for a member of a group

1. In the Active Roles console, right-click the group and click **Properties**.
2. In the list on the **Members** tab in the **Properties** dialog box, click the member and then click the **Temporal Membership Settings** button.
3. Use the **Temporal Membership Settings** dialog box to view or modify the start or end time settings.

The **Temporal Membership Settings** dialog box provides the following options:

- **Add to the group | Now** Indicates that the object should be added to the group at once.
- **Add to the group | On this date** Indicates the date and time when the object should be added to the group.
- **Remove from the group | Never** Indicates that the object should not be removed from the group.
- **Remove from the group | On this date** Indicates the date and time when the object should be removed from the group.

Regular members have the **Add to group** and **Remove from group** options set to **Already added** and **Never**, respectively. You can set a particular date for any of these options in order to convert a regular member to a temporal member.

NOTE:

- You can view or modify the start time and end time settings by managing an object rather than groups in which the object has memberships. Open the **Properties** dialog box for that object, and then, on the **Member Of** tab, select the group for which you want to manage the object's start or end time setting and click **Temporal Membership Settings**.
- On the **Members** or **Member Of** tab, you can change the start or end time setting for multiple members or groups at a time. From the list on the tab, select two or more items and click **Temporal Membership Settings**. Then, in the **Temporal Membership Settings** dialog box, select check boxes to indicate the settings to change and make the changes you want.

Removing temporal members

You can remove temporal group members in the same way as regular group members. Removing a temporal member of a group deletes the temporal membership settings for that object with respect to that group. As a result, the object will not be added to the group. If the object already belongs to the group at the time of removal, then it is removed from the group.

To remove a temporal member of a group

1. In the Active Roles console, right-click the group, and then click **Properties**.
2. On the **Members** tab in the **Properties** dialog box, click the member, click **Remove**, and then click **Apply**.

NOTE: You can remove an object that is a temporal member of a group by managing the object rather than the group. Open the **Properties** dialog box for that object, and then, on the **Member Of** tab, select the group from the list and click **Remove**.

9

Group Family

- Understanding Group Family
- Creating a Group Family
- Administering Group Family
- Scenario: Departmental Group Family

Understanding Group Family

- You can view or modify the start time and end time settings by managing an object rather than groups in which the object has memberships. Open the **Properties** dialog box for that object, and then, on the **Member Of** tab, select the group for which you want to manage the object's start or end time setting and click **Temporal Membership Settings**.
- On the **Members** or **Member Of** tab, you can change the start or end time setting for multiple members or groups at a time. From the list on the tab, select two or more items and click **Temporal Membership Settings**. Then, in the **Temporal Membership Settings** dialog box, select check boxes to indicate the settings to change and make the changes you want.
- provides for a separate category of rule-based policies specific to group auto-provision. Each policy of that category, referred to as *Group Family*, acts as a control mechanism for creating and populating groups.
- Group Family automatically creates groups and maintains group membership lists in compliance with configurable rules, allowing group membership to be defined as a function of object properties in the directory. Group Family also allows for creation of new groups based on new values encountered in object properties.

For instance, in order to manage groups by geographical location, a Group Family can be configured to create and maintain groups for every value found in the "City" property of user accounts. Group Family discovers all values of that property in the directory and generates a group for each, populating the group with the users that have the same value of the "City" property. If a new value is assigned to the "City" property for some users, Group Family automatically creates a new group for those users. If a user has the value of

the “City” property changed, Group Family modifies the group membership for that user accordingly.

The configuration of a Group Family does not have to be limited to a single property of objects. Rather, it can combine as many properties as needed. For example, a Group Family can be set up to look at both the “Department” and “City” properties. As a result, Group Family creates and maintains a separate group for each department in each geographical location.

Design overview

The key design elements of Group Family are as follows:

- **Scoping by object location** This determines the directory containers that hold the objects to be managed by Group Family. The scope of Group Family can be limited to certain containers, thereby causing it to affect only the objects in those containers.
- **Scoping by object type and property** This determines the type of objects, such as User or Computer, to be managed by Group Family. Thus, the scope of Group Family can be limited to a set of objects of a certain type. The scope can be further refined by applying a filter in order for Group Family to manage only those objects that meet certain property-related conditions.
- **Grouping by object property** Group Family breaks up the set of managed objects (scope) into *groupings*, each of which is comprised of the objects with the same combination of values of the specified properties (referred to as *group-by properties*). For example, with Department specified as a group-by property for user objects, each grouping only includes the users from a certain department.
- **Creating or capturing groups** For each grouping, Group Family normally creates a new group to associate (link) with the grouping, and ensures the members of the grouping are the only members of that group. When creating groups to accommodate groupings, Group Family uses group naming rules that are based on the values of the group-by properties. Another option is to manually link existing groups with groupings; this operation is referred to as *capturing groups*.
- **Maintaining group membership lists based on groupings** During each subsequent run of Group Family, the groupings are re-calculated, and their associated groups are updated to reflect the changes in the groupings. This process ensures that the group associated with a given grouping holds exactly the same objects as the grouping. If a new grouping found, Group Family creates a group, links the group to the new grouping, and populates the group membership list with the objects held in that grouping.
- **Adjusting properties of generated groups** When Group Family creates a new group to accommodate a given grouping, the name and other properties of the new group are adjusted in compliance with the rules defined in the Group Family configuration. These rules are also used to determine the container where to create new groups, the group type and scope settings, and Exchange-related settings such as whether to mail-enable the generated groups.

- **Running on a scheduled basis** Group Family is a state-based policy by nature. During each run, it analyses the state of directory data, and performs certain provisioning actions based on the results of that analysis. Group Family can be scheduled to run at regular intervals, ensuring that all the groups are in place and the group membership lists are current and correct. In addition, Group Family can be run manually at any time.
- **Action summary log** Active Roles provides a log containing summary information about the last run of Group Family. The log includes descriptions of the error situations, if any occurred during the run, and summarizes the quantitative results of the run, such as the number of updated groups, the number of created groups, and the number of objects that have group memberships changed.

How it works

The Group Family configuration specifies rules to determine:

- **Scope** The set of directory objects managed by Group Family is referred to as *scope*. The scope can be limited to objects of a certain category (such as User objects) located in certain organizational units. Filtering can be applied to further refine the scope.
- **Groupings** Group Family divides the scope into sub-sets referred to as *groupings*. Each grouping consists of objects with the same values of certain properties, referred to as *group-by properties*. Each grouping is identified by a certain combination of values of the group-by properties, with a list of all the combinations being stored and maintained as part of the Group Family configuration.
- **Group names** Unless otherwise specified, Group Family creates a new group for each new grouping found, with the group name being generated in accordance with the group naming rules. It is also possible to manually assign existing groups to some groupings, causing Group Family to *capture* those groups.
- **Links** For each grouping, Group Family creates or captures a group, links the group to the grouping, and populates the group with the objects found in the grouping. During each subsequent run, Group Family uses the link information to discover the group linked to the grouping, and updates the membership list of that group to reflect the changes in the grouping. The groups known to Group Family via the link information are referred to as *controlled groups*.

So, during the first run, Group Family performs as follows:

1. The scope is calculated and analyzed to build a list of all the existing combinations of values of the group-by properties. The list is then added to the Group Family configuration.
2. For each combination of values, a grouping is calculated consisting of all objects in the scope that have the group-by properties set to the values derived from that combination.

3. For each grouping, a group is created or captured, and linked to the grouping. The Group Family configuration is updated with information about those links. Whether to create or capture a group is determined by the Group Family configuration.
4. For each group linked to a certain grouping (controlled group), the membership list is updated to only include the objects found in that grouping. All the existing members are removed from the group and then all the objects found in the grouping are added to the group.

During a subsequent run, Group Family performs as follows:

1. The scope is calculated and analyzed to build up a list of all the existing combinations of values of the group-by properties. The Group Family configuration is then updated with that list.
2. For each combination of values, a grouping is calculated consisting of all objects in the scope that have the group-by properties set to the values derived from that combination.
3. For each grouping, a link information-based search is performed to discover the group linked to that grouping. If the group has been found, its membership list is updated so the group only includes the objects found in the grouping. Otherwise, a group is created or captured, linked to the grouping, and populated with the objects found in the grouping.

When creating a group to accommodate a given grouping, Group Family uses the group naming rules to generate a name for that group. The rules define a name based on the combination of values of the group-by properties that identifies the grouping. The group naming rules are stored as part of the Group Family configuration.

When capturing an existing group to accommodate a given grouping, Group Family uses a group-to-grouping link created manually and stored as part of the Group Family configuration. The link specifies the combination of values of the group-by properties to identify the grouping, and determines the group to be linked to that grouping.

Cross-domain Group Family

When you configure a Group Family, you choose containers that hold the objects you want Group Family to assemble into groups (managed object containers) as well as the container to hold those groups (controlled group container). The Group Family policy has the option allowing you to select managed object containers from any domains registered with Active Roles. With this option, managed object containers may be from different domains and the domain of the controlled group container may be different from the domain of the managed object containers. Depending upon the location of the managed object containers, the groups controlled by Group Family can include objects from domains other than the domain that holds the controlled group container (external domains).

Active Directory has restrictions regarding the types of groups that can have members from external domains, and the types of groups that can have membership in other groups. All these restrictions apply to the groups controlled by Group Family. Thus, Active Roles does not allow Group Family to add objects from external domains to global groups, nor does it allow Group Family to add domain local groups to a global group. With these natural

restrictions, you can configure Group Family so that its controlled groups include members from any domains registered with Active Roles.

As stated above, whether managed object containers can be selected from external domains depends upon the Group Family policy. If you want to use this capability, select the **Enable cross-domain membership** policy option (see Group Family policy options).

Group Family policy options

Group Family policy options determine the Group Family processing behavior. For instance, there is a policy option that determines whether controlled groups can have members from external domains.

You can view or change Group Family policy options in the Active Roles console as follows:

1. In the console tree, select **Configuration | Policies | Administration | Builtin.**
2. In the details pane, double-click **Built-in Policy - Group Family**.
3. In the **Built-in Policy - Group Family Properties** dialog box, click the **Policies** tab, select the policy, and click **View/Edit**.
4. In the **Policy Properties** dialog box that appears, click the **Policy Settings** tab.

The **Policy Settings** tab includes the following options:

- **Enable cross-domain membership**

  Select this option if you want Group Family to support the grouping of objects from external domains. When selected, this option allows each Group Family instance to have managed object containers from any domains that are registered with Active Roles. If this option is not selected, the managed object containers must be from the domain of the Group Family configuration storage group.

  Selecting this option should be considered a long-term commitment to scenarios where objects managed by Group Family may reside in domains other than the domain of the Group Family configuration storage group—external domains. Once you have enabled cross-domain membership, you can configure Group Family instances to look for managed objects in any domains registered with Active Roles. However, if you later decide to un-select this policy options, the Group Family instances that were configured to look for managed objects in external domains will cease to function. You will have to inspect and, if needed, reconfigure your existing Group Family instances to limit scope of managed objects to the domain of the Group Family configuration storage group.

- **Enable support for non-stored virtual attributes**

  When selected, this option makes it possible for Group Family to perform grouping based on custom non-stored virtual attributes-the attributes that have their value calculated by a certain policy rather than stored in the Active Roles database. This option can have a negative effect on Group Family performance, so select it only if you have any of the Group Family group-by properties implemented as a custom non-stored virtual attribute.

This option is normally not selected for performance reasons, which causes Group Family not to create controlled groups that use a custom non-stored virtual attribute as a group-by property. You need to select this option if you want Group Family to create controlled groups by grouping objects based on custom non-stored virtual attributes.

Creating a Group Family

Creation of a Group Family is a two-step process that includes:

1. Creating the Group Family configuration
2. Running the Group Family to initially create or capture groups

The Active Roles console provides the New Group Family wizard for creating the Group Family configuration. The wizard creates a group, referred to as *configuration storage group*, and populates that group with the configuration data you specify.

Note that you can create any number of Group Families, with each Group Family intended to control a certain collection of groups. When linking a group to a grouping, the Group Family engine ensures the group is under the control of only the Group Family that created the link, thereby avoiding conflicts.

NOTE: Groups created through Group Family does not support group name with special characters, such as, /\[]:;|=*?<>".

Start the New Group Family wizard

You can start the New Group Family wizard in the Active Roles console by using the **New | Group Family** command on the organizational unit in which you want to place the configuration storage group.

To start the New Group Family wizard

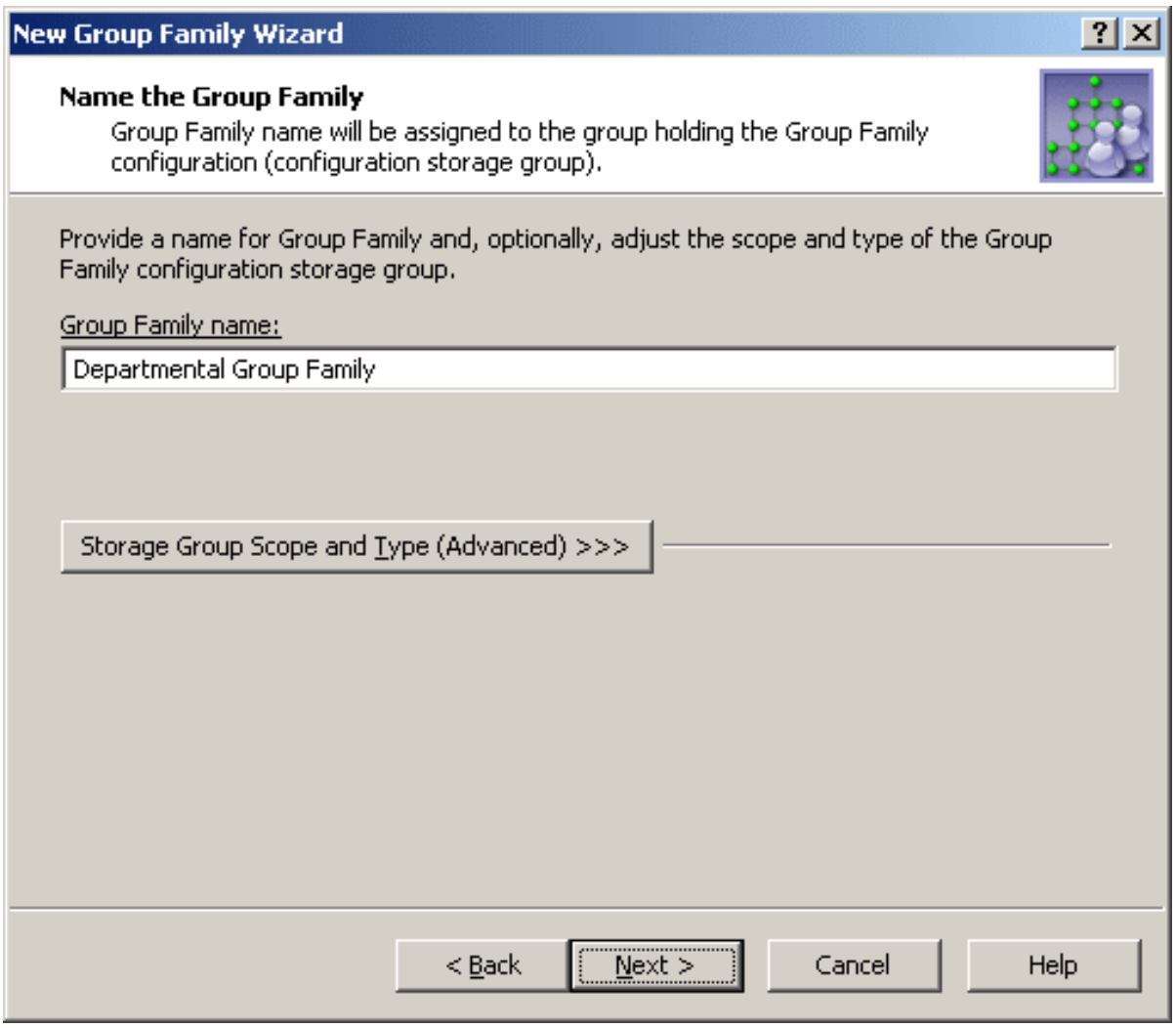
- Right-click the organizational unit to hold the Group Family configuration storage group, and select **New | Group Family**.

Name the Group Family

The first page following the Welcome page is used to provide a name for the new Group Family. The name is assigned to the group that stores the Group Family configuration data (configuration storage group).

You can also use this page to adjust the type and scope of the configuration storage group. These are set to Security and Global by default, and normally need not be modified.

Figure 96: Group Family name

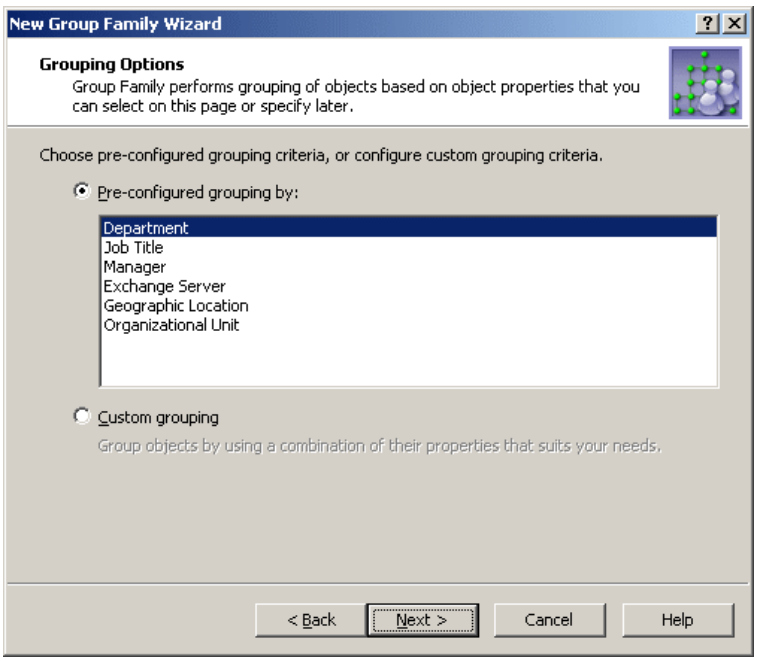


Type in a Group Family name, and then click **Next** to continue.

Grouping Options

The next page provides a list of commonly used grouping criteria. Group Family creates groupings based on the properties you can select on this page or specify later.

Figure 97: Grouping options



You can choose one of these options:

- **Pre-configured grouping** Provides a list of commonly-used group-by properties, such as Department, Title, or Geographic Location. Select an entry from the list to specify the group-by properties. Later, on the **Group-by Properties** page, the wizard will allow you to view or modify the list of the group-by properties you have selected.
- **Custom grouping** Lets you proceed without selecting group-by properties at this stage. The wizard will prompt you to set up a list of group-by properties on the **Group-by Properties** page.

Location of managed objects

The next page prompts you to specify the directory containers that hold the objects to be managed by this Group Family. The scope of the Group Family can be limited to certain containers, thereby causing it to take effect on only the objects in those containers.

The page lists the containers to be included in the scope of the Group Family. Each entry in the list identifies a container by name, and provides the path to the container's parent container.

To add a container to the list, click **Add** and select the container. This will cause the Group Family scope to include objects held in that container.

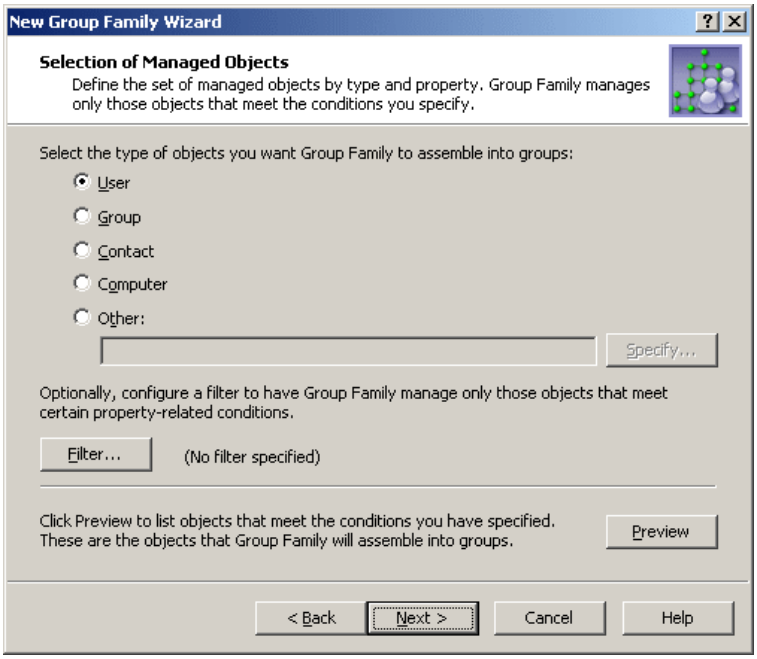
To remove containers from the list, select them and click **Remove**. This will cause the Group Family scope to no longer include the objects held in those containers.

To view of modify properties of a container, select it from the list and click **Properties**.

Selection of managed objects

The next page prompts you to specify the type of objects, such as User or Computer, to be managed by Group Family. In this way, the scope of the Group Family is limited to objects of a certain type. The scope can be further refined by applying a filter in order for the Group Family to manage only those objects that meet certain property-related conditions.

Figure 98: Selection of managed objects



You can select the type of objects you want the Group Family scope to include:

- **User** The Group Family scope only includes user accounts.
- **Group** The Group Family scope only includes groups. Note that with this option the Group Family creates groups and adds existing groups to the newly created groups.
- **Contact** The Group Family scope only includes contact objects.
- **Computer** The Group Family scope only includes computer accounts.
- **Other** The Group Family scope only includes the directory objects of the type you select. Click **Specify** and select an object type.

You have the option to further refine the Group Family scope by applying a filter. To do so, click **Filter**. This displays a window where you can view or modify filtering criteria. The label next to the **Filter** button provides a visual indication of whether any filtering criteria are specified.

In the **Filter** window, you can set up a list of filtering criteria, also referred to as *conditions*. Each condition specifies a property, operator and value, and evaluates to either TRUE or FALSE depending on the actual value of the property. For example, the following condition evaluates to TRUE for any object that has Description set to Full Time Employee:

Table 63: Filtering conditions

Property	Condition	Value
Description	Starts with	Full Time Employee

If any conditions are specified, a filter is applied so that the Group Family scope only includes the objects for which all conditions evaluate to TRUE.

With an empty list of conditions, the Group Family scope includes all objects of the specified type held in the specified containers. In other words, this results in no filtering being applied.

When you apply a filter, only the objects that meet the filter conditions are added to the controlled groups. By default, no filter is applied, which causes the controlled groups to include any objects of the specified type. You can configure a basic filter by selecting properties and specifying conditions and values to search for on the selected properties.

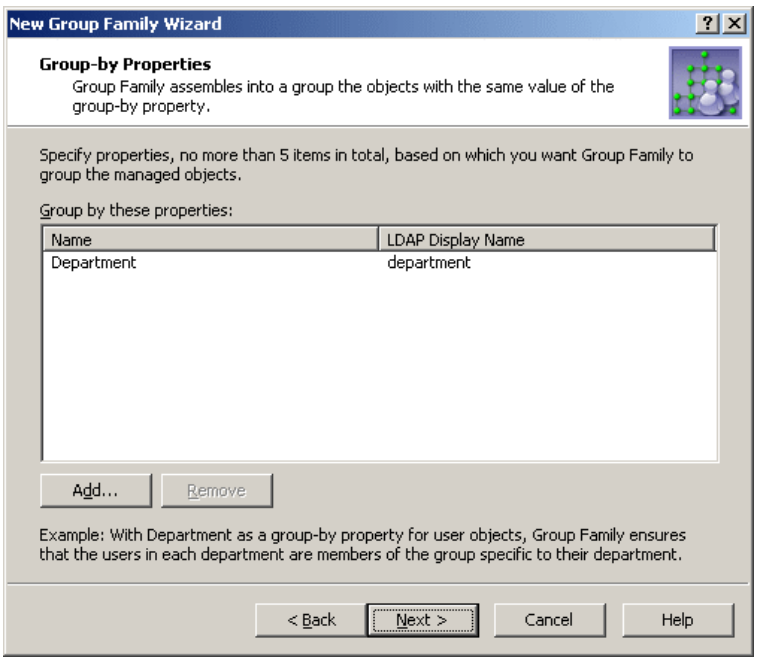
In addition, you have the option to configure an advanced filter by entering an appropriate LDAP query. To do so, click the **Advanced** button in the **Filter** window. Note that the basic and advanced filter options are mutually exclusive. If you have applied an advanced filter, the basic filter settings are disregarded. To return to the basic filter option, click the **Basic** button in the Filter window—this will override the LDAP query that the advanced filter is based upon.

By clicking **Preview** on the **Selection of Managed Objects** page, you can display a list of objects currently included in the Group Family scope. The **Preview** window lists the objects the Group Family is going to assemble into groups.

Group-by properties

The next page lets you set up the list of group-by properties. The Group Family breaks up the set of managed objects (scope) into groupings, each of which is comprised of the objects with the same combination of values of the specified group-by properties. For example, with Department specified as a group-by property for user objects, each grouping only includes the users from a certain department. Then, the Group Family ensures the members of each grouping belong to the group linked to that grouping.

Figure 99: Group by properties



The page lists of the currently selected group-by properties, and allows you to modify the list by adding or removing properties.

> IMPORTANT: The changes you make to the list on this page reset the Group Family options that are dependent on the group-by properties. These options include the group naming rules and the list of groups to capture (see the following two sections). If you add or remove a group-by property, the current naming rules are replaced by the default naming rule and the list of groups to capture is erased.

About multi-valued group-by properties

Group Family supports the use of multi-valued group-by properties, such as Keywords (edsvaKeywords). With Group Family configured to perform the grouping by a multi-valued property, Active Roles creates a separate group for each value of that property and populates the group with the objects whose multi-valued property in question contains the given value. Thus, by choosing edsvaKeywords as a group-by property, you can configure Group Family to create a separate group for each keyword of the objects held in a certain container. For each of those objects, Active Roles ensures that the object has membership in each of the groups corresponding to the keywords of that object. To take an example, consider a container that holds 3 objects with the following keywords:

- Object1 has Keyword1 and Keyword2
- Object2 has Keyword1 and Keyword3
- Object3 has Keyword1 and Keyword3

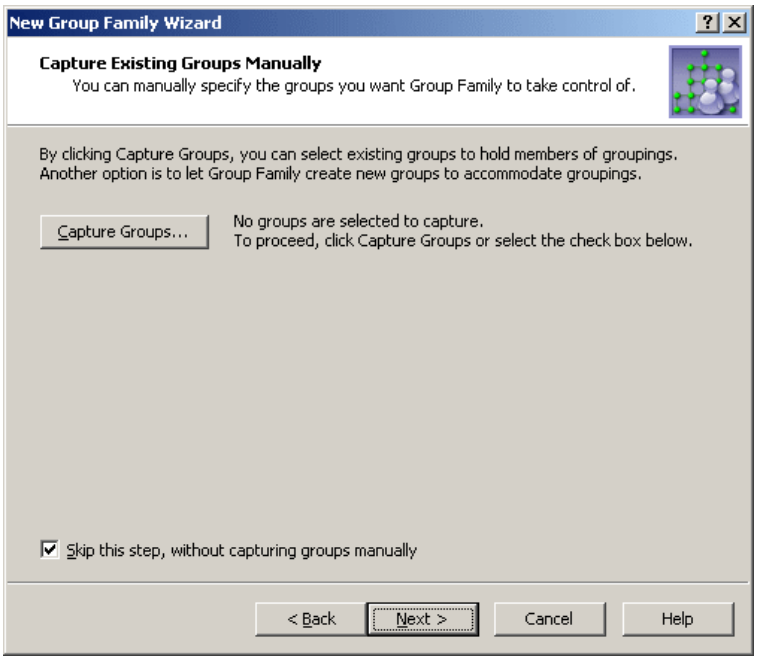
You can configure Group Family so that Active Roles will create 3 groups, each corresponding to one of the three keywords, and populate the groups as follows:

- Add Object1, Object2 and Object3 to the Keyword1 group
- Add Object1 to the Keyword2 group
- Add Object2 and Object3 to the Keyword3 group

Capture existing groups manually

The next page gives you the option to link existing groups to groupings. Normally, the Group Family automatically creates and links a group to each grouping. To override this behavior for certain groupings, you can configure the Group Family to link those groupings to the existing groups you specify.

Figure 100: Capture existing groups manually



On this page, do one of the following:

- To let the Group Family automatically create and link a group to every grouping it discovers, select the **Skip this step, without capturing groups manually** check box.
- To manually establish one or more group-to-grouping links, click **Capture Groups**.

Clicking **Capture Groups** displays a window where you can view or modify a list of group-to-grouping links. Each entry in the list includes the following information:

- **Combination of values of the group-by properties** The combination of property values that identifies a grouping.
- **Group Name** Identifies the group linked to the grouping.
- **In Folder** The canonical name of the container holding the group.

The **Capture Groups** window provides the following buttons for managing the list of group-to-grouping links:

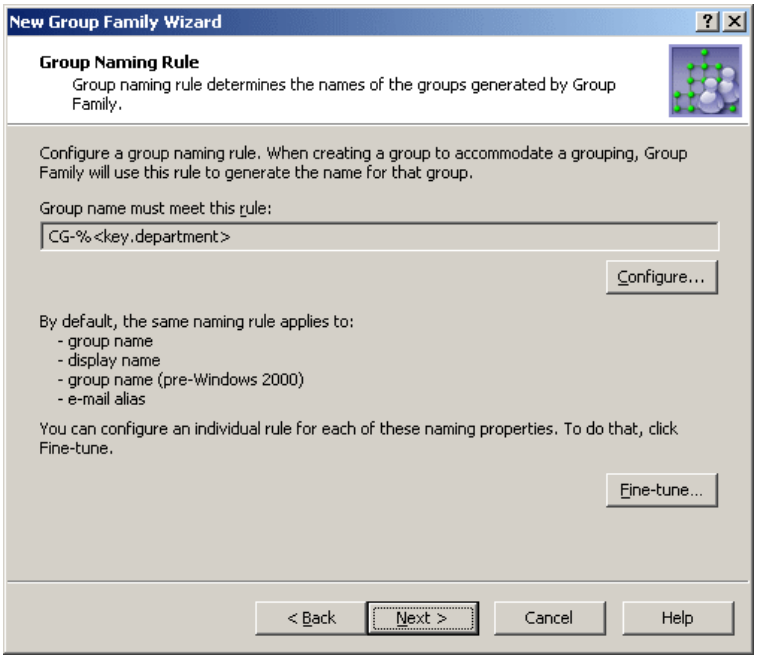
- **Add** Opens a window where you can select a group and specify a grouping. To specify a grouping, you need to enter a certain value of each of the group-by properties. The result is that the group you select is linked to the grouping identified by the combination of values you have entered.
- **Edit** Allows you to modify an entry you select from the list. Opens a window where you can select a different group, or specify a different grouping by making changes to the combination of values of the group-by properties.
- **Remove** Deletes the links you select from the list. The result is that the Group Family will create new groups for the groupings you remove from the list.

Group naming rule

On the next page of the wizard, you can view or modify the group naming rules used by the Group Family.

When creating a new group, the Group Family generates the group naming properties such as Group name, Display name, Group name (pre-Windows 2000) and, optionally, E-mail alias. Unless otherwise specified, the Group Family uses a certain default rule to generate those properties based on the values of the group-by properties.

Figure 101: Group naming rule



By default, the Group Family generates the group naming properties based on the following syntax: `CG-%<key.property1>-%<key.property2>...` In this syntax, CG is the abbreviation for Controlled Group, whereas each of the %<...> entries is used to represent a value of a certain group-by property. When creating a group for a given grouping, the Group Family substitutes the grouping-specific value of the group-by property for the entry containing the name of that property. For example, with a grouping identified by the **Operations** value of the **Department** property, the group name is set to **CG-Operations**. With two group-by properties, such as **Department** and **City**, an example of the group name could be **CG-Operations-London**.

You can modify the group naming rule by clicking the **Configure** button. This displays the **Configure Value** dialog box, discussed earlier in this document (see How to configure a Property Generation and Validation policy). You can use that dialog box to set up a value for the **'name' must be** condition, in the same way as you do when configuring a Property Generation and Validation policy.

A value is a concatenation of one or more entries. The **Configure Value** dialog box provides the **Add**, **Edit**, and **Remove** buttons for managing the list of entries. Clicking **Add** displays the **Add Entry** window.

In the **Add Entry** window, you can select the type of the entry to add, and then configure the entry. The available types of entries are as follows:

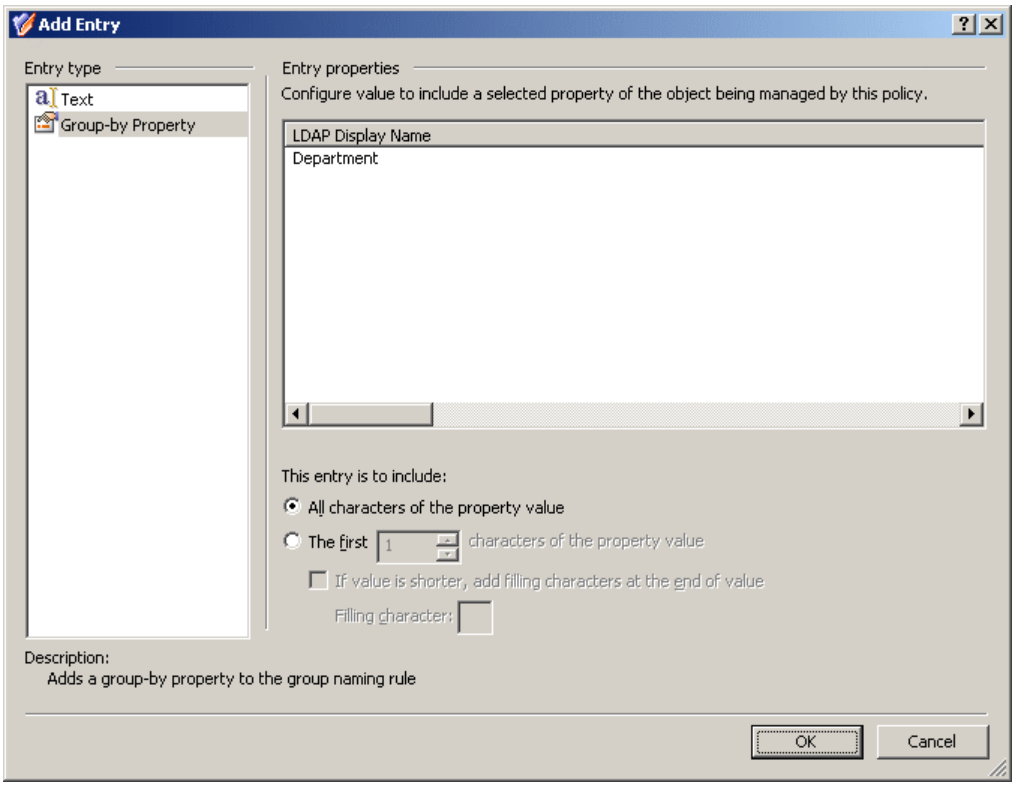
- **Text** Adds a text string to the group naming rule.
- **Group-by Property** Adds a group-by property or a part of a group-by property to the group naming rule.

To add a text string, you simply type a text in **Add Entry** window. The next subsection elaborates on the **Group-by Property** entry.

Entry type: Group-by Property

When you select **Group-by Property** under **Entry type** in the **Add Entry** window, the **Entry properties** area looks similar to the following figure.

Figure 102: Group-by property



Using the **Group-by Property** entry type, you can add an entry representing a value (or a part of a value) of a group-by property. Select a group-by property from the list, and then do one of the following:

- If you want the entry to include the entire value of the property, click **All characters of the property value**.
- If you want the entry to include a part of the property value, click **The first**, and specify the number of characters to include in the entry.

If you choose the second option, you can select the **If value is shorter, add filling characters at the end of value** check box, and type a character in the **Filling character** box. This character will fill the missing characters in the value of the property if the value is shorter than specified in the box next to **The first**. For example, if you specify **The first 12 characters** and enter **0** as the filling character, the **Accounting** property value results in the **Accounting00** entry.

When you are done configuring an entry, click **OK** to close the **Add Entry** window. The entry is added to the **Configure Value** dialog box. When you have completed the list of entries, click **OK** to close that dialog box. Note that the naming rule must include an entry for each of the group-by properties.

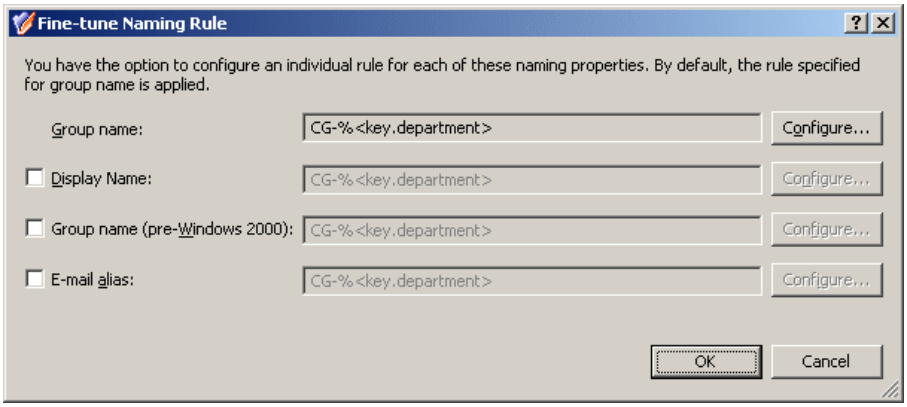
Separate rule for each naming property

By default, the same rule applies to these naming properties:

- Group name
- Group name (pre-Windows 2000)
- Group display name
- E-mail alias (if the Group Family is configured to create mail-enabled groups, as described later in this chapter)

You have the option to configure an individual rule for each of these naming properties. To do so, click **Fine-tune** on the **Group Naming Rule** page. This displays a window where you can select a naming property and configure a rule for that property the same way as you do for Group name. The window looks similar to the following figure.

Figure 103: Fine-tune naming rule

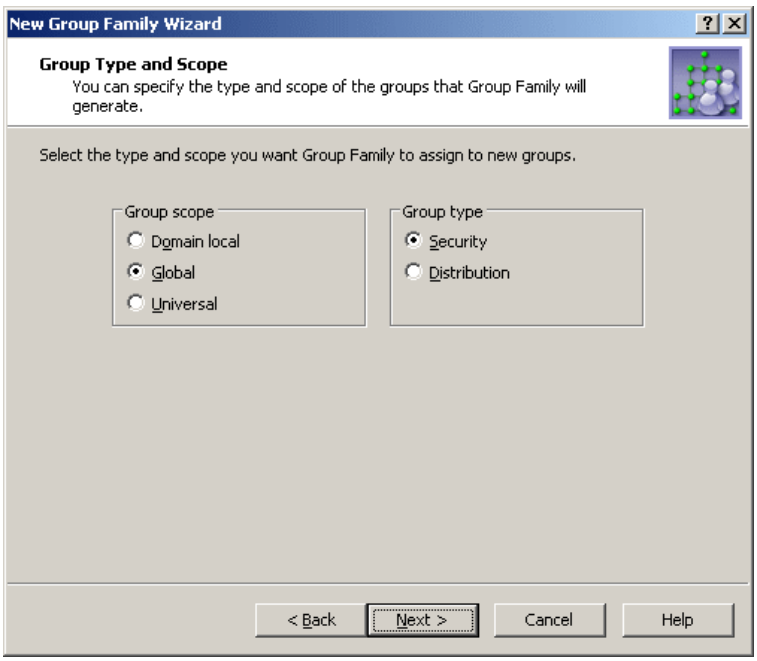


You may need to configure a separate rule for a certain property, considering restrictions imposed on that property. For example, Group name (pre-Windows 2000) must be less than 20 characters. In order to meet this requirement, select the **Group name (pre-Windows 2000)** check box and click **Configure** to set up an appropriate rule. When configuring entries to include group-by properties, limit the number of characters in each entry by using the option **The first** in the **Add Entry** window.

Group type and scope

On the next page, you can specify the group scope and group type you want to be assigned to the groups generated by the Group Family.

Figure 104: Group type and scope

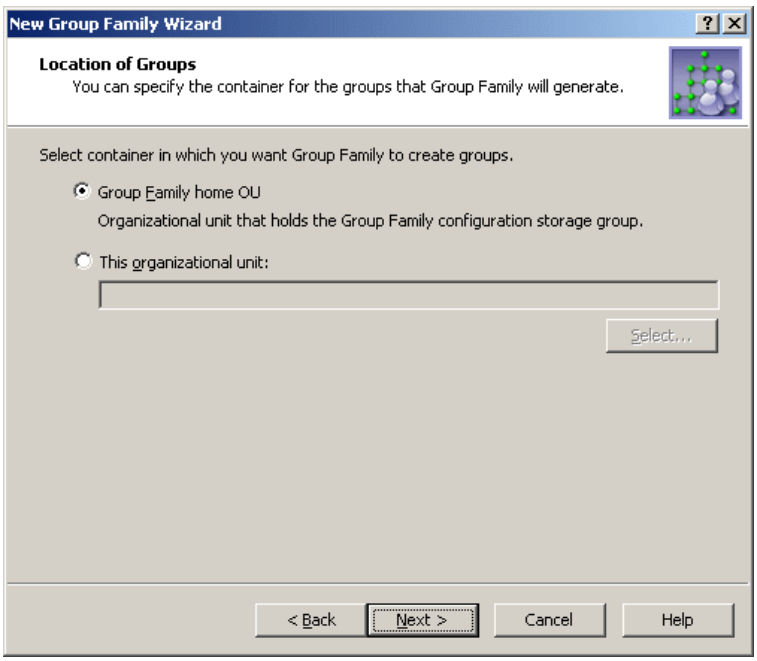


Available are the standard options for the group scope and group type. The Group Family creates groups of the scope and type you select.

Location of groups

On the next page, you can specify the container you want to hold the groups generated by the Group Family.

Figure 105: Location of groups



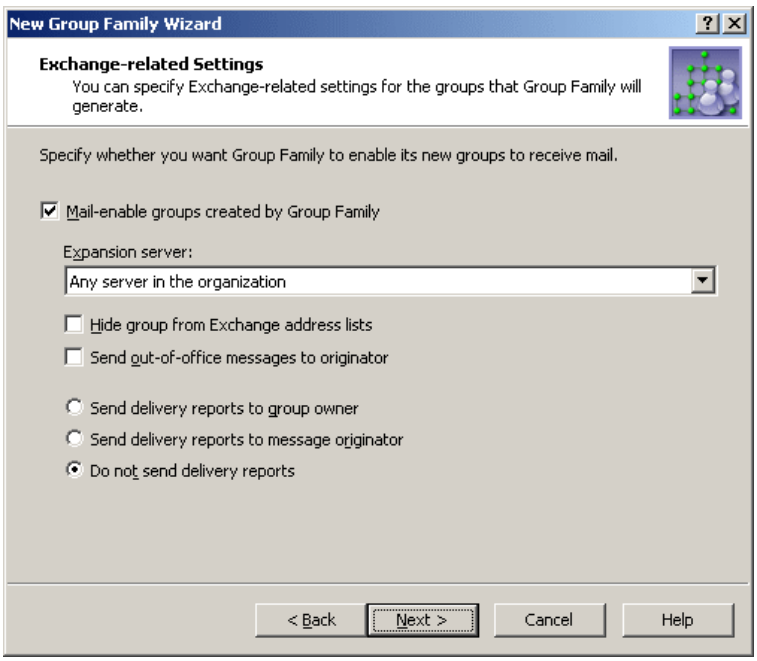
You can choose one of these options:

- **Group Family home OU** The Group Family creates groups in the container that holds the configuration storage group for that Group Family (see Start the New Group Family wizard earlier in this chapter).
- **This organizational unit** The Group Family creates groups in the container specified. This must be an organizational unit or container from the domain of the Group Family configuration storage group. Click **Select** to choose the desired organizational unit or container.

Exchange-related settings

On the next page, you can specify whether you want the groups generated by the Group Family to be mail-enabled, and set up Exchange-related properties to assign to those groups upon their creation.

Figure 106: Exchange-related settings



If you want the Group Family groups to be mail-enabled, select the **Mail-enable groups created by Group Family** check box. Then, you can set up the following Exchange-related properties for the Group Family groups:

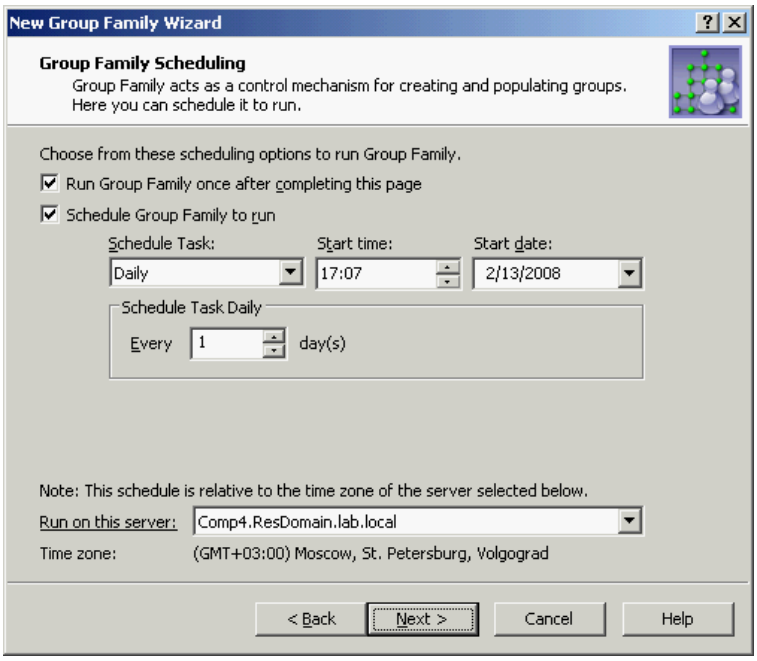
- **Expansion server** The Exchange server used to expand a Group Family group into a list of group members.
- **Hide group from Exchange address lists** Prevents the Group Family groups from appearing in address lists. If you select this check box, each of the groups will be hidden from all address lists.
- **Send out-of-office messages to originator** Select this check box if you want out-of-office messages to be sent to the message originator, when a message is sent to a Group Family group while one or more of the group members have an out-of-office message in effect.
- **Send delivery reports to group owner** Use this option if you want delivery reports to be sent to the group owner, when a message sent to a Group Family group is not delivered. This lets the group owner know that the message was not delivered.
- **Send delivery reports to message originator** Use this option if you want delivery reports to be sent to a message originator, when a message sent to a Group Family group is not delivered. This lets the message originator know that the message was not delivered.
- **Do not send delivery reports** Use this option if you do not want delivery reports to be sent, even if a message sent to a Group Family group is not delivered.

Group Family scheduling

On the next page, you can schedule the Group Family to run. During each run, the Group Family performs as described in the How it works section, earlier in this chapter.

When setting up the schedule options, take into account that a Group Family run is a lengthy and resource intensive operation. Therefore, a Group Family run should be scheduled for a time that it will have the minimum impact on users.

Figure 107: Group family scheduling



Select the first check box to run the Group Family right after you complete the wizard and whenever the Group Family is modified by managing the configuration storage group (see Administering Group Family later in this chapter).

Select the second check box to set up schedule options. As long as this check box is selected, the Group Family runs at specified time.

From the **Run on this server** list, you can select the Administration Service to run the Group Family. It is advisable to choose the least loaded Service.

Steps for creating a Group Family

Creation of a Group Family is a two-step process that includes:

1. Creating the Group Family configuration
2. Running the Group Family to initially create or capture groups

The Active Roles console provides the New Group Family wizard for creating the Group Family configuration. The wizard creates a group, referred to as *configuration storage group*, and populates that group with the configuration data you specify. The wizard also allows you to run the Group Family immediately or schedule the Group Family to run on a regular basis.

To create the Group Family configuration and run the Group Family

1. In the console tree, right-click the organizational unit in which you want to create the Group Family configuration storage group, and select **New | Group Family** to start the New Group Family wizard.
2. Follow the instructions on the wizard pages.
3. On the **Name the Group Family** page, specify a name for the Group Family.

   The wizard creates the Group Family configuration storage group with the name you specify on this page.
4. On the **Grouping Options** page, do one of the following, and then click **Next**:
   - Click **Pre-configured grouping by**, and then select a pre-configured grouping criteria from the list.
   - Click **Custom Grouping** to configure custom grouping criteria in later steps of the wizard.
5. On the **Location of Managed Objects** page, do the following, and then click **Next**:
   - Click **Add**, and then select a container that holds the objects to be assembled into groups.
   - Click **Remove** to remove a selected container from the **Containers** list.
6. On the **Selection of Managed Objects** page, do the following, and then click **Next**:
   - Select a type of objects by clicking one of the four topmost options; or click **Other**, and then click **Specify** to choose an object type from the **Object Types** list.
   - Click **Filter**, and complete the **Filter** dialog box (see instructions later in this topic).
   - Click **Preview** to view the list of objects that meet the specified conditions.
7. On the **Group-by Properties** page, do the following, and then click **Next**:
   - Click **Add**, and select an object property from the **Object property** list.
8. On the **Capture Existing Groups Manually** page, select **Skip this step, without capturing groups manually**, and then click **Next**.
9. On the **Group Naming Rule** page, do the following, and click **Next**:
   - Click **Configure**, and complete the **Configure Value** dialog box (see instructions later in this topic).
   - Click **Fine-tune Naming Rule**, and complete the **Fine-tune Naming Rule** dialog box (see instructions later in this topic).

10. On the **Group Type and Scope** page, do the following, and then click **Next**:
    - In the **Group scope** area, select a group scope.
    - In the **Group type** area, select a group type.
11. On the **Location of Groups** page, do one of the following, and then click **Next**:
    - To have the Group Family create new groups in the OU that holds the Group Family configuration storage group, click **Group Family home OU**.
    - To have the Group Family create new groups in a different OU, click **This organizational unit**, and then click **Select** to choose the OU.
12. On the **Exchange-related Settings** page, do the following, and then click **Next**:
    - Select or clear the **Mail-enable groups created by Group Family** as appropriate. If you select this check box, set up the Exchange-related options on this page.
13. On the **Group Family Scheduling** page, do the following, and then click **Next**.
    - If you want the Group Family to run once you have completed the wizard, select **Run Group Family once after completing this page**.
    - If you want the Group Family to run on a schedule basis, select **Schedule Group Family to run**, and then set the appropriate date, time, and frequency of runs by using the options below this check box.
    - From the **Run on this server** list, select the Administration Service you want to run the Group Family.
14. On the last page of the wizard, click **Finish**.

To complete the Filter dialog box

1. Select an object property under **Select Property**.
2. Select an operator from the **Select operator** drop-down list.
3. In **Specify value (case-insensitive)**, type in a value for the selected property.
4. Click **Add** to add the filter condition that you have just specified, to the **Conditions** list.
5. To add multiple filter conditions, repeat steps 1-4.

To complete the Configure Value dialog box

1. Click **Add**.
2. In the **Add Entry** dialog box, do one of the following, and then click **OK**:
    - To configure a text entry, click **Text** under **Entry type**, and then type a value in the **Text value** box.
    - To configure a group-by property entry, click **Group-by Property** under **Entry Type**, and then, under **Entry properties**, select a property from the list and do one of the following:
        - If you want the entry to include the entire value of the property, click **All characters of the property value**.

- If you want the entry to include a part of the property value, click **The first**, and specify the number of characters to include in the entry.

3. Optionally, do the following:
   - Add more entries, delete or edit existing ones, and use the arrow buttons to move entries up or down in the list.
   - Paste the Clipboard contents to the list of entries by clicking the button next to the **Configured value** box.
4. Click **OK**.

To complete the Fine-tune Naming Rule dialog box

1. Select the check box and click the **Configure** button next to the naming property that you want to configure, and then complete the **Configure Value** dialog box by using the procedure outlined above.
2. Click **OK**.

Administering Group Family

Most of the tasks related to Group Family administration are performed by using the **Properties** command on the groups used to store Group Family configurations. In the Active Roles console, such groups are marked with a special icon, to distinguish them from regular groups.

So, when you create a Group Family, a group is created to store the Group Family configuration. The group is assigned the name you have provided for the Group Family, and marked with the Group Family icon:

To facilitate Group Family administration, the **Properties** dialog box for a configuration storage group includes a number of Group Family-specific tabs:

- General tab Displays the name of the Group Family and allows the administrator to view or modify the description, group type, and group scope of the storage group.
- Controlled Groups tab Lists the groups that are under the control of the Group Family, and allows the administrator to view or modify the group-to-grouping links and group creation-related rules.
- Groupings tab Allows the administrator to view or modify the Group Family scope and the list of group-by properties.
- Schedule tab Displays Group Family schedule-related information, and allows the administrator to view or modify scheduling settings.
- Action Summary tab Displays information about the last run of the Group Family, and allows the administrator to view a log detailing results of the run.

These tabs are discussed in more detail later in this section.

NOTE: Changes to the regular, group-related properties of the configuration storage group do not affect the Group Family. For example, you can rename or move the configuration storage group without any impact on the process and results of Group Family operation. Renaming the configuration storage group only changes the display name of the Group Family.

The **Action** menu on each Group Family configuration storage group includes the **Force Run** command, so you can run the Group Family if you want to update it right away, without waiting for the scheduled run time.

Controlled groups

To help distinguish the groups that are under the control of a Group Family (controlled groups), the Active Roles console marks them with a special icon. For example, the following icon is used to indicate a global group that is under the control of a Group Family:

In addition, an explanatory text is added to the **Notes** field for such groups, stating that the Group Family will override any changes made directly to the group membership list.

In the Active Roles console, the **Properties** dialog box for controlled groups includes a Group Family-specific tab named **Controlled By**. From that tab, you can manage the configuration of the Group Family that controls the group.

The **Controlled By** tab displays the name and path of the group that stores the configuration of the Group Family. To view or change the configuration of the Group Family, click the **Properties** button.

So, there are two ways to access the **Properties** dialog box of the Group Family configuration storage group:

- On the **Controlled By** tab in the **Properties** dialog box for any group controlled by the Group Family, click **Properties**
- Right-click the Group Family configuration storage group, and click **Properties**

The following sections elaborate on the Group Family-specific tabs found in the **Properties** dialog box for the Group Family configuration storage group.

General tab

The **General** tab displays the Group Family name, and allows you to edit the description. This tab cannot be used to modify the Group Family name. You can change the name by using the **Rename** command on the Group Family configuration storage group.

By clicking the **Storage Group Scope and Type (Advanced)** button, you can view or modify the group scope and group type of the configuration storage group. Changes to these settings do not affect the Group Family. The group type and group scope are set to Security and Global by default, and normally need not be modified.

Controlled Groups tab

The **Controlled Groups** tab lists the groups that are controlled by this Group Family. The tab includes the following items:

Table 64: Controlled groups tab items

Item	Description
Controlled groups	This is a list of all groups that are under the control of this Group Family. For each group, the list displays the name of the group along with the path and name of the container that holds the group.
Capture Groups	Click this button to examine the list of controlled groups in detail. For each of the controlled groups, you can identify the grouping assigned to that group.
Manage Rules	Click this button to view or change the Group Family settings that determine properties of the controlled groups such as the naming properties, the group type and scope, the container that holds the groups, and Exchange-related properties.

Each of the groups listed on this tab is either created or captured by the Group Family, and linked to a certain grouping. You can view or modify those links by clicking **Capture Groups**.

> NOTE: For a newly created Group Family configuration, the list on this tab only includes the groups specified in the **Capture Existing Groups Manually** step of the New Group Family wizard. If that step was skipped, the list is empty until the Group Family has been run.

Clicking **Capture Groups** displays a window where you can view the list of controlled groups in more detail. The **Capture Groups** window allows you to add, modify, or remove entries from that list.

The **Capture Groups** window lists all the controlled groups. For each group, you can see which grouping is linked to that group. As usual, groupings are identified by combinations of values of the group-by properties. Thus, each entry in the list includes the following information:

- **Combination of values of the group-by properties** The combination of property values that identifies a grouping.
- **Group Name** Identifies the group linked to the grouping.
- **In Folder** The canonical name of the container holding the group.
- **Last Update.** The date and time the group was last updated by the Group Family. The update occurs during a Group Family run, when any changes to the grouping are

detected and the membership list of the group is modified so as to reflect those changes.

- **Members** The number of members that the group holds after the last update. Equals to the number of objects the Group Family found in the grouping as of the time of the last update.

The **Capture Groups** window provides these buttons for managing the list:

- **Add** Opens a window where you can select a group and specify a grouping to which you want to link (assign) an existing group. To specify a grouping, you need to enter a certain value of each of the group-by properties. The result is that the group you select is linked to the grouping identified by the combination of values you have entered.
- **Edit** Allows you to modify an entry you select from the list. Opens a window where you can select a different group, or specify a different grouping by making changes to the combination of values of the group-by properties.
- **Remove** Deletes the entries you select from the list. The result is that the Group Family will create new groups for the groupings you remove from the list.
- **Scan** Detects new combinations of values of group-by properties, and displays them in the list so that you can link existing groups to new combination manually if you do not want the Group Family to create new groups for those combinations.

When managing the list of groups in the **Capture Groups** window, consider the following:

- You can assign an existing group to a grouping regardless of whether the grouping actually exists in the directory. For example, you can assign a group to a grouping with a Department property value that is not encountered in the directory. Once the Department property for some users is set to that value, the Group Family will add those users to the specified group instead of creating a new group for the new Department.
- Only one group can be assigned to a grouping. If the list already includes a given grouping, you will not be allowed to add a new entry referring to that same grouping. In this case, you have the option to use the **Edit** button, to link a different group to the grouping.
- When you edit a list entry to link a different group to a grouping, the group that was earlier linked to the grouping remains intact. It neither is deleted nor has the membership list updated. In other words, the members of the grouping still belong to the group even though you have removed that group from the list, and thus from under the control of the Group Family.
- When you remove an entry from the list, the group that the entry refers to is not deleted. During a subsequent run, the Group Family will detect a grouping that has no group assigned and try to create a group for that grouping. This operation may fail due to a name conflict so long as there is an existing group with the same name—the group that was earlier linked to the grouping. To avoid name conflicts, rename or delete the groups you remove from under the control of the Group Family.

Group creation-related rules

When a Group Family discovers a grouping that is not linked to any group, it creates a new group, links the new group to the grouping, and adds the members of the grouping to that group. The Group Family configuration specifies a number of rules on how to set up certain properties for new groups.

The rules that control the group creation process are defined when the Group Family configuration is created. You can examine or modify those rules by using the **Manage Rules** button on the **Controlled Groups** tab, in the **Properties** dialog box of the Group Family configuration storage group.

- The **Manage Rules** button gives you access to a series of pages that are similar to those of the New Group Family wizard discussed earlier in this chapter. Clicking **Manage Rules** starts a step-by-step process organized into these pages:
- **Group Naming Rule** Group Family uses this rule to generate the Group name, Display name, Group name (pre-Windows 2000), and E-mail alias when creating new groups. For details, refer to the Group naming rule section earlier in this chapter.
- **Group Type and Scope** The group type and group scope that is assigned to the groups created by the Group Family.
- **Location of Groups** The rule that determines the container in which the Group Family creates new groups. For details, refer to the Location of groups section earlier in this chapter.
- **Exchange-related Settings** The rule that determines whether the groups created by the Group Family are mail-enabled, and a number of options pertinent to mail-enabled groups. For details, refer to the Exchange-related settings section earlier in this chapter.

You can navigate through these pages by using the **Back** and **Next** buttons. The **Finish** button on the last page commits the changes, if any, from all pages to the **Properties** dialog box, and completes the task of managing the group creation rules. The changes are applied when you click **OK** or **Apply** in the **Properties** dialog box, and can be discarded by clicking **Cancel**.

Groupings tab

From the **Groupings** tab, you can view or change the Group Family settings that control the Group Family calculation processes.

During each run, the Group Family re-calculates groupings by breaking up the set of managed objects (scope) into sub-sets, with each sub-set consisting of the objects that have a particular combination of values assigned to the group-by properties.

The scope and the group-by properties are specified when the Group Family configuration is created, and can be changed on the pages that appear when you click **Configure** on the **Groupings** tab. By clicking the **Configure** button, you can view or change the following settings:

- **Location of Managed Objects** The containers that hold the objects to be managed by this Group Family. For details, see Location of managed objects earlier in this chapter.
- **Selection of Managed Objects** The rules that determine what objects are to be managed by this Group Family. For details, see Selection of managed objects earlier in this chapter.
- **Group-by Properties** The list of properties based on which the Group Family calculates groupings. For details, see Group-by properties earlier in this chapter.

  If you add or remove a group-by property, the naming rules that currently exist are replaced with the default naming rule and the list of groups to capture is erased.

Schedule tab

The **Schedule** tab displays Group Family schedule-related information, and allows you to view or modify scheduling settings.

The tab displays the following information:

- **Schedule** The Group Family is scheduled to run as indicated by this statement.
- **Run on this server** The Administration Service that performs all operations needed to run the Group Family.
- **Last run time** The date and time the Group Family was last run.
- **Next run time** The date and time that the Group Family is next scheduled to run.

You can use the **Configure** button to examine the Group Family schedule in more detail, and make changes to the schedule as needed.

Clicking **Configure** displays the **Group Family Scheduling** page, similar to that of the New Group Family wizard discussed earlier in this chapter (see the Group Family scheduling section). View or modify the schedule settings on that page, and click the **Finish** button to commit your changes to the **Properties** dialog box. The changes are applied when you click the **OK** or **Apply** button, and can be discarded by clicking **Cancel**.

Action Summary tab

The **Action Summary** tab displays quantitative information about the Group Family run.

Use the **Action Summary** tab to see the following information about the last run of the Group Family:

- **Last run started** The date and time the run was started.
- **Last run finished** The date and time the run was finished.
- **Managed objects** The number of objects found in the Group Family scope.
- **Valid groupings** The number of groupings calculated during the run.

- **Failed groupings** The number of groupings the Group Family failed to identify due to invalid combinations of group-by property values. An example of an invalid combination occurs when values for one or more properties are missing from the combination.
- **Groups created** The number of groups the Group Family created during the run.
- **Groups updated** The number of groups for which the Group Family updated the membership lists during the run.
- **Updates in group memberships** The number of objects the Group Family added or removed from groups during the run.
- **Errors** The number of error encountered during the run.

To examine this information in more detail, click the **View Log** button.

Action summary log

Clicking the **View Log** button displays a log containing summary information about the last run of the Group Family. The log includes descriptions of the error situations, if any occurred during the run, and summarizes the quantitative results of the run, such as the number of updated groups, the number of created groups, and the number of objects that have group memberships changed.

The log can be divided into three sections: Prolog, Error List, and Epilog. The Prolog and Epilog sections are always present in the log, whereas the Error List section only appears if any errors or warnings occurred during the run.

The Prolog section provides the following information:

- The date and time the run was started
- The number of managed objects found in the Group Family scope
- The total amount of groupings found by analyzing the group-by properties

The Epilog section provides the following information:

- The number of errors, if any occurred
- The number of invalid combinations of group-by property values, if any detected
- The number of groups the Group Family created during the run
- The number of groups the Group Family updated during the run

The Error List section provides information about all errors and warnings the Group Family encountered during the run.

Steps for administering a Group Family

This topic covers some task-specific procedures that you can use to change configuration and properties of an existing Group Family.

To open the property sheet for a Group Family

- Right-click the Group Family configuration storage group, and then click **Properties**.

To view or modify grouping rules

1. Open the property sheet for the Group Family (see instructions earlier in this topic).
2. Click the **Groupings** tab, and then click **Configure**.
3. Follow Steps 5 through 7 of the procedure for creating a Group Family (see Steps for creating a Group Family).
4. On the **Group-by Properties** page, click **Finish**.
5. Click **OK** to close the property sheet.

To view or modify group creation-related rules

1. Open the property sheet for the Group Family (see instructions earlier in this topic).
2. Click the **Controlled Groups** tab, and then click **Manage Rules**.
3. Follow Steps 9 through 12 of the procedure for creating a Group Family (see Steps for creating a Group Family).
4. On the **Exchange-related Settings** page, click **Finish**.
5. Click **OK** to close the property sheet.

To manually add a group to a Group Family

1. Open the property sheet for the Group Family (see instructions earlier in this topic).
2. Click the **Controlled Groups** tab, and then click **Capture Groups**.
3. In the **Capture Groups** window, click **Add**.
   a. In the **Assign Group to Grouping** dialog box, do the following, and then click **OK**:
   b. Click **Select**, and then select the group you want to add.
4. In **Group-by property**, type a value of the group-by property. If multiple group-by properties are defined, type a value for each, so as to determine the grouping to which you want the group to be assigned.
5. Click **OK** to close the **Capture Groups** window.
6. Click **OK** to close the property sheet.

To remove a group from a group family

1. Open the property sheet for the Group Family (see instructions earlier in this topic).
2. Click the **Controlled Groups** tab, and then click **Capture Groups**.
3. In the **Capture Groups** window, select the group you want to remove from the Group Family, click **Remove**, and then click **OK**.
4. Click **OK** to close the property sheet.

To schedule a Group Family update

1. Open the property sheet for the Group Family (see instructions earlier in this topic).
2. Click the **Schedule** tab, and then click **Configure**.
3. On the **Group Family Scheduling** page, do the following, and then click **Finish**:
   a. Select **Schedule Group Family to run**, and then set the appropriate date, time, and frequency of Group Family update.
   b. If you also want the Group Family to run one time immediately after you close the property sheet, select **Run Group Family once after completing this page**.
   c. From the **Run on this server** list, select the Administration Service you want to run the Group Family.
4. Click **OK** to close the property sheet.

To view results of a Group Family update

1. Open the property sheet for the Group Family (see instructions earlier in this topic).
2. Click the **Action Summary** tab, and then click **View Log**.

To delete a Group Family

- Right-click the Group Family configuration storage group, and then click **Delete**.

NOTE: Deleting a Group Family only deletes the configuration storage group of the Group Family. This operation does not delete the controlled groups of the Group Family. Later, you can configure another Group Family to take control of those groups.

Scenario: Departmental Group Family

Suppose the organizational unit (OU) named Users contains a number of user accounts. Also assume that for each of the values listed below there are one or more user accounts in the Users OU with the Department property set to that value. Thus, the following values of the Department property are encountered in the user accounts held in the Users OU:

- Accounting
- Executive Services
- Facilities
- Finance
- Government Services
- Human Resources
- Information Technology
- Operations

In this section, you can find the instructions on how to implement a Group Family that creates and maintains a separate group for users in each of those departments. The Group Family configuration storage group will be created in the organizational unit named Groups. The Group Family will be configured to create the departmental groups in that same OU.

Open the Active Roles console, and perform the following steps to implement the Group Family.

To create and run the Departmental Group Family

1. Right-click the **Groups** OU and select **New | Group Family**.

   This will start the New Group Family wizard. The remaining steps apply to that wizard.

2. On the Welcome page, click **Next**.
3. In the **Group Family name** box, type **Departmental Group Family**. Click **Next**.
4. Click the **Pre-configured grouping by** option, click **Department** in the list under that option, and then click **Next**.
5. Remove the **Groups** OU from the **Containers** list, and add the **Users** OU to that list. Click **Next**.
6. Click the **User** option, and then click **Next**.
7. Verify that the **Group by these properties** list includes the only entry—**Department**. Click **Next**.
8. Select the **Skip this step, without capturing groups manually** check box. Click **Next**.
9. Click **Next** to accept the default rule for group naming: **CG-%<key.department>**
10. Click **Next** to accept the default group scope and type.
11. Click **Next** to accept the default location for the controlled groups: **Group Family home OU**
12. Click **Next** to accept the default settings related to Exchange.
13. Select the **Run Group Family once after completing this page** check box. Click **Next**.
14. Click **Finish**.

Once you have completed these steps, the Group Family performs all the necessary processing to create the groups, one group per department, and adds users to the appropriate groups based on the **Department** property.

You might look at the contents of the **Groups** OU in the Active Roles console to verify that the departmental groups are created successfully. You might also examine properties of a group generated by the Group Family, to verify that the membership list of the group is correct. For example, the membership list of the **CG-Executive Services** group consists of the user accounts that have the **Department** property set to **Executive Services**.

10

Dynamic Groups

- Understanding dynamic groups
- Dynamic groups policy options
- Managing dynamic groups
- Scenario: Automatically moving users between groups

Understanding dynamic groups

Active Directory allows groups (herein called basic groups) to include members statically—select objects and add them to groups. Active Roles provides a flexible, rules-based mechanism for populating groups. Once set up, the process automatically adds and removes members from groups.

Active Roles provides rules-based groups called dynamic groups. Membership rules determine whether an object is a member of a dynamic group. A membership rule may take a form of search query, object static inclusion and exclusion rule, and group member inclusion and exclusion rule. As the environment changes, the memberships of objects in dynamic groups automatically change to adapt to the new environment.

Active Roles dynamic groups reduce the cost of maintaining lists and groups, while increasing the accuracy and reliability of this maintenance. Furthermore, it automatically keeps distribution lists and security groups up to date, eliminating the need to add and remove members manually.

To automate the maintenance of group membership lists, dynamic groups provide the following features:

- Rules-based mechanism that automatically adds and removes objects from groups whenever object attributes change in Active Directory.
- Flexible membership criteria that enable both query-based and static population of groups.

In the Active Roles console, dynamic groups are marked with the following icon:

When you convert a basic group to a dynamic group, the group loses all members that were added to the group when it was a basic group. This is because members of a dynamic group can be defined only by membership rules.

When you convert a dynamic group to a basic group, the group retains all its members included due to the membership rules, and loses the membership rules only.

When a member of a dynamic group, such as a user or another group, is deprovisioned, the dynamic group is automatically updated to remove that member. Hence, deprovisioning a user or group removes that user or group from all dynamic groups. This behavior is by design.

Cross-domain membership

When you configure a dynamic group, you choose containers that hold the objects you want to be included or excluded from the group. For example, you could configure a dynamic group to include all users held in a particular Organizational Unit that meet certain conditions. These parent containers of dynamic group members can be selected from any domains registered with Active Roles. Depending upon the location of the members' parent container, the dynamic group can include objects from domains other than the domain in which the group resides (external domains).

Active Directory has restrictions regarding the types of groups that can have members from external domains, and the types of groups that can have membership in other groups. All these restrictions apply to dynamic groups. Thus, Active Roles disregards membership rules that would add external domain users to a global group. With these natural restrictions, you can configure membership rules for a dynamic group to have members from any domains that are registered with Active Roles.

Whether dynamic groups can have external members depends upon the Dynamic Groups policy. If you want dynamic groups to include objects from external domains, ensure that the **Enable cross-domain membership** policy option is selected (see Dynamic groups policy options).

Dynamic groups policy options

The behavior of dynamic groups is defined by the policy held in the build-in Policy Object called "Dynamic Groups." The policy ensures that any changes made to a dynamic group with any other tool used to manage Active Directory will be discarded. The group mActive Rolesembership lists are determined by membership rules.

To view or modify the policy, display the **Properties** dialog box for the **Built-in Policy - Dynamic Groups** Policy Object (located in container **Configuration/Policies/Administration/Builtin**), go to the **Policies** tab, select the policy, and click **View/Edit**. This displays the **Policy Properties** dialog box.

On the **Policy Settings** tab in the **Policy Properties** dialog box, you can select the following options:

- **Enable cross-domain membership** When selected, this option enables dynamic groups to have members from external domains. When cleared, it restricts the membership of each dynamic group to the objects from the domain in which the group resides.
- **Receive directory changes from DirSync control** Ensures that the policy correctly populates membership lists regardless of what tools are used to manage Active Directory. When this check box is not selected, some rules-based membership lists may be incompatible with membership rules. In this case, the policy only reapplies membership rules when directory changes are made by using Active Roles.
- **Include only mailbox-enabled users in dynamic distribution groups** Prevents the policy from adding users without Exchange mailbox to the distribution groups configured as Dynamic Groups.
- **Add this message to the Notes field for each dynamic group** Puts the message text to the **Notes** property of every dynamic group. (The **Notes** property is displayed on the **General** tab in the group **Properties** dialog box.)

Selecting the option that enables cross-domain membership should be considered a long-term commitment to scenarios where members of a dynamic group may reside in domains other than the domain of the dynamic group—external domains. Once you have enabled cross-domain membership, you can configure dynamic groups to include or exclude objects from any domains registered with Active Roles. However, if you later decide to un-select this policy option, the dynamic groups that were configured to include or exclude objects from external domains will cease to function. You will have to inspect and, if needed, reconfigure your existing dynamic groups to ensure that the membership rules of each dynamic group match only objects from the domain of the dynamic group itself.

Managing dynamic groups

This section guides you through the Active Roles console to administer dynamic groups. The following topics are covered:

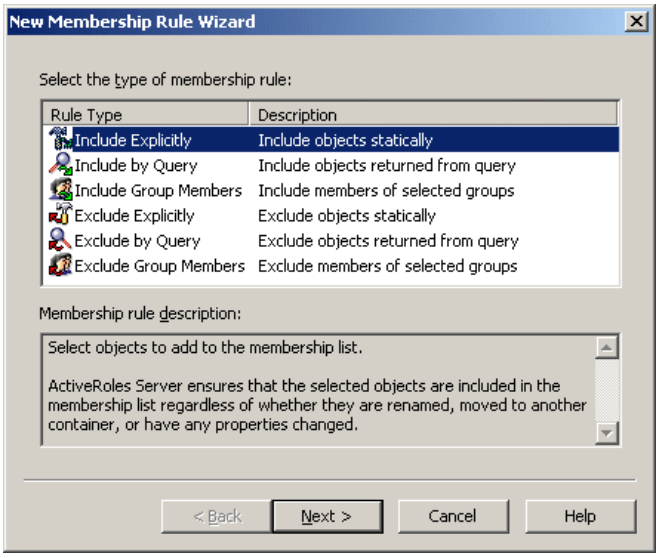
- Converting a basic group to a dynamic group
- Displaying the members of a dynamic group
- Adding a membership rule to a dynamic group
- Removing a membership rule from a dynamic group
- Converting a dynamic group to a basic group
- Modifying, renaming, or deleting a dynamic group

Converting a basic group to a dynamic group

To convert a basic group to a dynamic group, right-click the group, and then click **Convert to Dynamic Group** to start the New Membership Rule wizard. The first page of the wizard

looks as shown in the following figure.

Figure 108: Convert to dynamic group



On the first page of the wizard, you can select the type of the membership rule you want to configure. The text under **Membership rule description** explains which membership rules can be created using the rule type you select.

The **Include Explicitly** membership rule allows you to select objects to be statically added to the group. Active Roles ensures that the selected objects are included in the group regardless of whether they are renamed, moved to another container, or have any properties changed. With the **Include Explicitly** rule type the dynamic group behaves like a basic group.

The **Include by Query** membership rule allows you to define criteria the objects must match to be included in the group. Active Roles dynamically populates the group membership list with the objects that have certain properties. When an object is created, or when its properties are changed, Active Roles adds it to, or removes it from, the group depending on whether the object's properties match the defined criteria.

The **Include Group Members** membership rule allows you to select the groups which members you want to include in the dynamic group. Active Roles dynamically populates the group membership list with the objects that belong to the selected groups. When an object is added or removed from the selected groups, Active Roles adds or removes that object from the dynamic group.

The **Exclude Explicitly** membership rule allows you to select objects to be statically excluded from the group. Active Roles ensures that the selected objects are excluded from the group membership list regardless of whether they are renamed, moved, or have any properties changed. Because the **Exclude Explicitly** rule takes precedence over all other types of rule, the selected objects will be excluded from the group even if another rule states that they should be included.

The **Exclude by Query** membership rule allows you to define criteria the objects must match to be excluded from the group. Active Roles ensures that the objects with certain properties are excluded from the group membership list. Active Roles automatically

removes objects from the group depending on whether the objects' properties match the defined criteria.

The **Exclude Group Members** membership rule allows you to select groups whose members will be excluded from the given group. Active Roles ensures that the members of the selected groups are removed from the group membership list. When an object is added to any one of the selected groups, Active Roles automatically removes that object from the dynamic group.

On the first page of the wizard, select a rule type, and then click **Next**. On the next page of the wizard, click **Add** to configure the membership rule.

If you have selected the **Include Explicitly** or **Exclude Explicitly** rule type, you are presented with the **Select Objects** dialog box that lists users, groups, contacts, and computers. Select the objects you want to include or exclude from the dynamic group, click **Add**, and then click **OK**.

If you have selected the **Include Group Members** or **Exclude Group Members** rule type, the **Select Objects** dialog box appears. The list of objects in that dialog box consists of groups. Select groups, click **Add**, and then click **OK**. All members of the selected groups will be included or excluded from the dynamic group.

If you have selected the **Include by Query** or **Exclude by Query** rule type, the **Create Membership Rule** dialog box, similar to the **Find** dialog box, is displayed. In that dialog box, define the criteria that objects must match to be included or excluded from the dynamic group.

Click **Finish** to complete the New Membership Rule wizard.

> NOTE: After you have created a dynamic group with the first rule added to the group, you can add additional rules by managing properties of the group.

If you add several membership rules and some of them conflict with each other, then the conflict is resolved by a rule that defines the following order of precedence:

1. Exclude Explicitly
2. Include Explicitly
3. Exclude by Query
4. Exclude Group Members
5. Include by Query
6. Include Group Members

According to this, for example, the **Exclude Explicitly** rule takes precedence over all other types of rule. Therefore, the selected objects will be excluded from the dynamic group even if another rule states that they should be included (for example, the objects that match the criteria defined in the **Include by Query** membership rule, or members of a group selected in the **Include Group Members** rule).

Displaying the members of a dynamic group

For a dynamic group, the **Membership Rules** tab is added to the **Properties** dialog box. This tab displays a list of membership rules defined for the group, and allows you to add, remove, and edit the rules.

The **Members** tab for a dynamic group displays a list of objects that match the criteria specified in the membership rules. On that tab, you cannot add or remove members as you can for a basic group. To add or remove particular members from a dynamic group, you might add an appropriate **Include Explicitly** or **Exclude Explicitly** membership rule.

Adding a membership rule to a dynamic group

To add a membership rule to a dynamic group, right-click the dynamic group, and then click **Add Membership Rule**. This starts the New Membership Rule wizard. Complete the wizard as described in Converting a basic group to a dynamic group earlier in this chapter. To add a membership rule to a dynamic group, you can also use the **Membership Rules** tab in the **Properties** dialog box.

To add a membership rule to a group

1. In the console tree, select the folder that contains the group to which you want to add a membership rule.
2. In the details pane, right-click the group, and do one of the following to start the New Membership Rule wizard:
   - If the group is a basic group, click **Convert to Dynamic Group**, and then click **Yes**.
   - If the group is a dynamic group, click **Add Membership Rule**.
3. On the first page of the wizard, select the type of the membership rule you want to create. Do one of the following, and then click **Next**:
   - To create a rule that statically adds members to the group, click **Include Explicitly**.
   - To create a rule that statically excludes members from the group, click **Exclude Explicitly**.
   - To create a rule that adds all members of a certain group to the selected group, click **Include Group Members**.
   - To create a rule that excludes all members of a certain group from the selected group, click **Exclude Group Members**.
   - To create a rule that populates the group with the objects that match certain search criteria, click **Include by Query**.

- To create a rule that prevents the group from including the objects that match certain search criteria, click **Exclude by Query**.

4. On the next page of the wizard, click **Add**.

   If you selected the **Include by Query** rule type or the **Exclude by Query** rule type in Step 3, the **Create Membership Rule** dialog box appears. Otherwise, the **Select Objects** dialog box appears.

5. Complete the **Create Membership Rule** or **Select Objects** dialog box using the procedures outlined below in this section.
6. Click **Finish** to close the wizard.

To complete the Create Membership Rule dialog box

1. From the **Find** list, select the class of objects you want the membership rule to include or exclude from the group. For example, when you select **Users**, the membership rule includes or excludes the users that match the conditions you specify.
2. From the **In** list, select the domain or folder that holds the objects you want the membership rule to include or exclude from the group. For example, when you select an Organizational Unit, the membership rule includes or excludes only the objects that reside in that Organizational Unit.

   To add folders to the **In** list, click **Browse** and select folders in the **Browse for Container** dialog box.

3. Define the criteria of the membership rule. For example, to include or exclude the objects that have the letter T at the beginning of the name, type T in **Name**. You can use an asterisk (*) to represent any string of characters.
4. Optionally, click **Preview Rule** to view a list of objects that match the criteria you have defined.
5. Click **Add Rule**.

To complete the Select Objects dialog box

1. In the **Look in** list, click the domain or folder that holds the objects you want to select. To add a folder to the list, click **Browse**.
2. Do one of the following, and then click **OK**.
   - In the list of objects, double-click the object you want to add.
   - In the lower box, type the entire name, or a part of the name, of the object you want to add. Then, click **Check Names**.

NOTE:

- The only way to populate dynamic groups is by adding membership rules. The members of a dynamic group are the objects that match the criteria defined by the membership rules.
- To convert a dynamic group back to a basic group, right-click the group, and click **Convert to Basic Group**. When converting a dynamic group to a basic group, Active Roles removes all membership rules from the group. No changes are made to the list of the current members for that group.
- The **Create Membership Rule** dialog box is similar to the **Find** dialog box you use to search for objects in the directory. Once you have specified your search criteria, the **Add Rule** function saves them as a membership rule. For more information on how to specify search criteria, see Finding objects.
- The **Find** list includes the **Custom Search** entry. Selecting that entry displays the **Custom Search** tab, enabling you to build custom membership rules using advanced options, as well as to build advanced membership rules using the Lightweight Directory Access Protocol (LDAP), which is the primary access protocol for Active Directory. For more information about using advanced search options, see Steps for using advanced search options and Steps for building a custom search.

Removing a membership rule from a dynamic group

To remove a membership rule from a dynamic group, open the **Properties** dialog box for the group. On the **Membership Rules** tab, select the membership rules you want to remove, and click **Remove**. When finished, click **OK** to close the **Properties** dialog box.

NOTE: Active Roles does not allow members to be removed from a dynamic group by directly managing the membership list of the group. To remove particular members, use **Exclude Explicitly** rules.

To remove a membership rule from a group

1. In the console tree, locate and select the folder that contains the group from which you want to remove a membership rule.
2. In the details pane, right-click the group and click **Properties**.
3. On the **Membership Rules** tab, select the membership rule, and click **Remove**.

NOTE: The **Properties** dialog box includes the **Membership Rules** tab if the selected group is a dynamic group. If you do not see the **Membership Rules** tab, then the selected group is a basic group.

Converting a dynamic group to a basic group

When converting a dynamic group to a basic group, the membership rules only are removed from the group. The group membership list remains unchanged. To convert a dynamic group to a basic group, right-click the group, and then click **Convert to Basic Group**. In the confirmation message box, click **Yes**.

When a group is no longer dynamic, it becomes a basic group with the following characteristics:

- The **Membership Rules** tab disappears from the **Properties** dialog box.
- The **Members** tab allows you to add and remove members (the **Add** and **Remove** buttons appear on the **Members** tab).

Modifying, renaming, or deleting a dynamic group

You can manage dynamic groups in the same way as you manage basic (regular) groups — rename, modify properties, assign a Trustee when delegating control, and delete. The instructions on how to perform such management tasks on a Dynamic Group are the same as for regular groups. For step-by-step instructions on how to manage groups, see the "Group Management Tasks" section in the Active Roles User Guide or Active Roles Help.

Scenario: Automatically moving users between groups

This scenario removes a user from the **Seattle** group and adds the user to the **Atlanta** group when the user relocates to Atlanta from Seattle.

Suppose user accounts of employees working in Seattle belong to the **Seattle** group, and user accounts of those working in Atlanta belong to the **Atlanta** group. The group to which the user belongs is defined by the city attribute: employees working in Seattle have user accounts with the value Seattle for the **City** attribute. For those working in Atlanta, the value is Atlanta.

To implement this scenario, you must perform the following actions:

1. Create the Seattle and Atlanta groups.
2. Configure membership rules to add users with a city value of Seattle to the Seattle group, and those with Atlanta to the Atlanta group.

As a result, only user accounts that currently have a city value of Seattle belong to the Seattle group. Thus, when an employee leaves Seattle for Atlanta, an administrator changes the **City** attribute from Seattle to Atlanta, and the user automatically moves to the

Atlanta group because of the membership rule. Conversely, when an employee leaves Atlanta for Seattle, the administrator changes the city attribute from Atlanta to Seattle, and the user automatically transfers to the **Seattle** group.

The following sections elaborate on the steps to implement this scenario.

Step 1: Creating the groups

To create the **Seattle** group, in the console tree, right-click the container where you want to add the group, and select **New | Group**. Follow the instructions in the New Object – Group wizard. In the **Group name** box, type **Seattle**.

To create the **Atlanta** group, in the console tree, right-click the container where you want to add the group, and select **New | Group**. Follow the instructions in the New Object – Group wizard. In the **Group name** box, type **Atlanta**.

Step 2: Configuring the membership rules

In this scenario, employees working in Seattle have user accounts with a value of Seattle for the **City** attribute. Those working in Atlanta have a value of Atlanta.

First, configure the membership rule for the **Seattle** group. Right-click the group and click **Convert to Dynamic Group**. In the confirmation message box, click **Yes**.

On the first page of the New Membership Rule wizard, click **Include by Query**, and then click **Next**.

On the second page, click **Add** to display the **Create Membership Rules** dialog box. Then, follow these steps to configure the membership rule:

1. In the **Find** list, click **Users**.
2. Click **Browse** and select the domain, OU, or Managed Unit that holds user accounts of the employees.
3. Click the **Advanced** tab.
4. Click **Field**, click **City**, and then click **OK** in the **Select Object Property** dialog box.
5. In the **Condition** list, click **Is (exactly)**.
6. In the **Value** box, type **Seattle**.
7. Click **Add**, and then click the **Add Rule** button.

When you are done, click **Finish** in the New Membership Rule wizard.

Repeat the same procedure for the **Atlanta** group, but type **Atlanta** in the **Value** box when configuring the membership rule.

11

Active Roles Reporting

- Introduction
- Collector to prepare data for reports
- Working with reports

Introduction

The Active Roles reporting solution leverages Microsoft SQL Server Reporting Services (SSRS) as a platform for managing, generating, and viewing reports.

Through the use of SSRS, Active Roles delivers enterprise reporting functionality that combines the strengths of Web-based features and traditional reporting. The use of Reporting Services provides a way to centralize report storage and management, enable secure access to reports, control how reports are processed and distributed, and standardize how reports are used.

A comprehensive collection of report definitions, referred to as the *Active Roles Report Pack*, are published to the report server, a component of Reporting Services. Installing the Report Pack creates published reports that can be accessed through Web addresses (URLs), through SharePoint Web parts, or through Report Manager, a Web-based report access and management tool included with SSRS.

Opening a published report from the report server generates the report in a format suitable for viewing. This action is referred to as *rendering a report*. Rendering a report also occurs upon subscription, when the report is delivered to an e-mail inbox or a file share in an output format specified by the report user.

The reports that can be generated once the Active Roles Report Pack is deployed are instrumental in change tracking audits, directory data monitoring and analysis, and assessment of Active Roles security and policy configurations. The reports fall into these categories:

- **Active Roles Tracking Log** Check what changes were made to directory data through the use of Active Roles, who made the changes, and when the changes were made.

- **Active Directory Assessment** Examine the state of directory data, such as properties of users, groups and other directory objects, group membership lists, and contents of organizational units.
- **Administrative Roles** View details on who has access to what data when using Active Roles, and what changes administrative users or groups are authorized to make.
- **Managed Units** View details on the Managed Units defined in the Active Roles environment, what policies are applied to Managed Units, and what users or groups have administrative access to what Managed Units.
- **Policy Objects** View details on what administrative policies are defined in the Active Roles environment, where particular policies are applied, and what policies are in effect on particular objects and containers.
- **Policy Compliance** View details on what data in the directory is not compliant with Active Roles policies that are in effect, and what policy rules are violated.

Reports are built on data prepared by the Active Roles Collector. For details about the Active Roles Collector, see Collector to prepare data for reports later in this chapter.

You can generate and view reports by using Report Manager, which is part of SSRS. For instructions on how to generate and view reports, see Working with reports later in this chapter.

Collector to prepare data for reports

The Active Roles Collector allows you to collect data from computers running the Administration Service and store it in an on-premises or Azure SQL database, making the data available for reporting.

NOTE: The Collector is installed as a separate component of Active Roles.

Data for reports is collected from the following sources:

- **Active Directory** The Collector accesses Active Directory through the Administration Service. Reports built on this data provide detailed information about domains, accounts, groups, and other Active Directory objects.
- **Active Roles configuration database** Reports built on this data provide detailed information about who can carry out what actions and to which directory objects using Active Roles, as well as information about the policies defined by Active Roles.
- **Event log on computers running the Administration Service** Reports built on this data provide detailed information about actions performed, the success or failure of each action, and object properties that were modified using Active Roles.

The scope of data that the Collector can retrieve from Active Directory is restricted by the access rights of the user account under which the Collector performs the data collection task. Therefore, reports based on Active Directory data only include information about the objects that the Collector is permitted to access in Active Directory.

For example, suppose the Collector performs a data collection task under the user account that is not permitted to access user account properties in Active Directory. As a result, the Collector will not be able to retrieve data related to user accounts, and reports will not display any information about user accounts, including the number of user accounts.

Starting the Active Roles Collector wizard

To start the Active Roles Collector wizard

- Depending upon the version of your Windows operating system, click **One Identity Active Roles | Active Roles Collector and Report Pack** on the **Apps** page or select **All Programs |One Identity Active Roles | Active Roles Collector and Report Pack** from the **Start** menu.

When started, the Collector wizard displays the **Select Task** page where you can select one of the following the tasks to perform:

- **Collect data from the network** Collect data and events from the computers running the Administration Service, and store the collected information in a database server to make the information available to the report server.
- **Process gathered events** Export selected events to another database server, or delete obsolete information from the database.
- **Import events from an earlier database version** As the current version of the Active Roles reports is only compatible with the database of the current Collector version, you need to import events from the database of an earlier version to the database of the current version if you want to use those events for reporting.
- **Deploy reports to Report Server** Setup only installs the Active Roles report definitions to the local computer. To use the reports, you need to publish them to your SQL Server Reporting Services (SSRS) Report Server.

Collecting data from the network

If you select the option **Collect data from the network** on the **Select Task** page, the Collector wizard displays the **Configure Connection** page on the next step.

On the **Configure Connection** page, the wizard prompts you to specify the basic options for managing the data: the database in which you want to store the collected data; the source computer running the Administration Service; and the credentials to log on to that computer.

To specify a database, click the button next to the **Database** box. In the dialog box that appears, you can specify the desired database and authentication option for connection to SQL Server.

In the **Active Roles Service** box, type the full name of the computer running the Administration Service from which you want to collect information.

In the **Log on as** area, specify the credentials that the Collector will use to connect to the Administration Service. You can choose one of the following options:

- **Current user** Connect to the Administration Service with the credentials of the user account under which the Collector is running.
- **Specified user** Connect to the Administration Service with the specified user name and password.

Click **Next** to proceed to the **Data Collection Tasks** page.

On the **Data Collection Tasks** page, the wizard prompts you to select the sources of the data you want Collector to retrieve:

- **Active Directory** Collect information about users, groups, computers, organizational units, and domains from Active Directory.
- **Policy Compliance Information** Collect data to determine whether directory objects comply with the policies defined by Active Roles. This option requires the **Active Directory** option to be selected.
- **Active Roles event log** Collect events from the Active Roles event log on the computers running the Administration Service.

Click **Next** to proceed to the **Data to Collect** page.

NOTE: The wizard only displays the **Data to Collect** page if you select the **Active Directory** check box on the **Data Collection Tasks** page.

On the **Data to Collect** page, the wizard prompts you to specify the categories of data you want to collect:

- **Access Templates** Information about Access Templates defined in your Active Roles environment.
- **Policy Objects** Information about Policy Objects defined in your Active Roles environment.
- **Managed Units** Information about Managed Units defined in your Active Roles environment.
- **Script Modules** Information about Script Modules defined in your Active Roles environment.

NOTE: If you select the **Policy Compliance Information** check box on the previous page, the wizard does not allow the **Policy Objects** check box to be cleared on the **Data to Collect** page.

Click **Next** to proceed to the **Select Domains or OUs** page.

On the **Select Domains or OUs** page, the wizard prompts you to specify the domains or containers from which you want to collect information. You can complete this page as follows:

- Click **Add** to select a domain or OU to add to the list on the page.
- Click **Remove** to delete a selected domain or OU from the list.

When selecting a domain or OU, you have the option to force the wizard to collect information about all objects held in the selected domain or OU: select the **Use subtree search** check box in the dialog box that appears when you click **Add**. If you clear the **Use subtree search** check box, the wizard only collects information about the immediate child objects of the selected domain or OU.

Click **Next** to proceed to the **Select Operation Mode** page.

On the **Select Operation Mode** page, you can specify whether to start the task execution immediately or schedule the task to run at a convenient time. You can also disable SID resolving for faster data collection.

If you want to start the collection process right now, select **Now** under **Run Active Roles Collector** and click **Next**. While the wizard performs the operation you can see the progress screen, showing you the progress details.

When the operation is completed, the wizard displays the final screen that shows you the operation results. You can click **View Log** to examine operation log for possible errors.

If you want to schedule the task, select **On a schedule** and click **Next.** This displays the **Schedule** page where you can specify the task schedule and logon account. Click **Add** to create a schedule for the task. In the **User account under which the task will run** area, supply the user name and password of the user account under which you want the task to run. Once the scheduling options are set, click **Next** to complete the wizard.

Steps for collecting data from the network

You use the Active Roles Collector to prepare data for reporting. The data is stored in the database you specify. Then, in order to make the data available to the report server, you need to configure the data source on the report server to connect to the database that holds the data. This topic provides instructions on how to prepare report data. For instructions on how to configure the data source for the Active Roles Report Pack, see Configuring the data source later in this chapter.

To collect data from the network, start the Collector wizard (see Starting the Active Roles Collector wizard), and complete the wizard pages as follows.

1. On the **Select Task** page, select the **Collect data from the network** option.
2. On the **Configure Connection** page, specify the database in which you want to store the collected data; the computer running the Administration Service; and the credentials to log on to that computer:
    a. To initially specify a database, or choose a different database, click the button next to the **Database** box, and then use the dialog box that appears to specify the required database type, database, and authentication option for connection to database server.
    b. In **Active Roles Service**, specify the full name of the computer running the Administration Service from which you want to collect information.

c. Under **Log on as**, click one of these options:

- **Current user** to connect to the Administration Service with the user account under which the Collector is running.
- **Specified user** to specify the user name and password you want the Collector to use when connecting to the Administration Service.

3. On the **Data Collection Tasks** page, specify the sources of data you want to collect. Select or clear these check boxes as appropriate:
   - **Active Directory** to collect information about users, groups, computers, organizational units, and domains from Active Directory.
   - **Policy Compliance Information** to collect information on whether Active Directory data is in compliance with the policies defined by Active Roles. If you select this check box, the **Active Directory** check box is selected as well.
   - **Active Roles event log** to collect information from the Active Roles event log on the computers running the Administration Service.
4. On the **Data to Collect** page, specify the categories of Active Roles data you want to collect. Select or clear these check boxes as appropriate:
   - **Access Templates** to collect information about Access Templates defined in your Active Roles environment.
   - **Policy Objects** to collect information about Policy Objects defined in your Active Roles environment.
   - **Managed Units** to collect information about Managed Units defined in your Active Roles environment.
   - **Script Modules** to collect information about Script Modules defined in your Active Roles environment.
5. On the **Select Domains or OUs** page, specify the domains or containers from which you want to collect information:
   - Click **Add** to select a domain or OU to add to the list on the page.
   - Click **Remove** to delete a selected domain or OU from the list.

   When selecting a domain or OU, you have the option to force the wizard to collect information about all child objects of the selected domain or OU: Select the **Use subtree search** check box in the dialog box that appears when you click **Add**. If you clear the **Use subtree search** check box, the wizard only collects information about the immediate child objects of the selected domain or OU.
6. On the **Select Operation Mode** page, specify whether to start the task execution immediately or schedule the task to run at a convenient time:
   - To start the collection process right now, click **Now**, and then click **Next**.
   - To schedule the task, select **On a schedule**, and then click **Next**.
7. If you selected the **On a schedule** option, then, on the **Schedule** page, specify the task schedule and logon account:
   - Click **Add** to create a schedule for the task.

- In the **User account under which the task will run** area, supply the user name and password of the user account under which you want the task to run.

The user account under which the task will run must have the "Log on as a batch job" right. Use Group Policy security settings to assign that right to the user account. Members of the Administrators or Backup Operators group have the "Log on as a batch job" right by default.

You can use the Task Scheduler console to examine the Collector task that you have scheduled. Task Scheduler allows you to view or change the task's properties such as task's name, description, security options, triggers, conditions, and settings. The task's history can also be viewed along with the properties. Task Scheduler tracks the task's history by events that are raised when the task is started, run, finished executing, and at other times as needed to track the task's history. Errors related to the task are also tracked in the task's history.

To view the task's properties and history by using Task Scheduler

1. If Task Scheduler is not open, start Task Scheduler.

   You can start Task Scheduler by entering **Taskschd.msc** at a command prompt.

2. In the console tree, select **Task Scheduler Library | Active Roles | Collector**.
3. In the console window, double-click the name of the task.

   The name of the task in the Task Scheduler console has the following format: **Active Roles Collector (** *<task name>* **)** where *<task name>* stands for the name you specified in the Collector wizard; for example, **Active Roles Collector (New Task)**.

4. In the dialog box that appears, click a tab to view or change the task's properties located on that tab.
5. Click the **History** tab to view the task's history.

   The **History** tab lists the events specific to the task you selected. Click an event in the list to view the description of the event.

Processing gathered events

If you select the option **Process gathered events** on the **Select Task** page, the Collector wizard displays the **Data Processing Task** page on the next step.

On the **Data Processing Task** page, the wizard prompts you to specify what you want to do with the events that were gathered from the Administration Service computers and stored in the database. You can choose one of the following options:

- **Export using date range** Specify the date range for the events you want to export. The time you specify is considered Greenwich Mean Time (GMT).
- **Export events older than** Specify the age limit for the events you want to export.
- **Delete events older than** Specify the age limit for the events you want to delete.

Click **Next** to proceed to the **Source database** page allowing you to specify the database from which you want to export or delete information. Click the button next to the **Database** field. In the dialog box that appears, you can specify the desired database and authentication option for connection to SQL Server.

When finished, click **Next** to continue.

If you have chosen the option to export information, the wizard displays the **Target Database** page prompting you to specify the database to which you want to export information. Click the button next to the **Database** field. In the dialog box that appears, you can specify the desired database and authentication option for connection to SQL Server.

When finished, click **Next** to start the operation.

While the wizard performs the operation you selected, you can see the progress screen, showing you the progress details. When the operation is completed, the wizard displays the final screen that shows you the operation results. You can click **View Log** to examine the operation log for possible errors.

Steps for processing gathered events

To process the gathered events, start the Collector wizard (see Starting the Active Roles Collector wizard), and complete the wizard pages as follows.

1. On the **Select Task** page, select the **Process gathered events** option.
2. On the **Data Processing Task** page, specify what you want to do with the events that were gathered from the Administration Service computers and stored in the database. Select one of the following options:
   - **Export using date range** Specify the date range for the events you want to export. The time you specify is considered Greenwich Mean Time (GMT).
   - **Export events older than** Specify the age limit for the events you want to export.
   - **Delete events older than** Specify the age limit for the events you want to delete.
3. On to the **Source database** page, click **Specify**, and supply the name and SQL Server of the database from which you want to export or delete the events. You can also choose the authentication option for connection to SQL Server.
4. On to the **Target Database** page, click **Specify** box, and supply the name and SQL Server of the database to which you want to export the events. You can also choose the authentication option for connection to SQL Server.

Importing events from an earlier database version

The new version of the Active Roles reports is incompatible with the database of an earlier Collector version. To create reports based on the events held in that database, you need to import the events to the database of the new Collector version, and then specify the database of the new Collector version as the data source for the reports of the new Report Pack version. For instructions on how to configure the data source, see Configuring the data source later in this document.

To import events from the database of an earlier Collector version, start the Collector wizard (see Starting the Active Roles Collector wizard), and complete the wizard pages as follows.

1. On the **Select Task** page, select the option **Import events from an earlier database version**.
2. On the **Source database** page, click **Specify**, and supply the name, database type, and SQL database server used by your Collector of an earlier version. You can also choose the authentication option for connection to SQL Server.
3. On the **Target Database** page, click **Specify**, and supply the name, database type, and database server of the database used by your Collector of the current version. You can also choose the authentication option for connection to SQL Server.

Deploying reports to the Report Server

Active Roles reports require Microsoft SQL Server Reporting Services (SSRS). Make sure that you have SSRS in your environment. To use Active Roles reports, you first need to deploy them to your SSRS Report Server by using the Collector wizard.

To deploy the Active Roles reports to the Report Server, start the Collector wizard (see Starting the Active Roles Collector wizard), and complete the wizard pages as follows.

1. On the **Select Task** page, select the **Deploy reports to Report Server** option.
2. On the **Report Server** page, type the URL of your SSRS Report Server in the **Report Server Web Service URL** box.

   By default, the URL is http://*<serverName>*/ReportServer. You can use the Reporting Services Configuration Manager tool to confirm the server name and URL. For more information about URLs used in Reporting Services, see the topic "Configure Report Server URLs (SSRS Configuration Manager)" at http://msdn.microsoft.com/library/ms159261.aspx.
3. Optionally, on the **Data Source** page, configure the data source for the Active Roles reports:

a. Click the **Configure Data Source** button.

b. Use the **Configure Data Source** dialog box to specify the Database Server instance that hosts the database you have prepared by using Collector, the name of the database type, and the authentication method to use for connection to the database.

Configuring the data source is an optional step. If you do not have a database prepared by Collector, you can configure the data source later, after you have deployed the reports. For instructions, see Configuring the data source later in this document.

Once you have deployed the reports to your SSRS Report Server and configured the data source, you can create and view Active Roles reports using Report Manager, a Web-based tool included with SSRS. For instructions, see Generating and viewing a report later in this document.

Working with reports

You use the Active Roles Collector to prepare data for reporting. The data is stored in the database you specify when configuring the data collection job (see Collector to prepare data for reports earlier in this chapter). In order to make the data available to the report server, the data source on the report server must be configured to connect to the database that holds the report data. Then, you can generate and view Active Roles reports.

Configuring the data source

You have the option to configure the data source when deploying Active Roles reports to the report server (see Deploying reports to the Report Server earlier in this document). If you have not configured the data source, or need to change the data source, then you can do this by using Report Manager on the report server on which the Active Roles reports were deployed.

To configure the data source by using SSRS Report Manager

1. Start SSRS Report Manager from your Web browser.

   Report Manager is installed during setup of SQL Server Reporting Services (SSRS) on the same computer as the report server. To start Report Manager, open your web browser and type the Report Manager URL in the browser address bar. By default, the URL is http://<*ComputerName*>/reports.

2. Perform the following steps on the Contents page that appears:

   a. Click **Active Roles**. The **Version** and **SharedDataResources** components are displayed.

b. Click **SharedDataSources**.

c. Click the data source named Active Roles 7.4 Report Data.

If the **SharedDataSources** item is not displayed, click **Details View**.

3. In the **Connection string** box on the **Properties** page that appears, specify the database server instance, database type, and the name of the database that holds the report data prepared by the Active Roles Collector.

   For example, if the name of the database is ARServerReporting and the database is on the SQL Server instance named MyServer\Enterprise, then the connection string is as follows:

   ```
   data source = MyServer\Enterprise; initial catalog = ARServerReporting
   ```

4. Click **Apply**.

Generating and viewing a report

You can generate and preview Active Roles reports using SSRS Report Manager. This section provides basic instructions on how to use Report Manager for this purpose.

The following instructions assume that report data has been prepared by using the Active Roles Collector and the data source on the report server has been configured to connect to the database that holds the report data.

To view a report by using SSRS Report Manager

1. Start SSRS Report Manager from your Web browser.

   To start Report Manager, open your web browser and type the Report Manager URL in the browser address bar. By default, the URL is http://<*ComputerName*>/reports.

2. Click **Active Roles** on the Contents page that appears.
3. Find a report by browsing folders or searching for a report by name.

   Browse folder contents by clicking a folder name or folder icon on the Contents page. Search for a report by typing all or part of the report name in the **Search** text box at the top of that page.

4. To view a report, click the name of the report.

   Some reports require you to provide parameter values. You can also apply filters to specify what data you want the report to include.

5. Click the **View Report** button at the top of the page.

For detailed instructions on how to use Report Manager, refer to Microsoft SQL Server Books Online.

Contents of the Active Roles Report Pack

This section lists the reports provided by the Active Roles Report Pack. The list is organized into sub-sections. Each sub-section heading identifies the path to a certain report folder, with the reports contained in that folder being listed under the sub-section heading.

Active Directory Assessment/Domains/

- **Domain Summary** Lists the Active Directory domains in your environment. For each domain, the following information is provided: description, canonical name, functional level, creation date and last change date, and statistical data about the number of accounts of different types held in the domain.
- **Domain Trusts** For each Active Directory domain, lists the domains that the given domain trusts (trusted domains) and the domains that trust the given domain (trusting domains).
- **Domain account SID resolution** For each security principal object, lists the Security ID (SID) along with the name of the object. Security principals are accounts in Active Directory that can be assigned permissions, such as user accounts, groups, or computer accounts. Active Directory automatically assigns a unique SID to each security principal object at the time the object is created.

Active Directory Assessment/Users/Account Information/

- **User account list** Lists the Active Directory domain user accounts held in a given domain or container (Organizational Unit).
- **User account options** Lists Active Directory domain user accounts along with information about the state of the account options such as "User must change password at next logon" and "Password never expires."
- **Password age information** Lists Active Directory domain user accounts along with information about the account's password age. For each listed account, its password age is calculated using the pwdLastSet attribute of the account. The password age information helps determine when the user last changed their password.
- **Bad password information** Lists Active Directory domain user accounts along with information about the number of times the user tried to log on to the account using an incorrect password and the last time the user tried to log on using an incorrect password.

Active Directory Assessment/Users/Exchange/

- **Mailbox information by user** Lists Active Directory user accounts along with information on whether the user account is mailbox-enabled (has an Exchange mailbox), allowing you to examine the user's mailbox-related information in detail.
- **Email delivery restrictions** Lists Exchange mailbox-enabled user accounts along with information on mailbox delivery restrictions such as the maximum size of incoming and outgoing messages for the mailbox, and from whom the mailbox can or cannot receive e-mail.
- **Email delivery options** Lists Exchange mailbox-enabled user accounts along with information on mailbox delivery options such as who is allowed to send messages on behalf of the mailbox user, the forwarding address for messages addressed to the mailbox, and the maximum number of recipients to whom the mailbox user can send a message.

Active Directory Assessment/Users/Obsolete Accounts/

- **Disabled user accounts** Lists Active Directory domain user accounts that are currently disabled, and allows you to examine each account in detail.
- **Expired user accounts** Lists Active Directory domain user accounts that are past their expiration date, and allows you to examine each account in detail.
- **Inactive user accounts** Lists Active Directory domain user accounts that have not been used to log on within a given time period, and allows you to examine each account in detail.
- **Locked user accounts** Lists Active Directory domain user accounts that are currently locked out because of a number of failed logon attempts, and allows you to examine each account in detail.
- **User accounts with expired password** Lists Active Directory domain user accounts whose password is past the password expiration date, and allows you to examine each account in detail.
- **Deprovisioned user accounts** Lists Active Directory domain user accounts that have been deprovisioned by Active Roles, and allows you to examine each account in detail.
- **All discontinued user accounts** Lists Active Directory domain user accounts that are not in use for whatever reason, such as accounts that are disabled, expired, locked, deprovisioned or accounts whose password has expired, and allows you to examine each account in detail.

Active Directory Assessment/Users/Miscellaneous Information/

- **Users with specified properties** Lists Active Directory domain user accounts that have the properties you specify, and allows you to examine each account in detail.
- **User profile information** Lists Active Directory domain user accounts along with information on their profile settings such as the path to the user's profile, the name of the logon script, and the path to the user's home folder.
- **Objects managed by user** Lists Active Directory domain user accounts along with information about their managed objects. For a given account, the list of managed objects contains the objects whose 'Managed By' property specifies that account.
- **Personnel Hierarchy** Lists Active Directory domain user accounts along with information about their manager and subordinates. The manager ID is retrieved from the account's 'Manager' property; the list of subordinates is based on the 'Direct Reports' property.

Active Directory Assessment/Groups/

- **Domain group statistics** Lists the number of groups in a given Active Directory domain, itemized by group type (security or distribution) and group scope (universal, global, or domain local). Allows you to view a list of all groups of a particular type and scope, along with detailed information about each group.
- **Group list with member statistics** Lists the groups defined in a given Active Directory domain, along with information on how many members each group contains. For every group, allows you to view a list of its members.
- **Group Hierarchy** Lists the groups defined in a given Active Directory domain, representing the group nesting structure in a tree-like view. For every group, allows you to view a list of its member groups, and to examine each group in detail.
- **Empty Groups** Lists the groups defined in a given Active Directory domain that have no members.

Active Directory Assessment/Group Membership/

- **Group membership by group** For each group defined in a given Active Directory domain, lists the members of that group. Allows you to configure the list to include only members of a certain type (such as user, computer, or group), only direct members, or both direct members and members that belong to the group through group nesting.
- **Group membership by user** For each user account defined in a given Active Directory domain, lists the groups to which the user account belongs as a member. Allows you to configure the list to include only groups of a certain type and scope,

only groups of which the user is a direct member, or both groups of which the user is a direct member and groups to which the user belongs through group nesting.

- **Users with domain administrative rights** For a given Active Directory domain, lists the user accounts that belong to the built-in Administrators group in that domain whether as direct members or as members of other groups nested into the Administrators group. Allows you to examine each of the listed accounts in detail.

Active Directory Assessment/Organizational Units/

- **Member statistics by OU** Provides information on how many objects are held in each Organizational Unit. The list is split by object type, allowing you to view the number of objects of each individual type, such as the number of users, computers, groups, contacts, printers and shared folders. By clicking a number in the list you can examine the objects represented by that number.
- **Organizational Unit membership** For each Organizational Unit (OU), lists the objects held in that OU. The report is split by object type, allowing you to view the objects of each individual type in a separate list. You can view information about the following objects: users, computers, groups, contacts, printers and shared folders.
- **Organizational Unit hierarchy** Lists the Organizational Units (OUs) defined in a given Active Directory domain, representing the parent-child structure of OUs in a tree-like view. You can use this report to determine all OUs that are descendants of a particular OU, observing the entire tree of the OUs rooted in that OU.

Active Directory Assessment/Other Directory Objects/

- **Active Directory Object Properties** Lists the objects that meet the conditions you specify. For each object, provides information about its properties, allowing you to choose the properties to be displayed.
- **Computer Accounts** Lists the computer accounts held in a given domain or container (Organizational Unit). You can filter the list by various characteristics, such creation date, status (enabled or disabled) or computer operating system, of computer accounts.
- **All discontinued computer accounts** Lists the computer accounts that are not in use for whatever reason, such as accounts that are disabled, expired, not used for logon during a certain time period, or accounts whose password has expired, and allows you to examine each account in detail.

Active Directory Assessment/Potential Issues/

- **Cycled Groups** Lists the Active Directory groups, if any, each of which is a member of itself. You can use this report to determine whether your Active Directory domain has any group configured to contain itself as a member (for instance, group A is a member of group B which in turn is a member of group A). Note that such a configuration may cause administrative issues.

Active Roles Tracking Log/Active Directory Management/

- **User attribute management** Lists the changes that were made to Active Directory domain user accounts via Active Roles, allowing you to determine when and by whom individual user properties were changed, and view the values to which the properties were changed. You can filter the list by time period when changes occurred, name of the person who made changes, and name of the properties that were changed.
- **Directory object management** Lists the changes that were made to any objects in Active Directory via Active Roles, allowing you to examine the changes in detail and determine when and by whom the changes were made. You can configure various conditions to filter the list by object type (such as user, computer, group, or any other object type), category of changes (such as creation, modification, or deletion of objects), object properties that were changed, time period when changes occurred, and name of the person who made changes.
- **Deprovisioning of User Accounts** Lists the Active Directory domain user accounts that were deprovisioned via Active Roles, allowing you to determine when and by whom individual user accounts were deprovisioned. You can filter the list by time period when user accounts were deprovisioned, name of the person who deprovisioned user accounts, and name and location of deprovisioned user accounts.

Active Roles Tracking Log/Dashboard/

- **User Account Management** Lets you see how many user management operations were performed during a certain time period. The following operation types are covered: (1) Create; (2) Modify (change properties); (3) Add to groups; (4) Remove from groups; (5) Add in place of current group members; (6) Delete. You can specify your preferred time period. For each operation type, the report displays a separate graph indicating the number of the operations performed at particular points in time within the specified time period.

Active Roles Tracking Log/Active Roles Events/

- **Active Roles startup failures** Lists occurrences of a situation where Active Roles Administration Service failed to start, along with information about the cause of each failure (failure reason). You can filter the list by time period when startup failures occurred.
- **Active Roles event statistics** Lists Active Roles events and groups them by date (when the events occurred), by user (who initiated events), by computer (where events were logged), or by event category. You can filter the list by time period, event category and event ID.

Active Roles Tracking Log/Active Roles Configuration Changes/

- **Control Delegation** Lists Active Roles Access Templates that are applied to configure administrative permissions in Active Roles. For each Access Template, the report lists the objects to which the Access Template is linked, and informs of when and by whom the Access Template link was created. You can filter the list by Access Template name, name of the object to which the Access Template is linked, time period when the link was created, and name of the user who created the link.
- **Policy Enforcement** Lists Active Roles Policy Objects that are applied to configure administrative policy in Active Roles. For each Policy Object, the report lists the objects to which the Policy Object is linked, and informs of when and by whom the Policy Object link was created. You can filter the list by Policy Object name, name of the object to which the Policy Object is linked, time period when the link was created, and name of the user who created the link.

Active Roles Tracking Log/Active Roles Workflow/

- **Approvals and Rejections** Lists operation requests that were submitted via Active Roles and approved or rejected during the specified period of time, allowing you to examine approver actions. You can filter the list by name of the person who approved or rejected requests (approver), name of the person whose requests were subject to approval (initiator), approval decision (approved or rejected requests), and name and location of operation target objects. You can group the list by approver, initiator, or operation target object.
- **Workflow Monitoring** Lists events specific to Active Roles workflow and groups them by operation that started workflow or by name of workflow, allowing you to monitor workflow instances. For each workflow instance, the report identifies the operation request that caused the instance to start, and lists the date and time that the instance was started, the person who submitted the operation request (initiator), the operation target object, the server intended to perform the request, along with

all events that occurred during the lifetime of the workflow instance. You can filter the list of workflow instances by various parameters, such as date and time, operation ID, workflow name, operation initiator, target object, event category, and event ID.

Administrative Roles/

- **Access Template Permissions** Lists Active Roles Access Templates, allowing you to examine each Access Template in detail. You can view the name, location and description the Access Template, along with all permission entries held in the Access Template.
- **Access Template summary** Lists Active Roles Access Templates along with quantitative information regarding Access Template links. For each Access Template, this report allows you to determine the number of links that use the Access Template and the number of objects (Trustees and Containers) to which the Access Template is linked.
- **Access Templates linked to Managed Units** Lists Active Roles Access Templates that are linked to Active Roles Managed Units. Identifies the name, location and description of each Access Template along with the fully qualified name of every Managed Unit to which the Access Template is linked. You can extend the list to include both the Managed Units to which the Access Template is linked and the Managed Units that are affected by the Access Template through permission inheritance.
- **Access Templates linked to Organizational Units** Lists Active Roles Access Templates that are linked to Active Directory Organizational Units. Identifies the name, location and description of each Access Template along with the fully qualified name of every Organizational Unit to which the Access Template is linked. You can extend the list to include both the Organizational Units to which the Access Template is linked and the Organizational Units that are affected by the Access Template through permission inheritance.
- **Control delegation by object** Lists Active Directory objects to which Active Roles Access Templates are linked. Identifies the name, location and description of each object along with the name of every Access Template linked to that object and the security principal (Trustee) whose administrative permissions are determined by that link through direct assignment (without considering permission inheritance).
- **Control delegation by object (with group hierarchy)** Lists Active Directory objects to which Active Roles Access Templates are linked. Identifies the name, location and description of each object along with the name of every Access Template linked to that object and the security principals (Trustees) whose administrative permissions are determined by that link through direct assignment or because of group memberships.
- **Control delegation by Trustee** Lists Active Directory security principals (Trustees) that have administrative permissions specified by applying Active Roles Access Templates. Identifies the name of each Trustee along with the name of every Access Template that determines the Trustee's administrative permissions in Active

Roles, as well as the name of the container or leaf object to which the Access Template is linked thereby providing the Trustee with administrative permissions over that container or leaf object.

- **Control delegation by Trustee (with container hierarchy)** Lists security principals (Trustees) that have administrative permissions specified by applying Active Roles Access Templates, and provides detailed information about securable objects and containers for which the Trustee has administrative permissions and Access Templates that determine the Trustee's permissions. You can filter the list of Trustees by various parameters, including Trustee name and type, securable object or container name and type, Access Template name and type, permission name and type, and permission inheritance type.

Managed Units/

- **Managed Unit members** Lists Active Roles Managed Units along with their members. For each Managed Unit, identifies its name, path and description as well as the name, type and description of every object held in that Managed Unit.
- **Managed Unit membership rules** Lists Active Roles Managed Units along with their membership rules. For each Managed Unit, identifies its name, path and description as well as the rules that determine what objects are included to, or excluded from, that Managed Unit.
- **Managed Unit summary** Lists Active Roles Managed Units along with quantitative information regarding Managed Unit members, membership rules, Trustees and policies. For each Managed Unit, identifies the number of its members and membership rules, the number of security principals (Trustees) that have administrative permissions for that Managed Unit, and the number of Active Roles Policy Objects that affect the Managed Unit.
- **Managed Units affected by Policy** Lists Active Roles Managed Units that are affected by Active Roles Policy Objects whether through a Policy Object linked to the Managed Unit itself or through a Policy Object linked to a container or another Managed Unit that holds the given Managed Unit. For each Managed Unit, identifies the name and description of every Policy Object that affects the Managed Unit as well as the container or Managed Unit from which the policy effect is inherited.
- **Managed Units with delegated control** Lists Active Roles Managed Units that have administrative control delegated by applying Active Roles Access Templates whether to the Managed Unit itself (direct permissions) or to a container or another Managed Unit that holds the given Managed Unit (inherited permissions). For each Managed Unit, identifies the security principals (Trustees) to which administrative control is delegated, the Access Templates that determine the administrative permissions, and whether those are direct or inherited permissions.

Policy Objects/

- **Linked Property Validation Settings** Lists object properties that are under the control of any Property Generation and Validation policy defined in Active Roles. For each property, lists the object classes possessing that property, identifies the Policy Objects that affect the property, the container to which the Policy Object is linked, and the policy conditions. The report only includes containers to which Policy Objects are linked directly, without considering policy inheritance. You can filter the list of properties by various parameters, such as property name, object class name, container name, and Policy Object name.
- **Linked Property Validation Settings (with inheritance)** Lists objects along with their properties that are under the control of any Property Generation and Validation policy defined in Active Roles. An object included in this report may have a Policy Object linked to the object itself (direct policy) or to a container that holds the object (inherited policy). The report groups the list of objects by property. For each property, the report lists the objects possessing that property, identifies the Policy Objects and policy conditions that affect each of the listed objects, and indicates whether this is a direct or inherited policy. You can filter the list of objects and object properties by various parameters, such as property name, object name and type, and Policy Object name.
- **Linked Script Settings (with inheritance)** Lists objects that are under the control of any script-based (Script Execution) policy defined in Active Roles. An object included in this report may have a Policy Object linked to the object itself (direct policy effect) or to a container that holds the object (inherited policy effect). The report identifies the script-based Policy Objects that affect each of the listed objects, along with the origin of the policy effect (direct or inherited). You can filter the list of objects by various parameters, such as object name, object class name, and Policy Object name.
- **Policy Object references** Lists Active Roles Policy Objects that are applied (linked) to any container or Managed Unit. For each Policy Object, identifies its name, description and category (provisioning or deprovisioning), and lists the container to which the Policy Object is linked. You can filter the list by Policy Object name, container or Managed Unit name, and Policy Object category.
- **Policy Object Settings** Lists Active Roles Policy Objects together with their policy entries. For each Policy Object, provides detailed information about all policies that are defined in the Policy Object. You can filter the list by Policy Object name, policy type, and policy entry name.
- **Policy Object summary** Lists Active Roles Policy Objects together with the following information for each Policy Object: name; type (provisioning or deprovisioning); number of directory objects to which the Policy Object is linked (reference number); total number of individual policies defined in the Policy Object (entry number); number of policies of each particular type defined in the Policy Object.
- **Policy Objects with Securable Objects** Lists Active Roles Policy Objects together with the directory objects that are affected by each Policy Object. A directory object included in this report may have a Policy Object linked to the object itself (direct

policy effect) or to a container that holds the object (inherited policy effect). For each directory object that is affected by a given Policy Object, the report identifies the object's canonical name, type and description, and indicates whether the policy effect is direct or inherited. You can filter the list by Policy Object name, policy type, and by directory object name and type.

- **Securable Objects (with inheritance)** Lists directory objects that are affected by Active Roles Policy Objects. For each directory object, identifies the Policy Objects that are linked to the directory object itself (direct policy effect) or to a container that holds the directory object (inherited policy effect). For each Policy Object that affects a given directory object, the report lists the Policy Object's name, path, description and policy entries, and indicates whether the policy effect is direct or inherited. You can filter the list by directory object name and type, Policy Object name and type, and by policy entry name.

Policy Compliance/

- **Objects violating Policy Rules** Lists directory objects and their properties that are not in compliance with policies determined by Active Roles Policy Objects. For each directory object, identifies the object's name, parent container, type and description, and indicates what properties violate policy rules and what Policy Objects define the policy rules that are violated.
- **Violated Policy Rules** Lists Active Roles Policy Objects whose policy rules are violated by certain directory objects. For each Policy Object, identifies the policies defined in that Policy Object, and, for every single policy, provides information about directory objects and their properties which are not in compliance with that policy.

12

Management History

- Understanding Management History
- Management History configuration
- Viewing change history
- Examining user activity

Understanding Management History

The Management History feature provides information on who did what and when it was done with regard to the Active Directory management tasks performed using Active Roles.

This feature gives you a clear log documenting the changes that have been made to a given object, such as a user or group object. The log includes entries detailing actions performed, success or failure of the actions, as well as which attributes were changed.

By using the Management History feature, you can examine:

- **Change History** Information on changes that were made to directory data via Active Roles.
- **User Activity** Information on management actions that were performed by a given user.

IMPORTANT:

- The reports produced by the Change History or User Activity command include information only about the changes that were made using a certain group of Administration Services (those Services that share a common database). As the Active Roles console or Web Interface automatically selects the Service to connect to, you may encounter different reports for the same target object or user account during different connection sessions.
- Active Roles uses the Management History storage to hold approval, temporal group membership, and deprovisioning tasks. Without synchronizing information between Management History storages, such a task created by one of the Administration Services may not be present on other Administration Services. As a result, behavior of the Active Roles console or Web Interface varies depending on the chosen Administration Service.

Both Change History and User Activity use the same source of information—the Management History log, also referred to as the Change Tracking log. The configuration settings of the Change Tracking log are discusses later in this chapter (see Management History configuration).

Active Roles also includes reports to examine management history by collecting and analyzing event log records (see Active Roles Reporting earlier in this document). However, the process of retrieving and consolidating records from the event log may be time-consuming and inefficient.

For more information on the impact of change on the Management History database, see the *Impact on management history data* topic in *Active Roles What's New Guide*.

Considerations and best practices

The Management History feature is designed to help promptly investigate what changes were recently made to directory data, as well as when it was done and by whom. As such, this feature is not intended for data change auditing nor is it intended to explore large volumes of data changes that occurred during a long period of time. For this reason, in addition to the Management History feature, Active Roles provides a suite of reports for change tracking and auditing, which is part of the Active Roles Report Pack. Each of these options: Management History and Report Pack, has its own advantages and limitations. Follow the recommendations in this section to choose the one that best suits your needs.

You can use the Management History feature to examine changes that were made to directory data via Active Roles. The feature is designed to help you answer the following typical questions:

- Who made the most recent changes to a given user or group object?
- Who modified a given user or group object during the last X days?
- What changes were made to a given user object last night (yesterday, the day before)?
- Have any planned modifications of a given user or group object actually been performed?
- What objects did a given delegated administrator modify during the last X days?

You can instantly access Management History whenever you need to quickly investigate or troubleshoot a problem that results from inappropriate modifications of directory data.

Management History includes a dedicated repository to store information about data changes, referred to as the Change Tracking log, and GUI to retrieve and display information from that repository. No additional actions, such as collecting or consolidating information, are required to build Management History results.

However, the advantages of the Management History feature also entail some limitations. Before you use the Management History feature, consider the following recommended best practices and limitations of using this feature.

The main factor to consider is the size of the Change Tracking log. To ensure real-time update of the log on all Administration Services, the log is normally stored in the Active Roles configuration database. This imposes some limitations on the log size.

By default, the Change Tracking log is configured to store information about changes that occurred within last 30 days. If you increase this setting, do it carefully; otherwise, you may encounter the following problems:

- Excessive increase in the log size significantly increases the time required to build and display Change History and User Activity results.
- As the log size grows, so does the size of the configuration database. This considerably increases the time required to back up and restore the database, and causes high network traffic replicating the database when you join an additional Administration Service to Active Roles replication.
- The GUI is not suitable to represent large volumes of Management History results in a manageable fashion. Since there is no filtering or paging capabilities, it may be difficult to sort through the results.

To address these limitations, Active Roles gives you a different means for change auditing, change-tracking reports, included with the Active Roles Report Pack. These reports are designed to help answer the following questions:

- What management tasks were performed on a given object within a certain period of time?
- What management tasks were performed on a given object during the object's entire life time?
- When was a certain attribute of a given object modified?

Change-tracking reports are based on data collected from event logs. A separate log is stored on each computer running the Administration Service, and each log only contains events generated by one Administration Service. Therefore, to use reports, the events from all event logs need to be consolidated to form a complete audit trail.

The process of consolidating events, referred to as the data collection process, is performed by a separate Active Roles component—Collector. With the Collector wizard, you can configure and execute data collection jobs, and schedule them to run on a regular basis.

The main limitation of change-tracking reports is the fact that the information needs to be collected and consolidated in a separate database before you can build the reports. The data collection process exhibits the following disadvantages:

- Collecting data may be a very lengthy operation and the database size may grow unacceptable when collecting all events that occurred within a long period of time in a large environment.
- Collecting data is impossible over slow WAN links. This limitation is inherent to the Active Roles component intended to collect data for reporting.

Management History configuration

The configuration of Management History includes the following elements:

- **Change-tracking Policy** Builds the data pertinent to history of changes made to directory objects, and specifies what changes are to be included in the reports on change history and user activity.
- **Change Tracking Log Configuration** Specifies how many change requests are to be stored in the log.
- **Replication of Management History Data** Specifies whether to synchronize Management History data between Administration Services that use different databases.

Change-tracking policy

The behavior of the Management History feature is defined by the policy held in the build-in Policy Object called **Built-in Policy - Change Tracking**. The policy determines the object types and properties for which to gather the management history information.

To view or modify the policy, display the **Properties** dialog box for the **Built-in Policy - Change Tracking** Policy Object (located in container **Configuration/Policies/Administration/Builtin**), go to the **Policies** tab, select the policy, and click **View/Edit**. This displays the **Policy Properties** dialog box. The **Object Types and Properties** in that dialog box lists the object types and properties included in Management History. Each entry in the list includes the following information:

- **Object Type** If an object of this type is modified via Active Roles, information about that action is recorded in the Change Tracking log on condition that the modification affects a property specified in the **Properties** column.
- **Properties** Information about changes to these properties is recorded in the Change Tracking log.

You can manage the list on the tab by using the buttons beneath the list:

- **Add** Displays the dialog box where you can select the object type and properties you want to include in Management History. You have an option to either select individual properties or select all properties.
- **Remove** Deletes the selected entries from the list.
- **View/Edit** Displays the dialog box where you can view or modify the properties for the selected list entry.

Change Tracking log configuration

One more configuration setting for Management History determines the size of the Change Tracking log. The log stores information about requests to change directory data, one record per request. Each record includes information about the changes to a certain object that were made in accordance with a certain change request.

You can configure the maximum number of records by managing properties of the **Change Tracking Log Configuration** object, located in the **Configuration/Server Configuration** container.

On the **Log Settings** tab in the **Properties** dialog box for that object, you can select one of the following options:

- **All requests that occurred during last** <*number*> **days** Information about change requests is written to the log so that new requests replace those that are older than the specified number of days.
- **This total number of most recent requests** The log stores not more than the specified number of change requests. When the limit is reached, each new request to make changes to directory data replaces the oldest request in the log.
- **This number of most recent requests per object** For every object, the log stores at most the specified number of change requests. When the limit is reached for a certain object, each new request to make changes to the object replaces the oldest request related to that object. The total number of requests depends on the number of objects that are modified via Active Roles.

By default, the Change Tracking log is configured to store information about requests that occurred within last 30 days. Information about change requests is written to the log so that new requests replace those that are older than 30 days. If you increase this number, do it carefully. Increasing this number significantly increases the size of the log. If you are planning to change this setting, you should first review the Considerations and best practices section earlier in this chapter.

> NOTE: The Change Tracking log is used as the source of information on both Change History and User Activity. The volume of requests held in the log equally determines the Change History retention time and the User Activity retention time.

On the **Log Record Size** tab, you can choose from the options that allow you to reduce the size of the Change Tracking log by logging detailed information about a limited number of change requests, having only basic information about the other change requests logged and thus included in the reports. If the log record of a given change request contains detailed information, then the report on that request provides information about all changes made, along with all policies and workflows performed, by Active Roles when processing the request. Otherwise, the report provides information only about the changes to the object properties made in accordance with the request. Although storing only basic log records results in fewer details in the reports, doing so may considerably decrease the size of the Management History database. The following options are available:

- **All requests** The Change Tracking log contains detailed information about all requests stored in the log.

- **Requests that occurred during last** <*number*> **days** Detailed information about requests is written to the log so that new requests with detailed information replace those that are older than the specified number of days.
- **This number of most recent requests** The log stores not more than the specified number of requests containing detailed information. When the limit is reached, each new request with detailed information replaces the oldest request in the log.
- **Don't log detailed information about any requests** The Change Tracking log contains only basic information about all requests stored in the log.

Replication of Management History data

NOTE: Active Roles does not support replication on Azure SQL databases.

In Active Roles version 7.4 and later, the Management History data is stored in the Active Roles Management history database. So, if you have Active Roles replication configured as described in the Configuring replication section later in this document, the Management History data is replicated between Administration Services along with the configuration data. Given a large volume of the Management History data, this may cause considerable network traffic.

You can turn off replication of Management History data so as to reduce network traffic. However, doing so causes each database server to maintain a separate Management History data store. The result is that you can use Management History to examine the changes that were made only through the Administration Services that use the same database as the Administration Service you are connected to.

To sum up, the implications of turning off replication of Management History data are as follows:

- The reports produced by the **Change History** or **User Activity** command include information only about the changes that were made using a certain group of Administration Services (those Services that share a common database).

  As the Active Roles console or Web Interface automatically selects the Service to connect to, you may encounter different reports for the same target object or user account during different connection sessions.

- The features of Active Roles such as Approval Workflow, Temporal Group Memberships, and Undo Deprovisioning may not work as expected. Some operations that rely on those features may not be processed or displayed in a consistent way by client interfaces connected to different Administration Services.

  Active Roles uses the Management History storage to hold approval, temporal group membership, and deprovisioning tasks. Without synchronizing information between Management History storages, such a task created by one of the Administration Services may not be present on other Administration Services. As a result, behavior of the Active Roles console or Web Interface varies depending on the chosen Administration Service.

Turning off replication of Management History data has no effect on replication of the other data pertinent to the configuration of Active Roles. Only the Management History-related portion of the configuration database is excluded from Active Roles replication.

The instructions on how to turn off replication of Management History data depend upon whether Active Roles replication is already configured.

Replication is not yet configured

When initially configuring Active Roles replication, you can ensure that the Management History data will not participate in Active Roles replication by assigning the Publisher role as follows (for definitions of the replication roles, see Configuring replication later in this document):

1. With the Active Roles console, connect to the Administration Service whose SQL Server you want to hold the Publisher role.
2. In the console tree, expand **Configuration | Server Configuration** and select the **Configuration Databases** container.

   > NOTE: **Replication Support** column is added under configuration databases container to indicate the replication support.
   >
   > If the value of this column is **Supported**, it indicates that the replication is allowed for the database. If the value of this column is **Unsupported** value indicates that the database does not allow replication.

3. In the details pane, right-click the database, and click **Promote**.
4. Wait while the console performs the Promote operation.
5. In the console tree, under **Server Configuration**, select the **Management History Databases** container.
6. In the details pane, right-click the database, and click **Demote**.
7. Wait while the console completes the Demote operation.

Then, you can configure Active Roles replication by using the Active Roles console as described in the Configuring replication section later in this document: Use the **Add Replication Partner** command on the database in the **Configuration Databases** container to add Subscribers to the Publisher you have configured.

Replication is already configured

This section outlines the instructions on how to turn off replication of Management History data in case that Active Roles replication is already configured as described in the Configuring replication section later in this document. You need to first delete all Subscribers for Management History data, and then demote the Publisher for Management History data. This only stops replication of Management History data, leaving the other replication functions intact.

To turn off replication of Management History data

1. With the Active Roles console, connect to the Administration Service whose SQL Server holds the Publisher role.
2. In the console tree, expand **Configuration | Server Configuration**, and select the **Management History Databases** container.
3. Use the **Delete** command on each of the Subscriber databases to delete all Subscribers in the **Management History Databases** container.
4. Right-click the Publisher database, and click **Demote**.
5. Wait while the console completes the Demote operation.

Re-configuring replication of Management History data

With replication of Management History data turned off, it is still possible to have multiple Administration Services maintain the same Change History log by configuring them to use the same database. Note that the Administration Service version 6.x allows you to install multiple Services with the option to connect to a single configuration database. Thus, you can install the first Service in your environment, having the Setup program create a database. Then, you can install one more Service, having the Setup program configure the new Service to use the same database as the existing Service.

However, if different Administration Services in your environment use different database servers, you may need to re-configure replication of Management History data in order to take full advantage of the Management History feature. You can do so by managing objects in the **Management History Databases** container as follows.

To re-configure replication of Management History data

1. With the Active Roles console, connect to the Administration Service whose SQL Server holds the Publisher role for configuration data.
2. In the console tree, expand **Configuration | Server Configuration**, and select the **Management History Databases** container.
3. In the details pane, right-click the database, and click **Promote**.
4. Wait while the console performs the Promote operation.
5. Use the **Add Replication Partner** command on the Publisher database in the **Management History Databases** container to add Subscribers for Management History data.

The **Add Replication Partner** command starts the wizard that is similar to that discussed in the Adding members to a replication group section later in this document. The only difference is that the list of Administration Services whose database servers can be designated as Subscribers for Management History data is limited to those Services that share the configuration data hosted on the Publisher you have selected.

Centralized Management History storage

With the default replication settings in Active Roles, the Management History data is synchronized between replication partners, along with the Configuration data. Given a large volume of Management History data, this behavior may result in high network traffic and may cause performance degradation of Active Roles in certain scenarios, such as when adding a new partner to the Active Roles replication group. Here you can find instruction on how to eliminate replication of Management History data by implementing a common storage of that data for all replication partners.

Synchronization of the Management History data can be removed from the Active Roles replication process by implementing a common storage of that data for all replication partners. The common storage ensures the consolidation of the portions of Management History data that are generated by different Administration Services, while eliminating the need to synchronize that data between multiple storages.

By default, Active Roles allows you to implement a centralized, common storage for the Management History data. In this way, all the Administration Services that share common configuration use the same Management History storage - the Management History database you created.

Importing data to the new Management History database

You may need to populate the newly created Management History database with your existing Management History data, so that the data remains available to the Active Roles user interfaces after you have configured the Administration Service to use the new Management History database. You can do this by using Active Roles Configuration Center on the computer running the Administration Service.

IMPORTANT: The reports produced by the Change History or User Activity command include information only about the changes made using a certain group of Administration Services that share a common database from the connected management history database. If the Change History data is not imported from the previously available database, the data is not displayed in the new Management History database. For more information on the implications of not importing the Change History data from the available database, see *Impact on management history data* in the *Active Roles What's New* guide.

To import Management History data

1. In the **Configuration Center** main window, under **Administration Service**, click **Manage Settings**.

   Start the Configuration Center by selecting **Active Roles7.5 Configuration Center** on the **Apps** page or **Start** menu, depending on the version of your Windows operating system.

2. On the **Administration Service** page, click **Import Management History** to open the Import Management History wizard.
3. On the **Source database** page, specify the database from which you want to import the management history data (source database):
   a. **Database Type**: Select the required database type from the drop-down (on premises or Azure SQL).
   b. **Database Server name**: Enter the name of the SQL Server instance that hosts the source database.
   c. **Database**: Enter the name of the source database.
4. Under **Connect using**, select the authentication option:
   - If your Windows logon account has sufficient rights to write data to the destination database, click **Windows authentication**.
   - If you have a SQL Server login with sufficient rights, click **SQL Server authentication** and enter the login name and password.
   - If you have an Azure AD login with sufficient rights, click **Azure Active Directory authentication** and enter the login name and password.

   NOTE: Azure databases can be connected using SQL Server authentication and Azure AD authentication. Windows authentication is applicable only for on-premises databases.

   NOTE: Azure AD authentication currently does not support Multi-Factor Authentication (MFA).
5. Click **Next**.

   The **Destination database** page identifies the database of the Administration Service to which you are going to import data (destination database), and allows you to select the authentication option.
6. Under **Connect using**, select the authentication option:
   - If your Windows logon account has sufficient rights to write data to the destination database, click **Windows authentication**.
   - If you have a SQL Server login with sufficient rights, click **SQL Server authentication** and enter the login name and password.
   - If you have an Azure AD login with sufficient rights, click **Azure Active Directory authentication** and enter the login name and password.

   NOTE: Azure databases can be connected using SQL Server authentication and Azure AD authentication. Windows authentication is applicable only for on-premises databases.

   NOTE: Azure AD authentication currently does not support Multi-Factor Authentication (MFA).
7. Click **Next**.

8. On the **Records to Import** page, specify whether you want to import all data records or only a certain range of the data records.

   NOTE: The data for unfinished temporal group membership is imported when the management history data is imported for a selected date range.

   You can choose not to import all the data records as importing a large volume of data can take hours. Later, you can import additional data by choosing a different range of data records. During subsequent import sessions, the wizard only imports the data records that were not imported earlier.

9. Click **Next** and follow the instructions in the wizard to complete the import operation.

The Import Management History wizard merges the Management History data found in an existing Active Roles database with the data stored in the Management History database. The wizard only adds new data, keeping intact any data that already exists in the Management History database. You may import Management History data at any convenient time after you have configured the Administration Service to use the new Management History database, without being afraid of losing any data.

Viewing change history

The Change History log can be accessed from the Active Roles console, allowing you to quickly examine what changes were made to a given user or group, as well as when it was done and by whom. For example, if someone reset a user's password via Active Roles, you might use change history to see when and by whom the password was reset.

To examine changes made to a given object, such as a user or group object, right-click it in the Active Roles console and click **Change History**. By default, the **Change History** window only displays basic options. You can display more choices by clicking the plus sign (+) in the top-left corner, next to the first column heading.

In the **Change History** window, you can find the following information:

- **Name** The name of the object for which you are examining change history.
- **Requested** The date and time that the changes were requested.
- **Requested by** The user account that requested the changes.
- **Completed** The date and time that the changes were applied.
- **Properties** The properties of the object that were changed, including information about the changed property values.
- **Status** Indicates whether the requested changes are applied (status COMPLETED) or waiting for approval (status PENDING).

The **Change History** window also includes the following areas:

- **Properties changed during this operation** Information about the object property values that were changed (old values), new values assigned to the properties, and the user account that was used to make the changes.

- **Workflow activities and policy actions** Detailed information about all policies and workflows performed by Active Roles when processing the change request.
- **Operation details** Additional information on when and by whom the changes were requested.

The **Workflow activities and policy actions** area displays a report of the policy actions and workflow activity actions. The report organizes the action results into sections, each containing report items specific to a single policy or activity. You can expand the area by clicking its title. To expand a section, click the title of the section. For certain items, the report provides the option to further expand the view and display additional information. The **List** option displays a list of items, such as user or group properties, affected by the policy or activity. By clicking the **Details** option, you can examine the policy or activity action result in more detail.

The following topics list the possible sections and report items in the **Workflow activities and policy actions** area. Each section in the report describes results of the action performed by a certain workflow activity or policy. The report items within the section inform about success or failure of the policy or activity action. In the event of a failure, the report item includes an error description.

Not all the listed sections and items must necessarily be present in a report. An actual report only includes the sections corresponding to the workflow activities and policies that Active Roles performed when processing the operation request.

The following topics elaborate on the report sections and report items you encounter in the **Workflow activities and policy actions** area:

- Workflow activity report sections
- Policy report items
- Active Roles internal policy report items

Workflow activity report sections

In a Change History report, the report sections specific to workflow activities list all activities that Active Roles executed when processing a given operation request. For each activity, from the respective report section you can determine whether the activity was completed successfully or returned an error. In case of error, the report section provides an error description. For activities requesting changes to directory data (for example, activities that create new objects or modify existing objects), you can examine the requested changes in detail by clicking the Operation ID number in the report section.

This topic lists the contents of the activity report sections you may encounter in a Change History report. Each report section has a header that identifies the name of the activity; the target object of the activity (the object, such as a user, group or computer that the activity is applied to or acts upon); the time that the activity was executed; and the name of the workflow containing that activity. If the activity encountered an error, then the text in the header of the activity report section is red. You can expand the report section by clicking the header to view the body of the report section. The contents of the body varies depending on the type of the activity. In case of an error condition, the body displays an error description.

The remainder of this topic covers the contents of the report section body for each activity type in situations where no errors have occurred.

"Approval" activity

The report section specific to an approval activity provides information about the approval task created by that activity, and varies depending on the state of the approval task. Normally, the activity does not create an approval task if the operation that is subject to approval was requested by an Active Roles administrator or an approver. In this case, the section body displays a message indicating that the activity is bypassed. Otherwise, the contents of the report section body is as follows.

Task status: Pending

The following information is displayed if the task is waiting for approver action.

- **Approval task details**
  - Task ID: <number>
  - Title: <title of the approval task>
  - Status: Pending
  - Requested: <date and time that the task was created>
  - Requested by: <name that identifies who requested the operation>

Task status: Completed

The following information is displayed if the approver allowed the requested operation.

- **Properties changed by approver**
  - Property <property of the operation target object set or changed by the approver>
  - Changed to <value of the property supplied by the approver>
- **Approval task details**
  - Task ID: <number>
  - Title: <title of the approval task>
  - Status: Completed
  - Requested: <date and time that the task was created>
  - Requested by: <name that identifies who requested the operation>
  - Completed: <date and time that the task was completed>
  - Completed by: <name of the approver who performed the task>
  - Completion reason: <text supplied by the approver>
  - Approver action: <resolution the approver chose to allow the operation>

Task status: Rejected

The following information is displayed if the approver denied the requested operation.

- **Approval task details**
  - Task ID: <number>
  - Title: <title of the approval task>
  - Status: Rejected
  - Requested: <date and time that the task was created>
  - Requested by: <name that identifies who requested the operation>
  - Rejected: <date and time that the task was completed>
  - Rejected by: <name of the approver who performed the task>
  - Rejection reason: <text supplied by the approver>
  - Approver action: <resolution the approver chose to deny the operation>

Task status: Canceled

The following information is displayed if the approval task is canceled.

- **Approval task details**
  - Task ID: <number>
  - Title: <title of the approval task>
  - Status: Canceled
  - Requested: <date and time that the task was created>
  - Requested by: <name that identifies who requested the operation>
  - Canceled: <date and time that the task was canceled>
  - Canceled by: <identifies who canceled the task>
  - Cancellation reason: <indicates why the task was canceled>

Task status: Any

The following information is always displayed in addition to the approval task details.

- **Approval task settings**
  - Approvers: <list of names that identify who is authorized to approve the operation>
  - Possible actions of approver: <list of resolutions the approver may choose from>
  - Approver is requested to supply or change these properties: <list of property names>
  - Approver is allowed to change properties submitted for approval: <Yes | No>

“Script” activity

If the activity did not encounter any errors, the report section body displays the following message:

- Activity successfully performed the script '*name*'.

Otherwise, a message is displayed stating that the activity encountered an error. You can view an error description in the report section body.

“Stop/Break” activity

The report section body displays the notification message provided by the activity. You can set up a notification message when configuring a Stop/Break activity.

“Add Report Section” activity

The header and the body of the report section display text information provided by the activity. You can set up the header and the body of the report section when configuring an Add Report Section activity.

“Create” activity

The body of the report section identifies the object created by the activity, and provides the following information:

- The type of the object (such as user, group or computer)
- Name: <name of the object>
- Operation ID: <number>
- Requested: <date and time that the operation was requested>
- Status: <indicates whether the operation is complete or pending>

You can click the Operation ID number to examine in detail the operation of creating the object. This displays a change history report containing information about all workflow activities and policy actions that Active Roles performed during that operation.

“Update” activity

The body of the report section identifies the object changed by the activity, and provides the following information:

- The type of the object (such as user, group or computer)
- Name: <name of the object>
- Operation ID: <number>
- Requested: <date and time that the operation was requested>
- List of object properties changed by the activity
- For each property, the value set by the activity (new value) and the value the property had before it was changed by the activity (old value)
- Status: <indicates whether the operation is complete or pending>

You can click the Operation ID number to examine in detail the operation of changing the object. This displays a change history report containing information about all workflow activities and policy actions that Active Roles performed during that operation.

"Add to group" activity

For every group to which the activity added the activity target object, the body of the report section displays the following information:

- The name of the object
- The name of the group
- Operation ID: <number>
- Requested: <date and time that the operation was requested>
- Status: <indicates whether the operation is complete or pending>

You can click the Operation ID number to examine in detail the operation of adding the object to the group. This displays a change history report containing information about all workflow activities and policy actions that Active Roles performed during that operation.

"Remove from group" activity

For every group from which the activity removed the activity target object, the body of the report section displays the following information:

- The name of the object
- The name of the group
- Operation ID: <number>
- Requested: <date and time that the operation was requested>
- Status: <indicates whether the operation is complete or pending>

You can click the Operation ID number to examine in detail the operation of removing the object from the group. This displays a change history report containing information about all workflow activities and policy actions that Active Roles performed during that operation.

“Move” activity

The body of the report section identifies the object the activity moved to a certain container (activity target object), and provides the following information:

- The type of the object (such as user, group or computer)
- Name: <name of the object>
- Moved to: <identifies the move destination container>
- Operation ID: <number>
- Requested: <date and time that the operation was requested>
- Status: <indicates whether the operation is complete or pending>

You can click the Operation ID number to examine in detail the operation of moving the object. This displays a change history report containing information about all workflow activities and policy actions that Active Roles performed in during that operation.

“Deprovision” activity

The body of the report section identifies the object deprovisioned by the activity, and provides the following information:

- The type of the object (such as user or group)
- Name: <name of the object>
- Operation ID: <number>
- Requested: <date and time that the operation was requested>
- Status: <indicates whether the operation is complete or pending>

You can click the Operation ID number to examine in detail the operation of deprovisioning the object. This displays a change history report containing information about all workflow activities and policy actions that Active Roles performed during that operation.

“Undo deprovision” activity

The body of the report section identifies the object the activity restored from the deprovisioned state (activity target object), and provides the following information:

- The type of the object (such as user or group)
- Name: <name of the object>
- Operation ID: <number>
- Requested: <date and time that the operation was requested>
- Status: <indicates whether the operation is complete or pending>

You can click the Operation ID number to examine in detail the operation of restoring the deprovisioned object. This displays a change history report containing information about all workflow activities and policy actions that Active Roles performed during that operation.

"Delete" activity

The body of the report section identifies the object deleted by the activity (activity target object), and provides the following information:

- The type of the object (such as user or group)
- Name: <name of the object>
- Operation ID: <number>
- Requested: <date and time that the operation was requested>
- Status: <indicates whether the operation is complete or pending>

You can click the Operation ID number to examine in detail the operation of deleting the object. This displays a change history report containing information about all workflow activities and policy actions that Active Roles performed during that operation.

Policy report items

This topic lists the Change History report items specific to the polices that are applied by using Policy Objects in Active Roles. When executing a given policy, Active Roles adds a report section to describe the actions performed by that policy. The report section identifies the policy category and the Policy Object containing the policy, and informs about success or failure of the policy action.

The following tables list the possible report items, one table per section. The items in each section describe the results of the actions that were taken in accordance with the respective policy. Report items also inform about success or failure of the policy action. In the event of a failure, the report item includes an error description.

Not all the listed items must necessarily be present in a report. An actual report only includes the report items corresponding to the policies that Active Roles performed when processing the operation request.

> NOTE: This topic covers the Active Roles provisioning policies. The report sections specific to deprovisioning policies are listed in the Report on deprovisioning results and Report on results of undo deprovisioning topics earlier in this document.

Report section: Executing the 'User Logon Name Generation' policy

Table 65: User Logon Name Generation' policy

Report Item (Success)	Report Item (Failure)
The user logon name (pre-Windows 2000) is set to '*value*'.	*Not applicable*

Report section: Executing the 'E-mail Alias Generation' policy

Table 66: E-mail Alias Generation' policy

Report Item (Success)	Report Item (Failure)
The e-mail alias is set to '*alias*'.	*Not applicable*
Property 'Alias (mailNickName)' is removed from the operation request as no Exchange tasks were requested.	*Not applicable*

Report section: Executing the 'Exchange Mailbox AutoProvisioning' policy

Table 67: Exchange Mailbox AutoProvisioning' policy

Report Item (Success)	Report Item (Failure)
The mailbox database is set to '*database name*'.	*Not applicable*
The option to create the mailbox is selected by default.	*Not applicable*
The option to create the mailbox is not selected by default.	*Not applicable*
Changing the option to create the mailbox is allowed.	*Not applicable*
Changing the option to create the mailbox is not allowed.	*Not applicable*

Report section: Executing the 'Group Membership AutoProvisioning' policy

Table 68: Group Membership AutoProvisioning' policy

Report Item (Success)	Report Item (Failure)
The object is added to the following groups. • List: *Group names*	Unable to add the object to the following groups. • List: *Group names and error description*
The object is not added to the following groups as it is already a member of those groups. • List: *Group names*	*Not applicable*
The object is removed from the following groups. • List: *Group names*	Unable to remove the object from the following groups. • List: *Group names and error description*
The object is not removed from the following groups as it is not a member of those groups. • List: *Group names*	*Not applicable*

Report section: Executing the 'Home Folder AutoProvisioning' policy

Table 69: Home Folder AutoProvisioning' policy

Report Item (Success)	Report Item (Failure)
The home folder is mapped to letter '*letter*' and connected to path '*UNC path*' in Active Directory.	*Not applicable*
Home folder '*name*' is to be created on the file server.	*Not applicable*
Home folder '*name*' is created on the file server.	Unable to create home folder '{0}' on the file server. Details: *Error description*
User permissions on the home folder are set by copying permissions from the parent	Unable to set user permissions on home folder '*name*' on the file server.

Report Item (Success)	Report Item (Failure)
folder.	Details: *Error description*
The home folder user is set as the owner of the home folder.	Unable to set user permissions on home folder '*name*' on the file server. Details: *Error description*
User permission option 'Grant Change Access' is applied to the home folder.	Unable to set user permissions on home folder '*name*' on the file server. Details: *Error description*
User permission option 'Grant Full Access' is applied to the home folder.	Unable to set user permissions on home folder '*name*' on the file server. Details: *Error description*
Home folder '*name*' is to be renamed to '*new name*' on the file server.	*Not applicable*
Home folder '*name*' is renamed to '*new name*' on the file server.	Unable to rename home folder '*name*' to '*new name*' on the file server. Details: *Error description*
Home share '*name*' is to be created on the file server.	*Not applicable*
Home share '*name*' is created on the file server.	Unable to create home share '*name*' on the file server. Details: *Error description*
The user limit is set to allow no more than '*name*' users to connect to the home share at a time.	*Not applicable*
The user limit is set to allow the maximum number of users to connect to the home share at a time.	*Not applicable*

Report section: Executing the 'Property Generation and Validation' policy

Table 70: Property Generation and Validation' policy

Report Item (Success)	Report Item (Failure)
Property '*name*' is set to '*value*'.	*Not applicable*
Property '*name*' is removed (cleared).	*Not applicable*

Report section: Executing policy script '*name*'

Table 71: policy script 'name'

Report Item (Success)	Report Item (Failure)
Policy script '*name*' completed successfully.	• *Error message returned by the policy* Details: *Error description* *The default error message reads as follows:* • The 'Script Execution' policy encountered an error when running the script '*name*'. Details: *Error description*

Active Roles internal policy report items

The Active Roles internal policies are mainly intended to perform Exchange recipient management tasks, such as the task of creating a mailbox or the task of establishing an e-mail address for a group. These policies are triggered by Active Roles' internal logic, and cannot be configured by the administrator. Active Roles performs its internal policies as appropriate to the given operation request. For example, when processing a request to create a mailbox-enabled user account, Active Roles triggers an internal policy that carries out all the actions needed to create the user mailbox on the Exchange server.

The following tables list the possible report items, one table per report section. The items in each section describe the results of the actions that were taken in accord with the respective internal policy. Report items also inform about success or failure of the policy action. In the event of a failure, the report item includes an error description.

Not all the listed items must necessarily be present in a report. An actual report only includes the report items corresponding to the policies that Active Roles performed when processing the operation request.

Report section: Creating user mailbox

Table 72: Creating user mailbox

Report Item (Success)	Report Item (Failure)
User mailbox *'name'* is created.	Unable to create user mailbox *'name'*. Details: *Error description*
Legacy mailbox '*name*' is created.	Unable to create legacy mailbox '*name*'.

Report Item (Success)	Report Item (Failure)
	Details: *Error description*
Mailbox alias is set to '*alias*'.	*Not applicable*
Mailbox database is set to '*database name*'.	*Not applicable*
The following mailbox properties are set. List: *Property names and values*	Unable to set the following properties of the mailbox. List: *Property names and error description*

Report section: Creating linked mailbox

Table 73: Creating linked mailbox

Report Item (Success)	Report Item (Failure)
Linked mailbox *'name'* is created.	Unable to create linked mailbox *'name'*. Details: *Error description*
Legacy mailbox '*name*' is created.	Unable to create legacy mailbox '*name*'. Details: *Error description*
Mailbox alias is set to '*alias*'.	*Not applicable*
Mailbox database is set to '*database name*'.	*Not applicable*
The mailbox is linked to external account '*name*'.	*Not applicable*
The following mailbox properties are set. List: *Property names and values*	Unable to set the following properties of the mailbox. List: *Property names and error description*

Report section: Creating equipment mailbox

Table 74: Creating equipment mailbox

Report Item (Success)	Report Item (Failure)
Equipment mailbox *'name'* is created.	Unable to create equipment mailbox *'name'*. Details: *Error description*
Mailbox alias is set to '*alias*'.	*Not applicable*
Mailbox database is set to '*database name*'.	*Not applicable*

Report Item (Success)	Report Item (Failure)
The following mailbox properties are set. List: *Property names and values*	Unable to set the following properties of the mailbox. List: *Property names and error descriptions*

Report section: Creating room mailbox

Table 75: Creating room mailbox

Report Item (Success)	Report Item (Failure)
Room mailbox *'name'* is created.	Unable to create room mailbox *'name'*. Details: *Error description*
Mailbox alias is set to '*alias*'.	*Not applicable*
Mailbox database is set to '*database name*'.	*Not applicable*
The following mailbox properties are set. List: *Property names and values*	Unable to set the following properties of the mailbox. List: *Property names and error description*

Report section: Creating shared mailbox

Table 76: Creating shared mailbox

Report Item (Success)	Report Item (Failure)
Shared mailbox *'name'* is created.	Unable to create shared mailbox *'name'*. Details: *Error description*
Mailbox alias is set to '*alias*'.	*Not applicable*
Mailbox database is set to '*database name*'.	*Not applicable*
Shared mailbox is configured to allow the following users to use this mailbox. List: *User names*	*Not applicable*
The following mailbox properties are set. List: *Property names and values*	Unable to set the following properties of the mailbox. List: *Property names and error description*

Report section: Moving mailbox

Table 77: Moving mailbox

Report Item (Success)	Report Item (Failure)
The following items apply to mailbox move operation on Exchange 2013 or later	
The mailbox move request for mailbox '*name*' is created.	Unable to create the mailbox move request for mailbox '*name*'. Details: *Error description*
The mailbox is being moved from database '*database name*' to database '*database name*'.	*Not applicable*

Report section: Deleting mailbox

Table 78: Deleting mailbox

Report Item (Success)	Report Item (Failure)
Mailbox *'name'* is deleted.	*Not applicable*

Report section: Removing Exchange attributes

Table 79: Removing Exchange attributes

Report Item (Success)	Report Item (Failure)
The following Exchange attributes are removed from *'name'*. List: *Attribute names*	*Not applicable*

Report section: Enabling mailbox for Unified Messaging

Table 80: Enabling mailbox for Unified Messaging

Report Item (Success)	Report Item (Failure)
Mailbox *'name'* is enabled for Unified Messaging.	*Not applicable*

Report Item (Success)	Report Item (Failure)
The following Unified Messaging mailbox policy is assigned to the mailbox: '*policy name*'	*Not applicable*
The following Unified Messaging mailbox properties are set. List: *Property names and values*	*Not applicable*

Report section: Disabling Unified Messaging for mailbox

Table 81: Disabling Unified Messaging for mailbox

Report Item (Success)	Report Item (Failure)
Unified Messaging is disabled for mailbox *'name'*.	*Not applicable*

Report section: Resetting Unified Messaging PIN

Table 82: Resetting Unified Messaging PIN

Report Item (Success)	Report Item (Failure)
The Unified Messaging PIN is reset for mailbox *'name'*.	*Not applicable*

Report section: Establishing e-mail address for group

Table 83: Establishing e-mail address for group

Report Item (Success)	Report Item (Failure)
An e-mail address is established for group *'name'*. The group is now mail-enabled.	Unable to establish an e-mail address for group '*name*'. Details: *Error description*
E-mail alias is set to '*alias*'.	*Not applicable*
The following properties of the group are set. • List: *Property names and values*	*Not applicable*

Report section: Creating query-based distribution group

Table 84: Creating query-based distribution group

Report Item (Success)	Report Item (Failure)
Query-based Distribution Group *'name'* is created.	Unable to configure Query-based Distribution Group '*name*'. Details: *Error description*
E-mail alias is set to '*alias*'.	*Not applicable*
The following properties of the group are set. List: *Property names and values*	*Not applicable*

Report section: Establishing e-mail address for user

Table 85: Establishing e-mail address for user

Report Item (Success)	Report Item (Failure)
An e-mail address is established for user *'name'*. The user is now mail-enabled.	Unable to establish an e-mail address for user '*name*'. Details: *Error description*
E-mail alias is set to '*alias*'.	*Not applicable*
The following properties of the user account are set. List: *Property names and values*	*Not applicable*

Report section: Establishing e-mail address for contact

Table 86: Establishing e-mail address for contact

Report Item (Success)	Report Item (Failure)
An e-mail address is established for contact *'name'*. The contact is now mail-enabled.	Unable to establish an e-mail address for contact '*name*'. Details: *Error description*

Report Item (Success)	Report Item (Failure)
E-mail alias is set to '*alias*'.	*Not applicable*
The following properties of the contact are set. List: *Property names and values*	*Not applicable*

Report section: Deleting e-mail address for group

Table 87: Deleting e-mail address for group

Report Item (Success)	Report Item (Failure)
The e-mail address for group *'name'* is deleted. The group is no longer mail-enabled.	*Not applicable*

Report section: Deleting e-mail address for user

Table 88: Deleting e-mail address for user

Report Item (Success)	Report Item (Failure)
The e-mail address for user *'name'* is deleted. The user is no longer mail-enabled.	*Not applicable*

Report section: Deleting e-mail address for contact

Table 89: Deleting e-mail address for contact

Report Item (Success)	Report Item (Failure)
The e-mail address for contact *'name'* is deleted. The contact is no longer mail-enabled.	*Not applicable*

Report section: Converting user mailbox to linked mailbox

Table 90: Converting user mailbox to linked mailbox

Report Item (Success)	Report Item (Failure)
User mailbox *'name'* is converted to a linked mailbox.	*Not applicable*
The mailbox is linked to external account '*name*'.	*Not applicable*

Report section: Converting linked mailbox to user mailbox

Table 91: Converting linked mailbox to user mailbox

Report Item (Success)	Report Item (Failure)
Linked mailbox *'name'* is converted to a user mailbox.	*Not applicable*
The mailbox is un-linked from external account '*name*'. The external account can no longer access the mailbox.	*Not applicable*

Examining user activity

The Change Tracking log also allows you to examine the changes that a given user made to directory data, that is, the management activity of the user. The management activity retention time depends on the Change Tracking log configuration (see Change-tracking policy earlier in this chapter).

To see what changes were made by a given user, right-click the user object in the Active Roles console and click **User Activity**.

By default, the **User Activity** window only displays basic options. You can display more choices by clicking the plus sign (+) in the top-left corner, next to the first column heading.

In the **User Activity** window, you can find the following information:

- **Name** The name of the object for which you are examining change history.
- **Requested** The date and time that the changes were requested.
- **Completed** The date and time that the changes were applied.

- **Properties** The properties of the object that were changed, including information about the changed property values.
- **Status** Indicates whether the requested changes are applied (status COMPLETED) or waiting for approval (status PENDING).

The window also includes the same additional sections as the **Change History** window (see Viewing change history).

13

Entitlement Profile

- Understanding entitlement profile
- Entitlement profile configuration
- Viewing entitlement profile

Understanding entitlement profile

The entitlement profile is a list of entitlements, each of which represents authorization to access, use or manage a particular information resource. A resource could be a single object in the directory, such as a user, group, contact or computer object, or it could be a server-based resource, such as an Exchange mailbox, user home folder, Web application or network file share. In case of a server-based resource, entitlement normally takes the form of user attributes or stems from membership in a certain group. In case of a directory object, entitlement refers to the manager or owner rights on that object.

Active Roles provides the ability to view the entitlement profile of any given user, both in the Active Roles console and Web Interface. The entitlement profile is implemented as a configurable report that displays information about resources to which a given user is entitled. Configuration of the entitlement profile specifies what resources are to be listed and what information about each resource is to be displayed in the report. Active Roles provides effective controls to manage configuration of the entitlement profile.

A user's entitlement profile is essentially a list of information resources to which the user is entitled. The resource can be one of the following:

- A personal resource, such as the user's mailbox, home folder, account enabled for Office Communications Server, or Unix-enabled account.
- A shared, network-based resource, such as a Web application or network file share, that the user has permission to access.
- A managed resource, such as a group or distribution list, for which the user is responsible as the manager or owner.

The way in which a user gets entitled to a given resource depends upon the type of the resource:

- For a personal resource, entitlement takes the form of certain attributes of the user's account in the directory.
- For a shared resource, entitlement is granted by adding the user to a certain security group in Active Directory.
- For a managed resource, entitlement is granted by assigning the manager or owner role for a certain object in Active Directory.

The building of a user's entitlement profile is done by applying entitlement rules to the entitlement target objects specific to that user. If a given entitlement target object matches the entitlement rules for a particular resource, then the user is regarded as entitled to the resource and information about that resource appears in the entitlement profile. The entitlement target object can be one of the following:

- The user's account in Active Directory; this object is used to discover the personal resources to which the user is entitled.
- An Active Directory group of which the user is a member; this object is used to discover the shared resources to which the user is entitled.
- An Active Directory object for which the user is assigned as the manager or owner; this object is used to discover the managed resources to which the user is entitled.

Active Roles stores the entitlement rules in configuration objects called entitlement profile specifiers. These objects are essential to the process of building and presenting the entitlement profile.

About entitlement profile specifiers

In Active Roles, entitlement profile specifiers are configuration objects that govern the process of building and presenting the entitlement profile. Each specifier holds information about a single resource that allows Active Roles to determine whether a given user is entitled to the resource and, if the user appears to be entitled, what information about that resource to include in the user's entitlement profile.

An entitlement profile specifier holds the following information:

- **Entitlement Type** Specifies a way in which a user gets entitled to the resource.
- **Entitlement Rules** Provide a way to determine whether a given user is entitled to the resource.
- **Resource Display** Specifies how to represent the resource in the entitlement profile.

The following topics elaborate on each of these information blocks.

Entitlement type

The entitlement type setting is basically intended to determine the entitlement target object—the object to which Active Roles applies the entitlement rules when building the

entitlement profile. Entitlement types can be classified by how a user's entitlement to a resource is configured:

- **Personal resource entitlement** Configured by setting certain attribute of the user's account itself. In this case, the user's account plays the role of the entitlement target object.
- **Shared resource entitlement** Configured by adding the user to a certain security group. In this case, the group plays the role of the entitlement target object.
- **Managed resource entitlement** Configured by assigning the user to the manager or owner role for a certain object. In this case, the object managed or owned by the user plays the role of the entitlement target object.

The following table summarizes the types of entitlement.

Table 92: Types of entitlement

Type	Configuration	Target Object
Personal resource entitlement	The user's account has certain resource-specific attributes set in the directory.	The user's account
Shared resource entitlement	The user's account belongs to a certain security group in Active Directory.	The user's group
Managed resource entitlement	The user's account is specified as the primary owner (manager) or a secondary owner of a certain object in the directory.	The object managed or owned by the user

Entitlement rules

When building a user's entitlement profile, Active Roles uses a specifier's entitlement rules to tell whether the user is entitled to the resource represented by that specifier. The rules are evaluated against the entitlement target object. If the object matches the rules, then Active Roles regards the user as entitled to the resource, and adds information about the resource to the user's entitlement profile.

Entitlement rules can be classified by rule condition as follows:

- **Explicit exclusion** The rule condition is a list of directory objects. If the entitlement target object occurs in that list, it is regarded as not matching the rules.
- **Explicit inclusion** The rule condition is a list of directory objects. If the entitlement target object occurs in that list, it is regarded as matching the rules.
- **Filter-based exclusion** The rule condition is one or more filters each of which represents certain requirements on an object's location and properties. If the entitlement target object satisfies the requirements of at least one filter, then it is regarded as not matching the rules.
- **Filter-based inclusion** The rule condition is one or more filters each of which represents certain requirements on an object's location and properties. If the

entitlement target object satisfies the requirements of at least one filter, then it is regarded as matching the rules.

For details on how Active Roles applies entitlement rules, see About entitlement profile build process later in this document.

Resource display

For each resource that is to be included in the entitlement profile, Active Roles applies entitlement rules to single out the appropriate specifier and then it uses the resource display settings of that specifier to build the entitlement profile's section that displays information about the resource.

The resource display settings include the following:

- **Resource type icon** Graphics that helps distinguish the type of the resource in the entitlement profile.
- **Resource type name** Text string that identifies the type of the resource in the entitlement profile.
- **Resource naming attribute** Entitlement target object's attribute whose value is used to identify the resource in the entitlement profile.
- **Other resource-related attributes** List of the entitlement target object's attributes whose values are to be displayed in the entitlement profile.

The entitlement profile's section for a given resource is divided into two areas:

- **Heading** Displays the resource type icon, resource type name, and value of the resource naming attribute.
- **Details** Lists the names and values of the resource-related attributes.

The **Details** area can be customized by adding HTML code to a certain attribute of the user account for which the entitlement profile is being built. The LDAP display name of that attribute should be supplied in the edsaHTMLDetailsAttribute of the entitlement profile specifier. As a result, Active Roles renders that HTML code instead of displaying the attributes list in the Details area.

About entitlement profile build process

When requested to build a user's entitlement profile, Active Roles performs the following steps.

1. Prepare a list of the user's groups, that is, a list of the security groups to which the user belongs whether directly or because of group nesting.
2. Prepare a list of the user's managed objects, that is, a list of the directory objects for which the user is assigned as the primary owner (manager) or a secondary owner.

3. For each entitlement profile specifier of the personal resource entitlement type, evaluate the entitlement rules of that specifier against the user's account. If the user's account matches the entitlement rules, then add information about the resource to the entitlement profile, presenting the resource in accordance with the resource display settings found in the specifier.
4. For each of the user's groups, apply the entitlement profile specifiers of the shared resource entitlement type as follows:
    a. For each specifier, evaluate the entitlement rules of that specifier against the group.
    b. Once a specifier has been found such that the group matches its entitlement rules, then add information about the resource to the entitlement profile, presenting the resource in accordance with the resource display settings held in the specifier.
    c. If the group matches the entitlement rules of more than one specifier, apply the first specifier found and disregard the others.
5. For each of the user's managed objects, apply the entitlement profile specifiers of the managed resource entitlement type as follows:
    a. For each specifier, evaluate the entitlement rules of that specifier against the managed object.
    b. Once a specifier has been found such that the managed object matches its entitlement rules, then add information about the resource to the entitlement profile, presenting the resource in accordance with the resource display settings held in the specifier.
    c. If the managed object matches the entitlement rules of more than one specifier, apply the first specifier found and disregard the others.

Entitlement rules play a central part in the process of building the entitlement profile. It is the entitlement rules that determine whether Active Roles regards a given user as entitled to a given resource, and thus adds information about that resource to the user's entitlement profile. When evaluating entitlement rules against a particular object, Active Roles performs the following steps.

1. Apply the explicit exclusion rules. If the object is in the list of excluded objects, then disregard the remaining rules, and mark the object as not matching the rules. Otherwise, proceed to the next step.
2. Apply the explicit inclusion rules. If the object is in the list of included objects, then disregard the remaining rules, and mark the object as matching the rules. Otherwise, proceed to the next step.
3. Apply the filter-based exclusion rules. If the object satisfies the rule condition, then disregard the remaining rules, and mark the object as not matching the rules. Otherwise, proceed to the next step.
4. Apply the filter-based inclusion rules. If the object satisfies the rule condition, then mark the object as matching the rules.

It may occur that the entitlement target object matches the entitlement rules of more than one specifier. In this case, Active Roles needs to choose a single specifier from those matching the entitlement target object. This is accomplished as follows:

1. Examine the edsaPriority attribute of each specifier, and look for specifiers that have edsaPriority not set. If no such specifier found, then proceed to Step 3. If a single specifier found, then apply that specifier. Otherwise, proceed to Step 2.
2. Range the specifiers that have edsaPriority not set in ascending alphanumeric order by name, and apply the specifier that goes first. Do not perform Steps 3–4.
3. Choose the specifiers with the lowest edsaPriority value. If a single specifier has the lowest edsaPriority value, then apply that specifier. Otherwise, proceed to the next step.
4. Range the specifiers with the lowest edsaPriority value in ascending alphanumeric order by name, and apply the specifier that goes first.

Note that the specifiers that have edsaPriority not set take precedence over those for which edsaPriority is set.

Once Active Roles has identified a single specifier for entitlement to a given resource, it uses the resource display settings of the specifier to build a section of the entitlement profile that displays information about the resource. If multiple resources match a particular specifier, then the sections specific to those resources are grouped together in an expandable block, to prevent the entitlement profile display from cluttering.

Entitlement profile configuration

In Active Roles, entitlement profile specifiers provide the ability to store the definition of entitlement to a particular resource in a single object. entitlement profile specifiers determine the contents of the entitlement profile.

When building the entitlement profile of a given user, Active Roles uses the entitlement profile specifiers to determine what resources the user is entitled to, and what information about each resource is to be shown in the entitlement profile.

Active Roles comes with a collection of pre-defined specifiers, and allows administrators to create additional specifiers or change existing specifiers. You can use the following instructions to create or change entitlement profile specifiers:

- Creating entitlement profile specifiers
- Changing entitlement profile specifiers

For a list of pre-defined specifiers, see Pre-defined specifiers.

Creating entitlement profile specifiers

Active Roles stores entitlement profile specifiers in the **Entitlement Profile Specifiers** container. You can access that container by expanding the **Configuration/Server Configuration** branch in the Active Roles console tree.

To create an entitlement profile specifier

1. In the console tree, under **Configuration/Server Configuration/Entitlement Profile Specifiers**, right-click the container in which you want to create a new specifier, and select **New | Entitlement Profile Specifier**.

   For example, if you want to create a new specifier in the root container, right-click **Entitlement Profile Specifiers**.

2. In the New Object - Entitlement Profile Specifier wizard, type a name and, optionally, a description for the new specifier.

   The name and description are used to identify the specifier object in the Active Roles console.

3. Click **Next**.
4. Choose the desired type of entitlement:
   - Select the **User attributes** option if the fact that a given user is entitled to the resource stems from certain attribute settings of the user's account in Active Directory. For example, this is the type of entitlement to an Exchange mailbox or to a home folder.
   - Select the **Group membership** option if the fact that a given user is entitled to the resource stems from membership of the user in a certain security group.
   - Select the **Manager or owner role assignment** option if entitlement of a given user to the resource means that the user is designated as the manager (primary owner) or a secondary owner of a certain object.
5. Click **Next**.
6. Set up the **Entitlement rules** list.

   In this step, you define the criteria that are used to determine whether a given user is entitled to the resource. The entitlement rules take the form of conditions that the entitlement target object must meet in order for the user to be regarded as entitled to the resource, and thus for information about the resource to appear in the entitlement profile of that user.

   Active Roles evaluates the entitlement rules against the entitlement target object when building a user's entitlement profile. Depending on the entitlement type, the entitlement target object is:

- In case of the **User attributes** entitlement type, the user account of the user whose entitlement profile is being built. (This entitlement type is referred to as *personal resource entitlement*.)
- In case of the **Group membership** entitlement type, any single group to which the user belongs, whether directly or because of group nesting. (This entitlement type is referred to as *shared resource entitlement*.)

7. You can define entitlement rules based on object properties, such as whether the object has certain attributes set or whether the object is a security group. The conditions take the form of LDAP filter based search criteria. With the "Include" rule type, the user is regarded as entitled to the resource if the entitlement target object

meets the search criteria. With the "Exclude" rule type, the user is regarded as not entitled to the resource if the entitlement target object meets the search criteria.

8. In addition to filter-based rules, you can configure rules on a per-object basis, so as to include or exclude individual objects from entitlement assignment explicitly. If Active Roles encounters a rule to include the entitlement target object, it considers the user as entitled to the resource. If Active Roles encounters a rule to exclude the entitlement target object, then it considers the user as not entitled to the resource.
9. Active Roles evaluates the entitlement rules in the following order:
   a. Explicit exclusion
   b. Explicit inclusion
   c. Filter-based exclusion
   d. Filter-based inclusion

   Once the entitlement target object matches a rule of a particular type, the rule types that stand lower in this list are not applied. This means that exclusion rules take precedence over inclusion rules and explicit selection of objects takes precedence over filter-based rules.

   Initially, no entitlement rules are configured, which is treated as an inclusion-type condition that evaluates to TRUE for any object. As a result, entitlement to the resource is established regardless of the properties of the entitlement target object. You can add entitlement rules in order to categorize entitlements based on properties of entitlement target objects.

   To add an entitlement rule, click **Include** or **Exclude** depending on the rule type you want, and then use the **Configure Entitlement Rule** dialog box to specify your search criteria. You can specify search criteria the same way you do when using the **Find** dialog box. Then, do one of the following:

- To add a rule based on the search criteria you specified, click **Add Rule**.
- To select specific objects, click **Find Now**, select check boxes in the list of search results, and then click **Add Selection**.

7. Click **Next**.
8. View or change the icon that is used to distinguish the type of the resource in the entitlement profile:
   - View the icon in the area next to the **Change** button.
   - To choose a different icon, click **Change** and then select the desired image file.
   - To revert to the default icon, click **Use Default Icon**.
9. Type the name of the resource type to be displayed in the entitlement profile.
10. Click **Select** to choose the attribute of the entitlement target object whose value will be used to name the resource in the entitlement profile.

    The resource type icon, display name, and naming attribute are used to identify the resource in the entitlement profile. If the evaluation of the entitlement rules for a given user indicates that the user is entitled to the resource, then information about the resource appears as a separate section in the entitlement profile of that user. The heading of the section includes the resource type icon, the display name of the

resource type, and the value of the naming attribute retrieved from the entitlement target object.

11. Click **Next**.
12. Set up the list of the resource-related attributes that will be displayed in the entitlement profile:
    - Use the **Add** or **Remove** button to add or remove attributes from the list.
    - Click **Add Separator** to divide the attribute list into sections in the entitlement profile.
    - Use the **Up** and **Down** buttons to arrange the attribute list order.

    The attributes held in the list will be displayed in the entitlement profile, beneath the heading of the section that provides information about the resource. For each of the listed attributes, the section displays the name and the value of the attribute retrieved from the entitlement target object.
13. Click **Next**, and then click **Finish**.

Changing entitlement profile specifiers

You can change an existing entitlement profile specifier by changing the specifier's name and description, entitlement type and rules, resource display settings, and resource attributes list. The entitlement profile specifier objects are located under **Configuration/Server Configuration/Entitlement Profile Specifiers** in the Active Roles console.

The following table summarizes the changes you can make to an existing entitlement profile specifier object, assuming that you have found the object in the Active Roles console. You can also disable or delete a specifier using the **Disable** or **Delete** command on the **Action** menu. Active Roles disregards the disabled specifiers when building the entitlement profile. A disabled specifier can be re-enabled by using the **Enable** command that appears on the **Action** menu for disabled specifiers.

Table 93: Entitlement profile specifier object changes

To change	Do this	Commentary
Name	Right-click the object and click **Rename**.	The name is used to identify the object, and must be unique among the objects held in the same container.
Description	Right-click the object, click **Properties** and make the necessary changes on the **General** tab.	The description is intended to help Active Roles administrators identify the purpose and the function of the object.
Entitlement type	Right-click the object, click **Properties**, click the	The entitlement type specifies how the user is entitled to the resource. You

To change	Do this	Commentary
	tab, and then select the appropriate option.	can choose whether the user is entitled to the resource by means of: • **User attributes** Entitlement to a personal resource such as a mailbox or home folder, controlled by certain attributes of the user account. • **Group membership** Entitlement to a shared resource such as a Web application or a network file share via membership in a security group. • **Manager or owner role assignment** Entitlement to act as the manager (primary owner) or a secondary owner of a directory object such as a group, distribution list, or computer.
Entitlement rules	Right-click the object, click **Properties**, click the **Rules** tab, and then add, remove, or modify entitlement rules by using the buttons below the rules list.	The entitlement rules are used to determine whether a given user is entitled to the resource. The entitlement rules take the form of conditions that the entitlement target object must meet in order for the user to be regarded as entitled to the resource, and thus for information about the resource to appear in the entitlement profile of that user. To add or change an entitlement rule, click **Include** or **Exclude** depending on the rule type you want, or click **View/Edit**, and then use the **Configure Entitlement Rule** dialog box to specify rule conditions. You can do this the same way you use the **Find** dialog box to configure and run a search. Note that you can change only filter-based rules. If you select an explicit inclusion or exclusion rule the **View/Edit** button is unavailable. You can use the **Remove** button to remove a rule of any type. For more information, see Step 6 in

To change	Do this	Commentary
		Creating entitlement profile specifiers.
Resource display settings	Right-click the object, click **Properties**, click the **Display** tab, and then view or change the icon and display name of the resource type, and the resource naming attribute.	The resource type icon, display name, and naming attribute are used to identify the resource in the entitlement profile. If the evaluation of the entitlement rules for a given user indicates that the user is entitled to the resource, then information about the resource appears as a separate section in the entitlement profile of that user. The heading of the section includes the resource type icon, the display name of the resource type, and the value of the naming attribute retrieved from the entitlement target object.
Resource attributes list	Right-click the object, click **Properties**, click the **Attributes** tab, and then add, remove, or change the order of attributes by using the buttons below the attributes list.	The tab lists the attributes of the entitlement target object that will be displayed in the entitlement profile, beneath the heading of the section that provides information about the resource. For each of the listed attributes, the section displays the name and the value of the attribute retrieved from the entitlement target object.

Pre-defined specifiers

Active Roles comes with a collection of pre-defined specifiers that determine the default resource profile configuration. The pre-defined specifiers are located in the **Configuration/Server Configuration/Entitlement Profile Specifiers/Builtin** container, and can be administered using the Active Roles console. You can make changes to a pre-defined specifier (see Changing entitlement profile specifiers) or you can apply the **Disable** command for the specifier to have no effect. Note that pre-defined specifiers cannot be deleted.

The pre-defined specifiers have a lower priority than customer-created specifiers. This means the entitlement rules of customer-created specifiers are evaluated first, so that if a given entitlement target object matches the entitlement rules of both a pre-defined specifier and a customer-created specifier, the latter specifier is applied. The priority of specifiers is governed by the edsaPriority attribute setting (see About entitlement profile build process).

The following table provides information about the pre-defined specifiers. For each specifier, the table lists the specifier's name, description, entitlement type and rules, and resource display settings.

Table 94: Pre-defined specifiers

Name and Description	Type and Rules	Resource Display Settings
Name: Self - Exchange Mailbox *Description*: Specifies user entitlement to Exchange mailbox.	*Type*: Personal resource entitlement *Rules*: Entitlement target object is an Exchange mailbox enabled user account.	*Resource type name*: Exchange Mailbox *Resource naming attribute*: mail *Other resource-related attributes*: • mail • homeMDB • displayName
Name: Self - Home Folder *Description*: Specifies user entitlement to home folder.	*Type*: Personal resource entitlement *Rules*: Entitlement target object has the homeDirectory attribute set.	*Resource type name*: Home Folder *Resource naming attribute*: homeDirectory *Other resource-related attributes*: • homeDirectory • homeDrive
Name: Self - Unix Account *Description*: Specifies user entitlement to Unix-enabled account.	*Type*: Personal resource entitlement *Rules*: Entitlement target object has the uidNumber attribute set AND has a loginShell attribute value other than /bin/false.	*Resource type name*: Unix-enabled Account *Resource naming attribute*: userPrincipalName *Other resource-related attributes*: • userPrincipalName • uidNumber • gidNumber • unixHomeDirectory • loginShell
Name: Self - OCS Account *Description*: Specifies user entitlement to	*Type*: Personal resource entitlement *Rules*: Entitlement	*Resource type name*: Enabled for Office Communications Server *Resource naming attribute*: msRTCSIP-PrimaryUserAddress *Other resource-related attributes*:

Name and Description	Type and Rules	Resource Display Settings
Office Communications Server enabled account.	target object has the msRTCSIP-UserEnabled attribute set to TRUE.	• msRTCSIP-PrimaryUserAddress • edsva-OCS-Pool
Name: Membership - Member of Security Group *Description*: Specifies entitlement to a resource via membership in a security group.	*Type*: Shared resource entitlement *Rules*: Entitlement target object is a security group. This specifier has the lowest priority as per the edsaPriority attribute setting, so the entitlement rules of any other specifier of the shared resource entitlement type are evaluated prior to the rules of this specifier.	*Resource type name*: Member of Security Group *Resource naming attribute*: name *Other resource-related attributes*: • name • displayName • description • info • edsvaResourceURL • managedBy • edsvaPublished • edsvaApprovalByPrimaryOwnerRequired • edsvaParentCanonicalName
Name: Membership - Access to SharePoint Site *Description*: Specifies entitlement to a SharePoint site via membership in a certain security group.	*Type*: Shared resource entitlement *Rules*: Entitlement target object is a security group that has the edsva-SP-MirrorType attribute set.	*Resource type name*: Access to SharePoint Site *Resource naming attribute*: name *Other resource-related attributes*: • name • edsva-SP-SiteName • edsva-SP-SiteURL • managedBy • edsvaPublished • edsvaApprovalByPrimaryOwnerRequired • edsvaParentCanonicalName

Name and Description	Type and Rules	Resource Display Settings
Name: Managed By - Owner of Security Group *Description*: Specifies entitlement to the manager or owner role for a security group.	*Type*: Managed resource entitlement *Rules*: Entitlement target object is a security group.	*Resource type name*: Owner of Security Group *Resource naming attribute*: name *Other resource-related attributes*: • name • displayName • description • info • edsvaResourceURL • managedBy • edsvaPublished • edsvaApprovalByPrimaryOwnerRequired • edsvaParentCanonicalName
Name: Managed By - Owner of Distribution List *Description*: Specifies entitlement to the manager or owner role for a distribution group.	*Type*: Managed resource entitlement *Rules*: Entitlement target object is an Exchange mail enabled (distribution) group.	*Resource type name*: Owner of Distribution List *Resource naming attribute*: displayName *Other resource-related attributes*: • displayName • mail • description • info • managedBy • edsvaPublished • edsvaApprovalByPrimaryOwnerRequired • edsvaParentCanonicalName
Name: Managed By - Owner of Resource Exchange Mailbox *Description*: Specifies entitlement to the	*Type*: Managed resource entitlement *Rules*: Entitlement target object is a	*Resource type name*: Owner of Resource Exchange Mailbox *Resource naming attribute*: displayName *Other resource-related attributes*: • displayName

Name and Description	Type and Rules	Resource Display Settings
owner role for a room, equipment, or shared mailbox.	user account associated with a room, equipment or shared mailbox.	• edsva-MsExch-MailboxTypeDescription • mail • description • homeMDB • edsvaParentCanonicalName
Name: Managed By - Owner of Exchange Contact *Description*: Specifies entitlement to the owner role for an Exchange mail contact.	*Type*: Managed resource entitlement *Rules*: Entitlement target object is an Exchange mail contact.	*Resource type name*: Owner of Exchange Contact *Resource naming attribute*: displayName *Other resource-related attributes*: • displayName • givenName • sn • mail • telephoneNumber • company • edsvaParentCanonicalName
Name: Managed By - Owner of Computer *Description*: Specifies entitlement to the manager or owner role for a computer.	*Type*: Managed resource entitlement *Rules*: Entitlement target object is a computer account.	*Resource type name*: Owner of Computer *Resource naming attribute*: name *Other resource-related attributes*: • name • dNSHostName • description • operatingSystem • edsvaParentCanonicalName
Name: Managed By - Default *Description*: Default specifier for entitlement to the manager or	*Type*: Managed resource entitlement *Rules*: No rules specified, which	*Resource type name*: Owner of *<target object class display name>* *Resource naming attribute*: name *Other resource-related attributes*: • name

Name and Description	Type and Rules	Resource Display Settings
owner role.	means that any object is regarded as matching the entitlement rules of this specifier. This specifier has the lowest priority as per the edsaPriority attribute setting, so the entitlement rules of any other specifier of the managed resource entitlement type are evaluated prior to the rules of this specifier.	• description • edsvaParentCanonicalName

Viewing entitlement profile

A user's entitlement profile can be accessed from the Active Roles console or Web Interface, allowing you to quickly examine resources to which the user is entitled:

- In the console, right-click the user and click **Entitlement Profile**. Alternatively, click the **Entitlement Profile** button on the **Managed Resources** tab in the **Properties** dialog box for the user account.
- In the Web Interface, click the user, and then choose **Entitlement Profile** from the list of commands.

This opens the **Entitlement Profile** page that lists the user's resources grouped in expandable blocks by resource type. Each block may be a section that represents a single resource, or it may comprise a number of sections each of which represents a single resource. The grouping of sections occurs for resources of the same type. For example, the security groups in which the user has membership may be grouped together in a single block, with each group being represented by a separate section.

Initially, each block or section displays only a heading that includes the following items:

- **Resource icon** Graphics that helps distinguish the type of the resource.
- **Resource type** Text string that identifies the type of the resource.

- **Resource name** Text string that identifies the name of the resource, or indicates that the block comprises multiple resource-specific sections.

To view resource details, click the heading of a block or section.

Out of the box, Active Roles is configured so that a user's entitlement profile displays the user's entitlements to the resources listed in the table that follows. Active Roles administrators can configure the entitlement profile to display information about additional resources. If a user is not entitled to any resources of a particular type, then the user's entitlement profile does not contain the sections specific to that resource type. For example, if a user does not have an Exchange mailbox, then the user's entitlement profile does not contain information about the user's mailbox.

Table 95: User resources

Resource Type	Resource Name	Resource Details
Exchange Mailbox	E-mail address of mailbox	• E-mail address • Mailbox store or database location • Mailbox user's display name
Home Folder	Path and name of home folder	• Path and name of home folder • Drive letter assigned to home folder
Unix-enabled Account	User principal name	• User principal name • Unix user ID (UID) • Unix primary group ID (GID) • Unix home directory • Unix login shell
Enabled for Office Communications Server	Live communications address	• Live communications address • Office Communications server or pool
Member of Security Group	Group name	• Group name • Group display name • Group description • Group notes • Resource address (URL) • Group's "Managed By" setting • Group's "Is Published" setting • Group's "Approval by Primary Owner Required" setting • Group location ("In Folder" setting)
Access to	Group name	• Group name

Resource Type	Resource Name	Resource Details
SharePoint Site		• SharePoint site name • SharePoint site address (URL) • Group's "Managed By" setting • Group's "Is Published" setting • Group's "Approval by Primary Owner Required" setting • Group location (group's "In Folder" setting)
Owner of Security Group	Group name	• Group name • Group display name • Group description • Group notes • Resource address (URL) • Group's "Managed By" setting • Group's "Is Published" setting • Group's "Approval by Primary Owner Required" setting • Group location ("In Folder" setting)
Owner of Distribution List	Group display name	• Group display name • Group e-mail address • Group description • Group notes • Group's "Managed By" setting • Group's "Is Published" setting • Group's "Approval by Primary Owner Required" setting • Group location ("In Folder" setting)
Owner of Resource Exchange Mailbox	Mailbox display name	• Mailbox display name • Mailbox type • E-mail address • Mailbox store or database location • Mailbox description • Mailbox location ("In Folder" setting)

Resource Type	Resource Name	Resource Details
Owner of Exchange Contact	Contact display name	• Display name • First name • Last name • E-mail address • Telephone number • Company • Location ("In Folder" setting)
Owner of Computer	Computer name	• Computer name • Computer DNS name • Computer description • Operating system • Location ("In Folder" setting)
Owner of Resource (default)	Managed object's name	• Managed object's name • Managed object's description • Managed object's location ("In Folder" setting)

Authorizing access to entitlement profile

By default, permission to view the entitlement profile is given to Active Roles Admin, the administrative account or group specified during Active Roles installation. Other users or groups can also be permitted to view the entitlement profile. A dedicated Access Template is provided for this purpose so that you can allow the use of the **Entitlement Profile** command by designated users or user groups.

To permit particular users or groups to view the entitlement profile of the users held in a certain container, such as an organizational unit or a Managed Unit, apply the Access Template as follows.

To authorize access to the entitlement profile

1. In the Active Roles console, right-click the container and click **Delegate Control** to display the **Active Roles Security** window.
2. In the **Active Roles Security** window, click **Add** to start the Delegation of Control wizard.
3. In the wizard, click **Next**.
4. On the **Users or Groups** page, click **Add**, and then select the desired users or groups.
5. Click **Next**.

6. On the **Access Templates** page, expand the **Active Directory | Advanced** folder, and then select the check box next to **Users - View Entitlement Profile (Extended Right)**.
7. Click **Next** and follow the instructions in the wizard, accepting the default settings.

After you complete these steps, the users and groups you selected in Step 4 are authorized to view the entitlement profile of the users held in the container you selected in Step 1, as well as in any sub-container of that container.

14

Recycle Bin

- Understanding Recycle Bin
- Finding and listing deleted objects
- Restoring a deleted object
- Delegating operations on deleted objects
- Applying policy or workflow rules

Understanding Recycle Bin

Active Roles builds on Active Directory Recycle Bin, a feature of Active Directory Domain Services introduced in Microsoft Windows Server 2008 R2, to facilitate the restoration of deleted objects. When Recycle Bin is enabled, Active Roles makes it easy to undo accidental deletions, reducing the time, costs, and user impact associated with the recovery of deleted objects in Active Directory.

The use of Active Roles in conjunction with Active Directory Recycle Bin helps minimize directory service downtime caused by accidental deletions of directory data. Recycle Bin provides the ability to restore deleted objects without using backups or restarting domain controllers and a user interface featured by Active Roles expedites locating and recovering deleted objects from Recycle Bin. Flexible and powerful mechanisms provided by Active Roles for administrative tasks delegation, enforcement of policy rules and approvals, and change tracking ensure tight control of the recovery processes.

To undo deletions, Active Roles relies on the ability of Active Directory Recycle Bin to preserve all attributes, including the link-valued attributes, of the deleted objects. This makes it possible to restore deleted objects to the same state they were in immediately before deletion. For example, restored user accounts regain all group memberships that they had at the time of deletion.

Active Roles can be used to restore deleted objects in any managed domain that has Active Directory Recycle Bin enabled. This requires the forest functional level of Windows Server 2012, so all the forest domain controllers must be running Windows Server 2012. In a forest that meets these requirements, an administrator can enable Recycle Bin by using the Active Directory module for Windows PowerShell in Windows Server 2012. For more

information about Active Directory Recycle Bin, see What's New in AD DS: Active Directory Recycle Bin (http://go.microsoft.com/fwlink/?LinkId=141392).

Finding and listing deleted objects

Once Active Directory Recycle Bin is enabled in a managed domain, Active Roles provides access to the **Deleted Objects** container that holds the deleted objects from that domain. In the Active Roles console tree, the container appears at the same level as the domain itself, under the **Active Directory** node. If multiple managed domains have Active Directory Recycle Bin enabled, then a separate container is displayed for each domain. To tell one container from another, the name of the container includes the domain name (for example, **MyDomain.MyCompany.com - Deleted Objects**).

Search pages in the Active Roles console facilitate finding deleted objects, enabling the use of very specific queries based on any object properties. It is also possible to examine and search a list of deleted objects that were in a particular Organizational Unit or Managed Unit at the time of deletion.

Searching the Deleted Objects container

The Active Roles console offers the **Deleted Objects** search category in the **Find** dialog box, which is intended to perform a search in the **Deleted Objects** container of any managed domain where Active Directory Recycle Bin is enabled.

To search the Deleted Objects container

1. In the console tree, right-click the **Active Directory** and click **Find**.
2. In the **Find** list, click **Deleted Objects**.
3. Do any of the following:
   - In **Name** or **Description**, type the name or description, or part of the name or description, of the object to find.

     When searching by name, Active Roles uses ambiguous name resolution (ANR) to find objects with not only name but also some other properties matching the string you type in the **Name** box. The properties used for ANR include name, first name, last name, display name, and logon name.
   - Click the button next to the **Deleted from** box and select the object that was the parent of the deleted object you want to find.

     By using the **Deleted from** search option you can find child objects that were deleted from a particular container object.
   - Use the **Advanced** tab to build a query based on other properties of the deleted object to find. For instructions, see Steps for using advanced search options and Steps for building a custom search earlier in this document.

4. Click **Find Now** to start the search.

When the search completes, the **Find** dialog box displays a list of deleted objects that match the search criteria.

If you double-click an object in the list of search results, the property pages for that object are displayed. If you right-click an object, the shortcut menu displays all the actions you can perform on that object.

Searching for objects deleted from a certain OU or MU

To view and search a list of objects that were deleted from a particular Organizational Unit (OU) or Managed Unit (MU), you can use the **View or Restore Deleted Objects** command. The command opens a dialog box that lists the deleted objects that were direct children of the corresponding OU or MU at the time of deletion. The **View or Restore Deleted Objects** dialog box can be used to search for deleted objects whose name matches a specific search string. It provides flexible matching by using support for ambiguous name resolution (ANR).

To search for objects deleted from a particular OU or MU

1. Right-click the OU or MU and click **View or Restore Deleted Objects**.
2. In **Look for**, type the search string that you want to use.
3. Click **Find Now** to start the search.

When the search completes, the list in the dialog box is limited to the deleted objects whose name, first name, last name, display name, logon name, or any other property used for ANR begins with the specified search string. To clear the search results and display all the deleted objects, click the **Clear Search** button.

> NOTE: The **View or Restore Deleted Objects** command is also available on domain and container objects, which allows you to find deleted objects that were direct children of a particular domain or container at the time of deletion.

Restoring a deleted object

For restoring deleted objects you can use the **Restore** command that is available from:

- The **View or Restore Deleted Objects** dialog box
- A list of search results prepared using the **Deleted Objects** search category in the **Find** dialog box
- A list of objects held in the **Deleted Objects** container, which is displayed in the details pane when you select the **Deleted Objects** container in the console tree

In the Active Roles console the command can be found on the shortcut menu, which appears when you right-click a deleted object.

To restore a deleted object

1. In the **View or Restore Deleted Objects** dialog box, click the deleted object and then click the **Restore** button.

   OR

   In a list of search results prepared using the **Deleted Objects** search category, or in a list of objects held in the **Deleted Objects** container, right-click the deleted object and click **Restore**.

2. Review and, if necessary, change the settings in the **Restore Object** dialog box, and then click **OK** to start the restore process.

The **Restore Object** dialog box prompts you to choose whether deleted child objects (descendants) of the deleted object should also be restored. The **Restore child objects** check box is selected by default, which ensures that the **Restore** command applied on a deleted container object restores the entire contents of the container.

To clarify, consider an example in which an administrator accidentally deletes an Organizational Unit (OU) called Sales_Department that contains a number of user accounts for sales persons along with another OU called Admins that, in turn, contains a user account for an administrative assistant. When applying the **Restore** command on the Sales_Department OU, with the option to restore child objects, Active Roles performs the following sequence of steps:

1. Restore the Sales_Department OU
2. Restore all the deleted user accounts that were direct children of the Sales_Department OU
3. Restore the Admins OU in the Sales_Department OU
4. Restore all the deleted user accounts that were direct children of the Admins OU

If you clear the **Restore child objects** check box, Active Roles performs only the first step, so the restored Sales_Department OU is empty.

> IMPORTANT: When restoring a deleted object, ensure that its parent object is not deleted. You can identify the parent object by viewing properties of the deleted object: the canonical name of the parent object, preceded with the "deleted from:" label, is displayed beneath the name of the deleted object on the **General** tab in the **Properties** dialog box. If the parent object is deleted, you need to restore it prior to restoring its children because deleted objects must be restored to a live parent.

Delegating operations on deleted objects

The delegation model based on the Active Roles Access Templates is fully applicable to the administrative tasks specific to deleted objects. A new Access Template called **All Objects - View or Restore Deleted Objects** makes it easy to delegate the following operations to selected users:

- Viewing deleted Active Directory objects
- Restoring a deleted Active Directory object

When applied to the **Deleted Objects** container, the Access Template gives the delegated users the right to view and restore any deleted object. With the Access Template applied to an Organizational Unit (OU) or a Managed Unit (MU), the delegated users are given the right to view and restore only those deleted objects that were located in that OU or MU at the time of deletion.

To delegate the operation of restoring deleted objects

1. In the console tree, select **Configuration | Access Templates | Active Directory**.
2. In the details pane, right-click **All Objects - View or Restore Deleted Objects** and click **Links**.
3. In the **Links** dialog box, click **Add**.
4. Click **Next** on the Welcome page in the Delegation of Control Wizard.
5. On the **Objects** page in the wizard, click **Add**; then, select the container in which you want to delegate the operation of restoring deleted objects:
   - To delegate restoring only those deleted objects that were in a particular Organizational Unit (OU) or Managed Unit (MU) at the time of deletion, select that OU or MU.
   - To delegate restoring any deleted objects in a particular managed domain, select either the object representing that domain or the **Deleted Objects** container for that domain.
   - To delegate restoring any deleted objects in any managed domain, select the **Active Directory** container.
6. Follow the instructions on the wizard pages to complete the Delegation of Control Wizard.
7. Click **OK** to close the **Links** dialog box.

Although it is possible to delegate the operation of restoring deleted objects in any managed domain, Organizational Unit or Managed Unit, a deleted object cannot be restored by using Active Roles unless the object belongs to a managed domain that has Active Directory Recycle Bin enabled. For instructions on how to enable Recycle Bin, see "Active Directory Recycle Bin Step-by-Step Guide" in Microsoft's documentation for Windows Server 2008 R2.

Applying policy or workflow rules

In addition to the delegation of administrative tasks, Active Roles provides the ability to establish policy-based control over the process of restoring deleted objects. Policy rules can be used to perform additional verifications or custom script-based actions upon the restoration of deleted objects. Workflow rules can be applied so as to require approval for the restore operation or notify of the restore operation completion via e-mail.

The policy or workflow rules to control the process of restoring or otherwise managing deleted objects can be defined on:

- The **Active Directory** node in the Active Roles console - The rules defined in this way affect all deleted objects in any managed domain that has Recycle Bin enabled.
- The node representing a domain or the **Deleted Objects** container for that domain in the Active Roles console - These rules affect all deleted objects in that domain only.
- An Organizational Unit (OU) or Managed Unit (MU) that held the object at the time of deletion. Although the deleted object no longer belongs to that OU or MU, Active Roles considers the former location of the object so that the rules applied on that location continue to affect the object after the deletion.

For example, an administrator could create a workflow to require approval for the restoration of any user account that was deleted from a certain Organizational Unit (OU). The workflow definition would contain an appropriate approval rule, and have that OU specified as the target container in the workflow start conditions.

Policy rules are defined by configuring and applying Policy Objects.

To apply a Policy Object to the Deleted Objects container

1. Right-click the **Deleted Objects** container and click **Enforce Policy**.
2. In the **Active Roles Policy** dialog box, click **Add**.
3. In the **Select Policy Objects** dialog box, select the check box next to the Policy Object you want to apply, and then click **OK**.
4. Click **OK** to close the **Active Roles Policy** dialog box.

For more information and instructions on configuring and applying Policy Objects, see Applying Policy Objects earlier in this document.

Workflow rules are defined by configuring workflow definitions and specifying the appropriate workflow start conditions.

To apply a workflow to the Deleted Objects container

1. In the console tree, select the workflow you want to apply.

   To select a workflow, expand **Configuration | Policies | Workflow**, and then click the workflow definition object under **Workflow** in the console tree.

2. In the details pane, click the **Workflow options and start conditions** button above the workflow process diagram, and then click **Configure**

   This displays the **Workflow Options and Start Conditions** page.

3. Click **Select Operation**, select the **Restore** option, and then click **Finish**.

   This will cause the workflow to start upon a request to restore a deleted object of the type specified.

4. Click **Add** under **Initiator Conditions**.
5. On the **Add Initiator Condition** page, click **Browse** and select the **Deleted Objects** container.

   You could select a container other than **Deleted Objects**. If you do so, the workflow starts only upon the restoration of an object that was deleted from the container you have selected.

6. Complete configuring workflow start conditions.

For more information about workflows, see the Workflows chapter earlier in this document.

15

AD LDS Data Management

- Registering an AD LDS instance
- Managing AD LDS objects
- Configuring Active Roles for AD LDS

Registering an AD LDS instance

Active Roles provides the ability to manage directory data in Microsoft Active Directory Lightweight Directory Services (AD LDS), an independent mode of Active Directory formerly known as Active Directory Application Mode (ADAM).

A running copy of the AD LDS directory service is referred to as a service instance (or, simply, *instance*). To use Active Roles for managing data hosted by the AD LDS directory service, you first need to register the instance that holds the data to manage.

Once an instance has been registered, the Active Roles client interfaces—Console, Web Interface and ADSI Provider—can be used to access, view and modify directory data in the application and configuration partitions found on the instance. The instances registered with Active Roles are referred to as *managed AD LDS instances*.

To register an AD LDS instance with Active Roles

1. Open the Active Roles console.
2. In the console tree, expand **Configuration | Server Configuration**, right-click **Managed AD LDS Instances (ADAM)**, and select **New | Managed AD LDS Instance (ADAM)** to start the Add Managed AD LDS Instance Wizard.
3. Follow the instructions on the wizard pages.
4. On the **AD LDS Instance to Register** page, specify the server name and port number of the AD LDS instance you want to register with Active Roles.

   In **Server**, type the fully qualified DNS name (for example, server.company.com) of the computer on which the instance is running. In **LDAP port**, type the number of the Lightweight Directory Access Protocol (LDAP) communication port in use by the instance (the default communication port for LDAP is 389). You can also click **Select** to locate and select the AD LDS instance you want to register.

5. On the **Active Roles Credentials** page, specify the credentials that Active Roles will use to access the instance.

   If you want each Administration Service to connect to the instance in the security context of its own service account, click **The service account information the Administration Service uses to log on**. With this option, different Administration Services may have different levels of access to the instance (the service account of one Service may have administrative rights on the instance while the service account of another Service may not). As a result, switching from one Administration Service to another may cause Active Roles to lose access to the instance.

   If you want each Administration Service to connect to the instance using the same user account, click The **Windows user account information specified below** and type in the user name, password, and domain name. In this way, you specify a so-called *override account*, thereby causing the access rights of Active Roles on the instance to be determined by the access rights of that user account (rather than by those of the service account of the Administration Service).

6. On the completion page, click **Finish** to start the registration process.

The override account you specify in Step 5 must, at a minimum, be a member of the following groups in the AD LDS instance:

- **Instances** (CN=Instances,CN=Roles) in the configuration partition
- **Readers** (CN=Readers,CN=Roles) in the configuration partition and in each application partition

If you choose not to specify an override account, you should add the service account to these groups.

To allow Active Roles full access to the AD LDS instance, add the service account or, if specified, the override account to the following group:

- **Administrators** (CN=Administrators,CN=Roles) in the configuration partition

If you add the account to the **Administrators** group, you don't need to add it to the **Instances** or **Readers** group.

Use the AD LDS ADSI Edit console to add the account to the appropriate groups prior to registering the instance with Active Roles.

After an AD LDS instance is registered, you can view or change its registration settings by using the **Properties** command on the object representing that instance in the **Managed AD LDS Instances (ADAM)** container. Thus, you can make changes to the choices that were made in Step 5 of the above procedure.

If you no longer want to manage an AD LDS instance with Active Roles, you can unregister the instance by using the **Delete** command on the object representing that instance in the **Managed AD LDS Instances (ADAM)** container. Unregistering an instance only removes the registration information from Active Roles, without making any changes to the directory data within that instance.

Managing AD LDS objects

The application and configuration partitions found in the managed AD LDS instances are grouped together in a top-level container, thus making it easy to locate the AD LDS data. Each partition is represented by a separate container (node) so you can browse the partition tree the same way you do for an Active Directory domain.

The Active Roles console supports a wide range of administrative operations on AD LDS users, groups and other objects, so you can create, view, modify, and delete directory objects, such as users, groups and organizational units, in the managed AD LDS instances the same way you do for directory objects in Active Directory domains.

To browse the directory tree and manage AD LDS objects

1. In the console tree under the console tree root, double-click the **AD LDS (ADAM)** container.
2. In the console tree under **AD LDS (ADAM)**, double-click a directory partition object to view its top-level containers.
3. In the console tree, double-click a top-level container to view the next level of objects in that container.
4. Do one of the following:
    - To move down a directory tree branch, continue double-clicking the next lowest container level in the console tree.
    - To administer a directory object at the current directory level, right-click the directory object in the details pane and use commands on the shortcut menu.

In the **AD LDS (ADAM)** container, each directory partition is identified by a label that is composed of the name of the partition, the DNS name of the computer running the AD LDS instance that hosts the partition, and the number of the LDAP port in use by the instance.

Normally, the console only displays the application directory partitions. To view the configuration partition, switch into Raw view mode: select **View | Mode**, click **Raw Mode**, and then click **OK**.

You can only perform the data management tasks to which you are assigned in Active Roles. Thus, you are only shown the commands you are authorized to use and the objects you are authorized to view or modify.

In addition to access control, Active Roles provides for policy enforcement on directory data. Policies may restrict access to certain portions of directory objects, causing data entry to be limited with choice constraints, auto-generating data without the ability to modify the data, or requiring data entry. The console provides a visual indication of the data entries that are controlled by policies: the labels of such data entries are underlined on the dialog boxes so that the user can examine policy constraints by clicking a label.

Adding an AD LDS user to the directory

To enable the creation of users in AD LDS, the administrator should first import the optional definitions of user object classes that are provided with AD LDS. These definitions are provided in importable .ldf files (ms-User.ldf, ms-InetOrgPerson.ldf, ms-UserProxy.ldf), which can be found on the computer running the AD LDS instance. Alternatively, the software designers can extend the AD LDS schema with their custom definitions of AD LDS user object classes. Details on how to extend the AD LDS schema can be found in Microsoft's documentation that comes with AD LDS.

To add an AD LDS user to the directory

1. In the console tree, under **AD LDS (ADAM)**, right-click the container to which you want to add the user, and then select **New | User** to start the wizard that will help you perform the user creation task.
2. Follow the instructions on the wizard pages to set values for user properties.
3. If you want to set values for additional properties (those for which the wizard pages do not provide data entries), click **Edit Attributes** on the completion page of the wizard.
4. After setting any additional properties for the new user, click **Finish** on the completion page of the wizard.

By default, an AD LDS user is enabled when the user is created. However, if you assign a new AD LDS user an inappropriate password or leave the password blank, the newly created AD LDS user account may be disabled. Thus, an AD LDS instance running on Windows Server 2003 automatically enforces any local or domain password policies that exist. If you create a new AD LDS user, and if you assign a password to that user that does not meet the requirements of the password policy that is in effect, the newly created user account will be disabled. Before you can enable the user account, you must set a password for it that meets the password policy restrictions. The instructions on how to set the password for an AD LDS user and how to enable an AD LDS user are given later in this section.

Adding an AD LDS group to the directory

AD LDS provides default groups, which reside in the **Roles** container of each directory partition in AD LDS. You can create additional AD LDS groups as necessary. New groups can be created in any container.

To add an AD LDS group to the directory

1. In the console tree, under **AD LDS (ADAM)**, right-click the container to which you want to add the group, and then select **New** | **Group** to start the wizard that will help you perform the group creation task.
2. Follow the instructions on the wizard pages to set values for group properties.

3. If you want to set values for additional properties (those for which the wizard pages do not provide data entries), click **Edit Attributes** on the completion page of the wizard.
4. After setting any additional properties for the new group, click **Finish** on the completion page of the wizard.

You can add both AD LDS users and Windows users to the AD LDS groups that you create. For instructions, see the sub-section that follows.

Adding or removing members from an AD LDS group

When adding members to an AD LDS group, you can add security principals that reside in AD LDS instances or in Active Directory domains. Examples of security principals are AD LDS users, and Active Directory domain users and groups.

1. To add or remove members to or from an AD LDS group
2. In the console tree, under **AD LDS (ADAM)**, locate and select the container that holds the group.
3. In the details pane, right-click the group, and click **Properties**.
4. On the **Members** tab in the **Properties** dialog box, click **Add**.
5. Use the **Select Objects** dialog box to locate and select the security principals that you want to add to the group. When finished, click **OK**.
6. On the **Members** tab, select the group members that you want to remove from the group, and then click **Remove**.
7. After making the changes that you want to the group, click **OK** to close the **Properties** dialog box.

When using the Select Objects dialog box to locate a security principal, you first need to specify the AD LDS directory partition or Active Directory domain in which the security principal resides: click **Browse** and select the appropriate partition or domain.

It is only possible to select security principals that reside in managed AD LDS instances or Active Directory domains; that is, you can select security principals from only the instances and domains that are registered with Active Roles.

Disabling or enabling an AD LDS user account

You can disable the account of an AD LDS user in order to prevent the user from logging on to the AD LDS instance with that account.

To disable or enable an AD LDS user account

1. In the console tree, under **AD LDS (ADAM)**, locate and select the container that holds the user account.
2. In the details pane, right-click the user account, and do one of the following to change the status of the account:
   - If the user account is enabled, click **Disable Account**.
   - If the user account is disabled, click **Enable Account**.

If the AD LDS user whose account you want to disable is currently logged on to the AD LDS instance, that user must log off for the new setting to take effect.

Normally, an AD LDS user is enabled when the user is created. However, if the password of a new AD LDS user does not meet the requirements of the password policy that is in effect, the newly created user account will be disabled. Before you can enable the user account, you must set a password for it that meets the password policy restrictions. For instructions, see the sub-section that follows.

Setting or modifying the password of an AD LDS user

Each AD LDS security principal, such as an AD LDS user, must be assigned an account and password, which AD LDS uses for authentication. You can use the Active Roles console to set or modify the password of an AD LDS user.

To set or modify the password of an AD LDS user

1. In the console tree, under **AD LDS (ADAM)**, locate and select the container that holds the user account of the AD LDS user for whom you want to set or modify the password.
2. In the details pane, right-click the user account, and then click **Reset Password**.
3. In the **Reset Password** dialog box, type a password for the user in **New password**, and retype the password in **Confirm password**, or click the button next to **New password** to generate a password.
4. Click **OK** to close the **Reset Password** dialog box.

The AD LDS user for whom you set or modify the password must use the new password the next time that the user logs on to AD LDS.

By default, an AD LDS instance running on Windows Server 2003 or later automatically enforces any local or domain password policies that exist. If you set a password for an AD LDS user that does not meet the requirements of the password policy that is in effect, Active Roles returns an error.

Adding an organizational unit to the directory

To keep your AD LDS users and groups organized, you may want to place users and groups in organizational units (OUs). In AD LDS, as well as in Active Directory or other Lightweight Directory Access Protocol (LDAP)-based directories, OUs are the most commonly used method for keeping users and groups organized. To create an organizational unit in AD LDS, you can use the Active Roles console as follows.

To add an organizational unit to the directory

1. In the console tree under **AD LDS (ADAM)**, right-click the container to which you want to add the OU, and select **New | Organizational Unit**.
2. Type a name for the new OU, click **Next**, and then click **Finish**.

By default, OUs can only be added under OU (ou=), country/region (c=), organization (o=) or domain-DNS (dc=) object classes. For example, you can add an OU to o=Company,c=US but not to cn=Application,o=Company,c=US. However, the schema definition of the OU object class can be modified to allow other superiors.

You can create new AD LDS users and groups in an AD LDS organizational unit by using the **New | User** or **New | Group** command on that organizational unit, as discussed earlier in this section.

You can move an existing AD LDS user or group to an organizational unit by using the **Move** command on that user or group in the Active Roles console, or by using the drag-and-drop feature of the console.

Adding an AD LDS proxy object (user proxy)

AD LDS proxy objects are used in special cases where an application can perform a simple LDAP bind to AD LDS but the application still needs to associate the AD LDS user with a security principal (user account) in Active Directory. A process through which AD LDS can accept a bind request from an application and redirect this bind request to Active Directory, based on the contents of a proxy object, is referred to as *bind redirection*.

Bind redirection occurs when a bind to AD LDS is attempted using a proxy object (user proxy) - an object in AD LDS that represents a user account in Active Directory. Each proxy object in AD LDS contains the security identifier (SID) of a user in Active Directory. When an application attempts to bind to a proxy object, AD LDS takes the SID that is stored in the proxy object, together with the password that is supplied at bind time, and presents the SID and the password to Active Directory for authentication.

A proxy object in AD LDS represents an Active Directory user account, and it can be augmented to store additional data related to that user account that is specific to the application. Through bind redirection, applications can take advantage of the identity store of Active Directory, while retaining the flexibility of using AD LDS as an application data store.

To add a proxy object to AD LDS

1. In the console tree, expand the **AD LDS (ADAM)** container.
2. In the console tree, under **AD LDS (ADAM)**, expand the directory partition to which you want to add a proxy object and locate the container to which you want to add the proxy object.
3. In the console tree, right-click the container to which you want to add the proxy object, and select **New** | **Proxy Object** to start the wizard that will help you create a proxy object.
4. Specify a name for the proxy object; then, click **Next**.
5. Click **Select** and choose the Active Directory domain user account you want to be represented by the proxy object; then, click **Next**.
6. If you want to set values for additional properties (those for which the wizard pages do not provide data entries), click **Edit Attributes** on the completion page of the wizard.
7. After setting any additional properties for the new object, click **Finish** on the completion page of the wizard.

You can examine an existing proxy object by using the **Properties** command on that object. The **Properties** dialog box allows you to view the user account that is represented by the proxy object. However, due to a limitation of AD LDS, this setting cannot be changed on an existing proxy object. You can select an Active Directory domain user account only at the time that the proxy object is created. After a proxy object is created, this setting cannot be modified.

When creating a proxy object, you can select a user account from any domain that is registered with Active Roles, provided that the domain is trusted by the computer on which the AD LDS instance is running.

A proxy object for a domain user cannot be created in an AD LDS directory partition that already contains a foreign principal object (FPO) or a proxy object for that same domain user.

For a given user account in Active Directory, you can view a list of proxy objects that represent the user account in AD LDS: In the **Properties** dialog box for the user account, go to the **Object** tab and click **AD LDS Proxy Objects**.

Configuring Active Roles for AD LDS

The Active Roles configuration-related tasks specific to AD LDS data management include the following:

- **Deploying rule-based administrative views** You can configure Managed Units in Active Roles to represent virtual collections of directory objects, from AD LDS, Active Directory or both, for distribution of administrative responsibilities and enforcement of business rules and policies.

- **Implementing role-based delegation** You can apply Active Roles Access Templates to delegate control of AD LDS data the same way as you do for the directory data held in Active Directory domains.
- **Policy-based control and auto-provisioning of directory data** You can apply Active Roles Policy Objects to establish policy-based control and perform auto-provisioning of AD LDS data the same way as you do for the directory data held in Active Directory domains.

This section elaborates on each of these tasks.

Configuring Managed Units to include AD LDS objects

By using the Active Roles console, you can configure Managed Units in Active Roles to represent virtual collections of directory objects, from AD LDS, Active Directory or both, for the distribution of administrative responsibilities and enforcement of business rules. By enabling Managed Units to include directory objects from any location, be it AD LDS or Active Directory, Active Roles provides the ability to implement role-based delegation and policy based administrative control of directory data where appropriate, without regard to directory boundaries.

You can use the following instructions to configure an existing Managed Unit so that it holds AD LDS objects such as AD LDS users, groups, or organizational units. For detailed instructions on how to create and administer Managed Units, see the Rule-based Administrative Views section earlier in this document.

To configure an existing Managed Unit to include AD LDS objects returned from a query

1. Right-click the Managed Unit and click **Properties**.
2. On the **Membership Rules** tab, click **Add**.
3. In the **Membership Rule Type** dialog box, click **Include by Query**, and then click **OK**.
4. Use the **Create Membership Rule** dialog box to set up the query:
    a. In the **Find** list, click **Custom Search**.
    b. Click **Browse** next to the **In** box.
    c. In the **Browse for Container** dialog box, expand the **AD LDS (ADAM)** container, expand the AD LDS directory partition containing the objects you want the query to return, and select the container that holds those objects. Then, click **OK**.
    d. Click the **Field** button, and select the type of the objects that you want the query to return and the object property that you want to query.

e. In **Condition**, click the condition for your query, and then, in **Value**, type a property value, in order for your query to return the objects that have the object property matching the condition-value pair you have specified.

f. Click the **Add** button to add this query condition to the query.

g. Optionally, repeat steps d) through f), to further define your query by adding more conditions. If you want the query to return the objects that meet all of the conditions specified, click **AND**. If you want the query to return the objects that meet any of the conditions specified, click **OR**.

h. Optionally, click **Preview Rule** to display a list of objects that your query returns. Note that the query results may vary depending on the current state of data in the directory. The Managed Unit will automatically re-apply the query whenever changes to directory data occur, in order to ensure that the membership list of the Managed Unit is current and correct.

i. Click the **Add Rule** button.

5. Click **OK** to close the **Properties** dialog box for the Managed Unit.

You can also configure membership rules of categories other than "Include by Query" in order to include or exclude AD LDS objects from a Managed Unit. To do so, select the appropriate category in the **Membership Rule Type** dialog box. Further steps for configuring a membership rule are all about using either the **Create Membership Rule** dialog box to set up a certain query or the Select Objects dialog box to locate and select a certain object.

Viewing or setting permissions on AD LDS objects

By using the Active Roles console, you can apply Active Roles Access Templates to delegate control of AD LDS data the same way as you do for the directory data held in Active Directory domains. By applying Access Templates to users or groups (Trustees) on AD LDS objects and containers, you can give the Trustees the appropriate level of access to directory data held in AD LDS, thus authorizing them to perform a precisely defined set of activities related to AD LDS data management.

Active Roles provides a rich suite of pre-configured Access Templates to facilitate delegation of AD LDS data management tasks. For a list of the AD LDS-specific Access Templates, refer to the Access Templates Available out of the Box document, which is part of the Active Roles documentation set. You can find those Access Templates in the **Configuration/Access Templates/AD LDS (ADAM)** container, in the Active Roles console.

You can use the following instructions to examine which Access Templates are applied to a given AD LDS object, such as an AD LDS user, group, organizational unit, container, or entire directory partition, and to add or remove Access Templates in order to change the level of access the Trustees have to that object.

For detailed instructions on how to create, configure and apply Access Templates, see the Role-based Administration section earlier in this document.

To view or modify the list of Access Templates on an AD LDS object

1. In the console tree, under **AD LDS (ADAM)**, locate and select the container that holds the object on which you want to view or modify the list of Access Templates.
2. In the details pane, right-click the object, and click **Properties**.
3. On the **Administration** tab in the **Properties** dialog box, click **Security**.
4. In the **Active Roles Security** dialog box, view the list of Access Templates that are applied to the AD LDS object, or modify the list as follows:
   - To apply an additional Access Template to the object, click **Add** and follow the instructions in the Delegation of Control Wizard.
   - To remove permissions specified by an Access Template on the object, select the Access Template from the list and click **Remove**.
5. Click **OK** to close the **Active Roles Security** dialog box.
6. Click **OK** to close the **Properties** dialog box for the AD LDS object.

In the Delegation of Control Wizard, you can select the users or groups (Trustees) to give permissions to, and select one or more Access Templates from the **Access Templates/AD LDS (ADAM)** container to define the permissions. As a result, the Trustees you select have the permissions that are defined by those Access Templates on the AD LDS object. The Trustees can exercise the permissions only within Active Roles as Active Roles does not stamp permission settings in AD LDS.

In the **Active Roles Security** dialog box, an Access Template can only be removed if it is applied to the object you have selected (rather than to a container that holds the object). To view the Access Templates that can be removed on the current selection, clear the **Show inherited** check box.

Instead of removing an Access Template in the **Active Roles Security** dialog box, you can select the Access Template and then click **Disable** in order to revoke the permissions on the object that are defined by the Access Template. In this way, you can block the effect of an Access Template regardless of whether the Access Template is applied to the object itself or to a container that holds the object. You can undo this action by selecting the Access Template and then clicking **Enable**.

Viewing or setting policies on AD LDS objects

By using the Active Roles console, you can apply Active Roles Policy Objects to establish policy-based control and perform auto-provisioning of AD LDS data the same way as you do for the directory data held in Active Directory domains. By providing the ability to strictly enforce operating policies and to prevent unregulated access to sensitive information stored in AD LDS, Active Roles helps ensure the security of your business-critical data. Policy Objects can be configured to determine a wide variety of policies as applied to AD LDS, including data format validation, rule-based auto-provisioning of certain portions of data in AD LDS, and script-based, custom actions on AD LDS data.

You can use the following instructions to view or modify a list of Policy Objects that are applied to a given AD LDS object, such as an AD LDS user, group, organizational unit, container, or entire directory partition. For detailed instructions on how to create, configure and apply Policy Objects, see the Rule-based AutoProvisioning and Deprovisioning section earlier in this document.

To view or modify the list of Policy Objects on an AD LDS object

1. In the console tree, under **AD LDS (ADAM)**, locate and select the container that holds the object on which you want to view or modify the list of Policy Objects.
2. In the details pane, right-click the object, and click **Properties**.
3. On the **Administration** tab in the **Properties** dialog box, click **Policy**.
4. In the **Active Roles Policy** dialog box, view the list of Policy Objects that have effect on the AD LDS object, or modify the list as follows:
   - To apply an additional Policy Object to the AD LDS object, click **Add**, select the Policy Object to apply, and then click **OK**.
   - To remove the effect of a Policy Object on the AD LDS object, select the Policy Object from the list and click **Remove**. Alternatively, select the **Blocked** check box next to the Policy Object name.
5. Click **OK** to close the **Active Roles Policy** dialog box.
6. Click **OK** to close the **Properties** dialog box for the AD LDS object.

In the **Active Roles Policy** dialog box, a Policy Object can only be removed if it is applied to the AD LDS object you have selected (rather than to a container that holds the AD LDS object). To view the Policy Objects that can be removed on the current selection, click **Advanced**, and then clear the **Show inherited** check box.

Instead of removing a Policy Object in the **Active Roles Policy** dialog box, you can select the **Blocked** check box in the list entry for that Policy Object in order to remove the effect of the Policy Object on the AD LDS object. In this way, you can remove the effect of a Policy Object regardless of whether the Policy Object is applied to the AD LDS object itself or to a container that holds the object. If you block a Policy Object on a given AD LDS object, the policy settings defined by that Policy Object no longer take effect on the AD LDS object. You can undo this action by clearing the **Blocked** check box.

16

One Identity Starling Join and Configuration through Active Roles

- Configure Join to Starling
- Prerequisites to configure One Identity Starling
- One Identity Starling Two-factor Authentication for Active Roles
- Disconnecting One Identity Starling from Active Roles
- Starling Two-Factor Authentication User Access template
- Allowing two-factor authentication for Active Roles users
- ARS 2FA Users group
- Steps to create ARS 2FA Users group manually
- Disallowing two-factor authentication for Active Roles users
- Logging in to Web interface through 2FA authentication
- Pre-requisites to use One Identity Starling 2FA
- Registering to One Identity Starling 2FA
- Disabling or Enabling Starling 2FA Users from Configuration Center

Configure Join to Starling

Active Roles 7.5 supports integration with One Identity Starling services. The Starling Join feature in Active Roles now enables you to connect to One Identity Starling, the Software as a Service (SaaS) solution of One Identity. The Starling Join feature enables access to the Starling services through Active Roles, allowing you to benefit from the Starling services such as Two-factor Authentication, Identity Analytics and Risk Intelligence, and Connect.

To start the wizard, in the Active Roles Configuration Center main window, on the **Dashboard** page, click **Configure** under **Starling**.

To use Starling 2FA with Active Roles, you first need to join One Identity Starling to Active Roles on the Active Roles Configuration Center. The Join to One Identity Starling wizard also includes links, which provide assistance for using Starling:

- The Online link displays information about the Starling product and the benefits you can take advantage of by subscribing to Starling services.
- The Trouble Joining link displays the Starling support page with information on the requirements and process for joining with Starling.

Prerequisites to configure One Identity Starling

Before you configure Starling using the Active Roles Configuration Center, ensure the following:

- Users must have acquired valid Starling Credentials, such as a Starling Organization Admin account or a Collaborator account. For more information on Starling, see the *One Identity Starling User Guide*.
- The computer running Active Roles must have TLS version 1.2 enabled. For more information, see How to enable TLS 1.2 on clients in the *Microsoft Core infrastructure documentation*.
- The computer running Active Roles must be able to connect directly to the web and reach the following web addresses at a minimum:
  - *.cloud.oneidentity.com
  - *.cloud.oneidentity.eu

  NOTE: Additional Microsoft URLs may be required depending on your Starling integration with Azure. For more information, see KB Article 229909 on the One Identity Support Portal.
- The Active Roles Administration Service must be running on the computer where you want to configure Starling.
- The Active Roles Administration Service must have a managed domain.
- You must disable **IE Enhanced Security Configuration** to allow the Starling Join process to complete. Once the Starling Join process has completed, you can re-enable this setting.
  Follow the steps to disable **IE Enhanced Security Configuration**.

To disable IE Enhanced Security Configuration

1. Open **Server Manager**.
2. On the left pane, select **Local Server**.
3. On the right pane, next to **IE Enhanced Security Configuration**, click **On** and in

the popup window, turn **Off** the appropriate connection type (Administrators or Users). If unsure, turn off both connection types.

Configuring Active Roles to join One Identity Starling

To configure Active Roles to join One Identity Starling

1. On the Active Roles Configuration Center, under Starling, click **Configure**.
2. Click **Join One Identity Starling**. The Get Started page on the Starling product is displayed.
3. On the Starling Get Started page, enter your work email address enabled with Starling, and click **Next**.
4. Enter the Starling credentials provided to you at the time of subscribing to Starling and follow the instructions displayed on the wizard to continue.

   NOTE:

   - If you have a Starling account, when a subscription is created for you, you will receive a Starling invitation email. Click the link in the email and log in to the Starling account.
   - If you do not have a Starling account, when a subscription is created for you, you will get a Starling Sign-up email to complete a registration process to create a Starling account. Complete the registration and log in using the credentials that you have provided during registration. For account creation details, see the *One Identity Starling User Guide*.

   The One Identity Starling dialog box in Active Roles with a progress message indicating the progress of joining Starling is displayed. A join confirmation page is displayed with the name of the Active Roles instance that is going to be joined to Starling .

   After the operation is completed successfully, the Starling tab is displayed with **Account Joined** success message.

To view the Starling 2FA settings

1. On the Active Roles Configuration Center, in the left pane, click **Starling**.
2. Click **Starling** tab.

   The status of the Starling connection is displayed.
3. Click **Starling 2FA** tab.

   The status Starling 2FA is displayed.

4. To disable the Starling 2FA feature click **Disable Starling 2FA**. To enable it again, click **Enable Starling 2FA**.

Disconnecting One Identity Starling from Active Roles

After you configure Active Roles to join Starling, in case you want to disconnect from Starling, on Starling tab in Starling page, click **Unjoin One Identity Starling**. Unjoin Starling operation will disconnect Active Roles from your subscription. You are prompted to confirm if you want to continue. Click **Yes** to disconnect Active Roles from your subscription and complete the Unjoin One Identity Starling operation.

One Identity Starling Two-factor Authentication for Active Roles

Since Active Roles manages confidential Active Directory user details in both on-premises and cloud based environments, it is appropriate and safer to have an additional security measure such as the two-factor authentication. Active Roles now supports One Identity's Starling Two-Factor Authentication service.

The Starling Two-factor authentication provides enhanced security by necessitating users to provide two forms of authentication to Active Roles, namely a user name and password combination along with a token response. The token response is collected through an SMS, Phone call, or push notification received on a physical device such as a mobile or any other device other than the browser.

Starling Two-Factor Authentication User Access template

On installing Active Roles on a computer, the Starling Join feature is included by default. The Starling Two-Factor Authentication User Access template is generated and displayed as part of the Builtin Access templates. The Starling - Two Factor Authentication User Access template provides the Active Roles users with minimal permissions that includes enabling of mobile and email property for the users.

ARS 2FA Users group

After the Starling Join operation is completed successfully, the ARS 2FA Users group is generated and displayed in the Builtin Container by default. All members of the 2FA group have the Starling Two-Factor Authentication User Access template applied by default.

Pre-requisites to use One Identity Starling 2FA

Active Roles users who can use the Starling Two-factor Authentication feature must satisfy the following conditions:

- The Active Roles users must be members of the ARS 2FA Users group.
- The Active Roles users must have Starling Two-Factor Authentication User Access template permissions applied.
- The Active Roles users must have their mobile number and Email address properties populated.

For information on the mobile number formats that are allowed, see the *One Identity Starling User Guide* on https://support.oneidentity.com/technical-documents.

Allowing two-factor authentication for Active Roles users

To allow Active Roles users to use two-factor authentication, add the users to the ARS 2FA Users group. Adding the users to the ARS 2FA Users group enables the minimal permissions on the users through the **Starling - Two Factor Authentication User Access** template to authorize the users for two-factor authentication.

In case of multiple managed domains, the ARS 2FA Users group must be created manually in each of the domains and the Starling - Two Factor Authentication User Access template must be applied on the group.

Steps to create ARS 2FA Users group manually

1. Create the **ARS 2FA Users** group in the Builtin container.
2. Apply the **Starling - Two Factor Authentication User** Access template to the Domain.
3. Run the following command in the Active Roles Management Shell:

   ```
   new-QARSAccessTemplateLink -AccessTemplate 'CN=All Objects - Read All
   Properties,CN=Active Directory,CN=Access Templates,CN=Configuration' -
   DirectoryObject 'CN=Starling Configuration,CN=Configuration' -Trustee
   ```

```
'Domain\ARS 2FA Users' -Proxy
```

4. Add AD users to the group.

Registering to One Identity Starling 2FA

In order to use Starling 2FA, you must first register to the product. When you register to Starling 2FA using your mobile number, an SMS is delivered with the mobile app download link. Click on the link to access the App Store or Play Store from where you can download the Starling mobile application. Alternatively, you can go to the App Store or Play Store and search and download the Starling.

The following 2FA options are supported:

- **Push Notification**: After the Starling app is downloaded and registered with user's email id and mobile number, the user will get a push notification to Approve or Deny Starling Authentication.
- **Voice**: The user will get a voice call on the registered mobile number and on call user will get an OTP.
- **SMS OTP**: The user will get an OTP through SMS on the registered mobile number.
- The user can open the Starling app and copy and paste the code form the Starling app to Active Roles, and then click on **Verify**.

Logging in to Web interface through 2FA authentication

When a Starling 2FA enabled user tries to log in to the Active Roles Web interface, the user is prompted to enter the Starling Two-factor token response. Based on the option selected by the user, the token response is provided through SMS, Phone Call or Push Notifications.

On entering the token response and after successful verification the Web interface is displayed.

NOTE: Push Notification works only if the Starling App is installed on the device with registered mobile number. The link to install the Starling App will be send to your registered mobile number at the time of registering to Starling.

Logging in to MMC interface through 2FA authentication

When a Starling 2FA enabled user tries to log in to the Active Roles MMC interface, the user is prompted to enter the Starling Two-factor token response. Based on the option selected by the user, the token response is provided through SMS, Phone Call or Push Notifications. After the token is generated the token request options are disabled.

In case the token must be generated again, you need to wait for the minimum notification retry interval for the request options to get enabled. The default value for the notification retry interval period is 30 seconds.

On entering the token response and after successful verification the MMC interface is displayed.

NOTE: Push Notification works only if the Starling App is installed on the device with registered mobile number. The link to install the Starling App will be send to your registered mobile number at the time of registering to Starling.

If the system is kept idle for more than 30 minutes, the 2FA session expires and the MMC console gets disconnected with a session timeout warning.

Disallowing two-factor authentication for Active Roles users

To disable Active Roles users for two-factor authentication, remove the users from the ARS 2FA Users group. Removing the users from the ARS 2FA Users group disables the minimal permissions on the users applied through the Starling - Two Factor Authentication User Access template that authorize the users for two-factor authentication.

Disabling or Enabling Starling 2FA Users from Configuration Center

In case of Starling outage, the Administrator has the privilege to disable Starling users from 2FA in the Web interface by clicking on the **Disable Starling 2FA** button in Starling 2FA tab in Starling page of Configuration Center.

To re-enable the Starling users to use 2FA again, administrators can click on the **Enable Starling 2FA** button in Starling 2FA tab in Starling page of Configuration Center.

18

Managing One Identity Starling Connect

Active Roles provides support to connect to Starling Connect to manage the user provisioning and deprovisioning activities for the registered connectors. Using the Starling Join feature in Active Roles, you can connect to One Identity Starling.

On joining to Starling, the registered connectors for the user are displayed if the Starling Connect subscription exists. If the subscription does not exist, visit the Starling site for Starling Connect subscription. The displayed connectors are available for provisioning or deprovisioning of users or groups through Active Roles.

Viewing Starling Connect settings in Active Roles Configuration Center

The Active Roles Configuration Center enables you to view the Starling Connect settings in order to manage the registered connected systems.

NOTE: Before you view the Starling Connect settings, Active Roles must be joined to One Identity Starling.

To view the Starling settings

1. On the Active Roles Configuration Center, in the left pane, click **Starling**.
2. On the **Starling** tab, click **Join One Identity Starling** to join Starling.

   NOTE: For more information on extending the Active Roles provisioning and account administration capabilities to your cloud applications, click **Learn More** in the Starling tab.

To view the Starling Connectors settings

1. On the Active Roles Configuration Center, in the left pane, click **Starling**.
2. Click **Starling Connectors** tab.

   The options specific to the page are displayed. The available options are **Connection Settings**, **Visit Starling Connect Online**, **Refresh Connectors**.
3. Click **Connection settings** to view the current settings.

   The Connection Settings wizard displays the current Starling connect settings, such as the Subscription ID, SCIM Client ID, Client secret, and token end point URL. The settings are not editable and the values are populated when you join Starling.
4. Click **Visit Starling Connect Online** to connect to the Starling Connect portal.

   The Starling Connect portal displays the registered connectors and enables you to add or remove connectors.

   NOTE: In case the connectors are not displayed on the Active Roles Starling Connect page, you can view the registered connectors on the Starling Connect portal.
5. Click **Refresh Connectors**, to view the latest connectors that are added or removed from the Starling Connect portal.

   NOTE:**Refresh Connectors** refreshes the Starling Connect policy to reflect the latest connector list.

Create Provisioning policy for Starling Connect

To create a Policy Object for Starling Connect

1. In the console tree, under **Configuration** | **Policies** | **Administration**, locate and select the folder in which you want to add the Policy Object.

   You can create a new folder as follows: Right-click **Administration** and select **New** | **Container**. Similarly, you can create a sub-folder in a folder: Right-click the folder and select **New** | **Container**.
2. Right-click the folder, point to **New**, and then click **Provisioning Policy**.
3. On the **Welcome** page of the wizard, click **Next**.

4. On the **Name and Description** page, do the following, and then click **Next**:
    a. In the **Name** box, type a name for the Policy Object.
    b. Under **Description**, type any optional information about the Policy Object.
5. On the **Policy to Configure** page, select **Autoprovisioning in SaaS products**, and click **Next** to configure policy settings.
6. On the Object Type Selection page, click **Select**.
    a. On the **Select Object Type**, from the Object types list, select **User** or **Group**, and click **OK**.
    b. Click **Next**.
    c. On the **Policy Conditions** page, from the **Starling Connect Connectors** list, select the connectors to be provisioned for the user or group as part of the policy. Click **Next**.
7. On the **Enforce Policy** page, you can specify the containers on which this Policy Object is to be applied:
    a. Click **Add**, and use the **Select Objects** to locate and select the objects you want.
    b. Click **Next**.
8. Click **Finish**.

IMPORTANT: Starling Connect policy have to be applied on the container for any SaaS operations to take place.

SaaS operations for each connector may vary from each other. Each connector may have a set of mandatory attributes to perform any operation.

The operation will fail in case any of the mandatory attributes are missing in the particular request. The notification will report the information of all the mandatory attributes missing in that event which caused the failure.

In that case, you must create the corresponding virtual attributes, customize the Web Interface to enter the value for the virtual attribute during the specified operation. Using this approach, the attribute value is passed as a part of the request.

Provision a new SaaS user using the Web interface

You can use the Active Roles Web Interface to create and enable a new user with Starling Connect management capabilities.

To provision a new SaaS products user in Active Roles

1. On the Active Roles Web interface Navigation bar, click **Directory Management**.
2. On the **Views** tab in the **Browse** pane, click **Active Directory**.

The list of Active Directory domains is displayed.

3. Click the domain in which you need to create a new user.
4. In the list of objects displayed, click the required Container or the Organizational Unit on which the Starling Connect Policy is applied.
5. In the **Command** pane, click **New User**.
6. In the **New User in <OU name>** | **General** wizard, enter the user details such as First Name, Last Name, Initials, and User logon name.
7. Click **Next**.
8. In the Account properties wizard, click **Generate** to generate a password for the Account, select the required **Account** options, and then click **Next**.

   The **SaaS Products** tab displays the list of registered Starling Connect connectors. The Starling Connect connectors for which you can provision users are displayed with selected check boxes.
9. Click **Finish**.

   The user is created successfully and provisioned on the selected connected systems as per the policy applied.

Provision an existing Active Roles user for SaaS products

You can use the Active Roles Web Interface to enable an existing Active Roles user with Starling Connect management capabilities.

To provision an existing Active Roles user for SaaS products

1. On the Active Roles Web interface Navigation bar, click **Directory Management**.
2. On the **Views** tab in the **Browse** pane, click **Active Directory**.

   The list of Active Directory domains is displayed.
3. Click the specific domain, Container or the Organizational Unit, and then select the check box corresponding to the specific user, which you want to provision for SaaS products.
4. In the **Command** pane, click **Provision Object in SaaS Products**.

   The **SaaS Products** tab displays the list of registered Starling Connect connectors. The Starling Connect connectors for which you can provision users are displayed with selected check boxes.
5. Click **Finish**.

   The user is provisioned on the selected connected systems as per the policy applied.

Update the SaaS product user properties

For an existing Active Roles Starling Connect user, you can use the Active Roles Web Interface to update the properties. When Active Roles user properties are updated, if the user property is mapped to Starling Connect User properties, then the changes are reflected for the selected connected system.

Delete the SaaS product user

You can use the Active Roles Web Interface to delete an existing Active Roles Starling Connect user. When the Active Roles user is deleted, then the user is deleted on the selected connected system.

Deprovision an existing Active Roles user for SaaS products

Active Roles provides the ability to deprovision SaaS product users. When an Active Roles user is deprovisioned, if the user is mapped to Starling Connect, then the user is deprovisioned from the selected connected system. This means the Active Role SaaS product user is prevented from logging on to the network and connecting to any of the connected systems through the registered connectors.

The Deprovision command on a user updates the account as prescribed by the deprovisioning policies.

Active Roles comes with a default policy to automate some commonly-used deprovisioning tasks, and allows the administrator to configure and apply additional policies.

To deprovision a user for a SaaS product

1. On the Active Roles Web interface **Navigation** bar, click **Directory Management**.
2. On the **Views** tab in the **Browse** pane, click **Active Directory**.

   The list of Active Directory domains is displayed.
3. Click the specific domain, Container or the Organizational Unit, and then select the check box corresponding to the specific user, which you want to deprovision for SaaS products
4. Select the user, and in the **Command** pane, click **Deprovision**.

   A message is displayed prompting you to confirm the account deprovision.

5. Click **Yes**, to continue

   Wait while Active Roles updates the user.

   After the task is completed, a message is displayed that the account is deprovisioned successfully from Active Roles.

   If the user is mapped to Starling Connect, then the user is deprovisioned from the connected systems.

To undo deprovision of a user for a SaaS product

1. On the Active Roles Web interface **Navigation** bar, click **Directory Management**.
2. On the **Views** tab in the **Browse** pane, click **Active Directory**.

   The list of Active Directory domains is displayed.
3. Click the specific domain, Container or the Organizational Unit, and then select the check box corresponding to the specific user, which you want to undo deprovision for SaaS products.
4. In in the **Command** pane, click **UndoDeprovisioning**.

   The **Password Options** dialog box is displayed.
5. Select the option to **Leave the Password** unchanged or **Reset** the password, and click **OK**.

Notifications for Starling operations

The **Notification** pane displays the notification specific to Starling operations. The notifications are classified into **Starling Connect** and **Updates**.

IMPORTANT:

- You must enable **Port 7465 (HTTP) TCP Inbound/Outbound** and **Port 7466 (HTTPS) TCP Inbound/Outbound** for the notifications to work. For more information, see Access to domain controllers.
- The Web Interface machine must be able to resolve Service machine name for Notifications to work.

To view the Starling Connect notification

1. On the Active Roles Web interface, click the notification icon.

   Starling Connect and **Updates** tabs are displayed.
2. Click the **Starling Connect** tab to view the notifications specific to SaaS operations.

   NOTE: The latest five notifications are sent only to the initiator of the operation.

To view the Updates

1. On the Active Roles Web interface, click the notification icon.

   Starling Connect and **Updates** tabs are displayed.
2. Click the **Updates** tab to view the important updates about Starling.
3. For more information on the notification, click **Read More**.

   NOTE: The notifications are sent to all the users who have joined Starling on the Administration website.

To view notifications on the Notifications page

1. On the Active Roles Web interface, click the notification icon.

   Starling Connect and **Updates** tabs are displayed.
2. Click the **Starling Connect** tab to view the notifications specific to SaaS operations.

   The latest five notifications are displayed with the configuration status and a brief description.
3. Click **View all notifications** to view the details about the notification.

   The **Notification** page is displayed.
4. Click **Filter** drop-down menu to filter the notifications based on time, connector name , status, and keywords.
5. Select the required notifications and click **Export to CSV** from the **Action** drop-down menu. Click **Go**. You can also delete a notification by selecting a particular checkbox.
6. Point the mouse to the notification in the **Message** column to view a detailed description. Expand the connector information available next to the connector check box to view the detailed description. The description pane gives the link to Change History of that particular object for more details. You can also copy the message in case of a failure.

Configuring notification settings

You can configure notifications settings from the **Home screen | Settings page** and **Home screen | Customization | Global Settings**.

To configure notification settings on the Settings page

1. On the Active Roles Web interface, click the **Settings**.

   The **Settings** page is displayed.
2. On the **Settings** page, enter the time in minutes for which the notification is to be visible in **Time (in minutes) for which the notification is visible** field.

   NOTE: By default, the time is set to 0 and the notifications do not expire. You can update the time to the required limit in minutes.

3. Enter the number of notifications to be stored in **Maximum number of notifications to be stored in Active Roles** field.

   NOTE: The maximum number of notifications that can be stored is 1000.

To configure notification settings on the Customization page

1. On the Active Roles Web interface, click the **Customization**.

   The **Customization** page is displayed.
2. On the **Customization** page, click **Global Settings**.
3. In the **Settings applied for every user of the Web Interface by default** section, enter the time in minutes for which the notification is to be visible in **Time (in minutes) for which the notification is visible** field.

   NOTE: By default, the time is set to 0. You can update the time to the required limit in minutes.
4. Enter the number of notifications to be stored in **Maximum number of notifications to be stored in Active Roles** field.

   NOTE: The maximum number of notifications that can be stored is 1000.

IMPORTANT: For notifications to work as expected, you must perform the following, if you are using ActiveRoles website over HTTPS:

- Import a valid certificate into Trusted Root Certificate Authority in the machine where ActiveRoles Service is installed.
- In the below command, substitute thumbprint of the newly added certificate to CERT_HASH.
- In the below command, substitute a Unique GUID to APP_ID.
- Execute the command below in PowerShell command interface:

  netsh http add sslcert ipport=0.0.0.0:7466 appid='{APP_ID}' certhash=<CERT_HASH>.

SCIM attribute mapping with Active Directory

Active Roles provides support to connect to Starling Connect to manage the user provisioning and deprovisioning activities for the registered connectors. This is achieved through the internal attribute mapping mechanism. The AD attributes are mapped to SCIM attributes to perform each operation.

SCIM attribute mapping with Active Directory for Users

SCIM	Active Directory
displayName	displayName
givenName	givenName
familyName	sn
middleName	middleName
title	title
password	edsaPassword
streetAddress	streetAddress
locality	city
postalCode	postalCode
region	state
country	c
active	edsaAccountIsDisabled
userName	edsvauserName
honorificPrefix	initials
formattedName	cn
emails	proxyAddresses,mail
preferredLanguage	preferredLanguage
description	description
emailEncoding	edsvaemailEncoding
alias	edsvaalias
division	division
company	company
department	department
homePage	wWWHomePage
lastLogon	lastLogon
accountExpires	accountExpires
timezone	edsvatimezone
entitlements	edsvaentitlements
employeeNumber	employeeNumber

cn	cn
userPermissionsMarketingUser	edsvauserPermissionsMarketingUser
userPermissionsOfflineUser	edsvauserPermissionsOfflineUser
userPermissionsAvantgoUser	edsvauserPermissionsAvantgoUser
userPermissionsCallCenterAutoLogin	edsvauserPermissionsCallCenterAutoLogin
userPermissionsMobileUser	edsvauserPermissionsMobileUser
userPermissionsSFContentUser	edsvauserPermissionsSFContentUser
userPermissionsKnowledgeUser	edsvauserPermissionsKnowledgeUser
userPermissionsInteractionUser	edsvauserPermissionsInteractionUser
userPermissionsSupportUser	edsvauserPermissionsSupportUser
userPermissionsLiveAgentUser	edsvauserPermissionsLiveAgentUser
locale	localeID
phoneNumbers	telephoneNumber,mobile,homePhone
manager	manager
desiredDeliveryMediums	edsvadesiredDeliveryMediums
nickname	edsvanickname

SCIM attribute mapping with Active Directory for Groups

SCIM	Active Directory
displayName	cn
members	member
email	mail
manager	managedBy

19

Azure AD, Office 365, and Exchange Online Management

Active Roles is an administrative platform that facilitates the administration and provisioning of Active Directory, Exchange, and Azure Active Directory (Azure AD) resources in on-premises, cloud-only and hybrid environments as well. You can manage all these resources through the Active Roles Web Interface.

- In an on-premises environment, when you create new Active Directory objects (users, guest users, groups, contacts, and so on), Active Roles creates and stores these new objects in the local infrastructure of your organization.
- In a cloud-only environment, when you create new Active Directory objects (users, guest users, groups, contacts, and so on), Active Roles creates and stores these new objects in the Azure Cloud.
- In hybrid environments, when you create new Active Directory objects (users, guest users, contacts, and so on) Active Roles synchronizes the on-premises Active Directory objects and their properties to the Azure AD cloud. This synchronization is performed by the Active Roles Synchronization Service between Active Roles and Microsoft Office 365, whenever you configure an Active Directory object with the Active Roles Web Interface.

NOTE: The Active Roles Web Interface supports Azure AD-related operations only on sites based on the Administrators template. While some of the configuration procedures listed in this page are also supported through the Management Shell, they are all described with using the Active Roles Web Interface.

The Office 365 / Azure AD capabilities of Active Roles support the following administrative tasks:

- Create an Office 365 user account associated with a given Active Directory user account.
- Synchronize user properties from Active Directory user accounts to their associated Office 365 user accounts.
- View or change the properties of the Office 365 user account associated with a given Active Directory user account.

- Assign Office 365 licenses to the Office 365 user account associated with a given Active Directory user account.
- Delete the Office 365 user account associated with a given Active Directory user account.
- Create an Office 365 security group or distribution group associated with a given Active Directory group.
- Synchronize group properties, including the members list, from Active Directory groups to their associated Office 365 groups.
- View or change the properties of the Office 365 group associated with a given Active Directory group.
- Delete the Office 365 group associated with a given Active Directory group.
- Create an Office 365 external contact associated with a given Active Directory contact.
- Synchronize contact properties from Active Directory contacts to their associated Office 365 external contacts.
- View or change the properties of the Office 365 external contact associated with a given Active Directory contact.
- Delete the Office 365 external contact associated with a given Active Directory contact.
- View Office 365 domain and license information.
- Create Office 365 users. When you create an Office 365 user, you can choose whether to license that user for Exchange Online.
- Create security groups and distribution groups in Office 365. You can choose the type of the Office 365 group that you want to create.
- Assign licenses to Office 365 users. When creating or administering a user, you can choose the Office 365 licenses that you want to assign to that user.
- Restrict the licenses for Office 365 users. You can configure a policy to specify what Office 365 licenses can be assigned depending on user location in Active Directory.
- View or change the Office 365 specific object properties. You can edit Office 365 users, groups and contacts.
- Examine Office 365 licenses and license usage. For each of your license subscriptions, you can view how many licenses are valid, expired or assigned. This information is displayed on the add-on application page in the Active Roles console.
- Examine Office 365 domains. For each of your Office 365 domains, you can view the status of the domain and see whether the domain is configured for single sign-on (federated). Azure Domains are listed in Azure Domains in Azure Configuration
- Associate existing Office 365 users with on-premises Active Directory users. The synchronization workflow uses the GUID and the primary SMTP address to match an existing Office 365 user to the appropriate on-premises Active Directory user.

Configuring Active Roles to Manage Hybrid AD Objects

When a user signs up for a Microsoft cloud service such as Azure Active Directory, details about the user's organization and the organization's Internet domain name registration are provided to Microsoft. This information is then used to create a new Azure AD instance for the organization. The same directory is used to authenticate sign in attempts when you subscribe to multiple Microsoft cloud services.

The Azure AD instance of the organization, also called the Azure AD tenant, stores the users, groups, applications, and other information pertaining to an organization and its security. To access the Azure AD tenant, we need an application that is registered with the tenant. Active Roles uses this application, also called the Azure AD application, to communicate to Azure AD tenant after providing the required consent.

The Active Roles Web Interface and Management Shell can be used to perform the Azure AD configuration tasks. The new feature in Active Roles enables you to add or modify existing tenants to the management scope through the web interface and Management Shell.

The latest release of Active Roles supports Multiple tenants model.

> NOTE: Administrative users or users with sufficient privileges only can view Azure configuration.

The following section guides you through the Active Roles web interface and Management Shell to configure Azure AD tenants and applications and synchronize existing AD objects to Azure AD.

- Configuring Active Roles to manage Azure AD using the GUI
- Configuring Active Roles to manage Hybrid AD using Management Shell
- Active Roles Configuration steps to manage Hybrid AD objects
- Active Roles Configuration to synchronize existing Azure AD objects to Active Roles
- Changes to Azure O365 Policies in Active Roles after 7.4.1

Configuring Active Roles to manage Azure AD using the GUI

Use the Active Roles Web Interface and the Active Roles Configuration Center to perform the following actions and configure Azure AD deployments:

- Configuring a new Azure tenant and consenting Active Roles as an Azure application
- Importing an Azure tenant and consenting Active Roles as an Azure application
- Viewing or modifying the Azure AD tenant type
- Removing an Azure AD tenant

- Delete an Azure AD Application
- View Azure Health for Azure AD tenants and applications
- View Azure Licenses Report

Configuring a new Azure tenant and consenting Active Roles as an Azure application

When installing Active Roles out-of-the-box, the **Directory Management** > **Tree** > **Azure** node of the Active Roles Web Interface only contains an empty **Azure Configuration** sub-node by default.

To manage Azure AD directory objects (Azure users, guest users, contacts, O365 groups and Azure Security groups), you must specify an Azure tenant and configure Active Roles as a consented Azure application for it in the Active Roles Configuration Center.

NOTE: If you have already used an Azure tenant (or tenants) in a previous version of Active Roles, you can import and reconfigure them in two ways:

- If you perform an in-place upgrade of Active Roles (that is, you install the latest version without uninstalling the previous version of Active Roles first in one of the supported upgrade paths), you can reauthenticate the existing Azure tenants with the **Upgrade configuration** wizard upon launching the Active Roles Configuration Center after installation.

  For more information on reauthenticating Azure tenants this way, see *Reconfiguring Azure tenants during upgrade configuration* in the *Active Roles 7.5 Quick Start Guide*. For more information on the supported upgrade paths, see *Version upgrade compatibility chart* in the *Active Roles 7.5 Release Notes*.

- If you install a new version of Active Roles to a machine that does not have any earlier versions of the software installed (either because it has been already uninstalled, or it has been installed on another machine), you can import your existing Azure tenant(s) by importing your Azure AD configuration. Following the import, you can reconsent your Azure tenants manually.

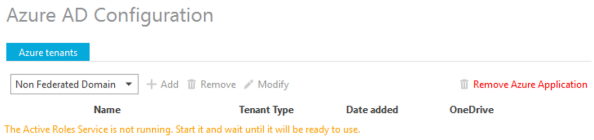
  For more information on importing existing Azure tenants this way, see Importing an Azure tenant and consenting Active Roles as an Azure application.

Prerequisites

The Active Roles Administration Service must be already running. If the service is not running, then:

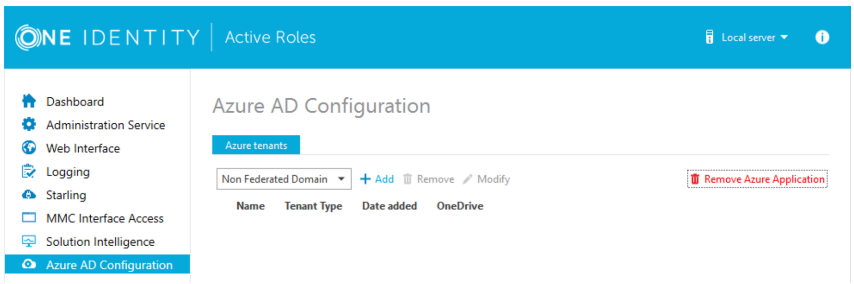
1. Open the Active Roles Configuration Center.
2. Navigate to the **Administration Service** page.
3. Click **Start**.

TIP: If the Active Roles Administration Service is not running, the **Azure AD Configuration** page indicates it with an on-screen warning.

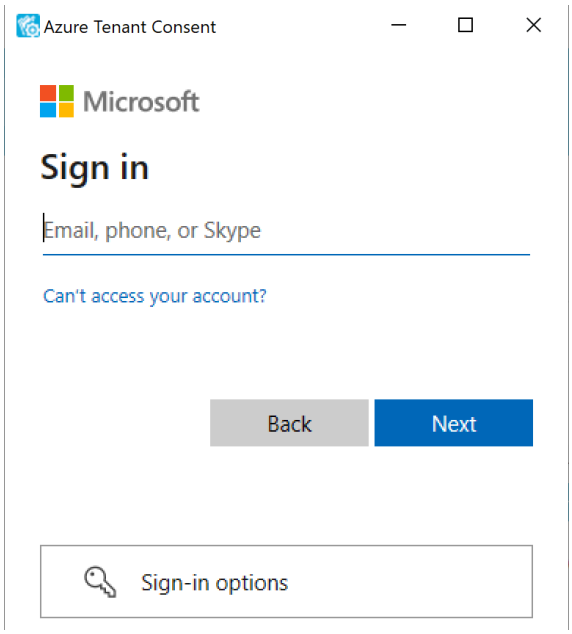


To configure a new Azure tenant (or tenants) and set Active Roles as a consented Azure application

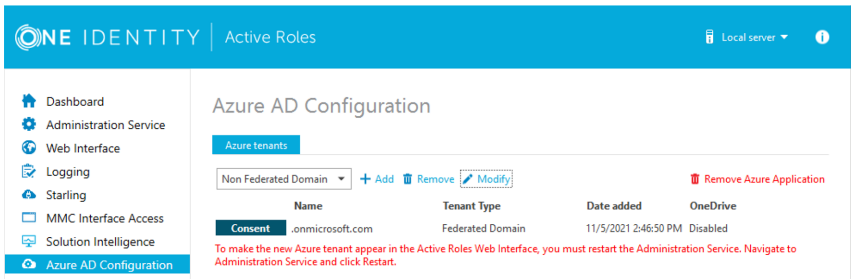
1. In the Active Roles Configuration Center, on the left pane, click **Azure AD Configuration**.



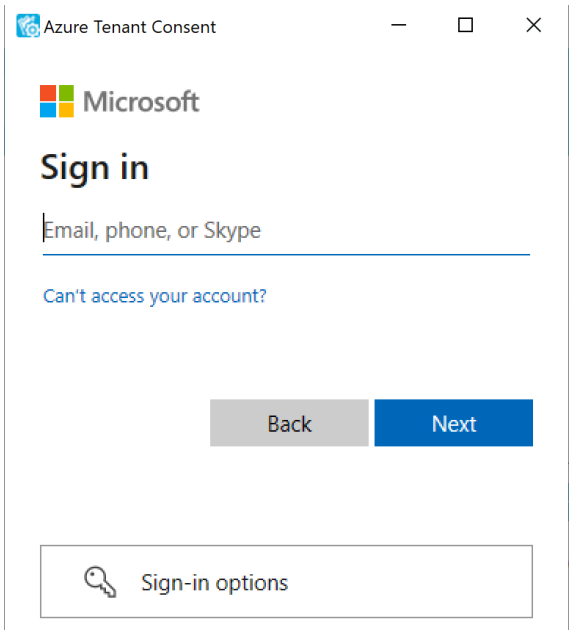
2. From the drop-down list, select the type of domain assigned to the new Azure AD tenant:
   - **Non-Federated Domain**: When selected, on-premises domains are not registered in Azure AD, and Azure AD Connect is not configured. Azure users and Azure guest users are typically created with the `onmicrosoft.com` UPN suffix.
   - **Federated Domain**: On-premises domains are registered in Azure AD and Azure AD Connect. Also, Active Directory Federation Services (ADFS) is configured. Azure users and Azure guest users are typically created with the UPN suffix of the selected on-premises domain.
   - **Synchronized Identity Domain**: On-premises domains may or may not be registered in Azure AD. Azure AD Connect is configured. Azure users and Azure guest users can be created either with the selected on-premises domain, or with the `onmicrosoft.com` UPN suffix.
3. To configure a new Azure tenant, click **Add**.
4. The standard Microsoft Azure **Select Account** or **Sign In** pop-up appears. Log in with the user name and password of your Azure AD administrator account.



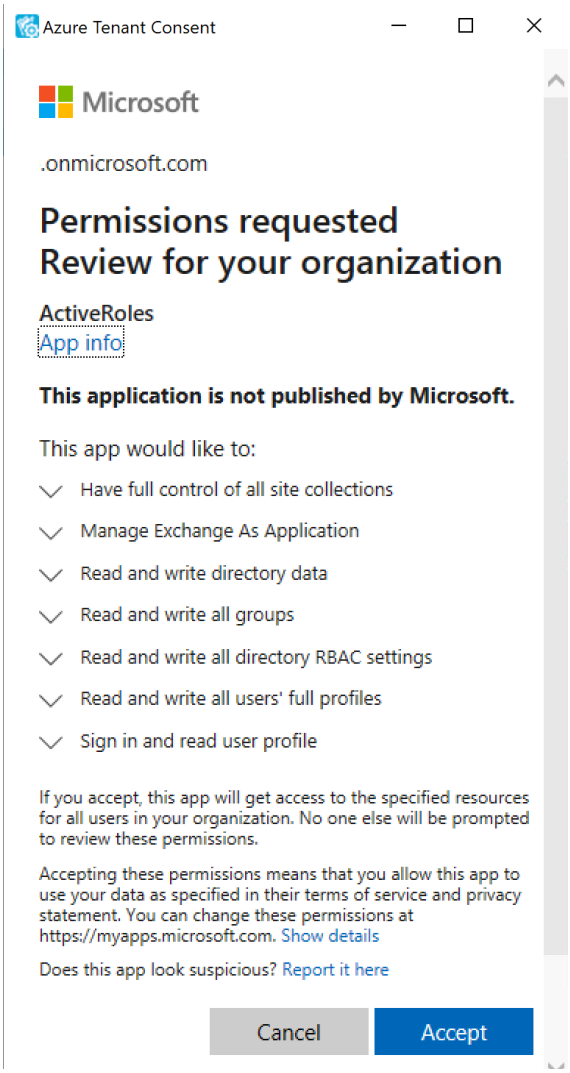
Upon successful authentication, the new Azure tenant appears in the list.



5. To be able to manage the Azure tenant and its contents in the Active Roles Web Interface, you must consent Active Roles as an Azure application. To do so, click **Consent** next to the Azure tenant.
6. The standard Microsoft Azure **Select Account** or **Sign In** pop-up appears. Log in with the user name and password of your Azure AD administrator account.



7. The standard Microsoft **Permissions requested** pop-up appears, listing all the permissions required for configuring Active Roles as an Azure application. To finish creating the Azure application, click **Accept**.



Active Roles then authenticates every Azure AD administrative operation performed in the Azure tenant with a set of generated client ID and client secret.

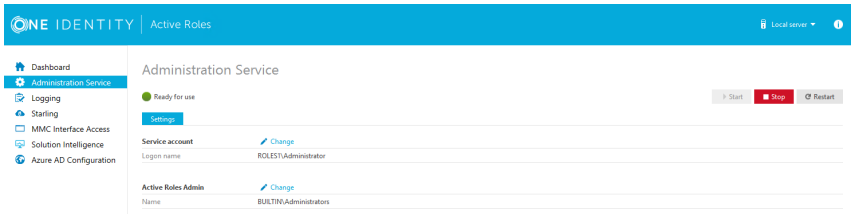
NOTE: Once you click **Accept**, Windows may show a **Security Warning** pop-up with the following message:

```
The current webpage is trying to open a site on your intranet. Do you want
to allow this?
```

In such cases, clicking either **Yes** or **No** could freeze the pop-up dialog, but consenting the Azure tenant will finish without problem.

This issue can occur in case the computer running Active Roles has incorrect browser settings. As a workaround, to get an up-to-date status of the state of the Azure tenant, close and restart the Active Roles Configuration Center after clicking **Yes** in the **Security Warning** pop-up.

8. If you have additional Azure tenants to add and consent, configure them as described in the previous steps of this procedure.
9. To make the configured Azure tenant(s) appear in the Active Roles Web Interface, you must restart the Administration Service, as indicated on the user interface. Click **Administration Service** on the left pane, then either click **Restart**, or first click **Stop** and then **Start**.



TIP: Once the Azure tenant or tenants are configured, and Active Roles is also set as a consented Azure AD application for it, you can view and modify the configured tenant(s) and their settings at the following locations:

- To change the domain type or OneDrive provisioning settings of an Azure tenant, in the Active Roles Configuration Center, navigate to **Azure AD Configuration**, select the Azure tenant, and click **Modify**. For more information, see Viewing or modifying the Azure AD tenant type.
- To check the connectivity status of the Azure configuration, in the Active Roles Web Interface, navigate to **Directory Management** > **Tree** > **Azure** > **Azure Configuration** > **Azure Health Check**. For more information, see View Azure Health for Azure AD tenants and applications .
- To check the Azure Licenses Report, in the Active Roles Web Interface, navigate to **Directory Management** > **Tree** > **Azure** > **Azure Configuration** > **Azure Licenses Report**. For more information, see View Azure Licenses Report .
- To check the Office 365 Roles Report, in the Active Roles Web Interface, navigate to **Directory Management** > **Tree** > **Azure** > **Azure Configuration** > **Office 365 Roles Report**. For more information, see View Office 365 Roles Report .

NOTE: Consider the following when configuring an Azure tenant:

- When Active Roles is registered as a consented Azure AD application, minimal permissions are assigned to it by default. To add additional permissions to the Azure application, sign in to the Azure Portal and add your required permissions there.
- Azure Multi-Factor Authentication (MFA) is automatically enforced for Azure users and Azure guest users added to the configured Azure tenant. To disable Azure MFA for the Azure tenant, sign in to the Azure Portal and navigate to **Tenant** > **Properties** > **Manage Security defaults** and set **Enable Security defaults** to **No**.

Importing an Azure tenant and consenting Active Roles as an Azure application

If you have previously managed an Azure AD deployment, but you are not upgrading from a previous version of Active Roles via in-place upgrade (for example, because the previous version of Active Roles has been uninstalled before installing the new version), you can import, reauthenticate and consent existing Azure tenants via the Active Roles Configuration Center.

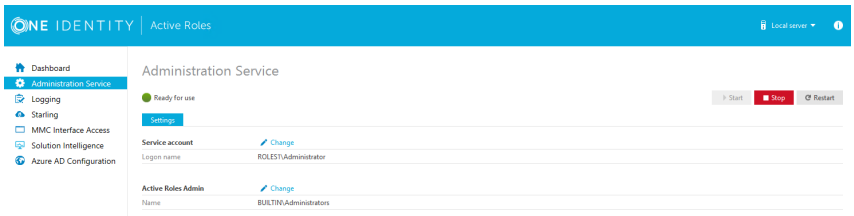
NOTE: Consider the following if you have not used any Azure tenants earlier, or if you installed the latest version of Active Roles via in-place upgrade:

- If you have installed Active Roles out-of-the-box, and no Azure AD environment has been used previously in your organization, you must specify a new Azure tenant to manage Azure directory objects (such as Azure users, guest users, contacts, O365 groups or Azure Security groups). For more information, see Configuring a new Azure tenant and consenting Active Roles as an Azure application.
- If you perform an in-place upgrade of Active Roles (that is, you install the latest version without uninstalling the previous version of Active Roles first in one of the supported upgrade paths), you can reauthenticate the existing Azure tenants with the **Upgrade configuration** wizard upon launching the Active Roles Configuration Center after installation.

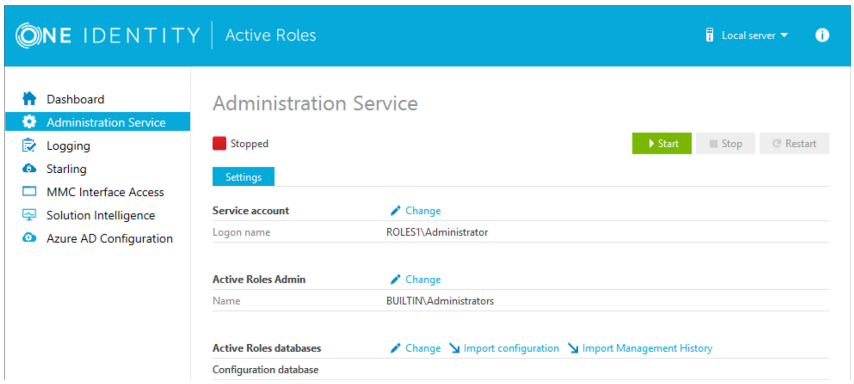
  For more information on reauthenticating Azure tenants this way, see *Reconfiguring Azure tenants during upgrade configuration* in the *Active Roles 7.5 Quick Start Guide*. For more information on the supported upgrade paths, see *Version upgrade compatibility chart* in the *Active Roles 7.5 Release Notes*.

To import and reauthenticate an Azure tenant and set Active Roles as a consented Azure application

1. Stop the Active Roles Administration Service. To do so, in the Active Roles Configuration Center, on the left pane, navigate to **Administration Service** and click **Stop**.



2. Once the Active Roles Administration Service stopped, open the **Import configuration** wizard by clicking **Active Roles databases** > **Import configuration**.

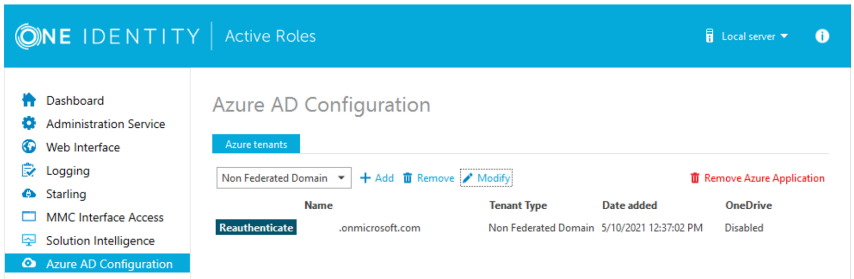


3. Perform the steps of the wizard. For more information, see Import configuration data or *Steps to deploy the Administration Service* in the *Active Roles Quick Start Guide*.

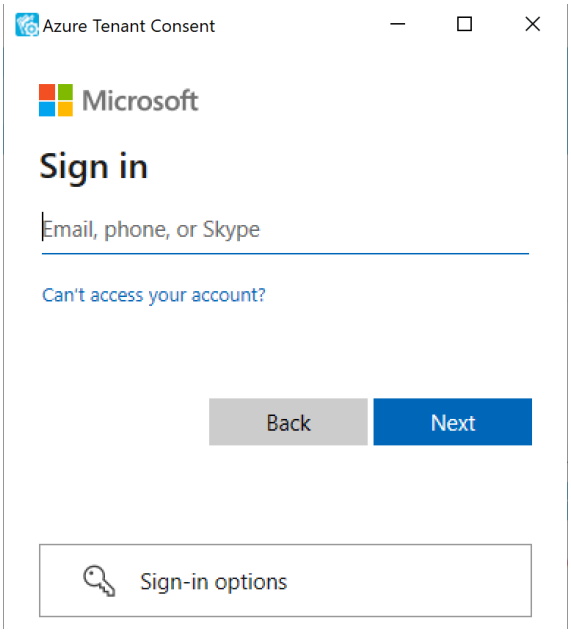
   CAUTION: Importing a configuration will overwrite every Azure tenant currently listed in the Azure AD Configuration page with those included in the imported configuration.

4. Once the import procedure finished, start the Active Roles Administration Service by clicking **Start** in the **Administration Service** page.
5. In the Active Roles Configuration Center, on the left pane, click **Azure AD Configuration**.

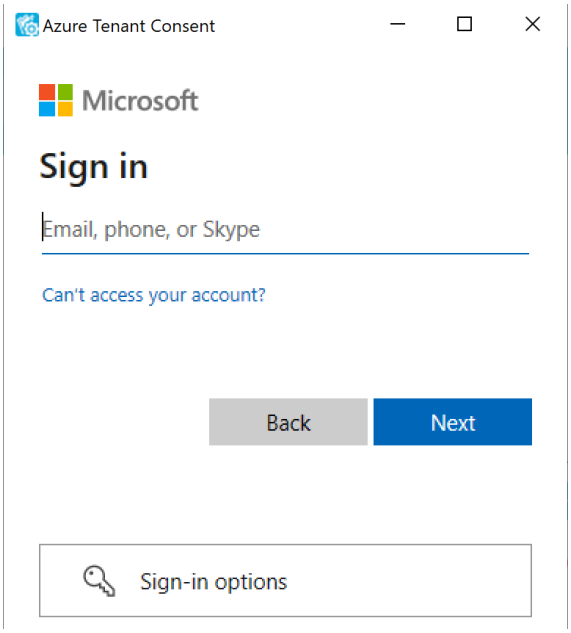
   The list of imported Azure tenants appears.



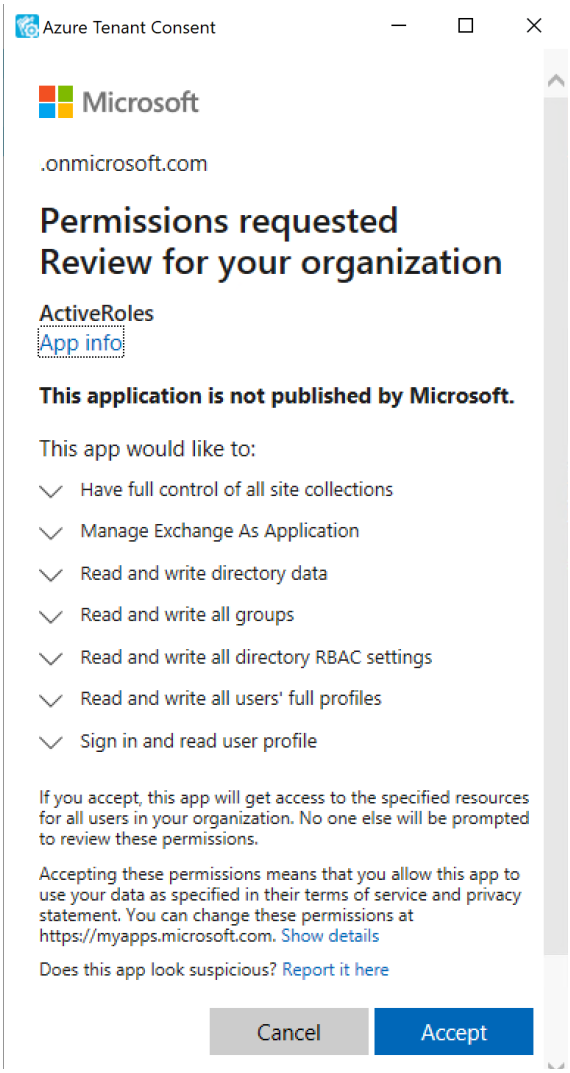
6. To configure an imported Azure tenant, click **Reauthenticate**.
7. The standard Microsoft Azure **Select Account** or **Sign In** pop-up appears. Log in with the user name and password of your Azure AD administrator account.



8. To be able to manage the Azure tenant and its contents in the Active Roles Web Interface, you must consent Active Roles as an Azure application. To do so, click **Consent** next to the Azure tenant.
9. The standard Microsoft Azure **Select Account** or **Sign In** pop-up appears. Log in with the user name and password of your Azure AD administrator account.



10. The standard Microsoft **Permissions requested** pop-up appears, listing all the permissions required for configuring Active Roles as an Azure application. To finish creating the Azure application, click **Accept**.



Active Roles then authenticates every Azure AD administrative operation performed in the Azure tenant with a set of generated client ID and client secret.

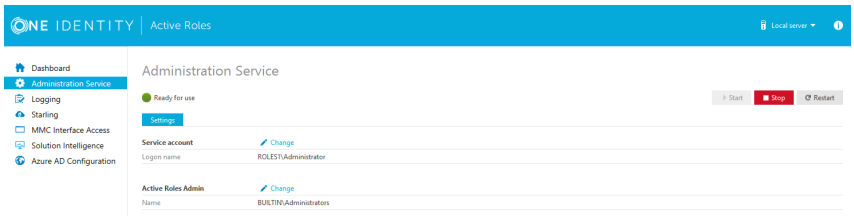
NOTE: Once you click **Accept**, Windows may show a **Security Warning** pop-up with the following message:

```
The current webpage is trying to open a site on your intranet. Do you want
to allow this?
```

In such cases, clicking either **Yes** or **No** could freeze the pop-up dialog, but consenting the Azure tenant will finish without problem.

This issue can occur in case the computer running Active Roles has incorrect browser settings. As a workaround, to get an up-to-date status of the state of the Azure tenant, close and restart the Active Roles Configuration Center after clicking **Yes** in the **Security Warning** pop-up.

11. To make the configured Azure tenant(s) appear in the Active Roles Web Interface, you must restart the Administration Service, as indicated on the user interface. Click **Administration Service** on the left pane, then either click **Restart**, or first click **Stop** and then **Start**.



TIP: Once the Azure tenant or tenants are configured, and Active Roles is also set as a consented Azure AD application for it, you can view and modify the configured tenant(s) and their settings at the following locations:

- To change the domain type or OneDrive provisioning settings of an Azure tenant, in the Active Roles Configuration Center, navigate to **Azure AD Configuration**, select the Azure tenant, and click **Modify**. For more information, see Viewing or modifying the Azure AD tenant type.
- To check the connectivity status of the Azure configuration, in the Active Roles Web Interface, navigate to **Directory Management** > **Tree** > **Azure** > **Azure Configuration** > **Azure Health Check**. For more information, see View Azure Health for Azure AD tenants and applications .
- To check the Azure Licenses Report, in the Active Roles Web Interface, navigate to **Directory Management** > **Tree** > **Azure** > **Azure Configuration** > **Azure Licenses Report**. For more information, see View Azure Licenses Report .
- To check the Office 365 Roles Report, in the Active Roles Web Interface, navigate to **Directory Management** > **Tree** > **Azure** > **Azure Configuration** > **Office 365 Roles Report**. For more information, see View Office 365 Roles Report .

NOTE: Consider the following when configuring an Azure tenant:

- When Active Roles is registered as a consented Azure AD application, minimal permissions are assigned to it by default. To add additional permissions to the Azure application, sign in to the Azure Portal and add your required permissions there.
- Azure Multi-Factor Authentication (MFA) is automatically enforced for Azure users and Azure guest users added to the configured Azure tenant. To disable Azure MFA for the Azure tenant, sign in to the Azure Portal and navigate to **Tenant** > **Properties** > **Manage Security defaults** and set **Enable Security defaults** to **No**.

Viewing or modifying the Azure AD tenant type

Use the Active Roles Administration Center to view or modify the tenant type of an existing Azure AD tenant. This is useful if you need to change the default domain settings of an Azure tenant due to an IT or organizational change.

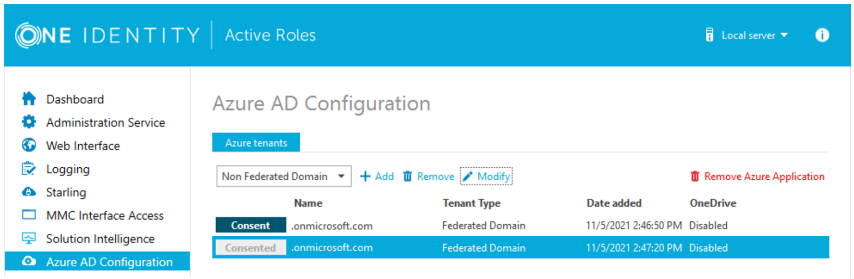
NOTE: Consider the following limitations when modifying the properties of the selected Azure AD tenant:

- If you set the tenant type of an on-premises or hybrid Azure AD to **Federated Domain** or **Synchronized Identity Domain**, then the **Azure properties** fields of the objects (Azure users, Azure guest users, groups and contacts) in the Azure tenant will be disabled and cannot be edited in the Active Roles Web Interface.
- You cannot modify the tenant ID and the authentication settings of the Azure AD tenant.

To view or modify the Azure AD tenant properties

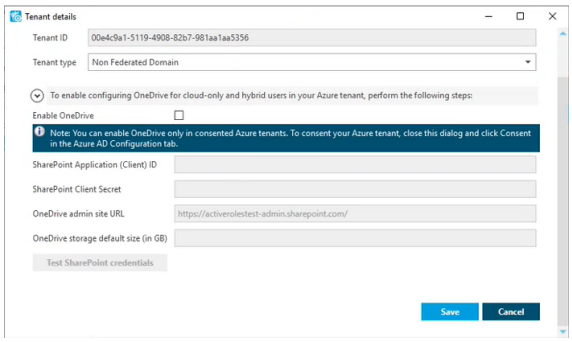
1. In the Active Roles Configuration Center, on the left pane, click **Azure AD Configuration**.

   The list of existing Azure AD tenants appears.



2. Select the Azure AD tenant you want to view or modify, then click **Modify**.

   The **Tenant details** window appears.



3. (Optional) To change the domain type of the Azure tenant, select the applicable type from the **Tenant type** drop-down list.
   - **Non-Federated Domain**: When selected, on-premises domains are not registered in Azure AD, and Azure AD Connect is not configured. Azure users and Azure guest users are typically created with the `onmicrosoft.com` UPN suffix.
   - **Federated Domain**: On-premises domains are registered in Azure AD and Azure AD Connect. Also, Active Directory Federation Services (ADFS) is configured. Azure users and Azure guest users are typically created with the

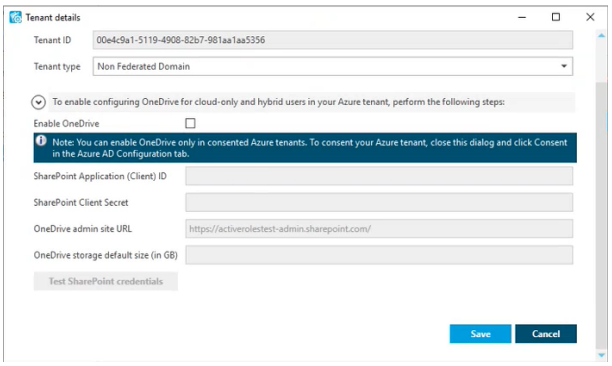
UPN suffix of the selected on-premises domain.

- **Synchronized Identity Domain**: On-premises domains may or may not be registered in Azure AD. Azure AD Connect is configured. Azure users and Azure guest users can be created either with the selected on-premises domain, or with the `onmicrosoft.com` UPN suffix.

4. (Optional) To enable, disable or modify the provisioned OneDrive storage of the Azure tenant, select or deselect **Enable OneDrive**, and (when selected), configure the SharePoint and OneDrive settings listed in the **Tenant details** window. For more information on configuring OneDrive storage in an Azure tenant, see Enabling OneDrive in an Azure tenant.
5. To close the **Tenant details** window without any changes, click **Cancel**. To apply your changes, click **Save**.

Enabling OneDrive in an Azure tenant

You can enable OneDrive in your consented Azure tenant(s) for cloud-only and hybrid Azure users in the **Azure AD Configuration** > **Tenant details** window of the Active Roles Configuration Center.



To enable OneDrive in an Azure tenant, you must:

1. Configure a SharePoint App-Only for authentication.
2. Specify the required application permissions for the configured SharePoint App-Only.
3. Specify the SharePoint admin site URL of your Azure tenant.
4. Configure the default size of the OneDrive storage provisioned for Azure users in the Azure tenant.

For the detailed procedure, see Configuring OneDrive for an Azure tenant.

NOTE: Once OneDrive is enabled, consider the following limitations:

- Active Roles supports creating OneDrive storage for new cloud-only and hybrid Azure users only if OneDrive is preprovisioned in your organization. For more information, see Pre-provision OneDrive for users in your organization in the official Microsoft documentation.

- When creating new cloud-only Azure users with OneDrive storage in the Active Roles Web Interface, make sure that the **General** > **Allow user to sign in and access services** setting is selected. Otherwise, Active Roles will not provision and create the OneDrive storage of the new Azure user. For more information on creating a new cloud-only Azure user in the Active Roles Web Interface, see Creating a new cloud-only Azure user.
- The **OneDrive admin site URL** and **OneDrive storage default size (in GB)** settings of the **Tenant details** window are applicable to cloud-only Azure users only, and do not affect OneDrive provisioning for hybrid users in your Azure tenant. To configure the OneDrive admin site URL and the default OneDrive storage size for hybrid users, you must set these settings in the Active Roles Console (also known as the MMC Interface) by configuring an **O365 and Azure Tenant Selection** policy for your Azure tenant, after configuring OneDrive in the Active Roles Configuration Center. For more information, see Configuring an O365 and Azure Tenant Selection policy.

Prerequisites of enabling OneDrive in an Azure tenant

Before configuring OneDrive for an Azure tenant in the Active Roles Configuration Center, make sure that the Azure tenant meets the following conditions:

- The Azure tenant is already consented. Attempting to enable OneDrive in an Azure tenant for which Active Roles was not consented as an Azure application will result in an error when testing the configured SharePoint credentials. For more information on consenting an Azure tenant, see Configuring a new Azure tenant and consenting Active Roles as an Azure application.
- The Azure tenant has the **Sites.FullControl.All** SharePoint application permission. Active Roles automatically configures this permission when consenting Active Roles as an Azure application for a newly-configured Azure tenant.

  However, if the Azure tenant for which you want to enable OneDrive has already been used in an Active Roles version earlier than Active Roles 7.5, you must add the **Sites.FullControl.All** SharePoint application permission manually for Active Roles in the Azure tenant. Failure of doing so will result in an error in the **Tenant Details** window of the Active Roles Configuration Center when testing the configured SharePoint credentials.

  For more information, see Checking and adding the Sites.FullControl.All permission for Active Roles.

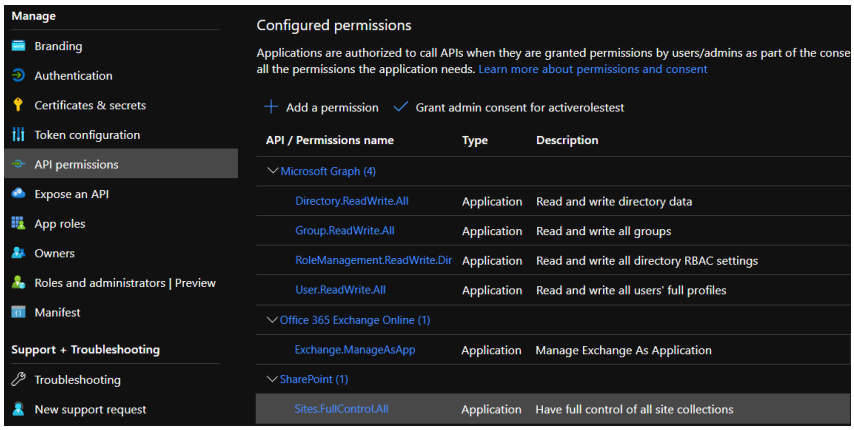
Checking and adding the `Sites.FullControl.All` permission for Active Roles

If the Azure tenant for which you want to enable OneDrive has already been used in an Active Roles version earlier than Active Roles 7.5, you must add the **Sites.FullControl.All** SharePoint application permission manually for Active Roles in the Azure tenant. Failure of doing so will result in an error in the **Tenant Details** window of the Active Roles Configuration Center when testing the configured SharePoint credentials.

To check that Active Roles has the Sites.FullControl.All application permission in an Azure tenant

1. Log in to Azure Portal.
2. Open the Azure tenant of your organization by clicking **Azure Active Directory** on the main screen.
3. To open the list of applications registered for your Azure tenant, navigate to **Manage** > **App registrations**.
4. Select your Active Roles deployment either by finding it in the **All applications** or **Owned applications** list, or by searching it in the search bar.
5. To open the list of API permissions, navigate to **Manage** > **API permissions**.
6. Check that the **Sites.FullControl.All** permission is listed under the **API / Permissions** name > **SharePoint** heading.

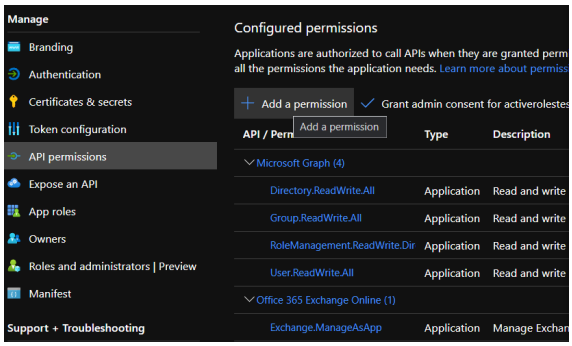
Figure 109: List of configured permissions under Azure Active Directory > Manage > API Permissions of Azure Portal



If **Sites.FullControl.All** is not listed, add it to Active Roles in the Azure tenant by completing the next procedure.

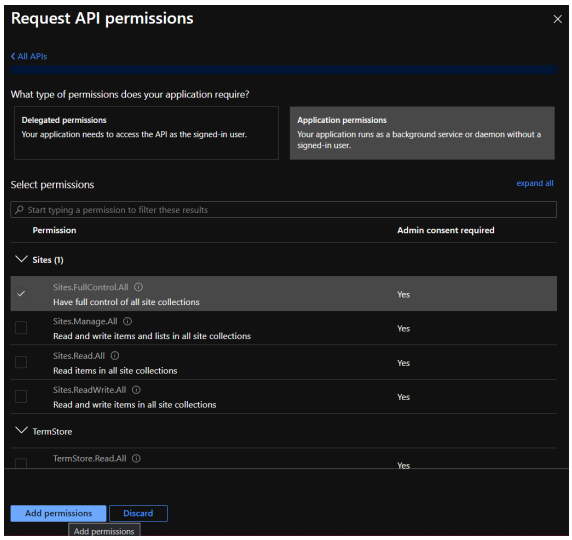
To add the Sites.FullControl.All application permission to Active Roles in an Azure tenant

1. In the **Configured permissions** list (available under **Manage** > **API permissions**) click **Add a permission**.

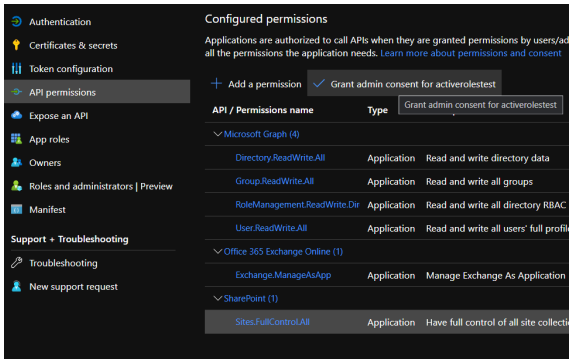


The list of available API permissions will appear on the right side of the screen under **Request API permissions**.

2. In the list of available API permissions, click **SharePoint**.
3. Click **Application permissions**.
4. Under **Select permissions** > **Sites**, select **Sites.FullControl.All** and click **Add permissions**.



5. To apply your changes, select **Sites.FullControl.All** under **Configured permissions** and click **Grant admin consent for <azure-tenant-name>**.



Configuring OneDrive for an Azure tenant

Use the **Azure AD Configuration** > **Modify** (**Tenant details**) window of the Active Roles Configuration Center to enable OneDrive storage for the cloud-only and hybrid users of your selected Azure tenant.

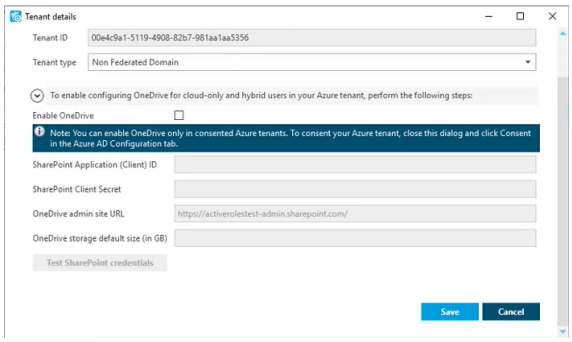
Prerequisites

Before beginning the configuration, make sure that the selected Azure tenant meets the requirements listed in Prerequisites of enabling OneDrive in an Azure tenant.

To enable OneDrive storage for Azure users in an Azure tenant

1. In the Active Roles Configuration Center, click **Azure AD Configuration**.
2. Select the Azure tenant for which you want to enable OneDrive storage, and click **Modify**. The **Tenant details** window appears.

   Figure 110: Active Roles Configuration Center > Azure AD Configuration > Modify

3. To start the configuration of the OneDrive storage, select **Enable OneDrive**.
4. To register Active Roles as a SharePoint App-Only for OneDrive authentication, open the SharePoint App-Only configuration site of your Azure tenant in your web browser:

   `<azure-tenant-name>.sharepoint.com/_layouts/15/appregnew.aspx`

   > TIP: To quickly open the SharePoint App-Only configuration site from the **Tenant details** window, expand the procedure overview above **Enable OneDrive** to access a clickable link.

5. On the SharePoint App-Only configuration site, configure the following settings:
   - **Client ID**: Generate a new client ID.
   - **Client Secret**: Generate a new client secret.
   - **Title**: Provide a name for the configuration (for example, `Active Roles SharePoint app`).
   - **App Domain**: Specify a custom application domain for the configuration.

> NOTE: Make sure that the specified **App Domain** is not a reserved domain (such as the domain of your Azure tenant), otherwise the SharePoint App-Only cannot be created. One Identity recommends specifying `https://www.localhost.com` as **App Domain**.

- **Redirect URI**: Specify a custom redirect URI for the configuration (such as `https://localhost`).

6. To apply your changes and create the SharePoint App-Only, click **Create**. Upon successful configuration, the SharePoint App-Only configuration site displays the configured settings with the following message:

```
The app identifier has been successfully created.
```

7. Copy the **Client ID** and **Client Secret** values to your clipboard or elsewhere, as they will be required for the next step.
8. Grant the required permissions for the configured SharePoint App-Only. To do so, open the application invitation page of the SharePoint administration site of your Azure tenant in your web browser with a Global Administrator user:

   `<azure-tenant-name>-admin.sharepoint.com/_layouts/15/appinv.aspx`

   > TIP: To quickly open the SharePoint administration site from the **Tenant details** window, expand the procedure overview above **Enable OneDrive** to access a clickable link.

9. On the SharePoint administration site, configure the following settings:
   - **App ID**: Paste the client ID generated on the SharePoint App-Only configuration site here.

     > TIP: To quickly fill the **Title**, **App Domain** and **Redirect URL** fields, click **Lookup** after pasting the client ID into the **App ID** field.

   - **Title**: Provide the name that you specified for the configuration on the SharePoint App-Only configuration site.
   - **App Domain**: Specify the custom application domain that you specified on the SharePoint App-Only configuration site.
   - **Redirect URL**: Specify the custom redirect URI that you specified on the SharePoint App-Only configuration site.
   - **Permission Request XML**: Paste the following XML code into the text box:

```
<AppPermissionRequests AllowAppOnlyPolicy="true">
  <AppPermissionRequest Scope="http://sharepoint/content/tenant"
Right="FullControl" />
  <AppPermissionRequest Scope="http://sharepoint/social/tenant"
Right="FullControl" />
</AppPermissionRequests>
```

10. To apply your changes and grant the application permissions, click **Create**.

11. In the **Tenant details** window of the Active Roles Configuration Center, configure the following settings:
    - **SharePoint Application (Client) ID**: Paste the client ID generated on the SharePoint App-Only configuration site.
    - **SharePoint Client Secret**: Paste the client secret generated on the SharePoint App-Only configuration site.
    - **OneDrive admin site URL**: Specify the URL of the SharePoint administration site of your Azure tenant. The URL has the following syntax: `<azure-tenant-name>-admin.sharepoint.com`
    - **OneDrive storage default size (in GB)**: Specify the default OneDrive storage size allocated for each Azure user in the Azure tenant. This field accepts only an integer and its value must be within the range of the storage size allowed by the OneDrive subscription in use within your organization.

      NOTE: The **OneDrive admin site URL** and **OneDrive storage default size (in GB)** settings of the **Tenant details** window are applicable to cloud-only Azure users only, and do not affect OneDrive provisioning for hybrid users in your Azure tenant. To configure the OneDrive admin site URL and the default OneDrive storage size for hybrid users, you must set these settings in the Active Roles Console (also known as the MMC Interface) by configuring an **O365 and Azure Tenant Selection** policy for your Azure tenant, after configuring OneDrive in the Active Roles Configuration Center. For more information, see Configuring an O365 and Azure Tenant Selection policy.
12. To check the SharePoint authentication configuration, click **Test credentials**.

    TIP: If the test fails for any reason, Active Roles indicates it with an error message. Typically, testing can fail for the following reasons:
    - The specified client ID and/or client secret is incorrect. To resolve the problem, double-check that they were copied correctly from the SharePoint App-Only configuration site.
    - The required application permissions were not granted in the SharePoint administration site of your Azure tenant. To resolve the problem, open the application invitation page of the SharePoint administration site of your Azure tenant, and copy the permission request XML code indicated in this procedure.
    - The Azure tenant is not consented. To resolve the problem, make sure that the Azure tenant is consented. For more information, see Configuring a new Azure tenant and consenting Active Roles as an Azure application.
    - If the Azure tenant for which you configure OneDrive has already been used in Active Roles versions earlier than 7.5, then the Azure tenant may not have the **Sites.FullControl.All** SharePoint permission granted. To resolve the problem, verify that the **Sites.FullControl.All** permission is granted for the Azure tenant. For more information, see Checking and adding the Sites.FullControl.All permission for Active Roles.

- The specified **OneDrive admin URL** is incorrect. To resolve the problem, double-check that the specified admin URL is correct and belongs to the Azure tenant for which OneDrive is configured.
- The specified **OneDrive storage default size** is incorrect (that is, the field is left empty, does not contain a numeric value, or the specified value is outside the storage size range available by the Microsoft 365 plan of your organization). To resolve the problem, specify a valid storage size.
- A problem occurred in your internet connection. To resolve the problem, check your internet connection and try again.

13. Once testing completed successfully, to apply your settings, click **Save**.

    NOTE: You can save the OneDrive configuration only if the test completes successfully.

14. (Optional) If you want to provision OneDrive storage for hybrid Azure users as well in your Azure tenant, then set up a new **O365 and Azure Tenant Selection** policy in the Active Roles Console (also known as the MMC Interface). For more information, see Configuring an O365 and Azure Tenant Selection policy.

NOTE: When creating a new hybrid or cloud-only Azure user in the Active Roles Web Interface after completing this procedure, make sure that you grant them the **SharePoint Online** license in the **Licenses** step. Otherwise, the configured OneDrive storage cannot be provisioned for the new Azure user. For more information, see Creating a new cloud-only Azure user.

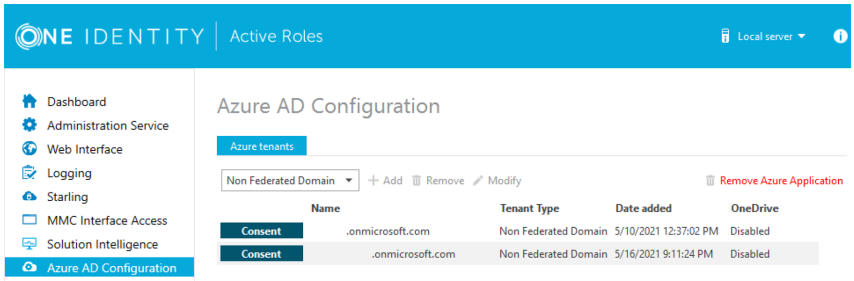
Removing an Azure AD tenant

You can use the Active Roles Configuration Center to delete an Azure AD tenant. This is typically required when an Azure tenant and its directory objects become obsolete because of organizational reasons.

To remove an Azure AD tenant

1. In the Active Roles Configuration Center, on the left pane, click **Azure AD Configuration**.

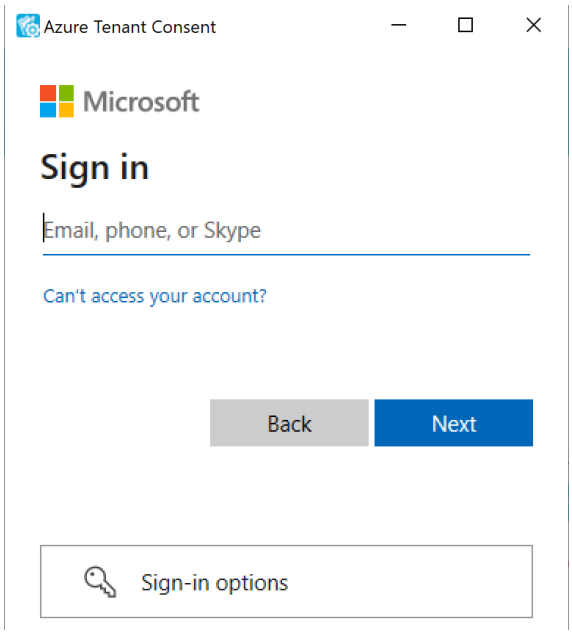
   The list of existing Azure tenants appears.



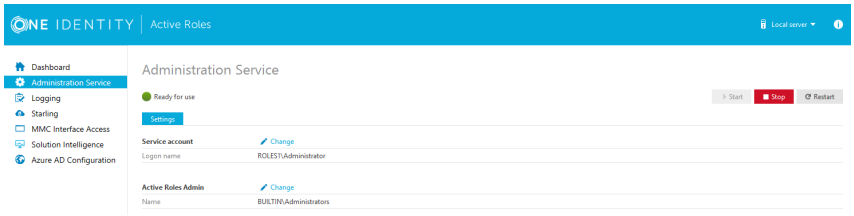
2. On the **Azure AD Configuration** page, from the list of Azure tenants, select the tenant that you want to remove.

3. Click **Remove**.

   The standard Microsoft Azure **Select Account** or **Sign In** pop-up appears. Log in with the user name and password of your Azure AD administrator account.



4. The Azure tenant and all the related domains and applications are then deleted upon successful login.
5. To apply the changes, you must restart the Administration Service, as indicated on the user interface. Click **Administration Service** on the left pane, then either click **Restart**, or first click **Stop** and then **Start**.



6. (Optional) If you want to force the deletion of the Active Roles Azure application on the Azure Portal for the removed Azure tenant, click **Remove Azure Application** and log in with the credentials of the removed Azure tenant.

   This is typically recommended as an extra housekeeping and security measure if the removed Azure tenant has been previously managed either in earlier Active Roles versions or on other machines as well, but the Azure tenant has not been removed from those Active Roles installations prior to uninstalling them (leaving their client secret intact on the Azure Portal).

CAUTION: Using the Remove Azure Application option will result in all Active Roles installations losing access to the specified Azure tenant. If this happens, users managing the Azure tenant in another Active Roles installation (for example, on another machine) can regain access to the Azure tenant if they:

1. **Remove the Azure tenant in the Azure AD Configuration tab of their Active Roles Configuration Center.**
2. **Add the Azure tenant again, as described in Configuring a new Azure tenant and consenting Active Roles as an Azure application.**

7. To confirm removal, check if the removed Azure tenant has disappeared from the list of Azure tenants in the **Azure AD Configuration** page of the Active Roles Configuration Center, and from the **Directory Management** > **Tree** > **Azure** node of the Active Roles Web Interface.

View Azure Health for Azure AD tenants and applications

Azure Health Check informs you about the Active Roles to Azure AD connectivity status, and the Active Roles Azure AD tenant and application heath status.

To view the Azure AD health status in Active Roles

1. On the Active Roles Web interface **Navigation** bar, click **Directory Management**.
2. On the **Views** tab in the **Browse** pane, click **Azure | Azure Configuration | Azure Health Check**. Select the Tenant name from the **Tenant** list drop down for which you want to view the Azure health status.

   The health status for the following services and resources is displayed:

   - **Graph Connectivity** – Green status indicates that the Active Roles connectivity to the Microsoft Graph API is successful
   - **Tenant Connectivity** – The tenant username and password are validated. Green status indicates that the Azure AD Tenant credentials are valid. The tenant connectivity is successful only if the Graph connectivity is successful
   - **Azure Application Connectivity** – The Azure AD applications are validated and verified if the applications are consented. Green status indicates that the Azure AD applications connectivity is successful. The application connectivity is successful only if both the Graph connectivity and tenant connectivity are successful.

View Azure Licenses Report

Azure Licenses Report displays the Office 365 licenses that are available and assigned to a user.

To view the Azure AD licenses report

1. On the Active Roles Web interface **Navigation** bar, click **Directory Management**.
2. On the **Views** tab in the **Browse** pane, click **Azure | Azure Configuration | Azure Licenses Report**. Select the Tenant name from the **Tenant** list drop down for which you want to view the Azure License Report

   The Azure Licenses Report wizard displays the list of Office 365 licenses available for the Azure AD domain. For each license the following information is displayed:

   - **Valid** – The total number of a specific license available for the Azure AD domain.
   - **Expired** – The number of licenses of a specific license type that are in renewal period or have expired.
   - **Assigned** – The number of licenses of a specific license type that have been assigned to any users in the domain.

View Office 365 Roles Report

Office 365 Roles Report displays the Office 365 roles that are available and assigned to a user.

To view the Office 365 roles report

1. On the Active Roles Web interface **Navigation** bar, click **Directory Management**.
2. On the **Views** tab in the **Browse** pane, click **Azure | Azure Configuration** | **Office 365 Roles Report**. Select the Tenant name from the **Tenant** list drop down for which you want to view the Office 365 roles report.

   The Office 365 Roles Report wizard displays the list of Office 365 roles and the users assigned with the roles for the Azure AD domain.

Azure Tenant Association

After the import or upgrade operation, a scheduled task is created for tenant association. This scheduled task runs at a specified date and time or you can also choose to run the task immediately to associate azure objects with the selected tenant.

The task can be scheduled to run at a specific time. To schedule the task, navigate to **Configuration | Server Configuration | Scheduled Tasks | Builtin container** in the Active Roles console.

After successfully completing the task, a log file is found in the Configuration Center logs folder with the name **SyncAssociatedTenantId.log**.

NOTE: Alternatively, Azure Tenant association can be run at any time using the template workflow **Update Azure Objects Associated Tenant Id** available in the Built-in Workflow Container. The parameter in the script used by the workflow can be configured with the required tenant ID. You can use the drop-down to select a default Azure Tenant from the list of available Azure Tenants. The script used by the workflow can be modified to Search Azure objects based on the requirement.

Configuring Active Roles to manage Hybrid AD using Management Shell

Active Roles Management Shell enables you to perform the following configuration tasks to manage Hybrid AD:

- Adding an Azure AD tenant
- Add an Azure AD Application

Adding an Azure AD tenant

Use the Active Roles Management Shell to add an Azure AD tenant. To do so, run the `New-QADAzureConfigObject` cmdlet on the Management Shell interface.

Description

`New-QADAzureConfigObject` lets you add an Azure AD tenant to Active Directory.

Usage Recommendations

Use `New-QADAzureConfigObject` to add an Azure AD tenant using the tenant ID provided by Microsoft for the default tenant (created at the time of the Microsoft Azure subscription).

Syntax

```
New-QADAzureConfigObject -type 'AzureTenant' -name 'Azuretenantname' -AzureTenantId
'AzureTenantGUID' -AzureTenantDescription 'AzureTenantDescription' -AzureAdminUserID
'AzureGlobalAdminUserID' -AzureAdminPassword 'AzureGlobalIDPassword' -
AzureADTenantType 'AzureTenantType'
```

Parameters

The `New-QADAzureConfigObject` cmdlet has the following parameters.

- `type (string)`: Specifies the object class of the directory object to be created (such as User or Group). The cmdlet creates a directory object of the object class specified with this parameter.

Table 96: Parameter: type (string)

Required	true
Position	named
Accepts pipeline input	false
Accepts wildcard characters	false

- `name (string)`: Sets the `name` attribute to the value of this parameter on the new object created by **`New-QADAzureConfigObject`** in the directory.

Table 97: Parameter: name (string)

Required	true
Position	named
Accepts pipeline input	false
Accepts wildcard characters	false

- `AzureTenantId (string)`: Specifies the Azure AD tenant ID obtained from the default tenant (created after subscribing to Microsoft Azure).

  NOTE: The Azure AD ID value configured for this parameter must match the tenant ID configured on the Azure AD side. Otherwise, attempts to create an Azure AD application or manage Azure AD objects will fail.

Table 98: Parameters: AzureTenantId (string)

Required	true
Position	named
Accepts pipeline input	false
Accepts wildcard characters	false

- `AzureTenantDescription`: Specifies the required description of the Azure AD tenant.

Table 99: AzureTenantDescription

Required	false
Position	named
Accepts pipeline input	false
Accepts wildcard characters	false

- `AzureAdminUserID`: Specifies the administrative user name for Microsoft Azure AD.

  NOTE: The administrative user must have the required privileges (for example, License Administrator, User Administrator or Groups Administrator roles) to perform license management or Azure user, guest user, and group management.

  For more information on the available privileges and for an overview of the various Azure and Azure AD administrative roles, see Azure AD built-in roles and Classic subscription administrator roles, Azure roles, and Azure AD roles in the official Microsoft documentation.

Table 100: Parameters: AzureAdminUserID

Required	true
Position	named
Accepts pipeline input	false
Accepts wildcard characters	false

- `AzureAdminPassword`: Specifies the administrative user password for Microsoft Azure AD.

Table 101: Parameters: AzureAdminPassword

Required	true
Position	named
Accepts pipeline input	false
Accepts wildcard characters	false

- `AzureADTenantType`: Specifies the Azure AD tenant type (Federated, Non-Federated, or Synchronized Identity).

  NOTE: Make sure that you select the tenant type corresponding to your organization environment.

Table 102: Parameters: AzureADTenantType

Required	true
Position	named
Accepts pipeline input	false
Accepts wildcard characters	false
Accepts value	• Federated • NonFederated • SynchronizedIdentity

Examples

See the following use cases for examples on how to use this cmdlet.

Example: Creating a new Azure AD tenant with a local user

To create a new Azure AD tenant with a locally logged on user

1. Connect to any available domain controller with the credentials of your local user.
2. Create a new Azure AD tenant with the following `New-QADAzureConfigObject` cmdlet:

```
C:\PS> New-QADAzureConfigObject -type 'AzureTenant' -name
'CompanyAzuretenant' -AzureTenantId 'CompanyAzureTenantID' -
AzureTenantDescription 'Azure tenant for Company' -AzureAdminUserID
'AzureAdminUser1' -AzureAdminPassword 'AzureAdminPassword1’ -
AzureADTenantType 'AzureTenantType'
```

Example: Creating a new Azure AD tenant with a specific user and then disconnecting

To create a new Azure AD tenant with a specific user and then disconnect

1. Connect to any available domain controller:

```
C:\PS> $pw = read-host "Enter password" -AsSecureString
```

2. Connect to the local Administration Service with a specific user of your choice:

```
C:\PS> connect-qadService -service 'localhost' -proxy -ConnectionAccount
'company\administrator' -ConnectionPassword $pw
```

3. Create the new Azure AD tenant:

```
C:\PS> New-QADAzureConfigObject -type 'AzureTenant' -name
'CompanyAzuretenant' -AzureTenantId 'CompanyAzureTenantID' -
AzureTenantDescription 'Azure tenant for Company' -AzureAdminUserID
'AzureAdminUser1' -AzureAdminPassword 'AzureAdminPassword1’ -
AzureADTenantType 'AzureTenantType'
```

4. Once the Azure AD tenant is created, disconnect your user:

```
C:\PS> disconnect-qadService
```

Add an Azure AD Application

You can use the Active Roles Management Shell to add an Azure AD application to the Azure AD tenant.

To add an Azure AD application

On the Management Shell interface, run the **New-QADConfigObject** cmdlet.

Synopsis

This cmdlet enables you to add an Azure AD application to the Azure AD tenant.

Syntax

```
New-QADAzureConfigObject -type 'AzureApplication' -name 'AzureApplication' -
DisplayName 'ApplicationDisplayName' -AzureTenantId 'AzureTenantGUID' -
AzureAppPermissions 'ApplicationPermission'
```

Description

Use this cmdlet to add an Azure AD application.

Parameters

- type (string)

  Use this parameter to specify the object class of the directory object to be created. This is the name of a schema class object, such as User or Group. The cmdlet creates

a directory object of the object class specified by the value of this parameter.

Table 103: Parameters: type (string)

Required	true
Position	named
Accepts pipeline input	false
Accepts wildcard characters	false

- name (string)

  Use this parameter to set the 'name' attribute to this parameter value on the new object created by this cmdlet in the directory.

  Table 104: Parameters: name (string)

Required	true
Position	named
Accepts pipeline input	false
Accepts wildcard characters	false

- AzureTenantId (string)

  Use this parameter to enter the Azure AD tenant ID obtained from the default tenant created after subscribing for Microsoft Azure.

  > NOTE: The values entered for configuring Azure AD tenant must exactly match the values configured for Azure AD, else Azure AD application creation and management of Azure AD objects fail.

  Table 105: Parameters: AzureTenantId (string)

Required	true
Position	named
Accepts pipeline input	false
Accepts wildcard characters	false

- Displayname (string)

  Use this parameter to specify the 'displayName' attribute to this parameter value.

Table 106: Parameters: Displayname (string)

Required	false
Position	named
Accepts pipeline input	false
Accepts wildcard characters	false

- AzureAppPermissions

  Use this parameter to specify the permission scope for applications for Azure AD.

  Table 107: Parameters:AzureAppPermissions

Required	true
Position	named
Accepts pipeline input	false
Accepts wildcard characters	false

- AzureApplicationDescription

  Use this parameter to specify the description of the Azure AD application.

  Table 108: Parameters: AzureApplicationDescription

Required	false
Position	named
Accepts pipeline input	false
Accepts wildcard characters	false

Example

Connect to any available domain controller with the credentials of the locally logged on user, and create a new Azure AD application:

```
C:\PS> New-QADAzureConfigObject -type 'AzureApplication' -name
'AzureApplication' -DisplayName 'ApplicationDisplayName' -AzureTenantId
'AzureTenantGUID' -AzureAppPermissions 'ApplicationPermission'
```

Example

Connect to the local Administration Service with the credentials of a specific user, create a new Azure AD tenant and then disconnect:

```
C:\PS> $pw = read-host "Enter password" -AsSecureString
```

```
C:\PS> connect-qadService -service 'localhost' -proxy -ConnectionAccount
'company\administrator' -ConnectionPassword $pw
C:\PS> New-QADAzureConfigObject -type 'AzureApplication' -name
'AzureApplication' -DisplayName 'ApplicationDisplayName' -AzureTenantId
'AzureTenantGUID' -AzureAppPermissions 'ApplicationPermission'
C:\PS> disconnect-qadService
```

Active Roles Configuration steps to manage Hybrid AD objects

To configure Active Roles to manage Hybrid AD objects, perform the following tasks:

1. Create an Azure AD tenant.
2. Create the Azure AD application.
3. Provide the administrator consent for the Azure AD application.
4. Enforce the **Built-in Policy - Azure - Default Rules to Generate Properties** Policy Object to the on-premises Active Directory containers, which are synchronized to Azure AD.

NOTE:

- After an upgrade the **edsvaAzureOffice365Enabled** is not available for viewing or editing from **Organizational Unit | Advanced Properties** or through the management shell command-let, however the organizational unit container continues to be an Azure enabled container as the azure policy is already applied.

For more information on Azure custom policies, see Changes to Azure O365 Policies in Active Roles after 7.4.1.

Configuring the Azure - Default Rules to Generate Properties policy

If you want to manage hybrid Azure objects (such as hybrid Azure users) in your Organization Unit (OU), then use the built-in **Azure - Default Rules to Generate Properties** Policy Object of the Active Roles Console (also known as the MMC Interface) to provision the default properties and accepted values or hybrid objects.

To configure the built-in Azure - Default Rules to Generate Properties policy

1. In the Active Roles Console, navigate to **Configuration** > **Policies** > **Administration** > **BuiltIn**.

2. Right-click on **Built-in Policy - Azure - Default Rules to Generate Properties** and click **Policy Scope**.
3. To open the **Select Objects** dialog for specifying the OU for provisioning, click **Add**.
4. To specify the OU for provisioning hybrid Azure users, click **Add**, browse the OU you want to provision, and click **Add**.

   TIP: If no elements are displayed in the **Select Objects** dialog, select **Click here to display objects**.
5. To apply the changes and close the dialog, click **OK**.

NOTE: The new provisioning policy settings will be applied automatically only to objects created after configuring the **Azure - Default Rules to Generate Properties** policy object.

To create cloud Azure users for existing on-premises users, you must configure the cloud Azure users manually for each existing on-premises user on the Active Roles Web Interface. To do so:

1. Navigate to the folder of the hybrid users of the OU under **Directory Management** > **Tree** > **Active Directory** > **<your-AD-folder>** > **<your-OU-folder>**.
2. Select the on-premises user for which you want to create a cloud Azure user.
3. To open the **New Azure User** dialog, on the right pane, click **Create Azure User**. For more information on the steps of creating a new cloud Azure user, see Creating a new cloud-only Azure user.

Active Roles Configuration to synchronize existing Azure AD objects to Active Roles

In any hybrid environment, on-premises Active Directory objects are synchronized to Azure AD using Azure AD Connect. When Active Roles is deployed in such a hybrid environment, the existing users and groups' information, such as Azure objectID, must be synchronized back from Azure AD to on-premises AD to continue using the functionality. To synchronize existing AD users and groups from Azure AD to Active Roles we must use the back-synchronization operation.

In Federated or synchronized Identity environment, while creating objects like users, groups, or contacts they are created in on-premise and then synchronized from on-premise to Azure using AAD Connect. Backsync operation is performed to obtain the ObjectID of these objects and update the `edsvaAzureObjectID` in Active Roles to allow further management.

The back-synchronization operation can be performed automatically or manually using the Active Roles Active Roles Synchronization Service Console:

- Automatic Back Synchronization is performed using the **Azure Backsync Configuration** feature in Active Roles Synchronization Service that allows you to configure the backsync operation in Azure with on-premises Active Directory objects

through the Active Roles Synchronization Service Console. After the backsync operation is completed successfully the Azure application registration and the required connections, mappings, and sync workflow steps are created automatically.

For information on configuring the backsync operation automatically using the Active Roles Synchronization Service Console, see Configuring Sync Workflow to back-synchronize Azure AD Objects to Active Roles automatically using the Active Roles Synchronization Service Console.

For more information on the results of the backsync operation see the *One Identity Active Roles Synchronization Service Administration Guide*.

- Manual Back Synchronization is performed by leveraging the existing functionality of Synchronization Service component of Active Roles. Synchronization workflows are configured to identify the Azure AD unique users or groups and map them to the on-premises AD users or groups. After the back-synchronization operation is completed, Active Roles displays the configured Azure attributes for the synchronized objects.

  For information on configuring Synchronization workflows for Azure AD, see *One Identity Active Roles Synchronization Service Administration Guide*.

Configuring Sync Workflow to back-synchronize Azure AD Objects to Active Roles automatically using the Active Roles Synchronization Service Console

Pre-requisites to configure the back-synchronization:

- The hybrid environment must have Azure AD Connect installed and configured.
- The user account used to perform back sync configuration must have the following privileges:
  - User Administrator
  - Privileged Role Administrator
  - Exchange Administrator
  - Application Administrator
- The Windows Azure Active Directory (Azure AD) module version 2.0.0.131 or later must be installed for the backsync feature to work successfully.
- Directory Writers Role must be enabled in Azure Active Directory. To enable the role use the following script:

```
$psCred=Get-Credential

Connect-AzureAD -Credential $psCred

$roleTemplate = Get-AzureADDirectoryRoleTemplate | ? { $_.DisplayName -eq
"Directory Writers" }

# Enable an instance of the DirectoryRole template
```

```
Enable-AzureADDirectoryRole -RoleTemplateId $roleTemplate.ObjectId
```

- For the back-synchronization to work as expected, the user in ARS must have write permissions for `edsvaAzureOffice365Enabled`, `edsaAzureContactObjectId` and `edsvaAzureObjectID`. The user must also have a local administrator privileges where the ARS synchronization service is running.

To configure Azure backsync in Active Roles Synchronization Service

1. In the upper right corner of the Synchronization Service Administration Console, select **Settings** | **Configure Azure BackSync**.

   The Configure BackSync operation in Azure with on-premises Active Directory objects dialog box is displayed.

2. In the dialog box that opens:

   a. Enter the Azure domain valid Account ID credentials, and click **Test Office 365 Connection**.

   b. Specify whether you want to use a proxy server for the connection. You can select one of the following options:

   - **Use WinHTTP settings**: Causes the connector to use the proxy server settings configured for Windows HTTP Services (WinHTTP).
   - **Automatically detect**: Automatically detects and uses proxy server settings.
   - **Do not use proxy settings**: Specifies to not use proxy server for the connection.

   On successful validation, the success message that the Office 365 Connection settings are valid is displayed.

   c. Enter the valid Active Roles account details and click **Test Active Roles Connection**.

   On successful validation the success message that the Active Roles connection settings are valid is displayed.

3. Click **Configure BackSync**.

   The Azure App registration is done automatically. The required connections, mappings, and workflow steps are created automatically.

   On successful configuration the success message is displayed.

   If the Azure BackSync settings are already configured in the system, a warning message is displayed to confirm if you want to override the existing backsync settings with the new settings. If yes, click **Override BackSync Settings**. Else, click **Cancel** to retain the existing settings.

Configuring Sync Workflow to back-synchronize Azure AD Objects to Active Roles manually

Prerequisites to configure the back-synchronization manually:

- The hybrid environment must have Azure AD Connect installed and configured.
- Synchronization Service Component must be installed and configured for Active Roles.
- Azure AD configuration and the Administrator Consent for Azure AD application through web interface must be complete.
- Azure AD built-in policy must be enforced for the container where the back-synchronization is performed.
- For the back-synchronization to work as expected, the user in ARS must have write permissions for `edsvaAzureOffice365Enabled`, `edsaAzureContactObjectId`, `edsvaAzureObjectID`, and `edsvaAzureAssociatedTenantId`. The user must also have a local administrator privileges where the ARS synchronization service is running.

To configure sync workflow to back-synchronize users and groups perform the following steps:

Step 1: Create Connection to Azure AD in the hybrid environment

Create a connection to Azure AD using the Azure AD Connector. The configuration requires the Azure domain name, the Client ID of an application in Azure AD, and the Client Key to establish the connection with Azure AD.

To configure an application:

1. Create an Azure Web application (or use any relevant existing Azure Web Application) under the tenant of your Windows Azure Active Directory environment.

   The application must have "Application Permissions" to "read" and "write" directory data in Windows Azure Active Directory.

   > NOTE: Alternatively, to assign the required permissions to the application by running a Windows PowerShell script, see the Creating a Windows Azure Active Directory connection section in the *Synchronization Service Administration Console*.

2. Open the application properties and copy the following:

- Client ID
- Valid key of the application

3. You need to supply the copied client ID and key when creating a new or modifying an existing connection to Windows Azure Active Directory in the *Synchronization Service Administration Console*.

NOTE: The Web Application that is created or is already available for Sync Service Azure AD Connector, is different from the application that is created while configuring Azure AD using Active Roles Web interface. Both the applications must be available for performing back-sync operations.

Step 2: Create Connection to Active Roles in the hybrid environment

Create a connection to Active Roles using the Active Roles Connector. The configuration requires the local domain details and Active Roles version used. Define the scope to select the container from which the objects for synchronization must be selected.

Step 3: Create Sync Work flow

Create a Sync Workflow using the Azure AD and Active Roles connections. Add a Synchronization step to Update Azure User/Group to Active Roles User/Group.

Set the **edsvaAzureAssociatedTenantId** attribute in Active Roles user/group to azure tenant id. If **edsvaAzureAssociatedTenantId** attribute is not configured , an error is logged in the event viewer for each object.

Configure the **Forward Sync Rule** to synchronize the following:

- Azure **ObjectID** property of a user/group to the Active Roles user/group **edsvaAzureObjectID** property.
- Set the **edsvaAzureOffice365Enabled** attribute in Active Roles user/group to **True**.
- Set **edsvaAzureAssociatedTenantId** with Azure Tenant ID.

Step 4: Create Mapping

Create a Mapping Rule which identifies the user/group in Azure AD and on-premises AD uniquely and map the specified properties from Azure AD to Active Roles appropriately.

For example, the property **userprincipalname** can be used to map users between on-premises AD and Azure AD in a federated environment.

NOTE:

- Based on the environment, make sure to create the correct Mapping rule to identify the user or group uniquely. In-correct mapping rule may create duplicate objects and the back-sync operation may not work as expected.
- Initial configuration and execution of back-sync operation for Azure AD users ID is a one-time activity.
- In Federated or Synchronized environments, Azure AD group creation is not supported. The group is created in Active Roles and is synchronized eventually to Azure using Microsoft Native tools, such as AAD Connect. To manage the Azure AD group through Active Roles, you must perform periodic back-synchronization to on-premise AD.
- Sync engine must be configured to synchronize the data back to AD based on the frequency of groups creation.

To configure sync workflow to back-synchronize contacts perform the following steps:

Step 1: Create Connection to Office 365 in the hybrid environment

Create a connection to Office 365 using the Microsoft Office 365 Connector. The configuration requires Microsoft Online Services ID, Password, Proxy server (if required) and Exchange Online services.

NOTE: Back synchronization of contacts uses Microsoft Office 365 Connector to establish connection to Office 365. Back synchronization of users and groups uses the Azure AD Connector to establish connection to Azure AD.

Step 2: Create Connection to Active Roles in the hybrid environment

Create a connection to Active Roles using the Active Roles Connector. The configuration requires the local domain details and Active Roles version used. Define the scope to select the container from which the objects for synchronization must be selected.

Step 3: Create Sync Workflow

Create a Sync Workflow using the Office 365 and Active Roles connections. Add a Synchronization step to Update Office 365 Contacts to Active Roles Contacts. Configure the **Forward Sync Rule** to synchronize the following:

- Azure **ExternalDirectoryObjectId** property of a contact to the Active Roles contact **edsaAzureContactObjectId** property.
- Set the **edsvaAzureOffice365Enabled** attribute in Active Roles contact to **True**.
- Set **edsvaAzureAssociatedTenantId** with Azure Tenant ID.

Step 4: Create Mapping

Create a Mapping Rule, which identifies the contact in Office 365 and on-premises AD uniquely and map the specified properties from Office 365 to Active Roles appropriately.

NOTE:

- Based on the environment, make sure to create the correct Mapping rule to identify the contacts uniquely. In-correct mapping rule may create duplicate objects and the back-sync operation may not work as expected.
- In Federated or Synchronized environments, Office 365 contact creation is not supported. The contact is created in Active Roles and is synchronized eventually to Office 365 using Microsoft Native tools, such as AAD Connect. To manage the Office 365 contact through Active Roles, you must perform periodic back-synchronization to on-premise AD.

Changes to Azure O365 Policies in Active Roles after 7.4.1

Active Roles 7.4.3 introduces support for Azure Multi tenant model. Multiple tenants can be configured on the Web Interface. Using this feature, the Azure objects from multiple tenants can be managed from the web interface.

The previous custom policies related to Azure Roles and licenses, and OneDrive are not valid and the policy evaluation is skipped after an import or upgrade. Active Roles 7.4.3 introduces a new Azure/Office 365 Tenant Management policy that encompasses all the previous Azure related policies such as Azure Roles and Licenses, and OneDrive policies. Configure the latest Azure/Office 365 Tenant Selection policies to proceed further. The Web Interface notifies the user if any older policies are applied on the OU. Deprovisioning policy for Azure license retention is invalid and must be created again and applied. For more information on the new policy, see Office 365 and Azure Tenant Selection.

Managing Hybrid AD Users

The Active Roles web interface enables you to perform administrative tasks such as create, read, update, deprovision, undo-deprovison, and delete Azure AD users in Hybrid environment. You can also perform other operations such as add and remove Azure AD users to Groups and assign Office 365 licenses to users. Some of the user operations can be performed using the Management Shell in addition to the web interface. The following section guides you through the Active Roles web interface and Management Shell to manage Azure AD users.

- Azure AD user management tasks using Web interface
- Hybrid User Management tasks using web interface
- Azure AD user management tasks using Management Shell interface
- Office 365 license management for hybrid environment users

Azure AD user management tasks using Web interface

Active Roles web interface enables you to perform the following management tasks for Azure AD users:

- Create a new Azure AD user
- View or update the Azure AD user properties
- Modify the Azure AD user Manager
- Disable or re-enable an Azure AD user
- Deprovision or undo deprovision of a Azure AD user
- Add or remove a Azure AD user from a group
- View the Change History and User Activity for an Azure AD user
- Delete an Azure AD user

Create a new Azure AD user

You can use the Active Roles Web Interface to create and enable a new Azure AD user. You can also assign Office 365 licenses to the new user.

To create a new Azure AD user

1. On the Active Roles Web interface **Navigation** bar, click **Directory Management**.
2. On the **Views** tab in the **Browse** pane, click **Active Directory**.

   The list of **Active Directory** domains is displayed.
3. Click the domain in which you need to create a new user.
4. In the list of objects displayed, click the required **Container** or the **Organizational Unit**.
5. In the **Command** pane, click **New User**.
6. In the **New User in <OU name>** | **General** wizard, enter the user details such as **First Name**, **Last Name**, **Initials, and User logon name.**
7. Click **Next**.
8. In the **Account** properties wizard, click **Generate** to generate a password for the Account, select the required Account options and then click **Next**.

   Alternatively, you can set the password manually and re-enter in the **Confirm Password** field to confirm the entered password.
9. In the **Create Azure Account** wizard, select the option **Create Azure Account**.

   The Azure AD account details for the new user are generated automatically and populated in the respective fields.

   > NOTE: The **Temporary Password** field is populated with the default password set for the Active Roles user. You can re-set the password for the Azure AD account if required.
10. Select the Tenant name from the Tenant list drop down. From the **User Principal Name** drop-down list, select the AD domain to which you want to associate the Azure AD user.
11. In the **Usage Location** field, enter the two-letter location code of the location where the product will be used.

    > NOTE: The **Usage Location** field is a mandatory field. The licenses cannot be assigned to the product if the product usage location information is not available. The local rules and regulations for usage of the product and services may vary based on the location.
12. Click **Next**.

    The **Licenses** wizard displays the Office 365 licenses, for example the Office 365 Business Essentials and Business Premium licenses, and the number of licenses that are available to assign to the user.
13. Select the check boxes corresponding to the license that needs to be assigned to the user, and click **Next**.

    The **O365 Roles** wizard displays the Office 365 roles, for example the Helpdesk Administrator, Directory Readers, and more.
14. Select the check boxes corresponding to the Office 365 roles that need to be assigned to the user, and click **Finish**.

The licenses assigned can be viewed on the user's **Azure Properties** | **Licenses** wizard.

The Office 365 roles assigned can be viewed on the user's **Azure Properties** | **O365 Roles** wizard.

The results can also be viewed on the Azure portal's Licenses and Directory role tabs.

View or update the Azure AD user properties

For an existing Azure AD user, you can use the Active Roles Web Interface to view or update the properties.

To view or modify the Azure AD user properties

1. On the Active Roles Web interface **Navigation** bar, click **Directory Management**.
2. On the **Tree** tab in the **Browse** pane, click **Active Directory | <*Domain*> |<*Organizational Unit*>**.

   The list of existing AD users are displayed.
3. Select the check box corresponding to the specific Azure AD user for which, you want to view or modify the Azure properties.
4. In the **Command** pane, click **Azure Properties**.

   The Azure Properties wizard for the Azure AD user is displayed.
5. Use the fields in the **Azure Properties** wizard to view or modify the properties of the Azure AD user.
6. After setting all the required properties, click **Save**.

   The modified settings can be viewed on the Azure Portal.

Modify the Azure AD user Manager

For an existing Azure AD user, you can use the Active Roles Web Interface to modify the Azure AD user Manager.

To view or modify the Azure AD user properties

1. On the Active Roles Web interface **Navigation** bar, click **Directory Management**.
2. On the **Views** tab in the **Browse** pane, click **Active Directory**.

   The list of Active Directory domains is displayed.
3. Click the specific domain, Container or the Organizational Unit, and then select the check box corresponding to the specific user for which you want to view or update the Manager information.

4. In the **Command** pane, click **General properties**.

   The **General Properties** dialog box for the user is displayed.

5. Navigate to the **Managed by** tab, and in the **Manager** field, click **Change**.
6. Use the **Select Objects** dialog box, to locate and select the Manger to assign to the user and click **OK**.

   The newly added Manager name is displayed in the **Manager** field.

7. Click **Save**.

   The **Manager ID** field in the **Azure Properties** wizard for the user is populated with the new Manager information.

   NOTE: To verify the changes in Microsoft Azure, go to the Azure Portal and view the Manger ID information for the specific user in the Work Info tab.

Disable or re-enable an Azure AD user

You can use the Active Roles Web Interface to disable a user for logon to Azure. This allows you to disable a previously enabled user in Azure AD while retaining all the Azure settings that were configured for the user. The Azure AD user settings are retained for a disabled account. Hence you can re-enable a disabled user again without having to reconfigure the user.

To disable or re-enable a previously enabled user for Azure

1. On the Active Roles Web interface **Navigation** bar, click **Directory Management**.
2. On the **Views** tab in the **Browse** pane, click **Active Directory**.

   The list of Active Directory domains is displayed.

3. Click the specific domain, Container or the Organizational Unit, and then the specific user to be disabled.
4. In the **Command** pane, click **Disable**.

   The account is disabled and marked with a disabled icon.

5. To enable a disabled account, select the check-box corresponding to the disabled account and in the **Command** pane click **Enable**.

   NOTE: The **Enable** command only appears for a disabled account.

   The account is enabled again.

Deprovision or undo deprovision of a Azure AD user

Active Roles provides the ability to deprovision rather than delete or only disable users. Deprovisioning a user refers to a set of actions that are performed by Active Roles in order to prevent the user from logging on to the network and accessing network resources such as the user's mailbox or home folder.

The Deprovision command on a user updates the account as prescribed by the deprovisioning policies. Active Roles comes with a default policy to automate some commonly-used deprovisioning tasks, and allows the administrator to configure and apply additional policies.

To deprovision a user for Azure

1. On the Active Roles Web interface **Navigation** bar, click **Directory Management**.
2. On the **Views** tab in the **Browse** pane, click **Active Directory**.

   The list of Active Directory domains is displayed.
3. Select the user, and in the **Command** pane, click **Deprovision**.

   A message is displayed prompting you to confirm the account deprovision.
4. Click **Yes**, to continue

   Wait while Active Roles updates the user.

   After the task is completed, a message is displayed that the account is deprovisioned successfully from Active Roles.

To undo deprovision of a user for Azure

1. On the Active Roles Web interface **Navigation** bar, click **Directory Management**.
2. On the **Views** tab in the **Browse** pane, click **Active Directory**.

   The list of Active Directory domains is displayed.
3. Select the user, and in the **Command** pane, click **Undo Deprovisioning**.

   The **Password Options** dialog box is displayed.
4. Select the option to **Leave the Password** unchanged or **Reset** the password, and click **OK**.

Add or remove a Azure AD user from a group

You can use the Active Roles Web Interface to add or remove an existing Azure AD user from a group.

To add an Azure AD user to a group

1. On the Active Roles Web interface **Navigation** bar, click **Directory Management**.
2. On the **Views** tab in the **Browse** pane, click **Active Directory**.

   The list of Active Directory domains is displayed.
3. Click the specific domain, Container or the Organizational Unit, and then the specific user that you want to add to a group.
4. Select the check-box corresponding to the user and in the **Command** pane click **Member Of**.

   The existing Group information for the user is displayed.
5. In the **<User> (objects found)** wizard, click **Add** to add the user to another group.
6. In the **Select Object** wizard, search and select the group to which you want to add the user.
7. In details pane, right-click the user, and then click **Add to a Group**.

   The **<User> (objects found)** wizard displays all the groups to which the account has been added as a member.

To remove an Azure AD user from a group

1. On the Active Roles Web interface **Navigation** bar, click **Directory Management**.
2. On the **Views** tab in the **Browse** pane, click **Active Directory**.

   The list of Active Directory domains is displayed.
3. Click the specific domain, Container or the Organizational Unit, and then the specific user that you want to add to a group.
4. Select the check-box corresponding to the user and in the **Command** pane click **Member Of**.

   The existing Group information for the user is displayed.
5. In the **<User> (objects found)** wizard, select the group from which you want to remove the user and click **Remove**.

   A message prompts you to confirm the action.
6. Click **Yes** to continue.

   The group information is removed from the <User> (objects found) wizard.

View the Change History and User Activity for an Azure AD user

You can use the Active Roles Web Interface to view the Change History and User Activity for an Azure AD user.

To view the Change History and User Activity of an Azure AD user

1. On the Active Roles Web interface **Navigation** bar, click **Directory Management**.
2. On the **Views** tab in the **Browse** pane, click **Active Directory**.

   The list of Active Directory domains is displayed.
3. Click the specific domain, Container or the Organizational Unit, and then the specific user.
4. In the **Command** pane, click **Change History** or **User Activity**.

   Selecting **Change History** displays the information on changes that were made to the user through Active Roles.

   Selecting **User Activity** displays information on management actions that were performed by a given user.

Delete an Azure AD user

You can use the Active Roles Web Interface to delete a user for logon to Azure.

To delete an Azure AD user

1. On the Active Roles Web interface **Navigation** bar, click **Directory Management**.
2. On the **Views** tab in the **Browse** pane, click **Active Directory**.

   The list of Active Directory domains is displayed.
3. Click the specific domain, Container or the Organizational Unit, and then the specific user to be deleted.
4. In the **Command** pane, click **Delete**.

   The account is deleted.

NOTE:

- Deleting a user is an irreversible operation. A new user with the same name as a deleted user does not automatically assume the permissions and memberships of the deleted account. For this reason, it is advisable to disable rather than delete accounts.
- In a hybrid environment, the user must be deleted in the on-premises AD first and then the changes must be synchronized with Azure AD. In case, the user is deleted in Azure AD first, the Active Roles web interface still displays the Azure properties link for the deleted user but with no information. Further modification of the Azure properties for the deleted user will not be valid.
- Only Global Admins can delete Azure users with any roles assigned to them.

Hybrid User Management tasks using web interface

Active Roles web interface enables you to perform the following Hybrid management tasks for hybrid users:

- Create a new Hybrid user using web interface
- Migrate an Exchange on-premise user to a Hybrid user
- View or modify the Exchange Online properties of an Office 365 User
- View the Mail Flow settings of an Office 365 User
- View or modify the Email Address settings for an Office 365 User
- View or modify the MailBox features for an Office 365 User
- View or modify the Mailbox settings for an Office 365 User
- View or Modify the MailBox Delegation settings for an Office 365 User

Create a new Hybrid user using web interface

You can use the Active Roles web interface to create and manage Hybrid users.

To create a new hybrid user

1. On the Active Roles Web interface **Navigation** bar, click **Directory Management**.
2. On the **Views** tab in the **Browse** pane, click **Active Directory**.

   The list of **Active Directory** domains is displayed.
3. Click the domain in which you need to create a new user.
4. In the list of objects displayed, click the required **Container** or the **Organizational Unit**.
5. In the **Command** pane, click **New User**.
6. In the **New User in <***OU name***>** | **General** wizard, enter the user details such as **First Name**, **Last Name**, **Initials, and User logon name.**
7. Click **Next**.
8. In the **Account** properties wizard, click **Generate** to generate a password for the Account, select the required Account options and then click **Next**.

   Alternatively, you can set the password manually and re-enter in the **Confirm Password** field to confirm the entered password.

9. Click Next.

   The Create Mailbox wizard is displayed. The **Create Exchange Mailbox** option is selected by default. This option enables creation of an on-premises exchange mailbox for the hybrid user.

   NOTE: To enable the creation of a remote mailbox for the hybrid user, clear the **Create Exchange Mailbox** check box.

10. In the **Create Azure Account** wizard, select the option **Create Azure Account**.

    The Azure AD account details for the new user are generated automatically and populated in the respective fields.

    NOTE: The **Temporary Password** field is populated with the default password set for the Active Roles user. You can re-set the password for the Azure AD account if required.

11. Select the Tenant name from the **Tenant** list drop down.
12. From the **User Principal Name** drop-down list, select the AD domain to which you want to associate the Azure AD user.
13. In the **Usage Location** field, enter the two-letter location code of the location where the product will be used.

    NOTE: The **Usage Location** field is a mandatory field. The licenses cannot be assigned to the product if the product usage location information is not available. The local rules and regulations for usage of the product and services may vary based on the location.

14. Click **Next**.
15. Select the Exchange Online license from the listed subscription in the Licenses wizard and click **Finish**.

    The assigned license can be viewed on the user's **Azure properties** | **Licenses** wizard.

Any license that creates a mailbox on the cloud must be assigned to the user for Remote Mailbox to be created.

NOTE: ARS service account must be a part of **Recipient Management group** to run exchange hybrid commands.

Migrate an Exchange on-premise user to a Hybrid user

An Exchange on-premise user can be converted to a hybrid user by migrating the Exchange on premise mailbox to Exchange Online.

To migrate an Exchange on-premise user in a Synchronized or Federated Environment to a Hybrid user:

1. On the Active Roles Web interface **Navigation** bar, click **Directory Management**.
2. On the **Views** tab in the **Browse** pane, click **Active Directory**.

The list of **Active Directory** domains is displayed.

3. Click the domain in which you need to create a new user.
4. In the list of objects displayed, click the required **Container** or the **Organizational Unit**.
5. Select an existing user with Exchange on premise and click **Azure Properties**.
6. In the **Azure Properties** wizard, select the **Licenses** wizard.
7. Select Exchange Online Plan from the License subscription list and click **Finish**.

   An Exchange online mailbox for the user is created.

   The Exchange Online created is available only after migration.

   Refer the Microsoft link to perform migration from Exchange on premise to Exchange Online mailbox.

   After the migration, the Exchange on premise properties updated in Web interface are synced to Office 365 portal through Microsoft Native Tools and can be viewed using Exchange Online Properties.

NOTE:

- The Exchange online properties on the Web Interface for Synchronized Identity and Federated Environments are editable. However, the Email addressed attribute is disabled, as that is synchronized to Exchange Online through Microsoft Native tools.

View or modify the Exchange Online properties of an Office 365 User

For an existing Office 365 user, you can use the Active Roles Web Interface to view or modify the Exchange Online properties.

NOTE:

- The Exchange online properties on the Web Interface for Synchronized Identity and Federated Environments are editable. However, the **E-mail addresses** attribute is disabled, as that is synchronized to Exchange Online through Microsoft Native tools.
- To manage Exchange or Exchange Online properties in Hybrid Exchange environment, Administrators must set the corresponding properties using the Exchange Properties tab. These properties are eventually synchronized to Exchange Online through Microsoft Native tools.

To view the Exchange Online properties of an Office 365 user

1. On the Active Roles Web interface **Navigation** bar, click **Directory Management**.
2. On the **Views** tab in the **Browse** pane, click **Active Directory**.

   The list of Active Directory domains is displayed.
3. Click the specific domain, Container or the Organizational Unit, and then select the check-box corresponding to the specific user with Exchange Online license for which you want to view the properties.
4. In the **Command** pane, click **Exchange Online Properties**.

   The **Exchange Online Properties** wizard displays the following Exchange Online properties for the Office 365 user.

   - Mail Flow Settings
   - Delegation
   - E-mail Addresses
   - Mailbox Features

     Mailbox Settings
5. Use the tabs in the **Exchange Online Properties** dialog box to view the following Exchange Online properties of the Office 365 user:
   - Mail Flow Settings
     - Message Size restrictions
       - Sending Message size
       - Receiving Message size.
     - Delivery Options
       - Send On behalf
       - Forwarding Address
       - Enabling or disabling of Delivery messages to the forwarding address and mailbox.
   - Delegation
   - E-mail Addresses
   - Mailbox Features
     - Exchange ActiveSync
     - Outlook Web App
     - MAPI
     - IMAP
     - POP3
     - Archive
   - Mailbox Settings
     - Messaging Records management

View the Mail Flow settings of an Office 365 User

For an existing Office 365 user, you can use the **Mail Flow settings** tab in the Exchange Online Properties wizard to view or set the message size restrictions and delivery options.

To view and modify the message size restrictions for an Office 365 user

1. In the Exchange Online Properties wizard of an Office 365 user, click **Mail Flow Settings**.
2. Under Mail flow settings, click **Message Size Restrictions** and then **Properties**.

   The Message Size Restrictions dialog box displays the sending and receiving message size restrictions.
3. To set or modify sending and receiving message size restrictions, select one of the following in the Message Size Restrictions dialog box:
   - Use default limit – Allows you to set the maximum size for the outgoing or incoming messages to the default value used in Exchange Online, which is applied through the built-in policy "**Built-in Policy - Exchange Online - Default Message Size Restrictions**" enforced on the container.
   - Maximum (KB) – Allows you to specify the maximum value for the outgoing or incoming message size.
4. Click **Save**.
5. Close the dialog box and click **Save**.

NOTE: The changes made to message size restrictions settings for the Office 365 user can be verified in the Microsoft Office 365 portal.

View or modify message delivery option for an Office 365 User

To view or modify the message delivery options for an Office 365 user

1. In the Exchange Online Properties wizard of an Office 365 user, click **Mail Flow Settings**.
2. Under Mail flow settings, click **Delivery Options** and then **Properties**.
3. To allow one or more users to send messages on behalf of the Office 365 user, in the Delivery Options dialog box, click **Add**, select one or more users from the **Select Object** list, and then click **OK**.
4. To limit users from sending messages on behalf of the Office 365 user, select the users in the **Name** list and click **Remove**.
5. To specify a forwarding address for messages addressed to the Office 365 user, select **Forward to**, and click **Modify**.

   Alternatively, to change the current forwarding address, click **Modify**.
6. From the Select Object wizard, select the users to whom the messages addressed to the mailbox can be forwarded and click **OK**.

7. Click **Save**.
8. Close the dialog box and click **Save**.

NOTE: The changes made to message delivery options for the Office 365 user can be verified in the Microsoft Office 365 portal.

View or Modify the MailBox Delegation settings for an Office 365 User

For an existing Office 365 user, you can use the MailBox delegation settings tab in the Exchange Online Properties wizard to view or modify other users or groups who can send mails or be provided full access to the user's mailbox.

NOTE: The modify option is applicable for Office 365 users on all environments, such as, Federated, Synchronized and non-federated.

To view or modify the MailBox delegation settings for an Office 365 user

1. In the Exchange Online Properties wizard of an Office 365 user, click **Delegation**.
2. To specify or modify the list of users or groups who can send mail from the Office 365 user’s mailbox, under **Send as**, click **Add**.
3. Select one or more users or groups from the **Select Object** list, and then click **OK**.
4. Alternatively, to limit users who can send emails from the Office 365 user’s mailbox, select the users in the **Name** list and click **Remove**.
5. Click **Properties** to view the general properties of the user added under the **Send as** option.
6. To specify Office 365 users or groups who can be provided full access to the user’s mailbox, under **Full Access**, click **Add**, select one or more users or groups from the **Select Object** list, and then click **OK**.
7. Alternatively, to limit users who can be provided full access to the user’s mailbox, select the users in the **Name** list and click **Remove**.
8. Click **Properties** to view the general properties of the user added under **Full Access** option.
9. Click **Save**.
10. Close the dialog box and click **Save**.

NOTE: The changes made to MailBox delegation settings for the Office 365 user can be verified in the Microsoft Office 365 portal.

View or modify the Email Address settings for an Office 365 User

For an existing Office 365 user, you can use the E-mail Address settings tab in the Exchange Online Properties wizard to view or set the email address settings.

IMPORTANT: The modify option is applicable for Office 365 users on a non-federated environment only.

To view or modify the email address settings for an Azure AD user

1. In the Exchange Online Properties wizard of an Office 365 user, click **E-mail Settings**.
2. To add email addresses, click **Add**.
3. In the E-mail Addresses dialog box, select the email address type, add the email address, and click **OK**.
4. To modify a selected email address, click **Edit**.
5. In the E-mail Addresses dialog box, edit the selected email address, and click **OK**.
6. To delete a selected email address, click **Remove**.
7. Click **Save**.
8. Close the dialog box and click **Save**.

NOTE: The changes made to email address settings for the Office 365 user can be verified in the Microsoft Office 365 portal.

View or modify the MailBox features for an Office 365 User

You can use the Exchange Features tab to manage a variety of mailbox features for the Office 365 mailbox user.

To view or modify the Mailbox features for an Office 365 user

1. In the Exchange Online Properties wizard of an Office 365 user, click **Mailbox Features**.

   The following mailbox features are displayed and can be managed for the Office 365 mailbox user:

   - **Exchange ActiveSync**: Allows the user to access the mailbox from a mobile device.
   - **Outlook Web App**: Allows the user to browse the mailbox with a cell phone or other wireless devices.
   - **MAPI**: Allows the user to access the mailbox from a MAPI client such as Microsoft Outlook.

- **IMAP4**: Allows the user to access the mailbox from an IMAP4 client such as Outlook Express.
- **POP3**: Allows the user to access the mailbox from a POP3 client such as Outlook Express.
- **Archive**: If the mailbox is archive-enabled, you can view or change the archive properties.

2. Under Mailbox Features, select the required feature you want to enable or disable for the Office 365 mailbox user, and then click **Enable** or **Disable** respectively.
3. Click **Save**.
4. Close the dialog box and click **Save**.

NOTE: The changes made to MailBox Features for the Office 365 mailbox user can be verified in the Microsoft Office 365 portal.

View, modify, or specify the Archive MailBox name for an Office 365 User

You can use the Exchange Features tab to view, modify, or specify the archive mailbox name of an archive-enabled Office 365 mailbox user.

To view, modify, or specify the archive mailbox name of an archive-enabled Office 365 mailbox user

1. In the Exchange Online Properties wizard of an Office 365 user, click **Mailbox Features**.
2. Under Mailbox Features, select **Archive**, click **Enable**, and then click **Save**.
3. Click **Properties**.

   The Exchange Online Archive Mailbox wizard is displayed, which allows you to view or modify the archive name derived from the Microsoft Office 365 portal and displayed in the **Specify a name for this archive** field.
4. Click **OK**, close the dialog box, and then click **Save**.
5. To specify a new name to the archive, select **Archive**, click **Enable**, and then click **Properties**.
6. On the Exchange Online Archive Mailbox wizard, in the **Specify a name for this archive** field, provide a name for the archive.
7. Click **OK**, close the dialog box, and then click **Save**.

NOTE: The changes made to MailBox Features for the Office 365 mailbox user can be verified in the Microsoft Office 365 portal.

View or modify the Mailbox settings for an Office 365 User

For an existing Office 365 user, you can use the Mailbox settings tab in the Exchange Online Properties wizard to view or modify the messaging records management settings.

NOTE: The modify option is applicable for Office 365 users on all environments, such as, Federated, Synchronized and non-federated.

To view or modify the messaging records management settings for an Office 365 user

1. In the Exchange Online Properties wizard of an Office 365 user, click **Mailbox Settings**.
2. To place the user mailbox on litigation hold, make sure that the **Exchange Online Plan 2** license is enabled for the Office 365 user.

   For information on enabling Office 365 licenses for a user, see Create a new Azure AD user.
3. Under **Messaging Records Management**, click **Properties**, and then select the **Enable litigation hold** check box.

   An error is displayed if the **Exchange Online Plan 2** license is not enabled for the user.
4. In the **Messaging records management description URL** text box, enter URL of the location where the deleted mailbox items are preserved and changes made to mailbox items are recorded.
5. In the **Comments** text box, enter the mailbox comments, and click **Save**.

   A message is displayed confirming the success of the operation.
6. Close the dialog box and click **Save**.

NOTE: The changes made to Mailbox settings for the Office 365 user can be verified in the Microsoft Office 365 portal.

Azure AD user management tasks using Management Shell interface

Active Roles enables you to perform the following management tasks for Azure AD users:

- Create a new Azure AD user
- Update the Azure AD user properties
- View the Azure AD user properties
- Delete an Azure AD user

Active Roles Management Shell enables you to perform the following management tasks for Azure AD users:

Create a new Azure AD user

You can use the Active Roles Management Shell to create a new user. To create a new user, on the Management Shell interface, run the **New-QADUser** cmdlet. Use this cmdlet with the additional Boolean parameters *AzureUserAccountEnabled*, *AzureOffice365Enabled*, and *AzureAssociateTenantId* to create and enable a new Azure AD user. To retrieve and update Azure properties *edsvaAzureObjectID* attribute with correct value is required.

For more information on creating a new Azure AD user using the Management Shell interface, see the Active Roles Management Shell Help.

Example

Create a new Azure AD user:

```
C:\PS> New-QADUser -name 'user64' -ParentContainer
'CN=Users,DC=SS64,DC=com' -UserPassword 'Pass123w0rd' -
AzureUserAccountEnabled $true -AzureOffice365Enabled $true -
AzureUserPrincipalName 'user64@Azuredomain'-AzureAssociatedTenantId
'f918cb6c-275a-4815-8863-d7cbb90598b2'
```

Example

You can add additional attribute using -attr @{}:

```
C:\PS> New-QADUser -name 'user64' -ParentContainer
'CN=Users,DC=SS64,DC=com' -UserPassword 'Pass123w0rd' -
AzureUserAccountEnabled $true -AzureOffice365Enabled $true -
AzureUserPrincipalName 'user64@Azuredomain' -AzureAssociatedTenantId
'f918cb6c-275a-4815-8863-d7cbb90598b2' -attr @
{edsaAzureUserGivenName='user64';edsaAzureUserUsageLocation='IN'}
```

Update the Azure AD user properties

You can use the Active Roles Management Shell to modify attributes of an Azure AD user in Active Directory. On the Management Shell interface, run the **Set-QADUser** cmdlet.

For more information on modifying an Azure AD user using the Management Shell interface, see the *Active Roles Management Shell Help*.

NOTE: Set-QADUser cmdlet does not work for Azure attributes in Synchronized Identity and Federated environment.

View the Azure AD user properties

You can use the Active Roles Management Shell to retrieve all Azure AD users in a domain or container that match the specified conditions. On the Management Shell interface, run the **Get-QADUser** cmdlet.

For more information on viewing the Azure AD users using the Management Shell interface, see the *Active Roles Management Shell Help*.

Delete an Azure AD user

You can use the Active Roles Management Shell to delete a user from Azure. To delete an Azure AD user, on the Management Shell interface, run the **remove-QADObject** cmdlet.

For more information on deleting a user from Azure using the Management Shell interface, see the *Active Roles Management Shell Help*.

NOTE: In Synchronized or Federated environment, **remove-QADObject** removes the user from AD and then gets synchronized to the Azure portal.

Office 365 license management for hybrid environment users

Active Roles enables you to perform the following Office 365 license management tasks for hybrid users:

- Assign Office 365 licenses to new hybrid users
- Assign Office 365 licenses to existing hybrid users
- Modify or remove Office 365 licenses assigned to hybrid users
- Update Office 365 licenses display names
- Office 365 Granular user license management

Assign Office 365 licenses to new hybrid users

To assign Office 365 license to new hybrid users

1. On the Active Roles Web interface, Create a new Azure AD user.
2. In the **Create Azure Account**| **Usage Location** field, enter the two-letter location code of the location where the product will be used.

   NOTE: The **Usage Location** field is a mandatory field. The licenses cannot be assigned to the user if the product usage location information is not available. The local rules and regulations for usage of the product and services may vary based on the location.

3. Click **Next**.

   The **Licenses** wizard displays the Office 365 licenses, for example the Office 365 Business Essentials and Business Premium licenses, and the number of licenses that are available to assign to the user.

4. Select the check boxes corresponding to the licenses that need to be assigned to the user, and click **Finish**.

   The licenses assigned can be viewed on the user's**Azure Properties| Licenses** wizard.

Assign Office 365 licenses to existing hybrid users

To assign Office 365 license to existing hybrid users

1. On the Active Roles Web interface **Navigation** bar, click **Directory Management**.
2. On the **Views** tab in the **Browse** pane, click **Active Directory**.

   The list of Active Directory domains is displayed.
3. Click the specific domain, Container or the Organizational Unit, and then select the check-box corresponding to the specific user for which you want to view or update the properties.
4. In the **Command** pane, click **Azure properties**.

   The Azure Propertiesdialog box for the user is displayed.
5. In the Azure Properties dialog box, click **Settings.**
6. If the usage location is not entered in the **Usage Location** field, enter the two-letter location code of the location where the product will be used, and click **Save**.

   NOTE: The **Usage Location** field is a mandatory field. The licenses cannot be assigned to the user if the product usage location information is not available. The local rules and regulations for usage of the product and services may vary based on the location.

   Alternatively, if the product usage location is entered for the user earlier, navigate to the **Licenses** wizard to assign the Office 365 license to the user.
7. Re-open the Azure Properties dialog box for the user, and click **Licenses.**

   The Licenses wizard displays the Office 365 licenses, for example Office 365 Business Essentials and Business Premium licenses, that are available for assigning to the user.
8. Select the check box corresponding to the license that is to be assigned to the user.
9. Click the drop-down arrow corresponding to the selected license to view the products included in the license.

   By default, all the products are enabled for the user.
10. De-select the check boxes corresponding to the products in the license that are to be disabled for the user.
11. Click **Save**.

Modify or remove Office 365 licenses assigned to hybrid users

To modify or remove the Office 365 license assigned to existing hybrid users

1. On the Active Roles Web interface **Navigation** bar, click **Directory Management**.
2. On the **Views** tab in the **Browse** pane, click **Active Directory**.

   The list of Active Directory domains is displayed.
3. Click the specific domain, Container or the Organizational Unit, and then select the check-box corresponding to the specific user for which you want to view or update the properties.
4. In the **Command** pane, click **Azure properties**.
5. In the Azure Properties dialog box, click **Licenses.**

   The Licenses wizard displays the Office 365 licenses, for example Office 365 Business Essentials and Business Premium licenses, that are available and assigned to the user.
6. Click the drop-down arrow corresponding to the available licenses.

   The products that are included and assigned to the user in the license are displayed.
7. Select or de-select the check box corresponding to the product included in the license that needs to be enabled or removed for the user.
8. Click **Save**.

NOTE:

- When a user is de-provisioned or deleted, all the licenses that were assigned to the user are removed and can be assigned to other hybrid users.
- On performing an undo-deprovision operation on a hybrid user, the license assignment gets restored to the user on successful completion of the operation.
- For information on Azure AD user De-provisioning policy for Office 365 licenses management see the Office 365 Licenses Retention section in the *Active Roles Administration Guide*.

Update Office 365 licenses display names

To update the names of the licenses displayed on Azure properties -> Licenses page of a hybrid user

1. On the system running the Active roles Service, go to **..\One Identity\Active Roles\7.4\Service\AzureLicenses.xml**..
2. Open the xml file and edit the required SKU with the new license display name.

NOTE: If the xml file with Azure licenses is not available or is not well formed, then the default SKUs as derived from Azure Graph APIs are displayed on the Azure properties | Licenses page for the Azure AD user.

The updated licenses display names can be viewed on the user's **Azure Properties| Licenses** wizard.

Unified provisioning policy for Azure O365 Tenant Selection, Office 365 License Selection, and Office 365 Roles Selection, and OneDrive provisioning

The provisioning policy **O365 and Azure Tenant Selection** is a unified policy for all O365 user license and user role management as well as OneDrive provisioning for Azure AD users. This O365 management for users is controlled or restricted by creating a new provisioning policy and applying the policy to the Organizational Unit.

How this policy works

The provisioning policy **O365 and Azure Tenant Selection** is a unified policy for Azure Office 365 management for users, controlled or restricted by creating a new provisioning policy and applying the policy to the Organizational Unit. This policy is used for tenant selection, Office 365 license selection, and Office 365 roles selection, and OneDrive provisioning for Azure AD users.

This policy is also used for tenant selection for Groups and contacts.

Configuring an O365 and Azure Tenant Selection policy

You can configure an **O365 and Azure Tenant Selection** policy in the Active Roles Console (also known as the MMC Interface) to:

- Validate the selected Azure tenants for Azure users, guest users, O365 Groups and contacts.
- Select O365 Licenses for Azure users and guest users.
- Select O365 Roles for Azure users and guest users.
- Preprovision OneDrive for Azure users.

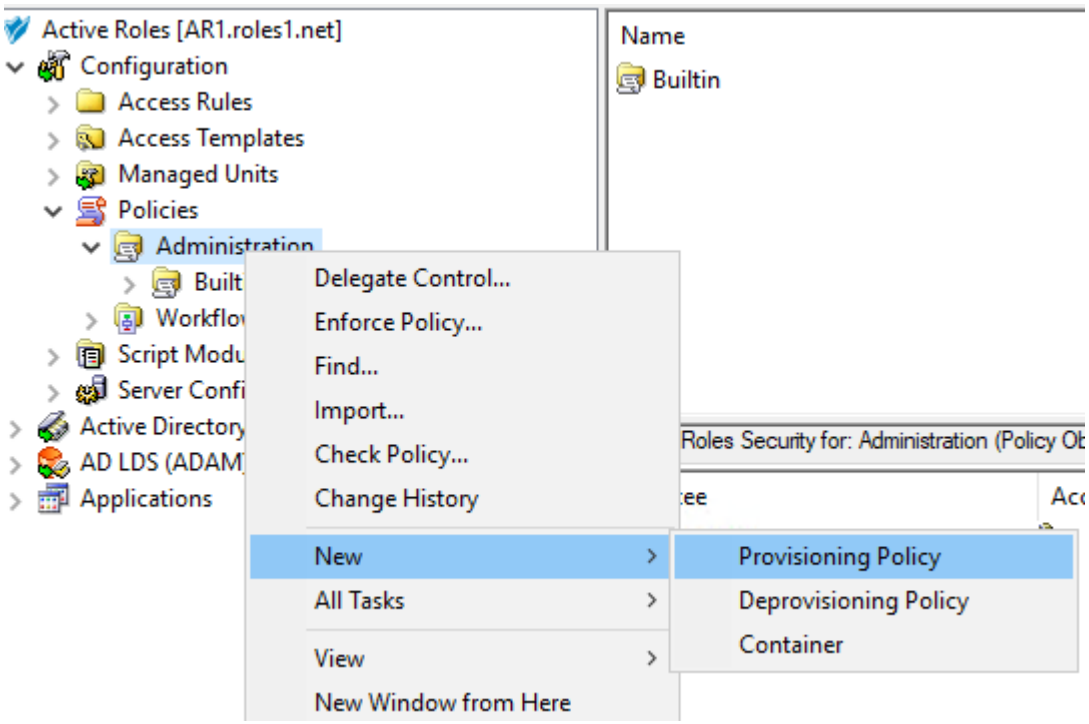
Prerequisites

Consider the following before configuring an **O365 and Azure Tenant Selection** policy:

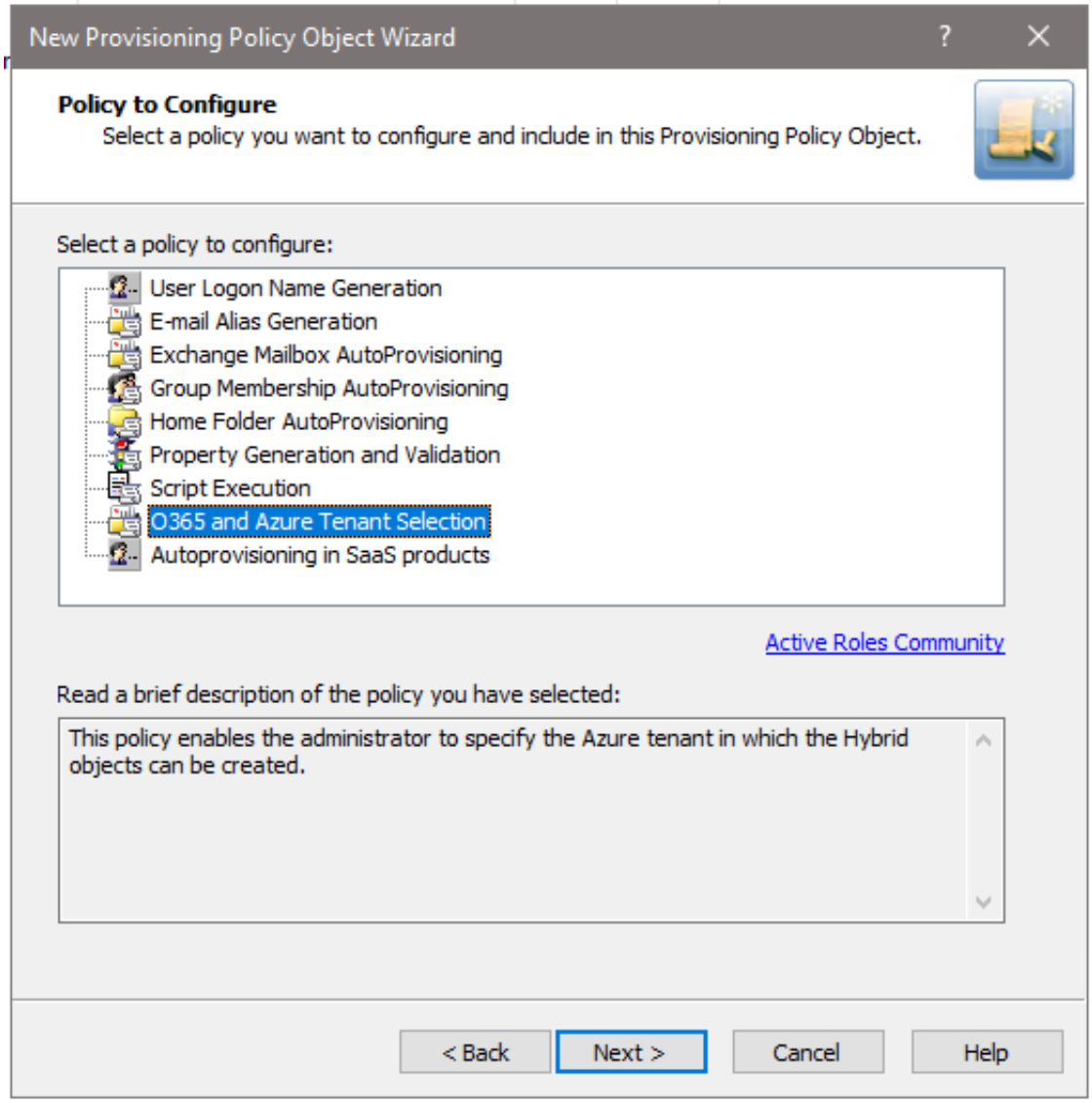
- The OneDrive settings of this policy are applicable to hybrid Azure users only, and will work only if you have already enabled OneDrive for your Azure tenant in the **Azure AD Configuration** > **Modify** (**Tenant details**) window of the Active Roles Configuration Center. For more information on enabling OneDrive for Azure users in an Azure tenant, see Enabling OneDrive in an Azure tenant.
- To configure an **O365 and Azure Tenant Selection** policy, your Organizational Unit (OU) must already have the **Azure - Default Rules to Generate Properties** built-in policy configured. For more information on configuring the policy, see Configuring the Azure - Default Rules to Generate Properties policy.

To configure an O365 and Azure Tenant Selection policy

1. Navigate to **Configuration** > **Policies** > **Administration**.
2. To open the **New Provisioning Policy Object Wizard** dialog, right-click in the middle pane to open the context menu, and then select **New** > **Provisioning Policy**.



3. On the **Name and Description** page, provide a unique **Name** for the new policy object. Optionally, also provide a **Description**. To continue, click **Next**.
4. On the **Policy to Configure** page, select **O365 and Azure Tenant Selection**, and click **Next**.

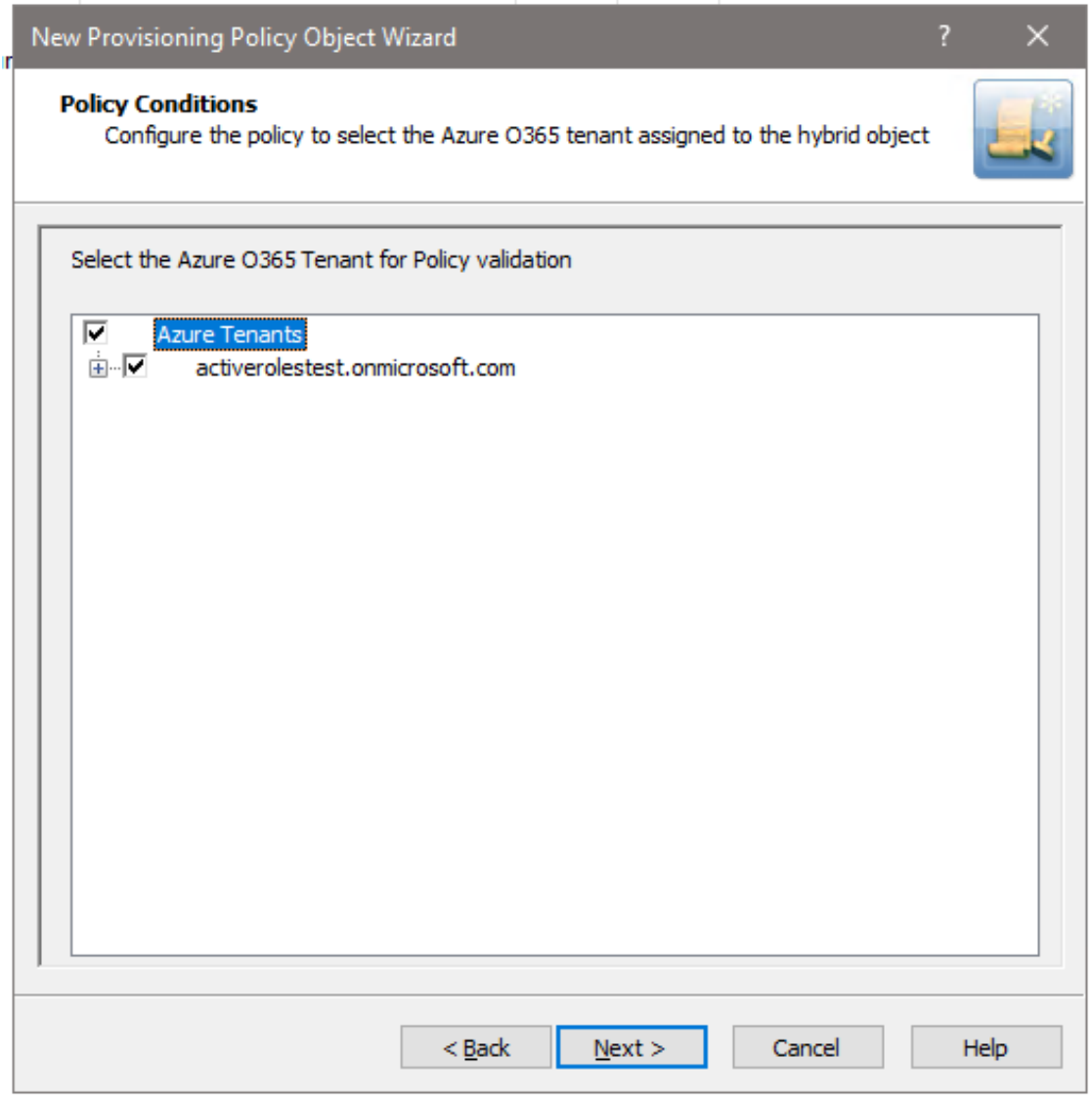


5. On the **Object Type Selection** page, to specify the type of object you want the policy to provision, click **Select**, then click **OK**.

   TIP: If you do not see the object type you need, expand the list by selecting **Show all possible object types**.

   NOTE: If you want to assign and validate Office 365 licenses and roles, or provision OneDrive storage as part of the configured policy, select the **User (user)** object type in this step. Office 365 license and role validation, and OneDrive provisioning are not applicable to Azure Groups and Azure Contacts.

6. On the **Policy Conditions** page, select your Azure tenant for which you want to set up the policy. To continue, click **Next**.

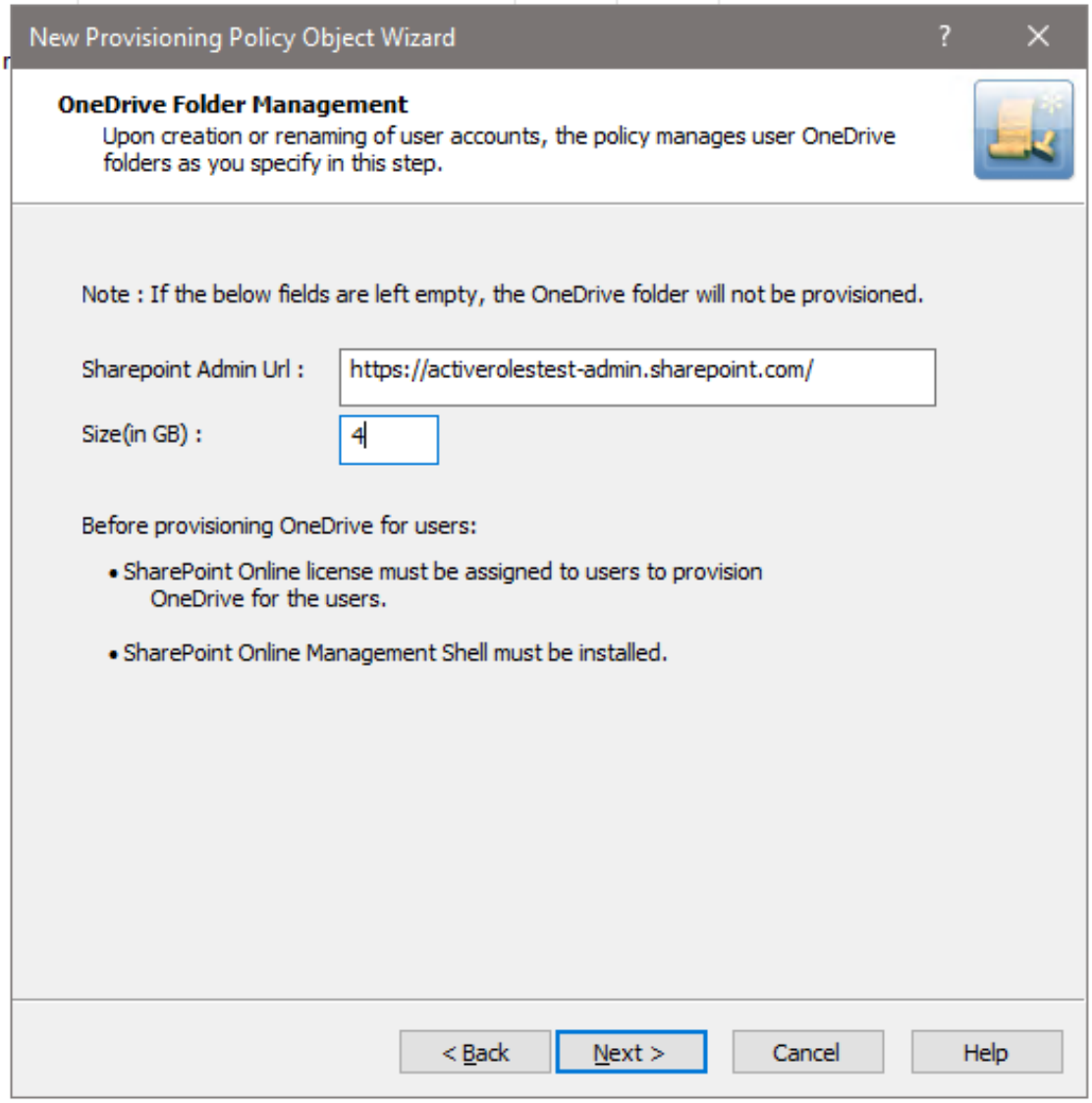


7. (Optional) On the next **Policy Conditions** page, select the licenses to validate and assign to new Azure users in the Azure tenant. To continue, click **Next**.

   NOTE: If OneDrive storage is planned to be provisioned in the selected Azure tenant for Azure users, make sure that you select the **SharePoint Online** license in this step. Otherwise, the configured OneDrive storage cannot be provisioned for Azure users created later. For more information, see Creating a new cloud-only Azure user.

8. (Optional) On the next **Policy Conditions** page, select the Office 365 roles to validate and assign to new Azure users in the Azure tenant. To continue, click **Next**.

9. (Optional) To configure OneDrive storage for the Azure users of the Azure tenant, configure the following attributes on the **OneDrive Folder Management** page:



- **SharePoint Admin URL**: Specify the URL of the SharePoint administration site of your Azure tenant. The URL has the following syntax: `<azure-tenant-name>-admin.sharepoint.com`
- **Size (in GB)**: Specify the default OneDrive storage size allocated for each Azure user in the Azure tenant.

If you do not need to provision OneDrive storage for users in the Azure tenant, leave the settings empty and click **Next**.

NOTE: If the wizard shows an error when clicking **Next** after configuring the OneDrive settings:

- Check that the specified SharePoint Admin URL is correct.
- Make sure that the specified OneDrive storage size is correct (that is, it is within the range of the individual cloud storage allowed for users in your organization).

10. On the **Enforce Policy** page, select the Organizational Unit (OU) for which the policy will be applied. To do so, click **Add** to open the **Select Objects** window, then select the OU from the list. To continue, click **OK** then **Next**.
11. To complete the wizard, click **Finish**.

Applying a new policy

Office 365 user license management

1. From the Web interface, assign, or modify the Office 365 license for an Azure AD User.

   The Policy is triggered for any Azure AD user in the Organization Unit for which the O365 and Azure Tenant selection policy is applied.

   If the policy conditions are not satisfied while assigning or modifying Azure AD User licenses, the following policy violation error is displayed:

   Provisioning policy failure. The 'O365 and Azure Tenant Selection' policy encountered an error. Exception in Azure Tenant Management Policy violation: The Azure user License(s) O365_BUSINESS_ESSENTIALS-PROJECTWORKMANAGEMENT, cannot be assigned. The policy prescribes that this Azure User requires only the specified license in the policy object to be assigned.

2. Right-click and click **Check Policy** to check if there are any policy violations

   For a container object, this displays the Check Policy dialog box.

3. Review the options in the **Check Policy** dialog box and click **OK**.

   The Policy Check Results window is displayed.

   IMPORTANT: Office 365 user license management now allows Administrator to select a subset of the licenses selected in policy during user creation or modification.

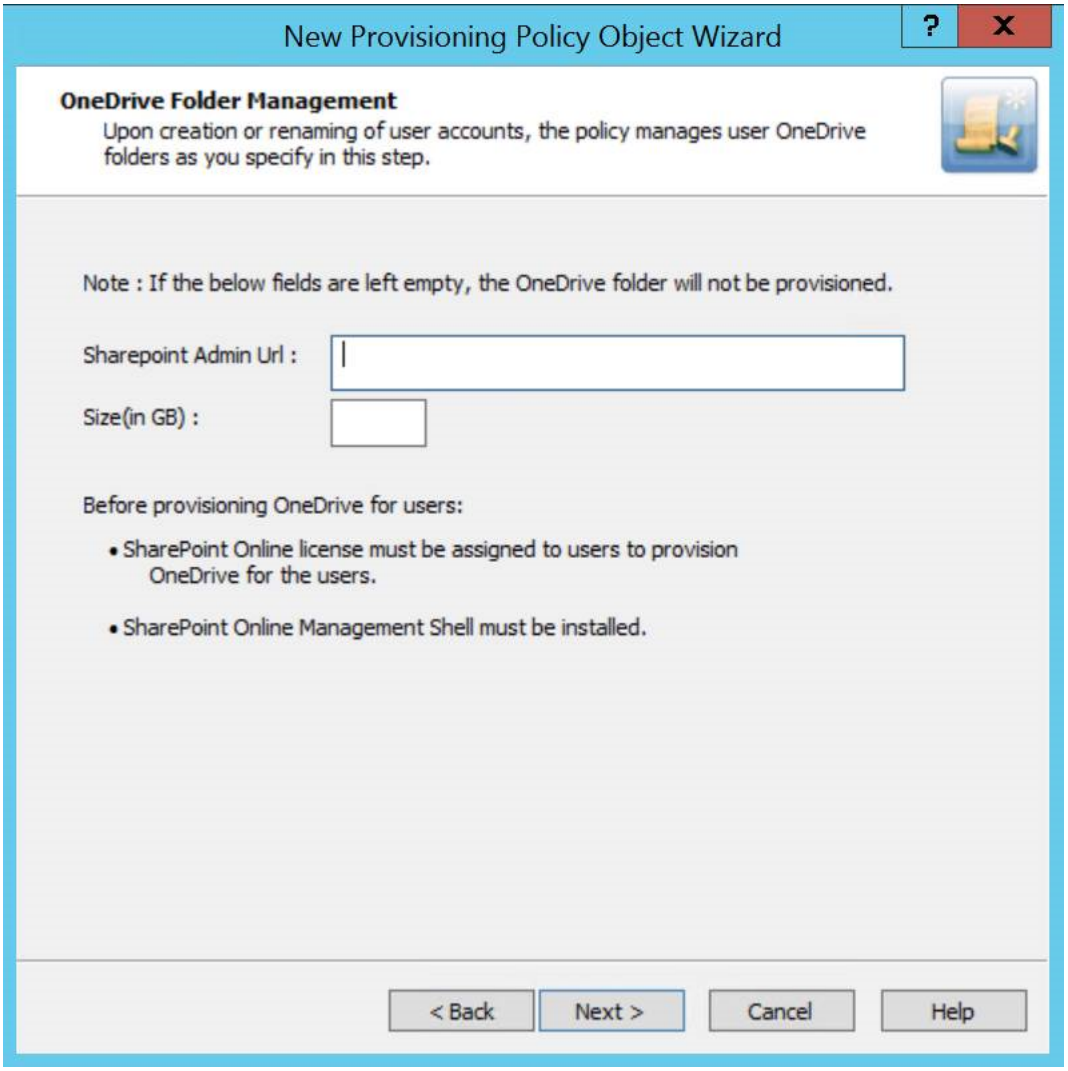
Office 365 user roles management through provisioning policy

From the Web interface, assign or modify the Office 365 roles for an Azure AD User.

While creating an Azure AD user from the Active Roles Web interface, if the policy conditions are not satisfied while assigning Azure AD User roles, the following policy violation error is displayed:

Provisioning policy failure. The 'O365 and Azure Tenant Selection' policy encountered an error. Exception in Azure Tenant Management Policy violation: The Azure user Role(s) cannot be assigned. The policy prescribes that this Azure User requires only the specified role in the policy object to be assigned.

Figure 111: OneDrive folder management wizard



Provisioning OneDrive for Azure AD users

1. From the Web interface, create an Azure AD User, and assign a valid SharePoint Online license.
2. After the user is created, the OneDrive provisioning process is performed in the background and after some time the process is completed.

   NOTE:

   - If the SharePoint Admin URL is incorrect then the OneDrive provisioning is not successful.
   - For an existing Azure AD user, during modification of user properties:

- If OneDrive is not provisioned, then OneDrive provisioning is triggered.
- If OneDrive is provisioned, and any changes are made to the OneDrive provisioning policy, then the policy changes are applied on the user.

3. To check the provisioning result, open Azure Properties window for the user from the Web interface, navigate to OneDrive tab.

   On successful provisioning of the user, the OneDrive URL, the used storage size, and the total storage size are displayed.

   NOTE: The storage size indicated in the policy gets synchronized to the Azure AD user's OneDrive.

Office 365 roles management for hybrid environment users

Active Roles enables you to perform the following Office 365 roles management tasks for hybrid users:

- Assign Office 365 roles to existing hybrid users
- Modify Office 365 roles assigned to hybrid users
- Office 365 user roles management

IMPORTANT:The Active Roles Web Interface only displays Azure roles that have been enabled. For the Office 365 Roles to be listed on the Web Interface, run the following commands.

- To get the guest inviter directory role template- `$roleTemplate = Get-AzureADDirectoryRoleTemplate | ? { $_.DisplayName -eq "Guest Inviter" }`.
- To enable an instance of the DirectoryRole template- `Enable-AzureADDirectoryRole -RoleTemplateId $roleTemplate.ObjectId`.

For more information on allowing the Azure roles to be listed on the Web Interface, see Enabling Azure Roles.

Assign Office 365 roles to existing hybrid users

To assign Office 365 roles to existing hybrid users

1. On the Active Roles Web interface **Navigation** bar, click **Directory Management**.
2. On the **Views** tab in the **Browse** pane, click **Active Directory**.

   The list of Active Directory domains is displayed.

3. Click the specific domain, Container or the Organizational Unit, and then select the check-box corresponding to the specific user for which you want to view or update the properties.
4. In the **Command** pane, click **Azure properties**.

   The Azure Properties dialog box for the user is displayed.
5. Click **O365 Roles** tab.

   The O365 Roles wizard displays the Office 365 roles, for example the Helpdesk Administrator, Directory Readers, and more.
6. Select the check boxes corresponding to the Office 365 roles that need to be assigned to the user, and click **Finish**.

Modify Office 365 roles assigned to hybrid users

To modify the Office 365 roles assigned to existing hybrid users

1. On the Active Roles Web interface **Navigation** bar, click **Directory Management**.
2. On the **Views** tab in the **Browse** pane, click **Active Directory**.

   The list of Active Directory domains is displayed.
3. Click the specific domain, Container or the Organizational Unit, and then select the check-box corresponding to the specific user for which you want to view or update the properties.
4. In the **Command** pane, click **Azure properties**.
5. In the Azure Properties dialog box, click **O365 Roles** tab.

   The O365 Roles wizard displays the Office 365 roles, for example the Helpdesk Administrator, Directory Readers, and more.
6. Select or clear the check boxes corresponding to the Office 365 roles that need to be assigned or removed for the user, and click **Finish**.

   The Office 365 roles assigned can be viewed on the user's **Azure Properties** | **O365 Roles wizard**.

   The results can also be viewed on the Azure portal's Licenses and Directory role tabs.

NOTE: When a user is de-provisioned, all the roles that were assigned to the user are retained.

Managing Office 365 Contacts

The Active Roles web interface enables you to perform administrative tasks such as create, read, update, and delete Office 365 contacts in Hybrid environment. You can also perform other operations such as add and remove Office 365 contacts to Groups.

Office 365 contact management tasks using Web interface

Active Roles web interface enables you to perform the following management tasks for Office 365 contacts:

- Create a new Office 365 contact
- Modify the Office 365 Contact Properties
- View the Change History for an Office 365 contact
- Delete an Office 365 contact

Create a new Office 365 contact

You can use the Active Roles Web Interface to create and enable a new Office 365 contact. .

To create a new Office 365 contact

1. On the Active Roles Web interface **Navigation** bar, click **Directory Management**.
2. On the **Views** tab in the **Browse** pane, click **Active Directory**.

   The list of **Active Directory** domains is displayed.
3. Click the domain in which you need to create a new contact.
4. In the list of objects displayed, click the required **Container** or the **Organizational Unit**.
5. In the **Command** pane, click **New Contact**.
6. In the **New Conatct in <OU name> ->General** wizard, enter the contact details such as **First Name**, **Last Name**, **Initials, and Display name.**
7. Click **Next**.
8. In the **Create Azure Account** properties wizard, select **Create Azure Contact** option.
9. Select the Tenant name from the **Tenant** list drop down.

10. In the **External e-mail address** field, enter the email address for the contact, and click **Finish**.

    The Office 365 account details for the new contact are generated automatically and populated in the respective fields.

NOTE: : In Federated or Synchronized environments, Office 365 contact creation is not supported. The contact is created in Active Roles and is synchronized eventually to Office 365 using Microsoft Native tools, such as AAD Connect. To manage the Office 365 contact through Active Roles, you must perform periodic back-synchronization to on-premise AD.

Modify the Office 365 Contact Properties

For an existing Office 365 contact, you can use the Active Roles Web Interface to modify the Office 365 contact properties.

To view or modify the Office 365 contact properties

1. On the Active Roles Web interface **Navigation** bar, click **Directory Management**.
2. On the **Views** tab in the **Browse** pane, click **Active Directory**.

   The list of Active Directory domains is displayed.
3. Click the specific domain, Container or the Organizational Unit, and then select the check box corresponding to the specific contact for which you want to view or update the Manager information.
4. In the **Command** pane, click **Azure properties**.

   The **Azure Properties** dialog box for the contact is displayed.
5. Use the tabs in the **Azure Properties** dialog box to view or modify properties of the Office 365 contact.
6. After setting all the required properties, click **Save**.

View the Change History for an Office 365 contact

You can use the Active Roles Web Interface to view the Change History for an Office 365 contact.

To view the Change History of an Office 365 contact

1. On the Active Roles Web interface **Navigation** bar, click **Directory Management**.
2. On the **Views** tab in the **Browse** pane, click **Active Directory**.

   The list of Active Directory domains is displayed.

3. Click the specific domain, Container or the Organizational Unit, and then the specific contact.
4. In the **Command** pane, click **Change History**.

   Selecting **Change History** displays the information on changes that were made to the contact through Active Roles.

Delete an Office 365 contact

You can use the Active Roles Web Interface to delete a contact for logon to Azure.

To delete an Office 365 contact

1. On the Active Roles Web interface **Navigation** bar, click **Directory Management**.
2. On the **Views** tab in the **Browse** pane, click **Active Directory**.

   The list of Active Directory domains is displayed.
3. Click the specific domain, Container or the Organizational Unit, and then the specific contact to be deleted.
4. In the **Command** pane, click **Delete**.

   The contact is deleted.

Managing Hybrid AD Groups

Active Roles provides the facility to perform administrative tasks such as create, read, update, and delete Groups in Azure Active Directory (Azure AD) through the web interface. You can also perform other operations such as add and remove members to Azure AD groups. Some of the group operations can be performed using the Management Shell in addition to the web interface. The following section guides you through the Active Roles web interface and Management Shell to manage Azure AD groups.

- Azure AD group management tasks using the Web interface
- Azure AD Group management tasks using Management Shell interface

Azure AD group management tasks using the Web interface

Active Roles enables you to perform the following management tasks for Azure AD groups:

- Create a new Azure AD Group
- View or modify Azure AD group properties
- Add or remove members to an Azure AD group
- View the Change History for an Azure AD Group
- Delete an Azure AD group

NOTE: : For the first time when Azure is configured, Office 365 Groups are not listed under the Office 365 Group container Refresh the page to resolve the issue.

Create an Azure AD group

You can use the Active Roles Web Interface to create and enable a new Azure AD group.

To create a new Azure AD group

1. On the Active Roles Web interface **Navigation** bar, click **Directory Management**.
2. On the **Views** tab in the **Browse** pane, click **Active Directory**.

   The list of **Active Directory** domains is displayed.
3. Click the domain in which you need to create a new group.
4. In the list of objects displayed, click the required **Container** or the **Organizational Unit**.
5. In the **Command** pane, click **New Group**.
6. In the **General** properties **New Group in <OU name>** wizard, enter the group details such as group name, pre-Windows 2000 group name, description, group scope, and group type.

   Group scope provides the option to create a Global or Universal group, and **Group type** enables you to create a Security or Distribution group.
7. Click **Next**.
8. In the **Create Azure Group** wizard, select the option **Create Azure Group**.

   Select the Tenant name from the Tenant list drop down. The Azure AD details for the new group are generated automatically and populated in the respective fields.

   NOTE: To set values for additional properties in the General Properties wizard, select the check-box corresponding to **Open properties for this object when I click Finish**
9. Click **Finish**.

NOTE: : In Federated or Synchronized environments, Azure AD group creation is not supported. The group is created in Active Roles and is synchronized eventually to Azure using Microsoft Native tools, such as AAD Connect. To manage the Azure AD group through Active Roles, you must perform periodic back-synchronization to on-premise AD.

View or modify Azure AD group properties

For an existing Azure AD group, you can use the Active Roles Web Interface to view or modify the properties.

To view or modify the Azure AD group properties

1. On the Active Roles Web interface **Navigation** bar, click **Directory Management**.
2. On the **Views** tab in the **Browse** pane, click **Active Directory**.

   The list of Active Directory domains is displayed.
3. Click the specific domain, Container or the Organizational Unit, and then the specific group for which you want to view or update the Azure AD group properties.
4. In the **Command** pane, click **Azure properties**.

   The **Azure Properties** wizard for the group account is displayed.
5. Use the tabs in the **Azure Properties** wizard to view or modify properties of the Azure AD group.
6. After setting all the required properties, click **Save**.

Add or remove members to an Azure AD group

You can use the Active Roles Web Interface to add or remove members from an Azure AD group.

To add a member to an Azure AD group

1. On the Active Roles Web interface **Navigation** bar, click **Directory Management**.
2. On the **Views** tab in the **Browse** pane, click **Active Directory**.

   The list of Active Directory domains is displayed.
3. Click the specific domain, Container or the Organizational Unit, and then the specific group to which you want to add members.
4. Select the check-box corresponding to the Azure AD group and in the **Command** pane click **Members**.

   The existing member information for the group is displayed.
5. In the **<Group> (objects found)** wizard, click **Add** to add a user to the group.
6. In the **Select Object** wizard, search and select the members you want to add to the group.

   NOTE: Click **Temporal Membership Settings** to specify the date and time when the selected members should be added or removed from the group.

7. Click **OK**.

   The **<Group> (objects found)** wizard displays all the members that are added to the group.

To remove a member from an Azure AD group

1. On the Active Roles Web interface **Navigation** bar, click **Directory Management**.
2. On the **Views** tab in the **Browse** pane, click **Active Directory**.

   The list of Active Directory domains is displayed.
3. Click the specific domain, Container or the Organizational Unit, and then the specific group from which you want to remove a member.
4. Select the check-box corresponding to the member and in the **Command** pane click **Members**.

   The existing member information for the group is displayed.
5. In the **<Group> (objects found)** wizard, select the member to be removed and click **Remove**.

   A message prompts you to confirm the action.
6. Click **Yes** to continue.

   The member information is removed from the <Group> (objects found) wizard.

View the Change History for an Azure AD Group

You can use the Active Roles Web Interface to view the Change History for an Azure AD group.

To view the Change History of an Azure AD group

1. On the Active Roles Web interface **Navigation** bar, click **Directory Management**.
2. On the **Views** tab in the **Browse** pane, click **Active Directory**.

   The list of Active Directory domains is displayed.
3. Click the specific domain, Container or the Organizational Unit, and then the specific user.
4. In the **Command** pane, click **Change History**.

   The information on changes that were made to the group properties through Active Roles is displayed.

Delete an Azure AD group

You can use the Active Roles Web Interface to delete an Azure AD group.

To delete an Azure AD group

1. On the Active Roles Web interface **Navigation** bar, click **Directory Management**.
2. On the **Views** tab in the **Browse** pane, click **Active Directory**.
3. The list of Active Directory domains is displayed.
4. Click the specific domain, Container or the Organizational Unit, and then the specific Azure AD group to be deleted.
5. In the **Command** pane, click **Delete**.

   A message prompts you to confirm the action.
6. Click **Yes** to continue.

   The Azure AD Group is deleted.

   > NOTE: Deleting a group account is an irreversible operation. A new group account with the same name as a deleted group account does not automatically assume the permissions and memberships of the deleted account. For this reason, it is advisable to disable rather than delete accounts.

Azure AD Group management tasks using Management Shell interface

Active Roles enables you to perform the following management tasks for Azure AD groups using the Management Shell interface:

- Create a new Azure AD Group
- Update the Azure AD Group properties
- Delete an Azure AD group
- Add a member to Azure AD Group
- Remove a member from Azure AD Group

Create a new Azure AD Group

You can use the Active Roles Management Shell to create a new user. To create a new group, on the Management Shell interface, run the **new-qadGroup** cmdlet. Use this cmdlet with the additional Boolean parameter **AzureOffice365Enabled** and **AzureAssociateTenantId** to create and enable a new Azure AD group.

For more information on creating a new Azure AD group using the Management Shell interface, see the Active Roles Management Shell Help.

Update the Azure AD Group properties

You can use the Active Roles Management Shell to modify attributes of an Azure AD user in Active Directory. On the Management Shell interface, run the **Set-QADGroup** cmdlet.

For more information on modifying an Azure AD user using the Management Shell interface, see the *Active Roles Management Shell Help*.

Delete an Azure AD group

You can use the Active Roles Management Shell to delete an Azure AD group. To delete an Azure AD group, on the Management Shell interface, run the **remove-QADObject** cmdlet.

For more information on deleting a group from Azure AD using the Management Shell interface, see the *Active Roles Management Shell Help*.

Add a member to Azure AD Group

You can use the Active Roles Management Shell to add a member to the Azure AD group. To add a member to an Azure AD group, on the Management Shell interface, run the **Add-QADGroupMember** cmdlet.

For more information on adding a member to an Azure AD group using the Management Shell interface, see the Active Roles Management Shell Help.

Remove a member from Azure AD Group

You can use the Active Roles Management Shell to remove a member from the Azure AD group. To remove a member from an Azure AD group, on the Management Shell interface, run the **Remove-QADGroupMember** cmdlet.

For more information on removing a member from an Azure AD group using the Management Shell interface, see the Active Roles Management Shell Help.

Managing Office 365 Groups

Active Roles supports CRUD (create, read, update and delete) operations for Office 365 (O365) groups and also lets you specify owners and add/remove members to or from existing O365 groups in your organization.

O365 groups facilitate teamwork within an organization by providing the same set of permissions to (guest) users, allowing you to provide access efficiently to various shared resources (such as a common Microsoft Outlook inbox and calendar, a shared OneNote

notebook, or other Microsoft 365 resources). For more information on O365 groups, see Overview of Microsoft 365 Groups for administrators in the *Microsoft 365 documentation*.

You can administer O365 groups either via the Active Roles Web Interface or through the Active Roles Management Shell.

- For more information on managing O365 groups with the Active Roles Web Interface, see Configuring O365 Groups with the Web Interface.
- For more information on managing O365 groups with the Active Roles Management Shell, see Office 365 Group management tasks using Management Shell interface.

Configuring O365 Groups with the Web Interface

You can use the Active Roles Web Interface to:

- Create, view, modify or delete O365 groups in your organization.
- Assign or remove owners and members to or from existing O365 groups.
- View the change history of existing O365 groups.

NOTE: You cannot use the Active Roles Web Interface to synchronize existing O365 groups. To synchronize O365 groups, configure an O365 synchronization schedule task with the Active Roles Console (also known as the MMC Interface). For more information, see Scheduling an O365 group synchronization task.

Creating an O365 Group with the Web Interface

You can use the Active Roles Web Interface to create and enable new Office 365 (O365) groups.

For more information on O365 groups, see Overview of Microsoft 365 Groups for administrators in the *Microsoft 365 documentation*.

To create a new O365 group

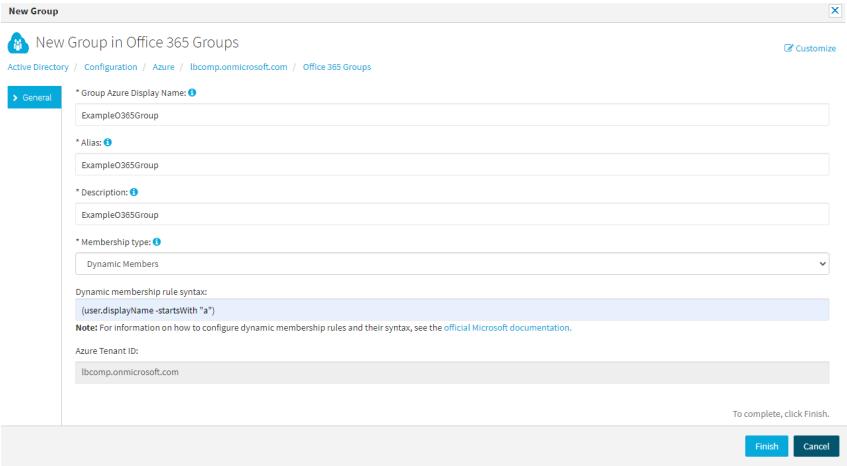
1. Navigate to **Directory Management** > **Tree** > **Azure** > **<azure-tenant-name>** > **Office 365 Groups**.

   The list of existing O365 groups in the selected Azure tenant appears.

   NOTE: When opening the list of **Office 365 Groups** the first time, Active Roles checks and fetches all existing O365 groups that may exist in the Azure cloud. This action is performed automatically and may take a few minutes to complete.

2. In the right-side pane, click **New Group**.

   The **New Group in Office 365 Groups** window appears.



3. Specify the **Group Azure Display Name** of the configured group.

   TIP: You can configure multiple groups with the same **Group Azure Display Name** in the same Azure tenant.

4. Specify the Exchange Online **Alias** of the O365 group. This value is used for naming the SharePoint site URL of the O365 group, and will also name the primary email address of the shared mailbox associated with the O365 group.

   TIP: The **Alias** of the O365 group must be unique within the Azure tenant.

5. Provide a short **Description** for the group.

6. Configure the **Membership type** of the group:

   - **Assigned**: When selected, you can add or remove members to or from the group manually later. For more information, see Adding or removing members from an O365 Group with the Web Interface.
   - **Dynamic Members**: When selected, Active Roles sets up the group as a dynamic membership group, and will automatically update group membership based on the configured **Dynamic membership rule syntax**.

   TIP: Consider the following when configuring the **Membership type**:

   - Select **Dynamic Members** to quickly configure a group based on a certain membership logic. For example, if you need to set up a group for employees from the same geographical location, business unit, or functional area, One Identity recommends configuring the group with **Dynamic Members**.
   - If you select **Dynamic Members**, you will not be able to manually add or remove members to or from the O365 group, unless you change its **Membership type** to **Assigned** later. However, you can still manually configure the owner(s) for a dynamic O365 group, as described in Adding or removing owners from an O365 Group with the Web Interface.
   - You can always change the **Membership type** later by navigating to the following option of the Active Roles Web Interface:

Directory Management > **Tree** > **Azure** > **<azure-tenant-name>** > **Office 365 Groups** > **<o365-group-name>** > **Azure Properties** > **General**.

- Changing the **Membership type** from **Dynamic Members** to **Assigned** later will keep the last set of members that were dynamically assigned to the group.

7. If you set the **Membership type** to **Dynamic Members**, specify the **Dynamic membership rule syntax**. Active Roles will send the logic configured in this field to Azure to automatically assign or remove members to or from the group later.

   NOTE: Consider the following when using the **Dynamic membership rule syntax** setting:

   - This setting is enabled only if **Membership type** is set to **Dynamic Members**. However, in that case, it is mandatory and cannot be empty.
   - The specified dynamic membership rule must meet all rule syntax requirements, otherwise the window will return an error. For more information on the available membership rule properties, operators and values, see Dynamic membership rules for groups in Azure Active Directory in the *Microsoft 365 documentation*.
   - Whenever you modify the dynamic membership rule of a dynamic O365 group, it can take several minutes for Azure to update the list of group members in the **Directory Management** > **Tree** > **Azure** > **<azure-tenant-name>** > **Office 365 Groups** > **<o365-group-name>** > **Dynamic Members** window.

8. To complete the configuration of the new O365 group, click **Finish**.

   The new O365 group will appear under the **Directory Management** > **Tree** > **Azure** > **<azure-tenant-name>** > **Office 365 Groups** node.

Modifying an O365 Group with the Web Interface

You can use the Active Roles Web Interface to modify the Azure properties of existing O365 groups in your Azure tenant. This is typically useful if you must:

- Modify the display name of the O365 group, for example because of an organizational change.
- Change the configured membership type (manually assigned or dynamic) of the O365 group.

NOTE: You cannot change the Exchange Online alias of an existing O365 group.

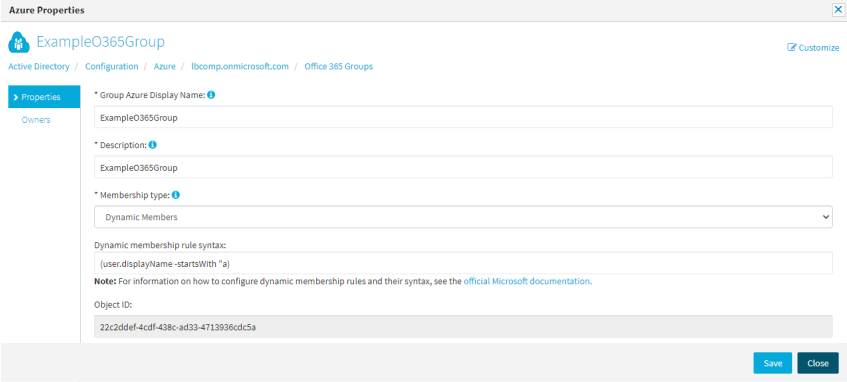
To modify the Azure properties of an O365 group

1. Navigate to **Directory Management** > **Tree** > **Azure** > **<azure-tenant-name>** > **Office 365 Groups**.

   The list of existing O365 groups in the selected Azure tenant appears.

NOTE: When opening the list of **Office 365 Groups** the first time, Active Roles checks and fetches all existing O365 groups that may exist in the Azure cloud. This action is performed automatically and may take a few minutes to complete.

2. Select the group that you want to configure.
3. In the right-side pane, click **Azure Properties**.
4. In the left-side pane of the Azure Properties window, click **Properties**.



5. (Optional) Specify the **Group Azure Display Name** of the configured group.

   TIP: You can configure multiple groups with the same **Group Azure Display Name** in the same Azure tenant.

6. (Optional) Provide a short **Description** for the group.
7. (Optional) Configure the **Membership type** of the group:
   - **Assigned**: When selected, you can add or remove members to or from the group manually later. For more information, see Adding or removing members from an O365 Group with the Web Interface.
   - **Dynamic Members**: When selected, Active Roles sets up the group as a dynamic membership group, and will automatically update group membership based on the configured **Dynamic membership rule syntax**.

   TIP: Consider the following when configuring the **Membership type**:
   - Select **Dynamic Members** to quickly configure a group based on a certain membership logic. For example, if you need to set up a group for employees from the same geographical location, business unit, or functional area, One Identity recommends configuring the group with **Dynamic Members**.
   - If you select **Dynamic Members**, you will not be able to manually add or remove members to or from the O365 group, unless you change its **Membership type** to **Assigned** later. However, you can still manually configure the owner(s) for a dynamic O365 group, as described in Adding or removing owners from an O365 Group with the Web Interface.
   - Changing the **Membership type** from **Dynamic Members** to **Assigned**

later will keep the last set of members that were dynamically assigned to the group.

8. (Optional) If you set the **Membership type** to **Dynamic Members**, specify the **Dynamic membership rule syntax**. Active Roles will send the logic configured in this field to Azure to automatically assign or remove members to or from the group later. For more information on how to specify a membership rule, see Dynamic membership rules for groups in Azure Active Directory in the *Microsoft 365 documentation*.
9. To apply your changes, click **Save**.

Adding or removing owners from an O365 Group with the Web Interface

You can use the Active Roles Web Interface to specify owners for an O365 group. Using the applicable options, you can either add or remove owners to or from the selected O365 group.

NOTE: Consider the following when configuring group ownership:

- You cannot specify a group as an owner of another group.
- Although Active Roles and Azure AD support specifying Azure guest users as group owners, One Identity recommends doing so only if assigning the ownership of a specific group to a guest user is in line with the security policies of your organization.

To add owners to an O365 group

1. Navigate to **Directory Management** > **Tree** > **Azure** > **<azure-tenant-name>** > **Office 365 Groups**.

   The list of existing O365 groups in the selected Azure tenant appears.

   NOTE: When opening the list of **Office 365 Groups** the first time, Active Roles checks and fetches all existing O365 groups that may exist in the Azure cloud. This action is performed automatically and may take a few minutes to complete.
2. Select the group that you want to configure.
3. In the right-side pane, click **Azure Properties**.
4. To list the owners of the selected group, click the **Owners** tab of the **Azure Properties** window.
5. Click **Add** to add a new owner (or owners) to the selected group.
6. In the **Select Object** page, use the search field to find the (guest) users in the Azure tenant that you want to specify as owners.

   The (guest) users meeting the search criteria will appear in the **Display Name** column.

7. Select the check boxes of the (guest) users you want to specify as owners of the group. The selected users will be listed in the lower pane of the **Select Object** page.
8. (Optional) To search for additional (guest) users, enter another search string. After that, select the (guest) users you want to add from the updated list.
9. To apply your changes, click **OK**. The **Owners** page will be updated with the new settings.

To remove owners from an O365 group

1. Navigate to **Directory Management** > **Tree** > **Azure** > **<azure-tenant-name>** > **Office 365 Groups**.

   The list of existing O365 groups in the selected Azure tenant appears.

   NOTE: When opening the list of **Office 365 Groups** the first time, Active Roles checks and fetches all existing O365 groups that may exist in the Azure cloud. This action is performed automatically and may take a few minutes to complete.
2. Select the group that you want to configure.
3. In the right-side pane, click **Azure Properties**.
4. To list the owners of the selected group, click the **Owners** tab of the **Azure Properties** window.
5. Select the owners whose ownership you want to revoke, and click **Remove**. The selected owners are removed from the list of owners.
6. To apply your changes, click **OK**. The **Owners** page will be updated with the new settings.

Adding or removing members from an O365 Group with the Web Interface

You can use the Active Roles Web Interface to add members to an existing Office 365 (O365) group with an **Assigned** membership setting. O365 groups support Azure AD users, Azure guest users, or external users as members.

NOTE: You cannot add or remove members manually to or from an O365 group with dynamic membership. To change the members of a dynamic group manually, first modify its membership type from **Dynamic Members** to **Assigned** membership. For more information, see Modifying an O365 Group with the Web Interface.

NOTE: Azure AD does not support adding O365 groups as members to other O365 groups. For more information, see the Add member page of the *Microsoft GRAPH REST API documentation*.

To add members to an O365 group

1. Navigate to **Directory Management** > **Tree** > **Azure** > **<azure-tenant-name>** > **Office 365 Groups**.

   The list of existing O365 groups in the selected Azure tenant appears.

   > NOTE: When opening the list of **Office 365 Groups** the first time, Active Roles checks and fetches all existing O365 groups that may exist in the Azure cloud. This action is performed automatically and may take a few minutes to complete.

2. Select the group that you want to configure.
3. In the right-side pane, click **Members**.

   The **Members** page then appears with the list of members in the selected group.

4. Click **Add** to add a new member (or members) to the group.
5. In the **Select Object** page, use the search field to find the (guest) users in the Azure tenant that you want to add as members.

   The (guest) users that meet the search criteria will appear in the **Display Name** column.

6. Select the check boxes of the (guest) users you want to add as members to the group. The selected (guest) users will be listed in the lower pane of the **Select Object** page.
7. (Optional) To search for additional (guest) users, enter another search string. After that, select the (guest) users you want to add as members from the updated list.
8. To apply your changes, click **OK**. The **Members** page will be updated with the new membership settings.

To remove members from an O365 group

1. Navigate to **Directory Management** > **Tree** > **Azure** > **<azure-tenant-name>** > **Office 365 Groups**.

   The list of existing O365 groups in the selected Azure tenant appears.

   > NOTE: When opening the list of **Office 365 Groups** the first time, Active Roles checks and fetches all existing O365 groups that may exist in the Azure cloud. This action is performed automatically and may take a few minutes to complete.

2. Select the group that you want to configure.
3. In the right-side pane, click **Members**.

   The **Members** page then appears with the list of members in the selected group.

4. To remove a member (or members) from the selected group, select the members from the **Members Name** list, and click **Remove**.

   The selected members are removed from the **Members Name** list.

5. To apply your changes, click **OK**. The **Members** page will be updated with the new membership settings.

Viewing the members of a dynamic O365 Group with the Web Interface

You can check the members of an O365 group with dynamic membership via the Active Roles Web Interface. This is useful if you want to get a quick update on the current membership status of the dynamic O365 group.

NOTE: You cannot add or remove members manually to or from an O365 group with dynamic membership. To change the members of a dynamic group manually, first modify its membership type from **Dynamic Members** to **Assigned** membership. For more information, see Modifying an O365 Group with the Web Interface.

To view the members of an O365 group with dynamic membership

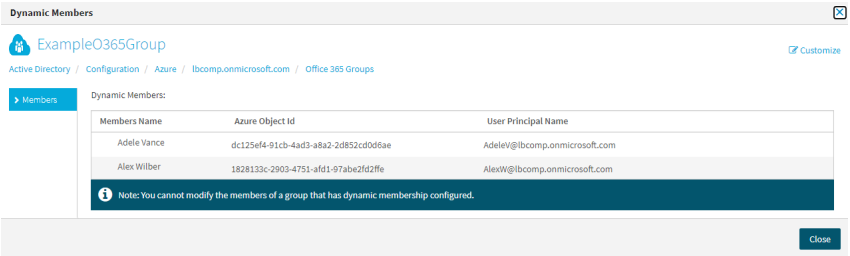
1. Navigate to **Directory Management** > **Tree** > **Azure** > **<azure-tenant-name>** > **Office 365 Groups**.

   The list of existing O365 groups in the selected Azure tenant appears.

   NOTE: When opening the list of **Office 365 Groups** the first time, Active Roles checks and fetches all existing O365 groups that may exist in the Azure cloud. This action is performed automatically and may take a few minutes to complete.

2. Select the group whose members you want to check.
3. In the right-side pane, click **Dynamic Members**.

   The **Dynamic Members** page then appears with the list of members in the selected group.



4. To exit the **Dynamic Members** window, click **Close**.

Viewing the change history of an O365 Group in the Web Interface

You can check the change history of an O365 group with the Active Roles Web Interface. This is useful if you want to view the list of changes that occurred to the selected O365 group, such as:

- Membership changes (that is, added or removed members).
- Membership type changes (that is, whether the group has been set to assigned or dynamic membership).

NOTE: The **Change History** option of the Active Roles Web Interface lists only group modifications that were performed in Active Roles. It does not list the changes of the group that were performed outside Active Roles, for example in Azure Portal.

To view the change history of an O365 group

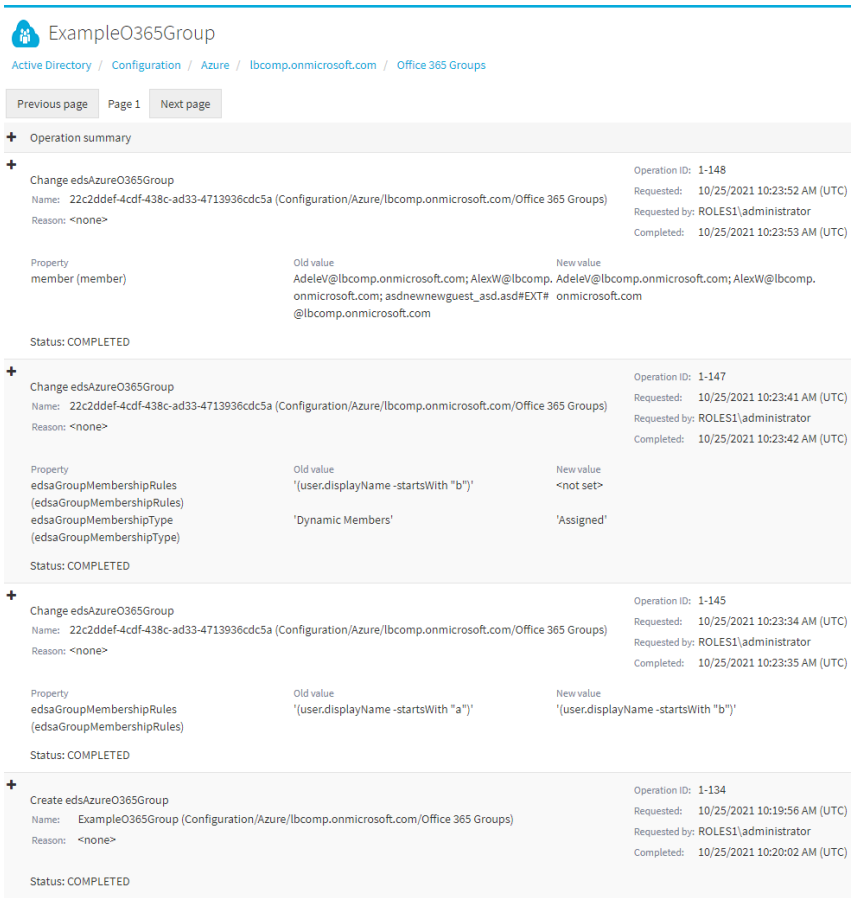
1. Navigate to **Directory Management** > **Tree** > **Azure** > **<azure-tenant-name>** > **Office 365 Groups**.

   The list of existing O365 groups in the selected Azure tenant appears.

   NOTE: When opening the list of **Office 365 Groups** the first time, Active Roles checks and fetches all existing O365 groups that may exist in the Azure cloud. This action is performed automatically and may take a few minutes to complete.

2. Select the group whose change history you want to check.
3. In the right-side pane, click **Change History**.

   The **Change History** page then appears, with the newest change of the group listed at the top of the page.



4. To close the **Change History** window, click any **Tree** node, or any option listed in the right-side pane.

Deleting an O365 Group with the Web Interface

You can use the Active Roles Web Interface to delete an O365 group from an Azure tenant. This is typically required when the O365 group becomes redundant or is otherwise no longer required, for example because of an organizational change.

CAUTION: Deleting an O365 group is a destructive operation that will delete the group from the Azure tenant on the Azure Portal as well.

To delete an O365 group

1. Navigate to **Directory Management** > **Tree** > **Azure** > **<azure-tenant-name>** > **Office 365 Groups**.

   The **Office 365 Groups** page then opens with the available Azure O365 Groups in the Azure tenant.
2. Select the group that you want to delete.
3. In the right-side pane, click **Delete**.
4. A confirmation dialog appears. To confirm the deletion of the group, click **Yes**.

The selected O365 group is then deleted from the Azure tenant.

Office 365 Group management tasks using Management Shell interface

Active Roles enables you to perform the following management tasks for Office 365 groups using the Management Shell interface:

- Create a new Office 365 Group
- Update the Office 365 Group properties
- Deleting an O365 Group with the Web Interface

Create a new Office 365 Group

You can use the Active Roles Management Shell to create a new group. To create a new group, on the Management Shell interface, run the **New-QADO365Group** cmdlet.

For more information on creating a new Office 365 group using the Management Shell interface, see the Active Roles Management Shell Help.

Update the Office 365 Group properties

You can use the Active Roles Management Shell to modify attributes of an Office 365 group. On the Management Shell interface, run the **Set-QADO365Group** cmdlet.

For more information on modifying an Office 365 user using the Management Shell interface, see the *Active Roles Management Shell Help*.

Delete an Office 365 group

You can use the Active Roles Management Shell to delete an Office 365 group. To delete an Office 365 group, on the Management Shell interface, run the **Remove-QADO365Group** cmdlet.

For more information on deleting a group from Office 365 using the Management Shell interface, see the *Active Roles Management Shell Help*.

Adding members to an Office 365 Group with the Management Shell

You can use the **Add-QADO365GroupMember** cmdlet on the Active Roles Management Shell to add new members to an Office 365 (O365) Group.

For more information on adding a member to an O365 Group using the Management Shell interface, see the *Active Roles PowerShell Reference Guide*.

> NOTE: Azure AD does not support adding O365 groups as members to other O365 groups. For more information, see the Add member page of the *Microsoft GRAPH REST API documentation*.

Get a member from Office 365 Group

You can use the Active Roles Management Shell to get a member from the Office 365 group. To get a member from an Office 365 group, on the Management Shell interface, run the **Get-QADO365GroupMember** cmdlet.

For more information on getting a member from an Office 365 group using the Management Shell interface, see the Active Roles Management Shell Help.

Get group from Office 365 Group

You can use the Active Roles Management Shell to get a group from the Office 365 group. To get a group from an Office 365 group, on the Management Shell interface, run the **Get-QADO365Group** cmdlet.

For more information on getting a group from an Office 365 group using the Management Shell interface, see the Active Roles Management Shell Help.

Removing members from an Office 365 Group with the Management Shell

You can use the **Remove-QADO365GroupMember** cmdlet on the Active Roles Management Shell interface to remove members from an Office 365 (O365) Group.

For more information on removing a member from an O365 Group using the Management Shell interface, see the *Active Roles PowerShell Reference Guide*.

Scheduling an O365 group synchronization task

You can use the **Sync Office 365 Groups** scheduled task of the Active Roles Console (also known as the MMC interface) to synchronize one or more O365 groups between the Azure Portal and the Active Roles database.

To configure a scheduled O365 group synchronization task

1. In the Active Roles Console, navigate to **Configuration** > **Server Configuration** > **Scheduled Tasks** > **Built-in container**.
2. Select **Sync Office 365 Groups**.
3. To customize the scheduling settings of the task, open the **Properties** > **Schedule** tab.
4. Use the **Schedule** tab to:
    - Set how frequently the task must run (daily, weekly, and so on).
    - Set the time and date of the first scheduled task running.
    - Set a timeout (that is, a duration after which the task stops if it runs for more time than the specified number of hours).

TIP: If the contents of the **Members** and/or **Azure Properties** pages in the Active Roles Web Interface for an O365 group differ from the membership and group properties information available on the Azure Portal, One Identity recommends running the scheduled **Sync Office 365 Groups** task manually to synchronize the O365 groups.

Managing Azure Security Groups

Active Roles supports CRUD (create, read, update and delete) operations for Azure AD Security groups and also lets you specify owners and add/remove members to or from existing Azure AD Security groups in your organization.

Azure Security groups are security principals used to secure objects (such as Azure users, Azure guest users, devices, applications, or other Azure Security groups) in Azure AD. Typically, Azure Security groups are set up to delegate application licenses or other resource permissions to users based on their group membership. For more information on Azure Security groups, see Groups in Microsoft 365 and Azure in the *Microsoft 365 community documentation*.

You can administer Azure Security groups via the Active Roles Web Interface.

Creating an Azure Security Group with the Web Interface

You can use the Active Roles Web Interface to create and enable new Azure security groups.

For more information on Azure Security groups, see Groups in Microsoft 365 and Azure in the *Microsoft 365 community documentation*.

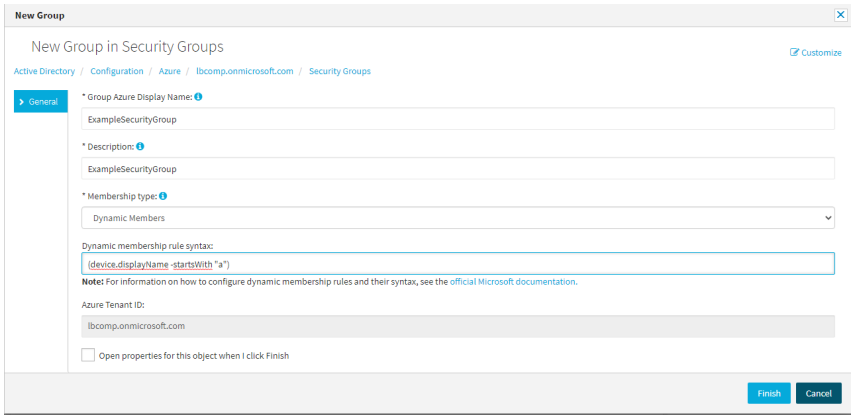
To create a new Azure Security group

1. Navigate to **Directory Management** > **Tree** > **Azure** > **<azure-tenant-name>** > **Security Groups**.

   The list of existing Azure Security groups in the selected Azure tenant appears.

2. In the right-side pane, click **New Group**.

   The **New Group in Security Groups** window appears.



3. Specify the **Group Azure Display Name** of the configured group.

   TIP: You can configure multiple groups with the same **Group Azure Display Name** in the same Azure tenant.

4. Provide a short **Description** for the group.
5. Configure the **Membership type** of the group:
   - **Assigned**: When selected, you can add or remove members to or from the group manually later. For more information, see Adding or removing members from an Azure Security Group with the Web Interface.
   - **Dynamic Members**: When selected, Active Roles sets up the group as a dynamic membership group, and will automatically update group membership based on the configured **Dynamic membership rule syntax**.

   TIP: Consider the following when configuring the **Membership type**:
   - Select **Dynamic Members** to quickly configure a group based on a certain membership logic. For example, if you need to set up a group for employees from the same geographical location, business unit, or functional area, One Identity recommends configuring the group with **Dynamic Members**.
   - If you select **Dynamic Members**, you will not be able to manually add or remove members to or from the Azure Security group, unless you change its **Membership type** to **Assigned** later. However, you can still manually configure the owner(s) for a dynamic Azure Security group, as described in Adding or removing owners from an Azure Security Group with the Web Interface.
   - Although the **Membership type** drop-down setting does not offer a separate **Dynamic Devices** option, you can actually set up dynamic Azure Security groups in Active Roles with the appropriate dynamic device membership rules (such as `device.displayName`). However, the Active Roles Web Interface cannot display member devices and applications.
   - You can always change the **Membership type** later by navigating to the following option of the Active Roles Web Interface:

     Directory Management > **Tree** > **Azure** > **<azure-tenant-name>** > **Security Groups** > **<azure-security-group-name>** > **Azure Properties** > **General**.
   - Changing the **Membership type** from **Dynamic Members** to **Assigned** later will keep the last set of members that were dynamically assigned to the group.

6. If you set the **Membership type** to **Dynamic Members**, specify the **Dynamic membership rule syntax**. Active Roles will send the logic configured in this field to Azure to automatically assign or remove members to or from the group later.

   NOTE: Consider the following when using the **Dynamic membership rule syntax** setting:

- This setting is enabled only if **Membership type** is set to **Dynamic Members**. However, in that case, it is mandatory and cannot be empty.
- The specified dynamic membership rule must meet all rule syntax requirements, otherwise the window will return an error. For more information on the available membership rule properties, operators and values, see Dynamic membership rules for groups in Azure Active Directory in the *Microsoft 365 documentation*.
- Whenever you modify the dynamic membership rule of a dynamic O365 group, it can take several minutes for Azure to update the list of group members in the **Directory Management** > **Tree** > **Azure** > **<azure-tenant-name>** > **Security Groups** > **<azure-security-group-name>** > **Dynamic Members** window.

7. To complete the configuration of the new Azure Security group, click **Finish**.

   The new Azure Security group will appear under the **Directory Management** > **Tree** > **Azure** > **<azure-tenant-name>** > **Security Groups** node.

Modifying an Azure Security Group with the Web Interface

You can use the Active Roles Web Interface to modify the Azure properties of an existing Azure Security group in your Azure tenant. This is typically useful if you have to:

- Modify the display name of the Azure Security group, for example because of an organizational or security policy change.
- Change the configured membership type (manually assigned or dynamic) of the Azure Security group.

To modify the Azure properties of an Azure Security group

1. Navigate to **Directory Management** > **Tree** > **Azure** > **<azure-tenant-name>** > **Security Groups**.

   The list of existing Azure Security groups in the selected Azure tenant appears.
2. Select the group that you want to configure.
3. In the right-side pane, click **Azure Properties**.
4. In the left-side pane of the Azure Properties window, click **Properties**.

5. (Optional) Specify the **Group Azure Display Name** of the configured group.

   TIP: You can configure multiple groups with the same **Group Azure Display Name** in the same Azure tenant.

6. (Optional) Provide a short **Description** for the group.
7. (Optional) Configure the **Membership type** of the group:
   - **Assigned**: When selected, you can add or remove members to or from the group manually later. For more information, see Adding or removing members from an Azure Security Group with the Web Interface.
   - **Dynamic Members**: When selected, Active Roles sets up the group as a dynamic membership group, and will automatically update group membership based on the configured **Dynamic membership rule syntax**.

   TIP: Consider the following when configuring the **Membership type**:
   - Select **Dynamic Members** to quickly configure a group based on a certain membership logic. For example, if you need to set up a group for employees from the same geographical location, business unit, or functional area, One Identity recommends configuring the group with **Dynamic Members**.
   - If you select **Dynamic Members**, you will not be able to manually add or remove members to or from the Azure Security group, unless you change its **Membership type** to **Assigned** later. However, you can still manually configure the owner(s) for a dynamic Azure Security group, as described in Adding or removing owners from an Azure Security Group with the Web Interface.
   - Changing the **Membership type** from **Dynamic Members** to **Assigned** later will keep the last set of members that were dynamically assigned to the group.

8. (Optional) If you set the **Membership type** to **Dynamic Members**, specify the **Dynamic membership rule syntax**. Active Roles will send the logic configured in this field to Azure to automatically assign or remove members to or from the group later. For more information on how to specify a membership rule, see Dynamic membership rules for groups in Azure Active Directory in the *Microsoft 365 documentation*.
9. To apply your changes, click **Save**.

Adding or removing owners from an Azure Security Group with the Web Interface

You can use the Active Roles Web Interface to specify owners for an Azure Security group. Using the applicable options, you can either add or remove owners to or from the selected Azure Security group.

NOTE: Consider the following when configuring group ownership:

- You cannot specify a group as an owner of another group.
- Although Active Roles and Azure AD support specifying Azure guest users as group owners, One Identity recommends doing so only if assigning the ownership of a specific group to a guest user is in line with the security policies of your organization.

To add owners to an Azure Security group

1. Navigate to **Directory Management** > **Tree** > **Azure** > **<azure-tenant-name>** > **Security Groups**.

   The list of existing Azure Security groups in the selected Azure tenant appears.
2. Select the group that you want to configure.
3. In the right-side pane, click **Azure Properties**.
4. To list the owners of the selected group, click the **Owners** tab of the **Azure Properties** window.
5. Click **Add** to add a new owner (or owners) to the selected group.
6. In the **Select Object** page, use the search field to find the (guest) users in the Azure tenant that you want to specify as owners.

   The (guest) users meeting the search criteria will appear in the **Display Name** column.
7. Select the check boxes of the (guest) users you want to specify as owners of the group. The selected users will be listed in the lower pane of the **Select Object** page.
8. (Optional) To search for additional (guest) users, enter another search string. After that, select the (guest) users you want to add from the updated list.
9. To apply your changes, click **OK**. The **Owners** page will be updated with the new settings.

To remove owners from an O365 group

1. Navigate to **Directory Management** > **Tree** > **Azure** > **<azure-tenant-name>** > **Security Groups**.

   The list of existing Azure Security groups in the selected Azure tenant appears.
2. Select the group that you want to configure.
3. In the right-side pane, click **Azure Properties**.

4. To list the owners of the selected group, click the **Owners** tab of the **Azure Properties** window.
5. Select the owners whose ownership you want to revoke, and click **Remove**. The selected owners are removed from the list of owners.
6. To apply your changes, click **OK**. The **Owners** page will be updated with the new settings.

Adding or removing members from an Azure Security Group with the Web Interface

You can use the Active Roles Web Interface to add members to an existing Azure Security group with an **Assigned** membership setting.

NOTE: Consider the following when managing the members of an Azure Security group in Active Roles:

- In the Active Roles Web Interface, you can only specify Azure users, Azure guest users, other Azure security groups and external users as group members for Azure Security groups with an **Assigned** membership setting. You cannot specify devices and applications. However, you can:
  - Configure Azure Security groups in the Active Roles Web Interface to have dynamic device membership by using the appropriate dynamic membership rules (such as `device.displayName`). For more information on the applicable membership rule syntax, see Dynamic membership rules for groups in Azure Active Directory in the *Microsoft 365 documentation*.
  - Configure device and application memberships later in Azure Portal for Azure Security groups created in Active Roles.
- You cannot add or remove members manually to or from an Azure Security group with dynamic membership. To change the members of a dynamic group manually, first modify its membership type from **Dynamic Members** to **Assigned** membership. For more information, see Modifying an Azure Security Group with the Web Interface.
- Although you can use the Active Roles Web Interface to manage Azure Security groups that also contain devices and applications, the Active Roles Web Interface cannot display the member devices and applications of such groups.

To add members to an Azure Security group

1. Navigate to **Directory Management** > **Tree** > **Azure** > **<azure-tenant-name>** > **Security Groups**.

   The list of existing Azure Security groups in the selected Azure tenant appears.
2. Select the group that you want to configure.
3. In the right-side pane, click **Members**.

The **Members** page then appears with the list of members in the selected group.

4. Click **Add** to add a new member (or members) to the group.
5. In the **Select Object** page, use the search field to find the (guest) users or Azure Security groups in the Azure tenant that you want to add.

   The (guest) users and Azure Security groups that meet the search criteria will appear in the **Display Name** column.
6. Select the check boxes of the (guest) users or Azure Security groups that you want to add to the group. The selected objects will appear in the lower pane of the **Select Object** page.
7. (Optional) To search for additional (guest) users or Azure Security groups, enter another search string. After that, select the objects you want to add from the updated list.
8. To apply your changes, click **OK**. The **Members** page will be updated with the new membership settings.

To remove members from an Azure Security group

1. Navigate to **Directory Management** > **Tree** > **Azure** > **<azure-tenant-name>** > **Security Groups**.

   The list of existing Azure Security groups in the selected Azure tenant appears.
2. Select the group that you want to configure.
3. In the right-side pane, click **Members**.

   The **Members** page then appears with the list of members in the selected group.
4. To remove a member (or members) from the selected group, select the members from the **Members Name** list, and click **Remove**.

   The selected members are removed from the **Members Name** list.
5. To apply your changes, click **OK**. The **Members** page will be updated with the new membership settings.

Viewing the members of a dynamic Azure Security Group with the Web Interface

You can check the member (guest) users and member Azure Security groups of an Azure Security group with dynamic membership via the Active Roles Web Interface. This is useful if you want to get a quick update on the current membership status of the dynamic Azure Security group.

NOTE: Consider the following when using dynamic Azure Security groups in Active Roles:

- You cannot add or remove members manually to or from an Azure Security group with dynamic membership. To change the members of a dynamic group manually,

first modify its membership type from **Dynamic Members** to **Assigned** membership. For more information, see Modifying an Azure Security Group with the Web Interface.

- Although you can use the Active Roles Web Interface to manage Azure Security groups that also contain devices and applications, the Active Roles Web Interface cannot display the member devices and applications of such groups.

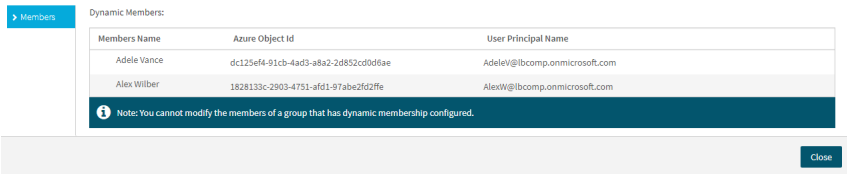
To view the members of an Azure Security group with dynamic membership

1. Navigate to **Directory Management** > **Tree** > **Azure** > **<azure-tenant-name>** > **Security Groups**.

   The list of existing Azure Security groups in the selected Azure tenant appears.

2. Select the group whose members you want to check.
3. In the right-side pane, click **Dynamic Members**.

   The **Dynamic Members** page then appears with the list of members in the selected group.



4. To exit the **Dynamic Members** window, click **Close**.

Viewing the change history of an Azure Security Group in the Web Interface

You can check the change history of an Azure Security group with the Active Roles Web Interface. This is useful if you want to view the list of changes that occurred to the selected Azure Security group, such as:

- Membership changes (that is, added or removed members).
- Membership type changes (that is, whether the group has been set to assigned or dynamic membership).

NOTE: The **Change History** option of the Active Roles Web Interface lists only group modifications that were performed in Active Roles. It does not list the changes of the group that were performed outside Active Roles, for example in Azure Portal.

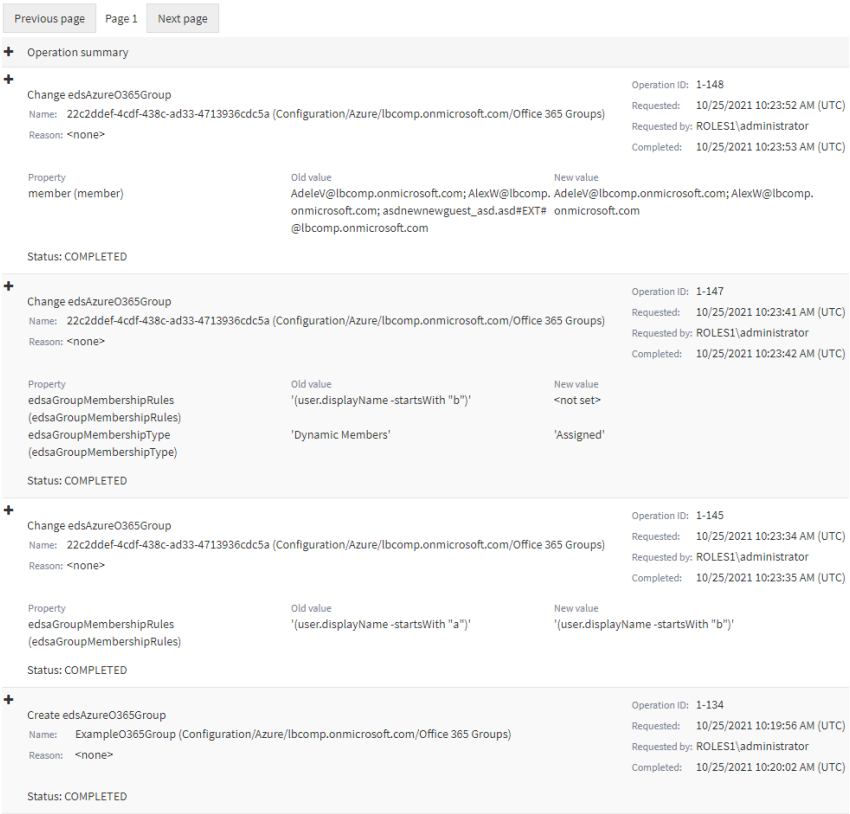
To view the change history of an Azure Security group

1. Navigate to **Directory Management** > **Tree** > **Azure** > **<azure-tenant-name>** > **Security Groups**.

The list of existing Azure Security groups in the selected Azure tenant appears.

2. Select the group whose change history you want to check.
3. In the right-side pane, click **Change History**.

   The **Change History** page then appears, with the newest change of the group listed at the top of the page.



4. To close the **Change History** window, click any **Tree** node, or any option listed in the right-side pane.

Deleting an Azure Security Group with the Web Interface

You can use the Active Roles Web Interface to delete an Azure Security group from an Azure tenant. This is typically required when the group becomes redundant or is otherwise no longer required, for example because of a security policy change.

CAUTION: Deleting an Azure Security group is a destructive operation that will delete the group from the Azure tenant on the Azure Portal as well.

To delete an Azure Security group

1. Navigate to **Directory Management** > **Tree** > **Azure** > **<azure-tenant-name>** > **Security Groups**.

   The list of existing Azure Security groups in the selected Azure tenant appears.
2. Select the group that you want to delete.
3. In the right-side pane, click **Delete**.
4. A confirmation dialog appears. To confirm the deletion of the group, click **Yes**.

The selected Azure Security group is then deleted from the Azure tenant.

Managing cloud-only Azure users

Active Roles provides the facility to perform administrative tasks such as create, read, update, and delete Azure users on cloud through web interface. You can also perform other operations such as viewing Azure membership details, Azure properties, Exchange online properties, change history, disabling the account, renaming the account, and password reset.

Viewing cloud-only Azure user

You can use the Active Roles Web Interface to view cloud-only Azure user information.

To view cloud-only Azure user

1. On the Active Roles Web interface **Navigation** bar, click **Directory Management**.
2. On the **Views** tab in the **Browse** pane, click | **Azure** |***<Azure tenant>*** |**Azure Users**.

   NOTE: Active Roles lists the available cloud-only Azure Users, Azure Guest Users, and Azure Contacts on the Active Roles Web Interface with the following restrictions:

   - Active Roles can initially list 999 items.
   - The items listed in the list have a sliding expiry of 8 hours, after which the objects that have not been accessed will be flushed.
   - Whenever you perform a search in the list, Active Roles will always fetch the list of objects from Azure to update the cache.

Creating a new cloud-only Azure user

You can use the Active Roles Web Interface to create and enable a new cloud-only Azure user.

To create a new cloud-only Azure user

1. On the Active Roles Web interface **Navigation** bar, click **Directory Management**.
2. On the **Views** tab in the **Browse** pane, click **Azure** |***<Azure tenant>*** |**Azure Users**.

   The Azure Users page is displayed and lists the cloud-only Azure users available in Azure.
3. In the **Command** pane, under Azure Users, click **New User**.
4. In the New User window, on the **General** tab, enter the appropriate text in the **Name**, **Alias**, and **Description** fields.
5. Click **Finish**.

   The Azure User page displays the newly added Azure users.

Viewing or modifying cloud-only Azure user properties

For an existing cloud-only Azure user, you can use the Active Roles Web Interface to view or modify the properties.

To view or modify Azure cloud-only user properties

1. On the Active Roles Web interface **Navigation** bar, click **Directory Management**.
2. On the **Views** tab in the **Browse** pane, click | **Azure** |***<Azure tenant>*** |**Azure users**.

   The Azure user page is displayed and lists the Azure users available in Azure.
3. Select the Azure user for which you want to view or modify the properties.
4. In the **Command** pane, click **Azure properties**.

   The **Azure Properties** wizard for the group account is displayed.
5. Use the tabs in the **Azure Properties** wizard to view or modify properties of the cloud-only Azure user.
6. After setting all the required properties, click **Save**.

Configuring Microsoft OneDrive for cloud-only Azure users

For cloud-only Azure users, you can use the Active Roles Web Interface to configure Microsoft OneDrive.

To configure Microsoft OneDrive

1. On the Active Roles Web interface **Navigation** bar, click **Directory Management**.
2. On the **Views** tab, click | **Azure Users** | **Azure** | | **Azure Configuration** | ***<Azure tenant>***.
3. Select the tenant and then click **OneDrive Configuration** available on the **Command** pane.
4. Provide the details in the **OneDrive Configuration** wizard and click **Save**.

IMPORTANT: The OneDrive Configuration here is applicable for cloud-only users. For OneDrive Configuration for hybrid users, see Configuring Active Roles to Manage Hybrid AD Objects .

Disabling cloud-only Azure user

You can use the Active Roles Web Interface to disable or enable an cloud-only Azure user.

To disable or re-enable a previously enabled cloud-only user for Azure

1. On the Active Roles Web interface **Navigation** bar, click **Directory Management**.
2. On the **Views** tab, click | **Azure** | ***<Azure tenant>*** | **Azure users**.

   The Azure user page is displayed and lists the Azure users available in Azure.
3. Select the Azure user to be disabled.
4. In the **Command** pane, click **Disable**.

   The account is disabled and marked with a disabled icon.
5. To enable a disabled account, select the check-box corresponding to the disabled account and in the **Command** pane click **Enable**.

   NOTE: The **Enable** command only appears for a disabled account.

   The account is enabled again.

Viewing and modifying Exchange Online properties

You can use the Active Roles Web Interface to create and view and modify the Exchange online properties of the new cloud-only Azure user.

To view the Exchange Online properties of a cloud-only Azure user

1. On the Active Roles Web interface **Navigation** bar, click **Directory Management**.
2. On the **Views** tab, click | **Azure** | ***<Azure tenant>*** | **Azure users**.

   The Azure user page is displayed and lists the Azure users available in Azure.
3. Select the check box corresponding to the specific cloud-only Azure user with Exchange Online license for which you want to view the properties.
4. In the **Command** pane, click **Exchange Online Properties**.

   The **Exchange Online Properties** wizard displays the following Exchange Online properties for the cloud-only Azure user.
   - Mail Flow Settings
   - Delegation
   - E-mail Addresses
   - Mailbox Features
   - Mailbox Settings
5. Use the tabs in the **Exchange Online Properties** dialog box to view the following Exchange Online properties of the cloud-only Azure user:
   - Mail Flow Settings
     - Message Size restrictions
       - Sending Message size
       - Receiving Message size.
     - Delivery Options
       - Send On behalf
       - Forwarding Address
       - Enabling or disabling of Delivery messages to the forwarding address and mailbox.
   - Delegation
   - E-mail Addresses
   - Mailbox Features
     - Exchange ActiveSync
     - Outlook Web App
     - MAPI

    - IMAP
    - POP3
    - Archive
- Mailbox Settings
    - Messaging Records management

Resetting password for a cloud-only Azure user

You can use the Active Roles Web Interface to reset the password for a cloud-only Azure user.

To reset password of the cloud-only Azure user

1. On the Active Roles Web interface **Navigation** bar, click **Directory Management**.
2. On the **Views** tab in the **Browse** pane, click **Azure** |***<Azure tenant>*** |**Azure Users**.

   The Azure Users page is displayed and lists the Azure users available in Azure.
3. In the **Command** pane, under Azure Users, click **Reset Password**.
4. In the **Password** field, provide the new password.
5. Reenter the password in the **Confirm password** field.
6. Select the relevant check box if you want users to change password during next sign-in.
7. Click **Finish**.

   The password is reset for the cloud-only Azure user.

Renaming Azure user

You can use the Active Roles Web Interface to rename an Azure user.

To rename an Azure user

1. On the Active Roles Web interface **Navigation** bar, click **Directory Management**.
2. On the **Views** tab in the **Browse** pane, click **Azure** |***<Azure tenant>*** |**Azure Users**.
3. Select the Azure user that need to be renamed.
4. In the **Command** pane, click **Rename**.

5. Enter the required name.
6. Click **Yes** to continue.

   The Azure user that are selected are renamed.

Viewing Azure membership

You can use the Active Roles Web Interface to view the Azure membership details of an cloud-only Azure user.

Viewing cloud-only Azure user membership details

1. On the Active Roles Web interface **Navigation** bar, click **Directory Management**.
2. On the **Views** tab in the **Browse** pane, click **Azure** |***<Azure tenant>*** |**Azure Users**.
3. In the **Command** pane, click **Azure member of**.

   You can view the Azure group to which the cloud-only Azure user is associated.

Viewing change history

You can use the Active Roles Web Interface to view the Change History and User Activity for a cloud-only Azure user.

To view the Change History and User Activity of a cloud-only Azure user

1. On the Active Roles Web interface **Navigation** bar, click **Directory Management**.
2. On the **Views** tab, click | **Azure** | ***<Azure tenant>*** | **Azure Users**.

   The Azure user page is displayed and lists the Azure users available in Azure.
3. Select the Azure user to view the history.
4. In the **Command** pane, click **Change History** or **User Activity**.

   Selecting **Change History** displays the information on changes that were made to the user through Active Roles.

Deleting an Azure user account

You can use the Active Roles Web Interface to delete an Azure user.

To delete an Azure user account

1. On the Active Roles Web interface **Navigation** bar, click **Directory Management**.
2. On the **Views** tab in the **Browse** pane, click **Azure** |***<Azure tenant>*** |**Azure Users**.
3. Select the Azure user that need to be deleted.
4. In the **Command** pane, click **Delete**.

   A message prompts you to confirm the action.
5. Click **Yes** to continue.

   The Azure user that are selected are deleted.

Managing cloud-only Azure guest users

You can invite (or re-invite), modify and remove cloud-only Azure guest users in the Azure AD of your organization with the Active Roles Web Interface.

An Azure guest user is a type of cloud-only Azure user that is not part of the organization domain for which you configure it.

When you create a new cloud-only Azure user for your organization, you must:

1. Specify a User Principal Name (UPN) and password for the Azure user.
2. Select the organization domain where the Azure user will be located within the Azure tenant.

However, when you create an Azure guest user, no domains are assigned to the user within the Azure tenant. Instead, the procedure has the following main steps:

1. You specify the basic permissions of the guest user, along with an email address to which Active Roles will send an invitation.
2. Using the link in the invitation e-mail, the guest user can gain the configured access with their account upon joining the organization.
3. Once the guest accepted the invitation, you can assign additional permissions (like roles, licenses, storage space, and so on) to the user, similarly to a regular cloud-only Azure user.

NOTE: Active Roles does not restrict the type of permissions that you can assign to Azure guest users. However, for security reasons, One Identity recommends that you assign only the rights and resources to guest users that external contractors typically receive in your organization.

Inviting an Azure guest user

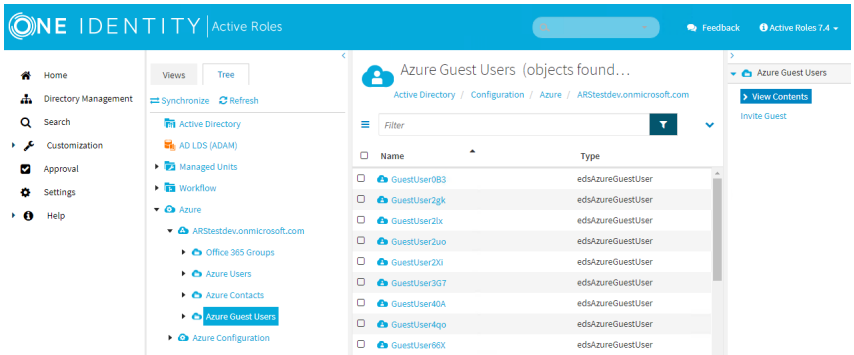
If an external user (such as a contractor, or other non-employee resource with limited permissions) must be added to the organization, invite them as Azure guest users to the Azure tenant of the organization using the Active Roles Web Interface.

To invite an Azure guest user

1. Navigate to **Directory Management** > **Tree** > **Azure** > **<azure-tenant-name>** > **Azure Guest Users**.

   The list of Azure guest users of the selected tenant is displayed.

   Figure 112: Directory Management > Tree View > Azure > <azure-tenant-name> > Azure Guest Users — Listing the Azure guest users in the tenant

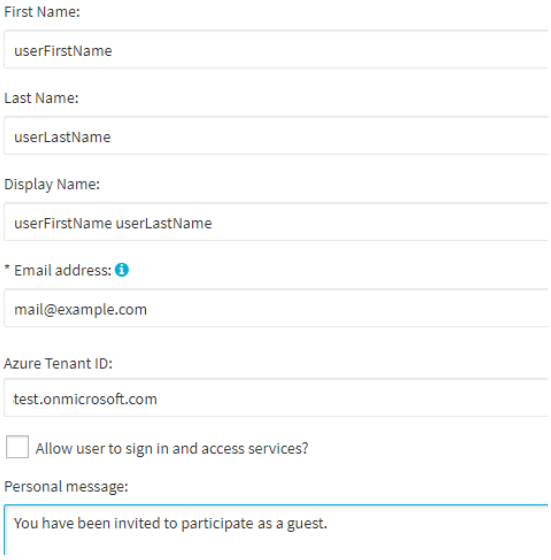
2. In the right-side pane, click **Invite Guest**.

   You will invite a new guest user, and set up their account, application licenses and various admin roles, too.

3. **Identity**

   Configure the settings required by your organization for setting up the identity of the guest user.

Figure 113: Azure Guest Users > Invite Guest > Identity – Configuring basic user account settings for the Azure guest user



a. (Optional) Enter the **First Name** of the Azure guest user.

   NOTE: If you do not enter a **First Name**, Active Roles will fill this field with the local part of the specified **Email address**.

b. (Optional) Enter the **Last Name** of the Azure guest user.

c. (Optional) Enter the **Display Name** of the Azure guest user.

   TIP: By default, the **Display Name** is automatically generated from the specified **First Name** and **Last Name**, but you can modify it to something else (such as a nickname).

d. Enter the **Email address** where Active Roles will send out the invitation. This field is mandatory and must be unique.

e. (Optional) Enter the **Azure Tenant ID** of the Azure tenant that will contain the guest user.

f. To grant the Azure guest user access to the configured licenses and admin roles, select **Allow user to sign in and access services**.

   - If this setting is selected during this step, the guest user will receive access as soon as they accept the invitation.
   - If left clear, you must manually grant access later by enabling this setting in the **Azure Properties** page of the guest user. For more information, see Viewing and updating the properties of an Azure guest user.

     TIP: Leaving this setting clear is useful if the account of the Azure guest user is created in advance, and they require access to the

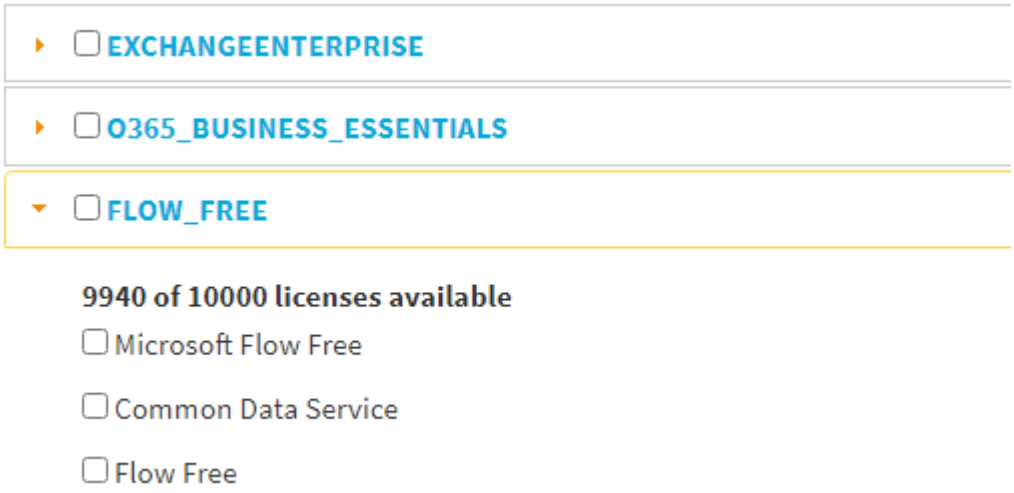
assigned resources only later (for example, because their contract project starts only at a later date).

g. (Optional) Enter a unique **Personal message** that the invitation email will contain.

4. **Licenses**

   Select the Microsoft application resources licensed in your organization that you want to assign to the configured Azure guest user.

   Figure 114: Azure Guest Users > Invite Guest > Licenses – Assigning application licenses to the Azure guest user



5. **O365 Admin Roles**

   Select the O365 role(s) that you want to grant for the Azure guest user.

Figure 115: Azure Guest Users > Invite Guest > O365 Admin Roles – Assigning Office 365 administrator roles to the Azure guest user

NOTE: You can assign roles to the Azure guest user in Active Roles without any limitation. However, One Identity recommends that you assign Azure guest users only the admin roles that external contractors typically receive in your organization.

6. (Optional) **Job Info**

   Enter the **Job Title** and the assigned **Department** of the guest user, if needed.

   Figure 116: Azure Guest Users > Invite Guest > Job Info – Specifying organizational information for the Azure guest user

7. To save your changes and send the invite email to the guest user, click **Finish**.

   NOTE: Consider the following when administering cloud-only Azure guest users:

   - You can resend the invitation later for the guest user, if needed. For more information, see Resending the invitation to an Azure guest user.

- You can modify the user account settings later, if needed. For more information, see Viewing and updating the properties of an Azure guest user.

Viewing Azure guest users

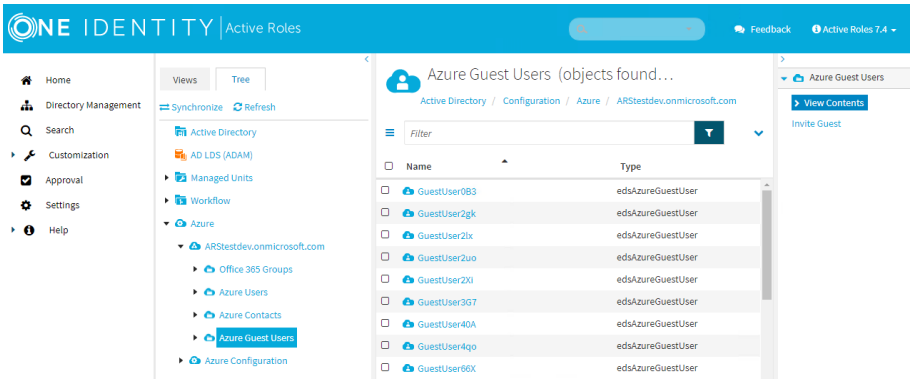
To list the configured cloud-only Azure guest users of an Azure tenant, and access their available configuration actions, expand the **Azure Guest Users** node of the Active Roles Web Interface.

To view the configured Azure guest users in an Azure tenant

Navigate to **Directory Management** > **Tree** > **Azure** > **<azure-tenant-name>** > **Azure Guest Users**.

The list of Azure guest users of the selected tenant is displayed.

Figure 117: Directory Management > Tree View > Azure > <azure-tenant-name> > Azure Guest Users — Listing the Azure guest users in the tenant



NOTE: Active Roles lists the available cloud-only Azure Users, Azure Guest Users, and Azure Contacts on the Active Roles Web Interface with the following restrictions:

- Active Roles can initially list 999 items.
- The items listed in the list have a sliding expiry of 8 hours, after which the objects that have not been accessed will be flushed.
- Whenever you perform a search in the list, Active Roles will always fetch the list of objects from Azure to update the cache.

Disabling or Enabling an Azure guest user

If you want to revoke the access of an Azure guest user from the resources, applications and roles assigned to them, you can disable their account without deleting them with the **Disable Account** action.

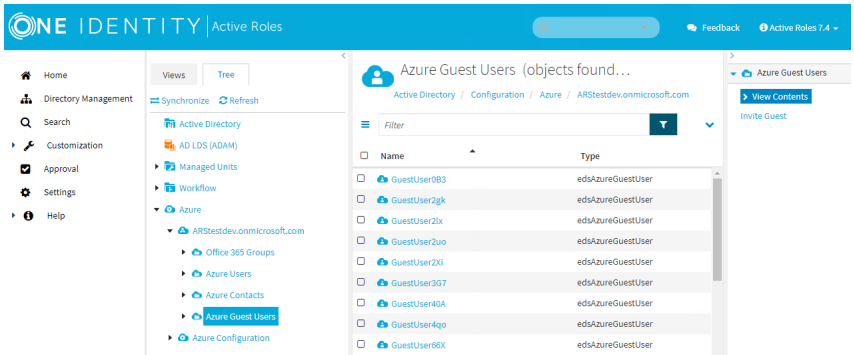
Likewise, once the revoked access rights of a disabled Azure guest user can be reinstated, you can re-enable them with the **Enable Account** action.

To disable or enable a cloud-only Azure guest user

1. Navigate to **Directory Management** > **Tree** > **Azure** > **<azure-tenant-name>** > **Azure Guest Users**.

   The list of Azure guest users of the selected tenant is displayed.

   Figure 118: Directory Management > Tree View > Azure > <azure-tenant-name> > Azure Guest Users — Listing the Azure guest users in the tenant



2. Select the Azure guest user that you want to enable or disable from the list.
3. Click the applicable option:
   - If the selected Azure guest user is enabled, click **Disable Account**.
   - If the selected Azure guest user is disabled, click **Enable Account**.

   NOTE: The available option changes depending on the state of the selected guest user account.
4. To confirm disabling/enabling the selected Azure guest user, click **Save**.

Revoking the session of an Azure guest user

You can revoke the current session of any selected cloud-only Azure guest user of an Azure tenant. When doing so, Active Roles clears the active login tokens of the guest user on all devices they have previously logged in from, forcing them to log in again and validate their credentials.

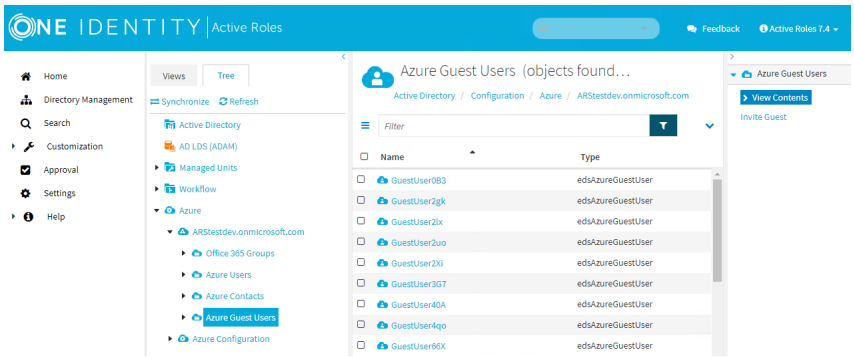
TIP: If any device that the Azure guest user has been previously logged in from has been compromised (for example, because the guest user has lost their notebook or cellphone), then One Identity recommends revoking their current session.

To revoke the active session of an Azure guest user

1. Navigate to **Directory Management** > **Tree** > **Azure** > **<azure-tenant-name>** > **Azure Guest Users**.

   The list of Azure guest users of the selected tenant is displayed.

   Figure 119: Directory Management > Tree View > Azure > <azure-tenant-name> > Azure Guest Users — Listing the Azure guest users in the tenant



2. Select the Azure guest user whose session you want to revoke.
3. Click **Revoke Session**.
4. To confirm revoking the session of the selected Azure guest user, click **Save**.

Resending the invitation to an Azure guest user

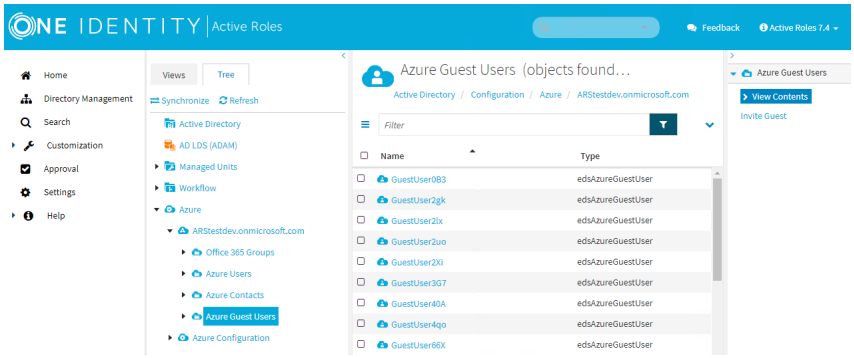
It can happen that the invitation email sent out at the end of the Inviting an Azure guest user procedure must be sent again to the Azure guest user (for example, because the guest user cannot access the specified email address for some reason, or because the previous invitation was accidentally deleted). In such cases, you can resend the invitation email.

To resend the invitation to an Azure guest user

1. Navigate to **Directory Management** > **Tree** > **Azure** > **<azure-tenant-name>** > **Azure Guest Users**.

   The list of Azure guest users of the selected tenant is displayed.

Figure 120: Directory Management > Tree View > Azure > <azure-tenant-name> > Azure Guest Users — Listing the Azure guest users in the tenant



2. Select the Azure guest user for which you want to resend the invitation.
3. Click **Resend Invitation**.

   Active Roles will then resend the invitation to the email address previously specified with the **Invite Guest** > **Email address** property.

Renaming an Azure guest user

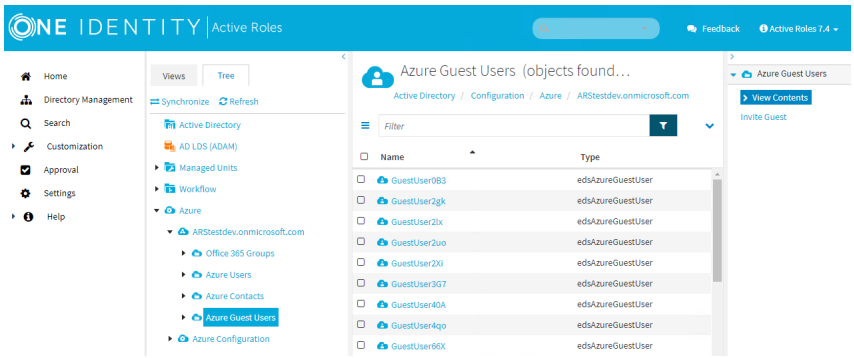
If an Azure guest user account must be renamed for any reason (for example, to fix a typo or an incorrect first/last name), you can use the **Rename** option of the Active Roles Web Interface.

To rename an Azure guest user

1. Navigate to **Directory Management** > **Tree** > **Azure** > **<azure-tenant-name>** > **Azure Guest Users**.

   The list of Azure guest users of the selected tenant is displayed.

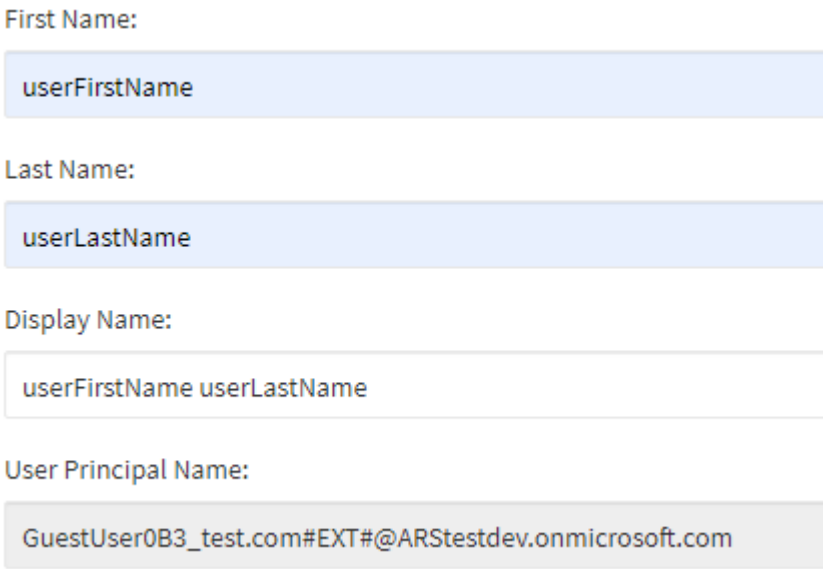
   Figure 121: Directory Management > Tree View > Azure > <azure-tenant-name> > Azure Guest Users — Listing the Azure guest users in the tenant



2. Select the Azure guest user that you want to rename.

3. To open the rename form, click **Rename**.

Figure 122: Azure Guest Users > Rename – Renaming an Azure guest user



4. Update the **First Name**, **Last Name** or **Display Name** of the guest user as needed.
5. To apply your changes, click **Finish**.

Viewing and updating the properties of an Azure guest user

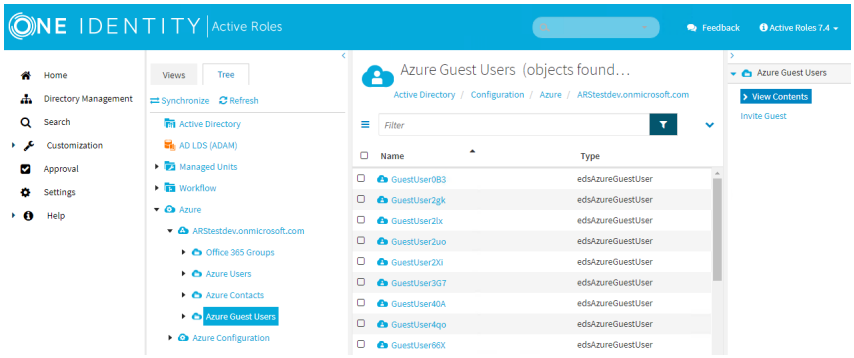
Once you configured and invited a new Azure guest user as described in Inviting an Azure guest user, you can modify their account settings with the **Azure Properties** option later if any change occurred to the user that must be reflected in their account.

To view and update the properties of an Azure guest user

1. Navigate to **Directory Management** > **Tree** > **Azure** > **<azure-tenant-name>** > **Azure Guest Users**.

   The list of Azure guest users of the selected tenant is displayed.

Figure 123: Directory Management > Tree View > Azure > <azure-tenant-name> > Azure Guest Users — Listing the Azure guest users in the tenant



2. From the list in the middle pane, select the Azure guest user that you want to update.
3. To open the properties of the selected Azure guest user, click **Azure Properties** on the right pane.
4. In the available **Azure Properties** pages, configure the Azure guest user settings that you want to change.

Table 109: Available Azure properties

Page	Description
Identity	View and configure user identity settings and information in this tab.
Settings	View and configure user authentication settings in this tab.
Job Info	View and configure job and organizational information in this tab.
Contact Info	View and configure contact and location information in this tab. NOTE: You can only update certain **Contact Info** properties (such as phone numbers or email addresses) for non-administrator Azure guest users, or for Azure guest users with a specific set of limited administrator roles. For more information on these roles, see the Update User page of the official Microsoft documentation. Attempting to update these properties for an Azure guest user with different administrative roles assigned to it will result in failure, and the following error log message appearing in the Windows Event Log: `Post-processing operation on object caused a policy violation.`
Licenses	View and configure the Microsoft application resources available in the organization to the Azure guest user.
O365 Admin Roles	View and configure the O365 roles in the organization granted for the guest user.

5. To apply your changes, click **Save**.

NOTE: Active Roles lists the available cloud-only Azure Users, Azure Guest Users, and Azure Contacts on the Active Roles Web Interface with the following restrictions:

- Active Roles can initially list 999 items.
- The items listed in the list have a sliding expiry of 8 hours, after which the objects that have not been accessed will be flushed.
- Whenever you perform a search in the list, Active Roles will always fetch the list of objects from Azure to update the cache.

Configuring the Identity settings of an Azure guest user

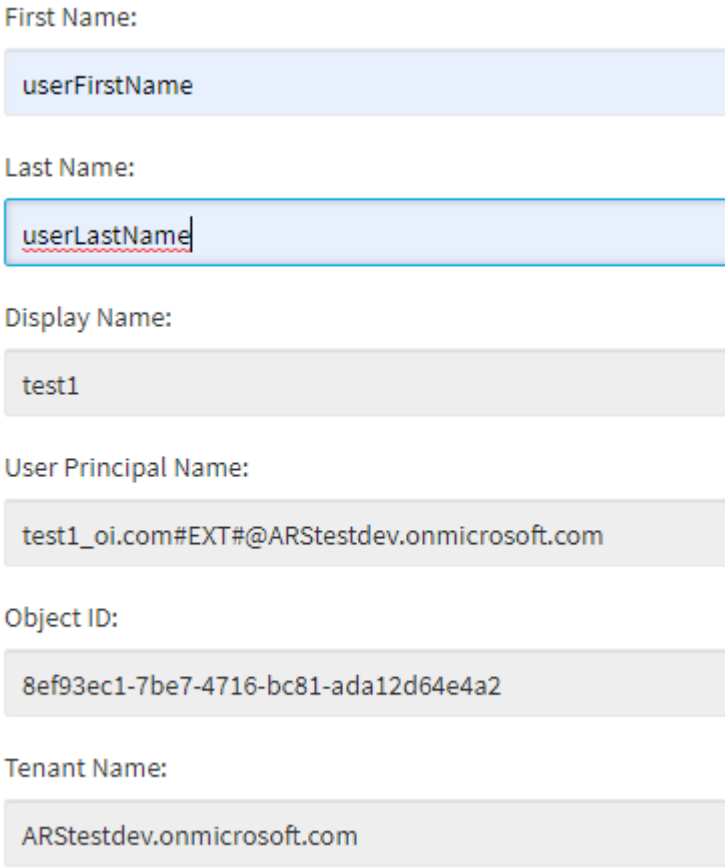
You can update the name settings of an Azure guest user in an Azure tenant with the **Azure Properties** > **Identity** tab.

NOTE: You can only change the **First Name** and **Last Name** settings of the guest user on this tab. You can change the rest of the identity settings when inviting the guest user. For more information, see Inviting an Azure guest user.

To update the settings of an existing Azure guest user

1. On the Active Roles Web Interface, navigate to **Directory Management** > **Tree View** > **Azure** > **<azure-tenant>** > **Azure Guest Users**.
2. From the list in the middle pane, select the Azure guest user that you want to update.
3. To open the properties of the selected Azure guest user, click **Azure Properties** on the right pane.
4. To open the name settings, click the **Identity** tab.

Figure 124: Directory Management > Tree View > Azure > *<azure-tenant>* > Azure Guest Users > *<azure-guest-user>* > Azure Properties > Identity — Configuring the identity-related information of an Azure Guest user



5. Enter the **First Name** of the Azure guest user. If no first name has been specified in this field when inviting the Azure guest user, this text box contains the local-part of the email address where the invite has been sent.
6. Enter the **Last Name** of the Azure guest user.
7. To apply your changes, click **Save**.

NOTE: You can also view the following identity properties of the selected Azure guest user on this page:

- **Display Name**: Shows the display name of the Azure guest user. By default, the display name consists of the specified **First Name** and **Last Name**.

  TIP: You cannot directly modify the **Display Name** of the guest user on this tab. To do that, use the **Rename** action. For more information, see Renaming an Azure guest user.

- **User Principal Name**: Displays the User Principal Name (UPN) of the Azure guest user. The UPN has the following syntax:

  ```
  <azure-guest-user-email-address>#EXT#@<azure-tenant>
  ```

- **Object ID**: Displays the object ID of the Azure guest user
- **Tenant Name**: Displays the Azure tenant containing the Azure guest user.

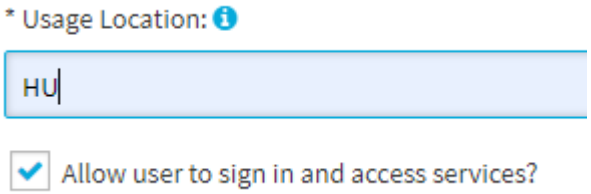
Configuring the Settings of an Azure guest user

You can update the authentication settings of an Azure guest user in an Azure tenant with the **Azure Properties** > **Settings** tab. You must modify these settings typically when the geographical location of the guest user has changed (for example, because they have moved to an office located in another country), or if the guest user has received no access to the configured roles and licenses when their account has originally been created.

To configure the authentication settings of an existing Azure guest user

1. On the Active Roles Web Interface, navigate to **Directory Management** > **Tree View** > **Azure** > **<azure-tenant>** > **Azure Guest Users**.
2. From the list in the middle pane, select the Azure guest user that you want to update.
3. To open the properties of the selected Azure guest user, click **Azure Properties** on the right pane.
4. To open the user authentication settings, click the **Identity** tab.

   Figure 125: Directory Management > Tree View > Azure > *<azure-tenant>* > Azure Guest Users > *<azure-guest-user>* > Azure Properties > Settings — Accessing the authentication settings of an Azure Guest user

5. To restrict the login attemps with the configured Azure guest user account to a specific geographical location, enter the corresponding ISO 3166 country code in the **Usage Location** field. Active Roles will then only allow the guest user to log in, if the login attempt occurs from the country that you specified.
6. (Optional) To grant the Azure guest user access to the configured licenses and admin roles, select **Allow user to sign in and access services**. If access has been granted previously, and must be revoked, then deselect this option.

> TIP: Leaving this setting clear is useful if the account of the Azure guest user is created in advance, and they require access to the assigned resources only later (for example, because their contract project starts only at a later date).

7. To apply your changes, click **Save**.

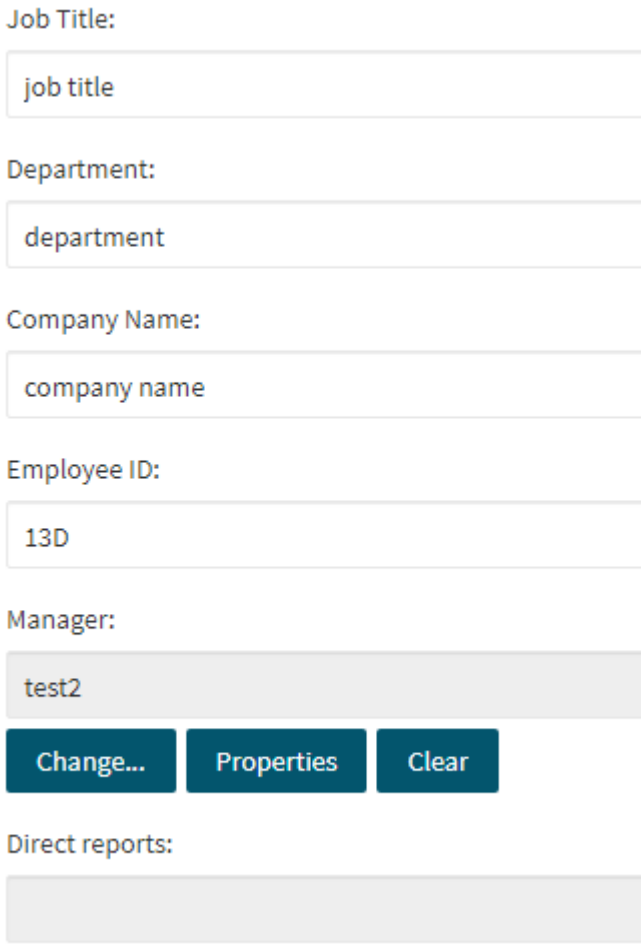
Configuring the Job Info settings of an Azure guest user

You can configure job and organizational information for an existing Azure guest user in an Azure tenant with the **Azure Properties** > **Job Info** tab. This is typically required if the employment status of the guest user changes, for example their position, assigned department or employee ID is modified for some reason.

To modify the job information of an existing Azure guest user

1. On the Active Roles Web Interface, navigate to **Directory Management** > **Tree View** > **Azure** > **<azure-tenant>** > **Azure Guest Users**.
2. From the list in the middle pane, select the Azure guest user that you want to update.
3. To open the properties of the selected Azure guest user, click **Azure Properties** on the right pane.
4. To open the job information settings, click the **Job Info** tab.

Figure 126: Directory Management > Tree View > Azure > <*azure-tenant*> > Azure Guest Users > <*azure-guest-user*> > Azure Properties > Job Info — Accessing the organizational settings of an Azure Guest user



5. (Optional) Specify the **Job Title** of the guest user.
6. (Optional) Specify the **Department** of the guest user to which they are assigned.
7. (Optional) Specify the assigned company of the guest user with the **Company Name** setting. For example, this can be either the external company that employs the guest user, or a specific company-size unit within your organization that is contracting them.
8. (Optional) Specify the **Employee ID** of the guest user, if they have one issued.
9. (Optional) Specify the **Manager** the guest user reports to. Use **Change...** to specify or change the manager, click **Properties** to view information about the currently specified manager, or click **Clear** to remove the current selection.
10. To apply your changes, click **Save**.

NOTE: The **Job Info** also has a **Direct reports** field that lists the employees or other guest users reporting to the selected guest user, if there are any.

Configuring the Contact Info settings of an Azure guest user

You can modify the contact and location information (such as phone number, address, office location) of an Azure guest user in an Azure tenant with the **Azure Properties** > **Contact Info** tab. This is typically required if the organization requires detailed contact information for the guest user, or if any previously-configured contact information has been changed.

NOTE: You can only update certain **Contact Info** properties (such as phone numbers or email addresses) for non-administrator Azure guest users, or for Azure guest users with a specific set of limited administrator roles. For more information on these roles, see the Update User page of the official Microsoft documentation.

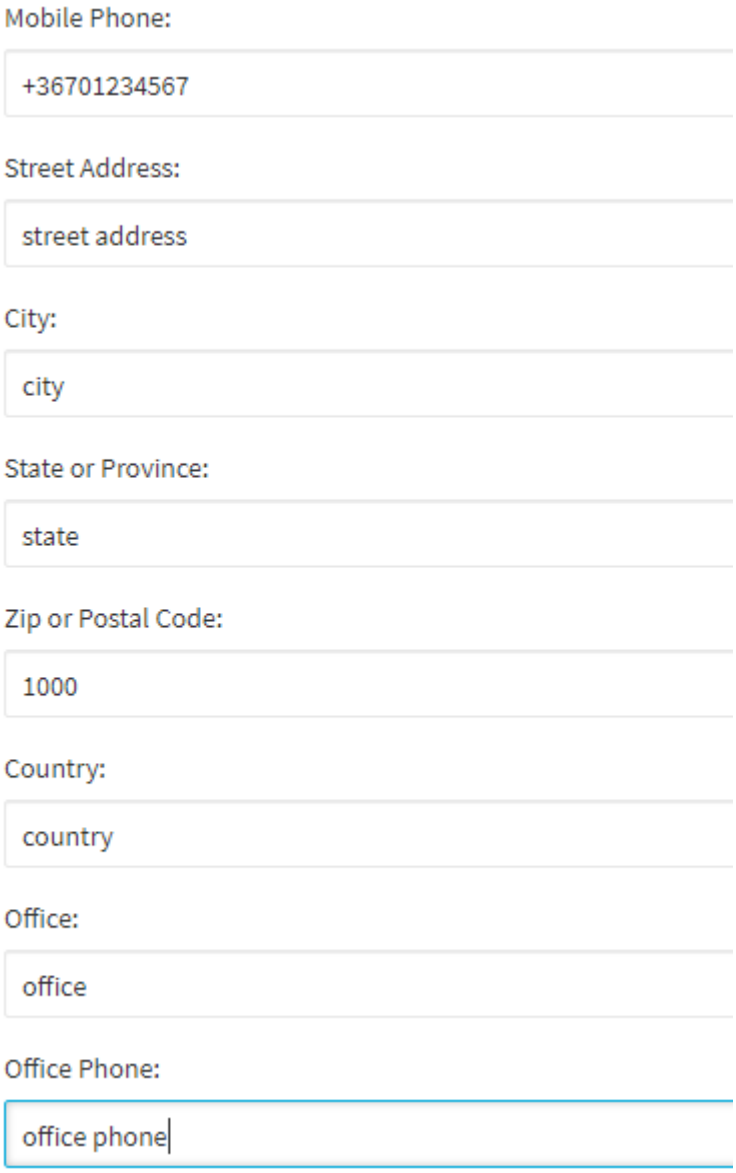
Attempting to update these properties for an Azure guest user with different administrative roles assigned to it will result in failure, and the following error log message appearing in the Windows Event Log:

```
Post-processing operation on object caused a policy violation.
```

To modify the contact information of an existing Azure guest user

1. On the Active Roles Web Interface, navigate to **Directory Management** > **Tree View** > **Azure** > **<azure-tenant>** > **Azure Guest Users**.
2. From the list in the middle pane, select the Azure guest user that you want to update.
3. To open the properties of the selected Azure guest user, click **Azure Properties** on the right pane.
4. To open the contact information settings, click the **Contact Info** tab.

Figure 127: Directory Management > Tree View > Azure > *<azure-tenant>* > Azure Guest Users > *<azure-guest-user>* > Azure Properties > Contact Info — Accessing the contact and location settings of an Azure Guest user



5. (Optional) Specify the **Mobile Phone** number of the guest user.
6. (Optional) Specify the **Street Address** of the guest user.
7. (Optional) Specify the **City** where the guest user is located.
8. (Optional) Specify the **State or Province** where the guest user is located.
9. (Optional) Specify the **Zip or Postal Code** of the location of the guest user.
10. (Optional) Specify the **Country** where the guest user is located.

11. (Optional) Specify the **Office** where the guest user is located.
12. (Optional) Specify the **Office Phone** number of the guest user, if one is issued to them.
13. To apply your changes, click **Save**.

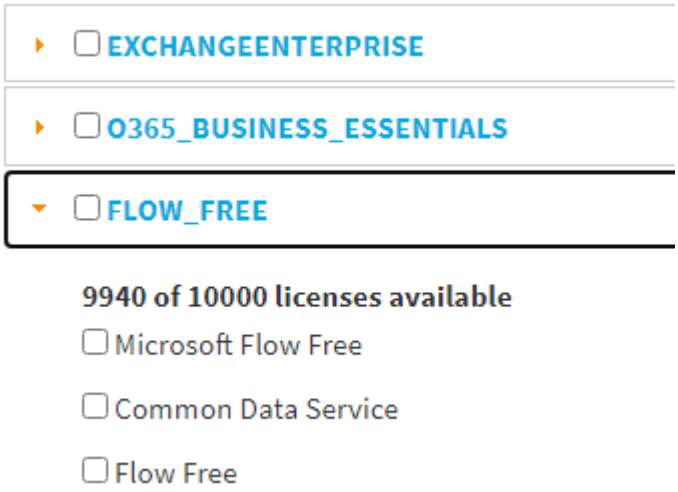
Configuring the Licenses settings of an Azure guest user

You can assign or unassign any of the Microsoft application resources in an organization to an existing Azure guest user in the **Azure Properties** > **Licenses** tab. This is typically required if the previously-configured application licenses must be modified, for example because of changes in the assignment of the guest user.

To configure the application licenses of an existing Azure guest user

1. On the Active Roles Web Interface, navigate to **Directory Management** > **Tree View** > **Azure** > **<azure-tenant>** > **Azure Guest Users**.
2. From the list in the middle pane, select the Azure guest user that you want to update.
3. To open the properties of the selected Azure guest user, click **Azure Properties** on the right pane.
4. To open the application license settings, click the **Licenses** tab.

   Figure 128: Directory Management > Tree View > Azure > *<azure-tenant>* > Azure Guest Users > *<azure-guest-user>* > Azure Properties > Licenses — Accessing the application license settings of an Azure Guest user

5. (Optional) If the available licenses are categorized into various headings, expand the list of the license(s) you want to add or remove from the guest user.

6. Select the license(s) you want to assign to the guest user, or deselect the one(s) you want to remove from them.
7. To apply your changes, click **Save**.

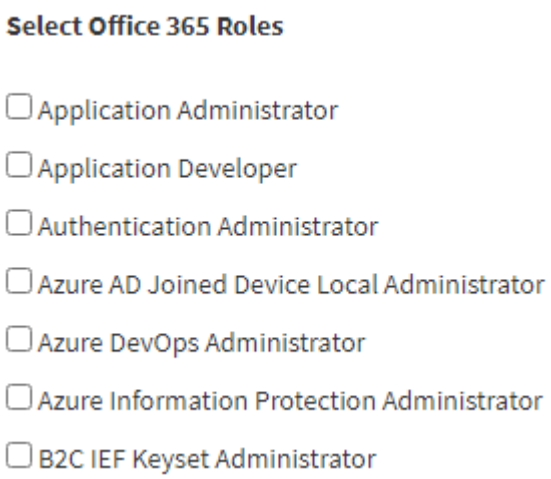
Configuring the O365 Admin Roles settings of an Azure guest user

You can grant (or revoke) O365 administration roles to (or from) an existing Azure guest user in the **Azure Properties** > **O365 Admin Roles** tab. This is typically required either when the assignment of a guest user changes, or when it is finished.

To configure the O365 admin roles of an existing Azure guest user

1. On the Active Roles Web Interface, navigate to **Directory Management** > **Tree View** > **Azure** > **<azure-tenant>** > **Azure Guest Users**.
2. From the list in the middle pane, select the Azure guest user that you want to update.
3. To open the properties of the selected Azure guest user, click **Azure Properties** on the right pane.
4. To open the administration role settings, click the **O365 Admin Roles** tab.

   Figure 129: Directory Management > Tree View > Azure > *<azure-tenant>* > Azure Guest Users > *<azure-guest-user>* > Azure Properties > O365 Admin Roles — Accessing the administrator role settings of an Azure Guest user

5. Select the administrator role(s) you want to grant for the guest user, or deselect the role(s) you want to revoke.

   NOTE: You can assign roles to the Azure guest user in Active Roles without any limitation. However, One Identity recommends that you assign Azure guest users only the admin roles that external contractors typically receive in your

organization.

6. To apply your changes, click **Save**.

Viewing or updating the Exchange Online properties of an Azure guest user

You can create, modify or view the Exchange Online properties of an existing Azure guest user with the **Exchange Online** option of the Active Roles Web Interface. With the Exchange Online properties, you can configure various mailbox-related settings for the guest user, such as:

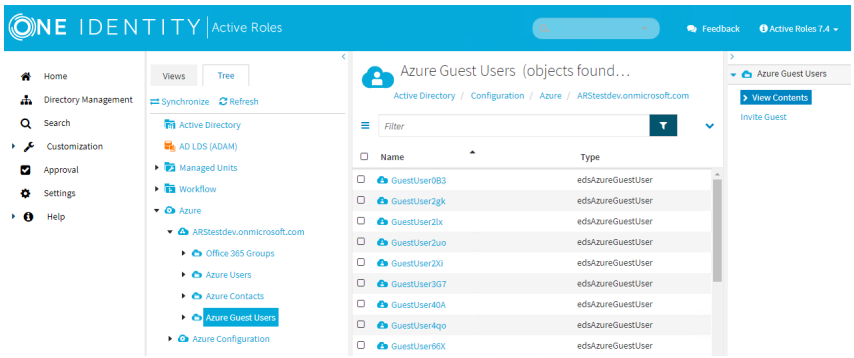
- The name of the email address.
- Message size and delivery rules.
- Setting up the guest user mailbox as a shared mailbox.
- Enabling or disabling various applications (such as Outlook Web App) or protocols (such as MAPI, IMAP4, or POP3) for the mailbox.
- Configuring Messaging Records Management (MRM) settings for the guest user mailbox.

To view and update the Exchange Online properties of an Azure guest user

1. Navigate to **Directory Management** > **Tree** > **Azure** > **<azure-tenant-name>** > **Azure Guest Users**.

   The list of Azure guest users of the selected tenant is displayed.

   Figure 130: Directory Management > Tree View > Azure > <azure-tenant-name> > Azure Guest Users — Listing the Azure guest users in the tenant



2. Select the guest user whose Exchange Online properties you want to check or update.
3. To access the Exchange Online-specific mailbox settings, click **Exchange Online Properties**.

4. In the available **Exchange Online Properties** tabs, configure the Exchange Online mailbox settings.

   Table 110: Available Exchange Online properties

Page	Description
Mail Flow Settings	View and configure rules for the emails that the Azure guest user sends or receives via the Exchange Online service of the organization.
Delegation	Configure the email account of the Azure guest user as a shared mailbox.
E-mail Addresses	View and configure email addresses for the selected Azure guest user.
Mailbox Features	View and configure various Exchange Online mailbox features for the Azure guest user.
Mailbox Settings	View and configure Messaging Records Management (MRM) settings for the Azure guest user.

5. To apply your changes, click **Close**.

Configuring the Mail Flow Settings of an Azure guest user

You can set up rules for the emails that the Azure guest user sends or receives in the organization in the **Exchange Online Properties** > **Mail Flow Settings** tab of the Active Roles Web Interface. Active Roles supports setting up two types of such rules:

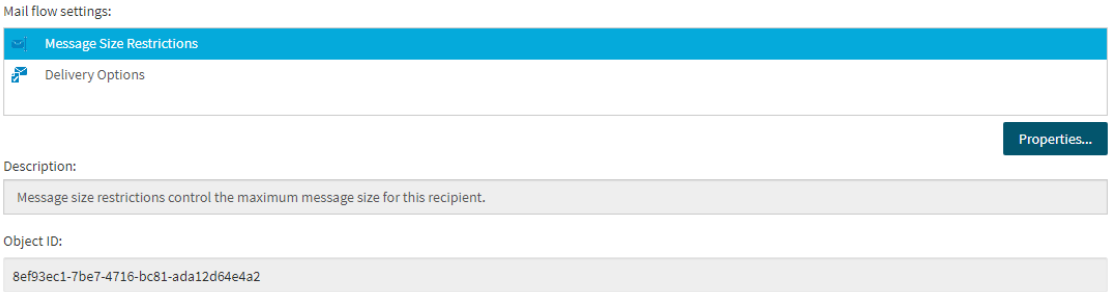
- Message size settings, specifying the size of the emails that the guest user can send or receive.
- Email delivery and forwarding settings, allowing the guest user to send emails on behalf of a specified group, or have their received emails automatically forwarded to an additional specified address.

Such mail flow settings are typically configured if the organization enforces specific email messaging policies for users (and guest users).

To configure the Mail Flow Settings for an existing Azure guest user

1. On the Active Roles Web Interface, navigate to **Directory Management** > **Tree View** > **Azure** > **<azure-tenant>** > **Azure Guest Users**.
2. From the list in the middle pane, select the Azure guest user that you want to update.
3. To open the Exchange Online properties of the selected Azure guest user, click **Exchange Online Properties** on the right pane.
4. To open the mail flow settings, click the **Mail Flow Settings** tab.

Figure 131: Directory Management > Tree View > Azure > <azure-tenant> > Azure Guest Users > <azure-guest-user> > Exchange Online Properties > Mail Flow Settings — Configuring the message size and forwarding settings of an Azure Guest user



5. Select **Message Size Restrictions**, and click **Properties...**.
6. Configure the size of the emails (in KB) that are sent or received by the Azure guest user. By default, both the **Sending message size** and the **Receiving message size** settings use the default limit of the Azure tenant.
7. To apply your changes and close the **Message Size Restrictions** dialog, click **Save**.
8. (Optional) Select **Delivery Options**, and click **Properties...** to configure the following email delivery and forwarding settings.
   - **Send on Behalf**: When configured, the guest user is allowed to send emails on behalf of the specified mailbox or group.
   - **Forwarding Address**: When configured, the emails received by the guest user are always forwarded to the specified email address.
9. (Optional) To apply any changes you made in the **Delivery Options** dialog, click **Save**.
10. To close the **Exchange Online Properties** window, click **Close**.

Configuring the Delegation settings of an Azure guest user

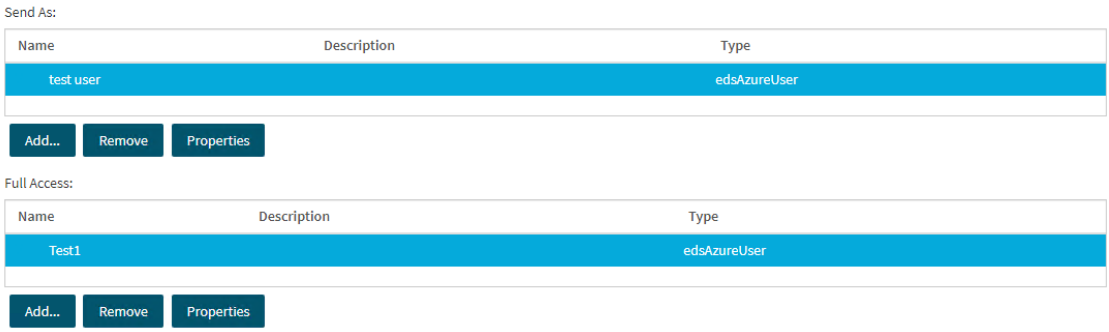
You can set up the email account of the selected Azure guest user as a shared mailbox in the **Exchange Online Properties** > **Delegation** tab of the Active Roles Web Interface. This is typically performed if the configured email account is used as a group account, such a common support or information email address.

Active Roles supports granting *Send as* and *Full access* permissions to the specified users. For more information on shared mailboxes and these permissions, see the Shared mailboxes in Exchange Online page of the official Microsoft documentation.

To configure the email delegation settings of an Azure guest user

1. On the Active Roles Web Interface, navigate to **Directory Management** > **Tree View** > **Azure** > **<azure-tenant>** > **Azure Guest Users**.
2. From the list in the middle pane, select the Azure guest user that you want to update.
3. To open the Exchange Online properties of the selected Azure guest user, click **Exchange Online Properties** on the right pane.
4. To open the delegation settings, click the **Delegation** tab.

   Figure 132: Directory Management > Tree View > Azure > <azure-tenant> > Azure Guest Users > <azure-guest-user> > Exchange Online Properties > Delegation — Accessing the email account delegation settings of an Azure Guest user



5. To delegate *Send as* permission to a user (or users), click **Add...** under the **Send As** list.
6. Select the user(s) you wish to grant *Send as* rights for the email address of the Azure guest user, then click **OK**.
7. To delegate *Send as* permission to a user (or users) click **Add...** under the **Full Access** list.
8. Select the user(s) you wish to grant *Full access* rights for the email address of the Azure guest user, then click **OK**.
9. To remove a delegated user either from the **Send As** or **Full Access** list, click **Remove** and select the user(s) you want to revoke the permission from.
10. To close the **Exchange Online Properties** window, click **Close**.

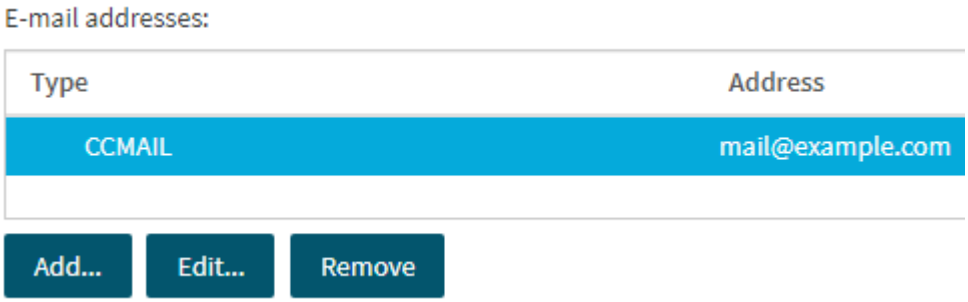
Configuring Email Address settings for Azure guest users

You can add, edit or remove an email address to or from an Azure guest user in the **Exchange Online Properties** > **E-Mail Addresses** tab of the Active Roles Web Interface. Adding, editing, or removing an email address to or from the guest user is typically required when their employment status changes (for example, they are either assigned to a new project, or their contract ends within the organization).

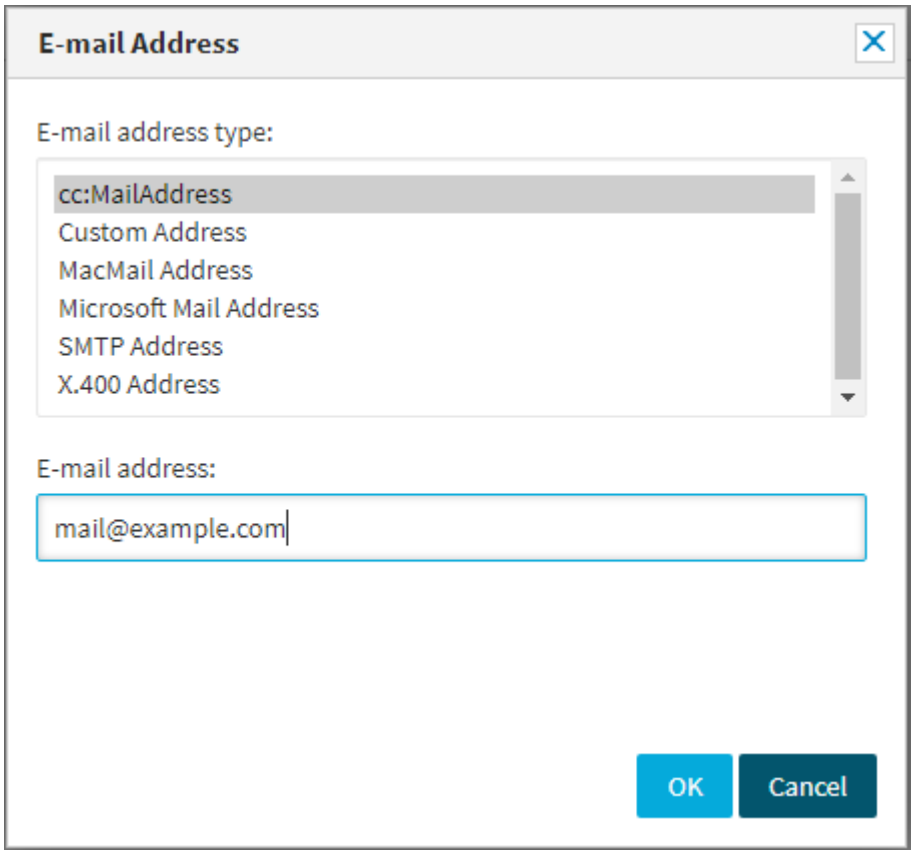
To add a new email address to an Azure guest user

1. On the Active Roles Web Interface, navigate to **Directory Management** > **Tree View** > **Azure** > **<azure-tenant>** > **Azure Guest Users**.
2. From the list in the middle pane, select the Azure guest user that you want to update.
3. To open the Exchange Online properties of the selected Azure guest user, click **Exchange Online Properties** on the right pane.
4. To open the email address settings, click the **E-Mail Addresses** tab.

   Figure 133: Directory Management > Tree View > Azure > <azure-tenant> > Azure Guest Users > <azure-guest-user> > Exchange Online Properties > E-mail Addresses — Accessing the email account settings of an Azure Guest user

5. Click **Add...**. The **E-mail Address** dialog then opens.



6. From the **E-mail address type** list, select the email account type applicable to your organization.
7. In the **E-mail address** text box, specify the address of the new account.
8. To apply your changes and create the new email account, click **OK**.
9. To close the **Exchange Online Properties** window, click **Close**.

To edit an existing email address of an Azure guest user

1. On the Active Roles Web Interface, navigate to **Directory Management** > **Tree View** > **Azure** > **<azure-tenant>** > **Azure Guest Users**.
2. From the list in the middle pane, select the Azure guest user that you want to update.
3. To open the Exchange Online properties of the selected Azure guest user, click **Exchange Online Properties** on the right pane.
4. To open the email address settings, click the **E-Mail Addresses** tab.
5. Click **Edit...**. The **E-mail Address** dialog then opens.
6. In the **E-mail address** text box, specify the address of the new account.

   NOTE: You cannot modify the **E-mail address type** of an existing email account. You can only change the existing address.
7. To apply your changes and create the new email account, click **OK**.
8. To close the **Exchange Online Properties** window, click **Close**.

To remove an existing email address of an Azure guest user

1. On the Active Roles Web Interface, navigate to **Directory Management** > **Tree View** > **Azure** > **<azure-tenant>** > **Azure Guest Users**.
2. From the list in the middle pane, select the Azure guest user that you want to update.
3. To open the Exchange Online properties of the selected Azure guest user, click **Exchange Online Properties** on the right pane.
4. To open the email address settings, click the **E-Mail Addresses** tab.
5. In the **E-mail addresses** list, select the account you want to remove.
6. Click **Remove** and confirm the deletion of the account.
7. To close the **Exchange Online Properties** window, click **Close**.

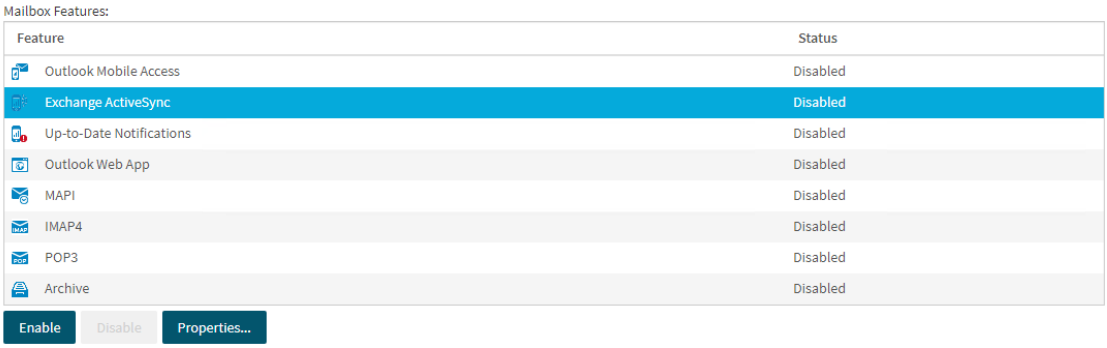
Configuring Mailbox Features for Azure guest users

You can enable or disable various Exchange Online mailbox features for the Azure guest user (such as Outlook Mobile Access or support for messaging protocols like IMAP4 or POP3) in the **Exchange Online Properties** > **Mailbox Features** tab of the Active Roles Web Interface. This is typically required if the organization supports specific applications and protocols for the user Exchange mailboxes created for the users and guest users.

To enable or disable Exchange Online Mailbox Features for an Azure guest user

1. On the Active Roles Web Interface, navigate to **Directory Management** > **Tree View** > **Azure** > **<azure-tenant>** > **Azure Guest Users**.
2. From the list in the middle pane, select the Azure guest user that you want to update.
3. To open the Exchange Online properties of the selected Azure guest user, click **Exchange Online Properties** on the right pane.
4. To open the mailbox feature settings, click the **Mailbox Features** tab.

Figure 134: Directory Management > Tree View > Azure > <azure-tenant> > Azure Guest Users > <azure-guest-user> > Exchange Online Properties > Mailbox Features — Configuring mailbox features for an Azure Guest user



5. Select the Exchange Online mailbox feature that you want to enable or disable:
   - **Outlook Mobile Access**: Enables or disables the Outlook Mobile Access (OMA) mobile browsing service for the account of the Azure guest user. Enabling this settings lets the Azure guest user use OMA on their mobile device to access their account.
   - **Exchange ActiveSync**: Enables or disables the Exchange ActiveSync synchronization protocol for the account of the Azure guest user. Enabling this setting lets the Azure guest user synchronize their configured mobile device with their mailbox.
   - **Up-to-Date Notifications**: Enables or disables the Up-to-date (UTD) feature notifications for the account of the Azure guest user.
   - **Outlook Web App**: Enables or disables access to the browser-based Outlook Web App for the account of the Azure guest user.
   - **MAPI**, **IMAP4**, **POP3**: Enables or disables support for the MAPI, IMAP4 or POP3 protocols for the account of the Azure guest user. If MAPI is enabled, the Azure guest user can access their mailbox through the Outlook desktop app (or other MAPI clients). If IMAP4 or POP3 is enabled, they are also able to access their mailbox with any IMAP4 or POP3 email client.
   - **Archive**: Enables or disables the archive mailbox feature for the account of the Azure guest user.
6. Click **Enable** to enable the selected mailbox feature, or **Disable** to disable it.
7. Once you are done with the configuration, click **Close**.
8. To close the **Exchange Online Properties** window, click **Close**.

Configuring Mailbox Settings for Azure guest users

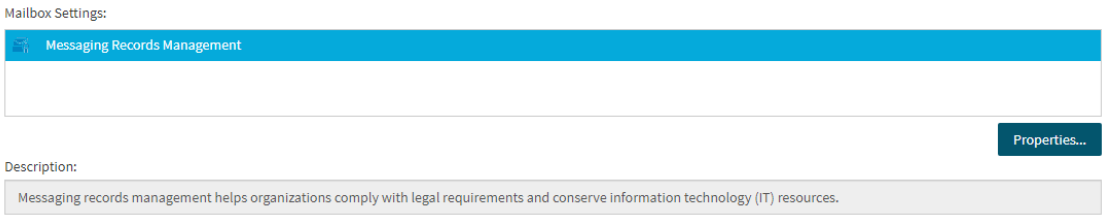
You can configure settings related to Messaging Records Management (MRM) for the email account of the selected Azure guest user in the **Exchange Online Properties** > **Mailbox Settings** tab of the Active Roles Web Interface. MRM settings must be typically configured to meet mailbox archiving policies in effect within the organization.

For more information about MRM in Exchange Online, see the Messaging records management page of the official Microsoft documentation.

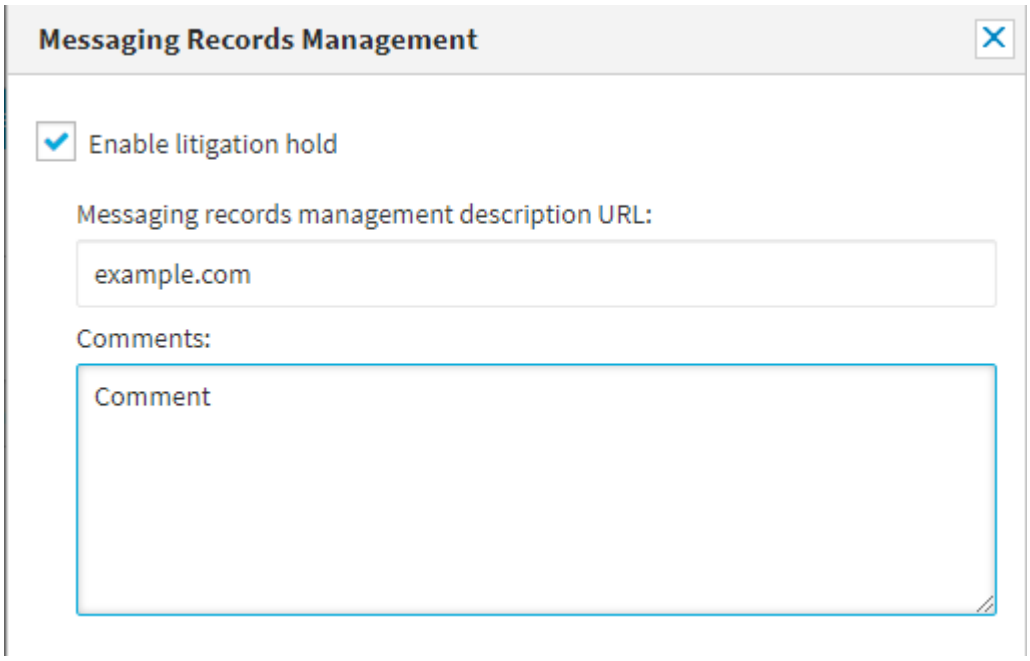
To configure Messaging Records Management settings for an Azure guest user

1. On the Active Roles Web Interface, navigate to **Directory Management** > **Tree View** > **Azure** > **<azure-tenant>** > **Azure Guest Users**.
2. From the list in the middle pane, select the Azure guest user that you want to update.
3. To open the Exchange Online properties of the selected Azure guest user, click **Exchange Online Properties** on the right pane.
4. To open the MRM settings, click the **Mailbox Settings** tab.

   Figure 135: Directory Management > Tree View > Azure > <azure-tenant> > Azure Guest Users > <azure-guest-user> > Exchange Online Properties > Mailbox Settings — Accessing the MRM settings of an Azure Guest user

5. Under **Mailbox Settings**, make sure that **Messaging Records Management** is selected, then click **Properties...**. The **Messaging Records Management** dialog opens.



6. To enable placing the entire contents of the Azure guest user mailbox on hold, enable the **Enable litigation hold** check box. For more information on the Litigation Hold feature of Exchange Online, see the In-Place Hold and Litigation Hold page of the official Microsoft documentation.
7. (Optional) If your organization has an internal resource on the litigation hold practices, specify its URL in the **Messaging records management description URL** text box.
8. (Optional) If you want to display a customized message in Outlook for the Azure guest user on the litigation hold, write the message in the **Comments** text box.
9. Click **Save** to apply your changes and close the **Messaging Records Management** dialog.
10. To close the **Exchange Online Properties** window, click **Close**.

Deleting an Azure guest user

You can delete an Azure guest user in the selected Azure tenant with the **Delete** option of the Active Roles Web Interface. This is typically performed if the guest user no longer works for the organization.

NOTE: You can only remove certain Azure guest users (for example, Global Administrators) if you have sufficient administrator roles. For more information on these role requirements, see the official Microsoft documentation.

Attempting to delete an Azure guest user without sufficient administrative privileges will result in failure, and the following error log message appearing in the Windows Event Log:

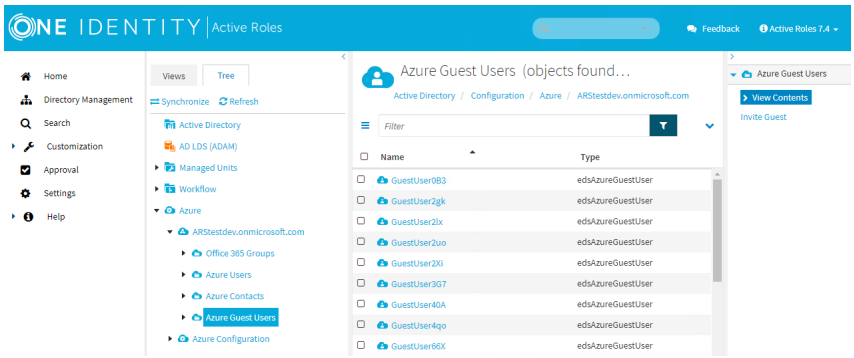
```
Post-processing operation on object caused a policy violation.
```

To delete an Azure guest user

1. Navigate to **Directory Management** > **Tree** > **Azure** > **<azure-tenant-name>** > **Azure Guest Users**.

   The list of Azure guest users of the selected tenant is displayed.

   Figure 136: Directory Management > Tree View > Azure > <azure-tenant-name> > Azure Guest Users — Listing the Azure guest users in the tenant



2. Select the Azure guest user that you want to delete.
3. Click **Delete**.
4. To confirm the removal of the guest user, click **Yes**.

Configuring the O365 Group membership of an Azure guest user

You can configure and view the Azure group membership(s) of an Azure guest user with the **Azure Member Of** option of the Active Roles Web Interface. Using this option, you can:

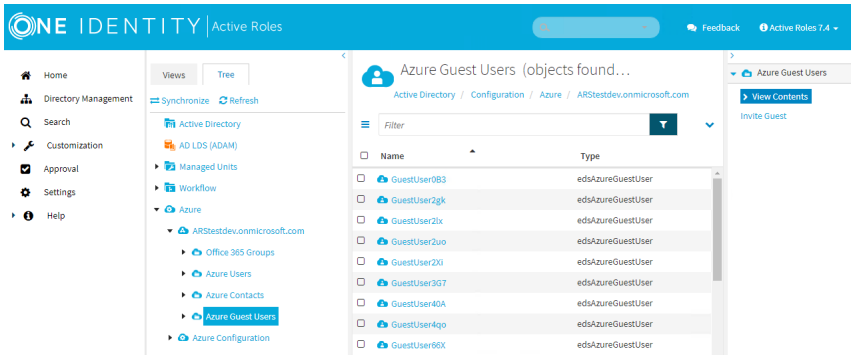
- View the existing O365 group membership(s) of the Azure guest user.
- Add or remove the Azure guest user to or from the selected Azure O365 Group(s).

To add or remove an existing Azure guest user to or from an O365 Group

1. Navigate to **Directory Management** > **Tree** > **Azure** > **<azure-tenant-name>** > **Azure Guest Users**.

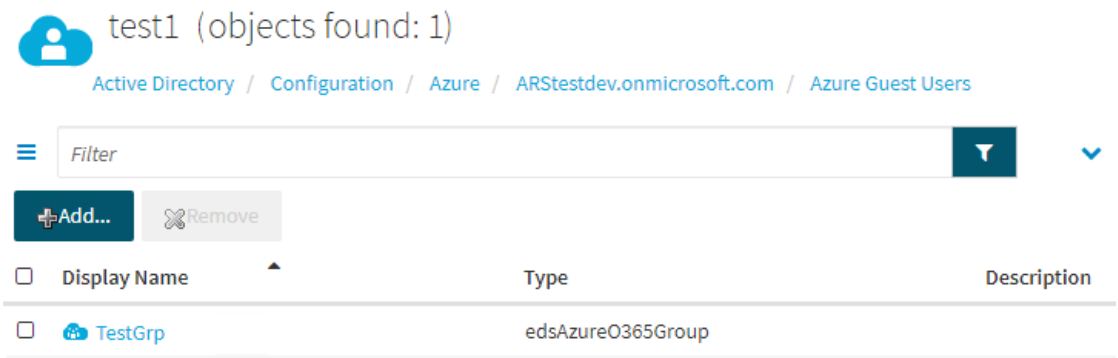
   The list of Azure guest users of the selected tenant is displayed.

Figure 137: Directory Management > Tree View > Azure > <azure-tenant-name> > Azure Guest Users — Listing the Azure guest users in the tenant



2. In the middle pane, select the Azure guest user whose membership you want to view or configure.
3. In the right pane, click **Azure Member Of**. The list of Azure O365 groups where the guest user has a membership then appears.

Figure 138: Directory Management > Tree View > Azure > <azure-tenant> > Azure Guest Users > Azure Member Of – Listing the Azure groups of the selected Azure Guest user



4. To add the Azure guest user to a new Azure O365 group of the Azure tenant, click **Add**.
5. In the **Select Object** page, select the O365 Group(s) you want the Azure guest user to be a member of, then click **OK** to apply your changes and return to the **Azure Member Of** page. The list is then updated with the new groups that you selected previously.
6. To remove the Azure guest user from any O365 Group(s), select the group(s) in the **Azure Member Of** page, and then click **Remove**. Click **OK** to confirm the removal from the group.

Viewing the change history of an Azure guest user

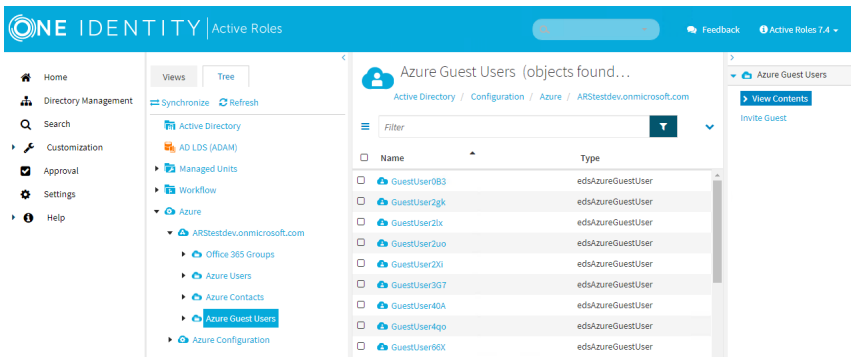
You can view the change history of an Azure guest user in the selected Azure tenant with the **Change History** option of the Active Roles Web Interface.

To view the change history of an Azure guest user

1. Navigate to **Directory Management** > **Tree** > **Azure** > **<azure-tenant-name>** > **Azure Guest Users**.

   The list of Azure guest users of the selected tenant is displayed.

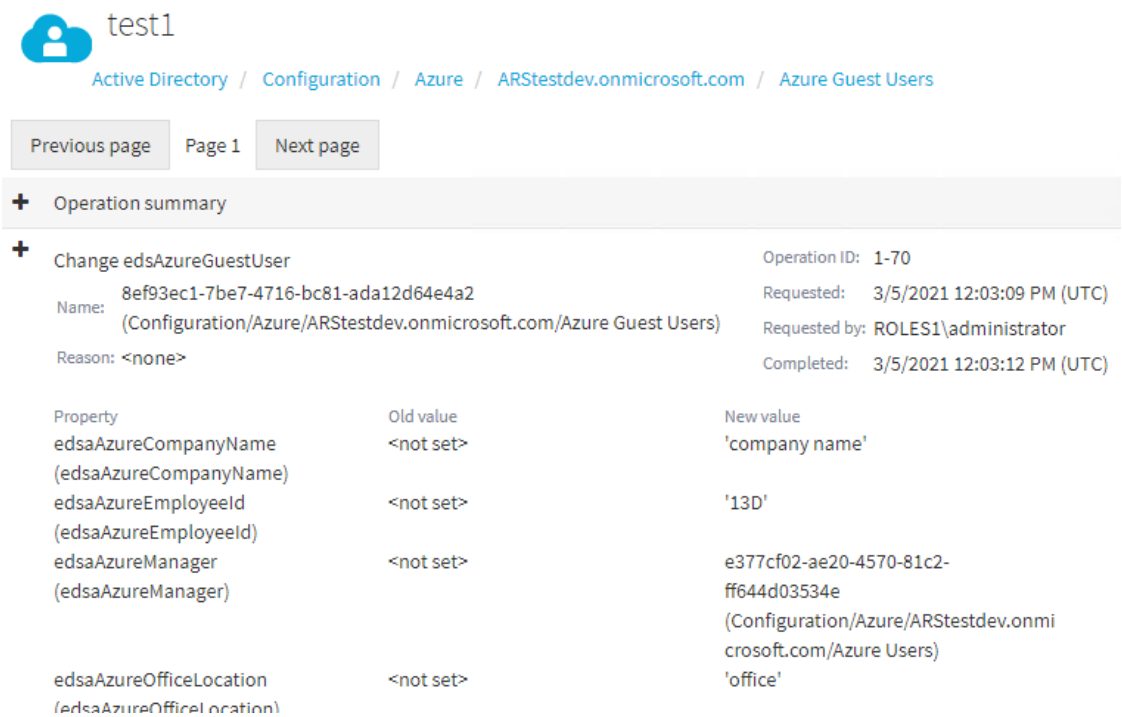
   Figure 139: Directory Management > Tree View > Azure > <azure-tenant-name> > Azure Guest Users — Listing the Azure guest users in the tenant



2. Select the Azure guest user whose change history you want to check.
3. Click **Change History**.

   The change history of the Azure guest user then appears.

Figure 140: Directory Management > Tree View > Azure > <azure-tenant> > Azure Guest Users > Change History – Viewing the change history of the selected Azure guest user



Managing cloud-only Azure contacts

Active Roles provides the facility to perform administrative tasks such as create, read, update, and delete Azure contact on cloud through web interface. You can also perform other operations such as viewing and modifying the Azure cloud-only contact, view change history, and other operations related to Azure cloud-only contacts.

NOTE: When creating a new cloud-only Azure contact or updating an existing one, it may take up to 15 minutes for the changes to appear on the Active Roles Web Interface. This is due to a replication delay present between PowerShell and the Microsoft Graph API.

View cloud only Azure contacts

You can use the Active Roles Web Interface to view the cloud only Azure contacts.

To view the cloud only Azure contacts

1. On the Active Roles Web interface **Navigation** bar, click **Directory Management**.
2. On the **Views** tab in the **Browse** pane, click | **Azure** |***<Azure tenant>*** |**Azure Contacts**.

   NOTE: Active Roles lists the available cloud-only Azure Users, Azure Guest Users, and Azure Contacts on the Active Roles Web Interface with the following restrictions:

   - Active Roles can initially list 999 items.
   - The items listed in the list have a sliding expiry of 8 hours, after which the objects that have not been accessed will be flushed.
   - Whenever you perform a search in the list, Active Roles will always fetch the list of objects from Azure to update the cache.

Create new cloud only Azure contacts

You can use the Active Roles Web Interface to create and enable new cloud only Azure contacts.

To create a new cloud only Azure contact

1. On the Active Roles Web interface **Navigation** bar, click **Directory Management**.
2. On the **Views** tab in the **Browse** pane, click **Azure** |***<Azure tenant>*** |**Azure Contacts**.

   The Azure Contacts page is displayed and lists the Azure cloud only contacts available in Azure.
3. In the **Command** pane, under Azure Contacts, click **New Contact**.
4. In the New Contact window, on the **General** tab, enter the appropriate text in the **Name**, **Alias**, and **Description** fields.
5. Click **Finish**.

   The Azure Contacts page displays the newly added Azure contact.

NOTE: When creating a new cloud-only Azure contact or updating an existing one, it may take up to 15 minutes for the changes to appear on the Active Roles Web Interface. This is due to a replication delay present between PowerShell and the Microsoft Graph API.

View or modify Azure contacts properties

For an existing Azure cloud only contact, you can use the Active Roles Web Interface to view or modify the properties.

To view or modify the Azure contacts properties

1. On the Active Roles Web interface **Navigation** bar, click **Directory Management**.
2. On the **Views** tab in the **Browse** pane, click | **Azure** |***<Azure tenant>*** |**Azure Contacts**.

   The Azure contacts page is displayed and lists the Azure contacts available in Azure.
3. Select the Azure contact for which you want to view or modify the properties.
4. In the **Command** pane, click **Azure properties**.

   The **Azure Properties** wizard for the contact is displayed.
5. Use the tabs in the **Azure Properties** wizard to view or modify properties of the Azure cloud only contact.
6. After setting all the required properties, click **Save**.

NOTE: When creating a new cloud-only Azure contact or updating an existing one, it may take up to 15 minutes for the changes to appear on the Active Roles Web Interface. This is due to a replication delay present between PowerShell and the Microsoft Graph API.

Renaming Azure cloud contacts

You can use the Active Roles Web Interface to rename an Azure cloud contacts.

To rename an Azure cloud contacts account

1. On the Active Roles Web interface **Navigation** bar, click **Directory Management**.
2. On the **Views** tab in the **Browse** pane, click **Azure** |***<Azure tenant>*** |**Azure Contacts**.
3. Select the Azure contact that need to be renamed.
4. In the **Command** pane, click **Rename**.
5. Enter the required name.
6. Click **Yes** to continue.

   The Azure cloud contacts that are selected are renamed.

NOTE: When creating a new cloud-only Azure contact or updating an existing one, it may take up to 15 minutes for the changes to appear on the Active Roles Web Interface. This is due to a replication delay present between PowerShell and the Microsoft Graph API.

Viewing and modifying Exchange Online properties

You can use the Active Roles Web Interface to create and view and modify the Exchange online properties of the new cloud-only Azure Contacts.

To view the Exchange Online properties of a cloud only Azure Contacts

1. On the Active Roles Web interface **Navigation** bar, click **Directory Management**.
2. On the **Views** tab, click | **Azure** | ***<Azure tenant>*** | **Azure Contacts**.

   The Azure contacts page is displayed and lists the Azure Contacts available in Azure.
3. Select the check-box corresponding to the specific cloud only Azure contacts for which you want to view the properties.
4. In the **Command** pane, click **Exchange Online Properties**.

   The **Exchange Online Properties** wizard displays the following Exchange Online properties for the cloud only Azure contact.

   - General
   - Mail tip
5. Use the tabs in the **Exchange Online Properties** dialog box to view the following Exchange Online properties of the cloud only Azure contact:

   - General

     Provide an Exchange online alias name in the **Alias** field. You can also choose to hide the alias name from the organizational address list.
   - Mail tip

     Provide an optional mail tip in the **Mail tip text** field.

NOTE: When creating a new cloud-only Azure contact or updating an existing one, it may take up to 15 minutes for the changes to appear on the Active Roles Web Interface. This is due to a replication delay present between PowerShell and the Microsoft Graph API.

Viewing change history

You can use the Active Roles Web Interface to view the Change History and User Activity for cloud only Azure contacts.

To view the Change History and User Activity of cloud only Azure contacts

1. On the Active Roles Web interface **Navigation** bar, click **Directory Management**.
2. On the **Views** tab, click | **Azure** | ***<Azure tenant>*** | **Azure Contacts**.

   The Azure Contacts page is displayed and lists the Azure contacts available in Azure.
3. Select the Azure contact to view the history.
4. In the **Command** pane, click **Change History** or **User Activity**.

   The information on changes made to the contact through Active Roles is displayed.

Deleting an Azure contact

You can use the Active Roles Web Interface to delete an Azure contact.

To delete an Azure contact

1. On the Active Roles Web interface **Navigation** bar, click **Directory Management**.
2. On the **Views** tab in the **Browse** pane, click **Azure** |***<Azure tenant>*** |**Azure Contacts**.

   The Azure Contacts page is displayed and lists the Azure contacts available in Azure.

1. Select the Azure contact that needs to be deleted.
2. In the **Command** pane, click **Delete**.

   A message prompts you to confirm the action.

3. Click **Yes** to continue.

   The Azure contacts that are selected are deleted.

Changes to Active Roles policies for cloud-only Azure objects

Active Roles 7.4.4 introduced support for cloud-only Azure objects: Azure users, guest users and contacts. To support the management of these cloud-only Azure objects, the existing Active Roles policies received the following updates:

- The Property Generation and Validation policy now supports specifying object property rules for cloud-only Azure objects. To get started with provisioning cloud-only Azure properties, Active Roles contains a new built-in policy for provisioning cloud-only Azure properties. Find the policy in the following node of the Active Roles MMC console:

  Configuration > **Policies** > **Administration** > **BuiltIn** > **Azure CloudOnly Policy - Default Rules to Generate Properties**

- The Group Membership AutoProvisioning policy now supports specifying group membership rules to automatically assign (or unassign) cloud-only Azure users and guest users to (or from) O365 Groups located in the same Azure tenant as the provisioned Azure objects.

  In the **New Provisioning Policy Wizard** of the Active Roles MMC console, the cloud-only Azure objects supported for provisioning are listed in the **Object Type Selection** > **Select Object Type** dialog, while the O365 Groups can be selected in the **Group Selection** > **Browse for Container** dialog.

- Script Execution policies now also support PowerShell and other custom scripts for provisioning cloud-only Azure objects. As part of this change, Active Roles contains a

new built-in script module that you can use to configure policies for generating cloud-only Azure user passwords complying with Azure AD password generation policies. This built-in script module is available at the following node of the Active Roles MMC console:

Configuration > **Script Modules** > **BuiltIn** > **Generate User Password - Azure only**

Managing room mailboxes

Room mailbox is a type of Exchange Online resource mailbox assigned to a physical location, such as a meeting room. Using room mailboxes that an administrator creates, users can reserve rooms by adding room mailboxes to meeting requests as an attendee or location.

In the Active Roles Web Interface, you can create, manage or delete room mailboxes in **Directory Management** > **Tree**> **Azure** > **Resource Mailboxes**. Room mailboxes created in the Active Roles Web Interface are synchronized to the Exchange admin center (admin.exchange.microsoft.com), where you can find them in **Home** > **Resources**.

For more information about room mailboxes, see Manage resource mailboxes in Exchange Online in the *Microsoft Exchange Online documentation*.

Creating a new room mailbox

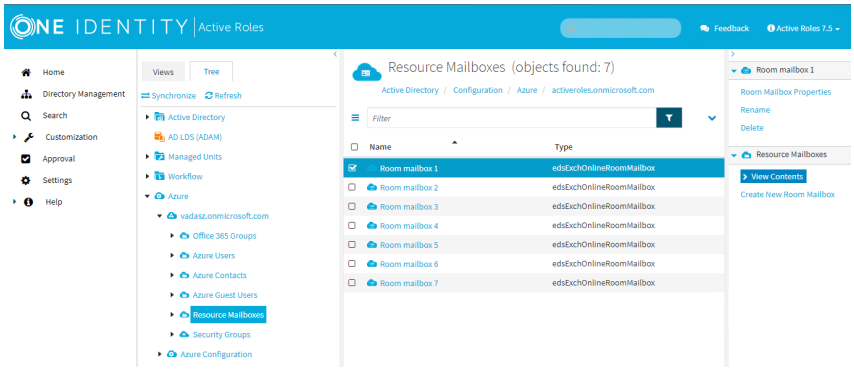
To create a new room mailbox in the Active Roles Web Interface, follow the steps.

To create a new room mailbox

1. In the Active Roles Web Interface, navigate to **Directory Management** > **Tree**> **Azure** > **Resource Mailboxes**.

   The list of resource mailboxes of the selected tenant is displayed.

Figure 141: Directory Management > Tree View > Azure > Resource Mailboxes — Listing the resource mailboxes in the tenant



2. In the right pane, click **Create New Room Mailbox.**

   The **Create New Room Mailbox** window opens.

3. On the **General** tab, set the following general details of the room:

   - (Optional) **Display name**
   - **Name**: Enter a name for the room.

     NOTE: If you enter a name that is already used, you will receive an error message and Exchange Online will not allow you creating the new room mailbox. To create a new room mailbox, enter a different name.

     NOTE: To change the name of an existing room mailbox:

     1. In the right pane, click **Rename**.
     2. **Display name**: Enter a new display name for the room.
     3. **Name**: Enter a new name for the room.
     4. Click **Finish**.

   - (Optional) **Primary SMTP Address (leave blank for default value)**

     To specify the domain, use the drop-down.

     The default value of the Primary SMTP Address is the name and the domain name of the room mailbox. For example, `roommailbox1@activeroles.onmicrosoft.com`, where `roommailbox1` is the name and `activeroles.onmicrosoft.com` is the domain name.

   - (Optional) **Capacity**
   - (Optional) **Hide from global address lists**

     Select this check box if you do not want the room mailbox to appear in the address book and other address lists defined in your Exchange organization.

     By default, this check box is not selected.

4. (Optional) On the **Calendar Processing** tab, set the following optional details of the room:
   - **Maximum duration (hours)**
   - **Booking window (days)**
   - **Allow repeating meetings**

     By default, this check box is selected.
   - **Allow scheduling only during work hours**

     By default, this check box is selected.
5. (Optional) On the **Location** tab, set the following optional details of the company:
   - **Department**
   - **Company Name**
   - **Street Address**
   - **City**
   - **State or Province**
   - **Zip or Postal Code**
   - **Country**: You must enter a valid country code or country name, for example: `US` or `United States of America (the)`.
6. Click **Finish**.

If the operation is successful, the newly-created room mailbox appears in the list of **Resource Mailboxes**.

In the right pane, the name of the room mailbox appears with the following available actions:

- **Room Mailbox Properties**
- **Rename**
- **Delete**

The newly-created room mailbox also appears in the Exchange admin center, in **Home** > **Resources**.

Viewing or modifying a room mailbox

To view or modify the properties of a room mailbox in the Active Roles Web Interface, follow the steps.

NOTE: To change the name of an existing room mailbox:

1. In the right pane, click **Rename**.
2. **Display name**: Enter a new display name for the room.

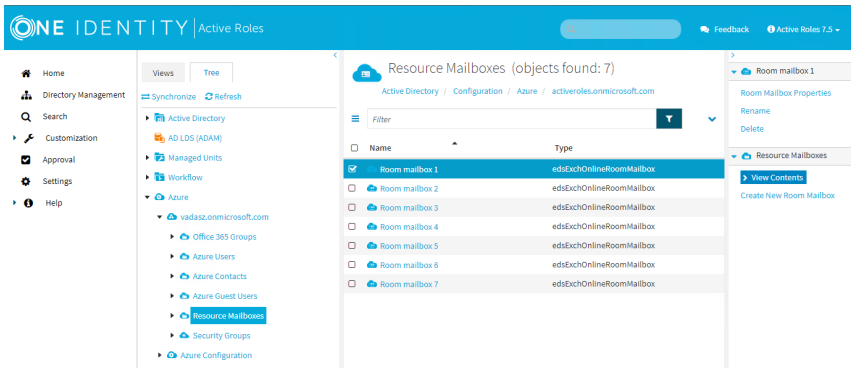
3. **Name**: Enter a new name for the room.
4. Click **Finish**.

To view or modify the properties of a room mailbox

1. In the Active Roles Web Interface, navigate to **Directory Management** > **Tree**> **Azure** > **Resource Mailboxes**.

   The list of resource mailboxes of the selected tenant is displayed.

   Figure 142: Directory Management > Tree View > Azure > Resource Mailboxes — Listing the resource mailboxes in the tenant



2. Select the room mailbox you want to view or modify.

   In the right pane, the name of the room mailbox appears with the following available actions:

   - **Room Mailbox Properties**
   - **Rename**
   - **Delete**

3. In the right pane, click **Room Mailbox Properties**.

   The **Room Mailbox Properties** window opens.

4. On the **General** tab, view or modify the following general details of the room:

   - **Display name**
   - **Object GUID**: The Exchange Online GUID of the mailbox object in the Exchange Cloud. You cannot modify this value.
   - **External directory ID**: The Azure Active Directory (AD) object of the user object connected to the mailbox object in Azure AD. You cannot modify this value.
   - **User Principal Name**: The room mailbox address in User Principal Name (UPN) format. You cannot modify this value.
   - **Primary SMTP Address**: You cannot modify this value.
   - **Capacity**

- **Hide from global address lists**

  Select this check box if you do not want the room mailbox to appear in the address book and other address lists defined in your Exchange organization.

  By default, this check box is not selected.

5. On the **Calendar Processing** tab, view or modify the following optional details of the room:
   - **Maximum duration (hours)**
   - **Booking window (days)**
   - **Allow repeating meetings**

     By default, this check box is selected.
   - **Allow scheduling only during work hours**

     By default, this check box is selected.
6. On the **Location** tab, view or modify the following optional details of the company:
   - **Department**
   - **Company Name**
   - **Street Address**
   - **City**
   - **State or Province**
   - **Zip or Postal Code**
   - **Country**: You must enter a valid country code or country name, for example: `US` or `United States of America (the)`.
7. To close the **Room Mailbox Properties** window:
   a. To update the properties of the room mailbox, click **Save**.
   b. To close the window without saving the changes, click **Cancel**.

If the operation is successful, the updated properties of the room mailbox appear both in the Active Roles Web Interface and in the Exchange admin center.

Deleting a room mailbox

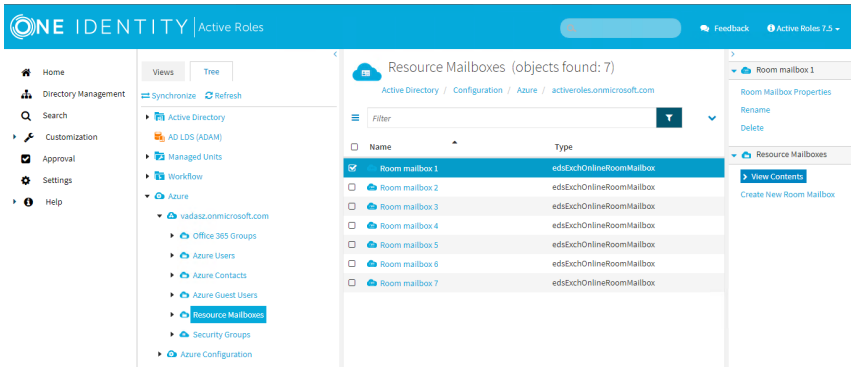
To delete a room mailbox in the Active Roles Web Interface, follow the steps.

To delete a room mailbox

1. In the Active Roles Web Interface, navigate to **Directory Management** > **Tree**> **Azure** > **Resource Mailboxes**.

   The list of resource mailboxes of the selected tenant is displayed.

Figure 143: Directory Management > Tree View > Azure > Resource Mailboxes — Listing the resource mailboxes in the tenant



2. Select the room mailbox that you want to delete.

   In the right pane, the name of the room mailbox appears with the following available actions:

   - **Room Mailbox Properties**
   - **Rename**
   - **Delete**

3. Click **Delete**.

   The following dialog appears:

   ```
   Are you sure you want to delete <room mailbox>?
   ```

4. Click **Yes** to confirm.

If the operation has been successful, the room mailbox is deleted and it disappears both from the **Resource Mailboxes** list in the Active Roles Web Interface, and from the **Resources** list in the Exchange admin center.

20

Managing Active Roles

This chapter covers the following management activities:

- Connecting to the Administration Service
- Adding and removing managed domains
- Using unmanaged domains
- Evaluating product usage
- Configuring replication
- Using AlwaysOn Availability Groups
- Using database mirroring
- Creating and using virtual attributes
- Examining client sessions
- Monitoring performance
- Customizing the console
- Using Configuration Center
- Changing the Active Roles Admin account
- Enabling or disabling diagnostic logs
- Active Roles Log Viewer

To manage the configuration of Active Roles, you must have the necessary permissions. It is sufficient to be a member of the Active Roles Admin account. The Active Roles Admin account is specified when configuring the Administration Service. It defaults to the Administrators group on the computer running the Administration Service.

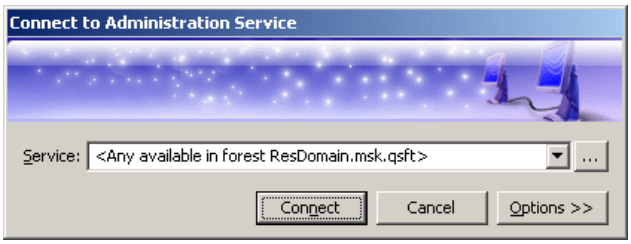
The authority to modify the Active Roles configuration can be delegated by applying the **Manage Configuration** Access Template to the **Server Configuration** container.

Connecting to the Administration Service

To configure a particular Administration Service using the Active Roles console, you need to manually specify the Administration Service to connect to. Otherwise, the console automatically selects the Administration Service.

You can use the **Connect to Administration Service** dialog box to select the appropriate Administration Service. To display this dialog box, right-click **Active Roles** in the console tree and click **Connect**. The dialog box looks as shown in the following figure.

Figure 144: Connect to Administration Service

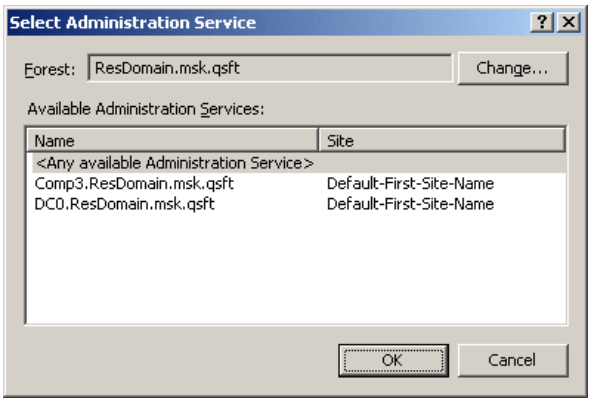


In the **Service** box, type or select the name of the computer running the Administration Service to connect to, and then click **Connect**. The **Service** box provides a list of names that were specified for previous connection sessions. The last selected name is displayed by default.

To select the Administration Service that is not in the list, click the **Select** button next to the **Service** box: ...

This displays the **Select Administration Service** dialog box, shown in the following figure.

Figure 145: Select Administration Service



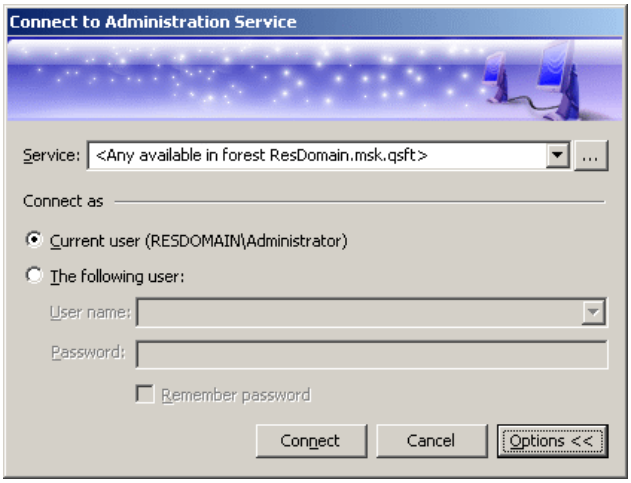
The **Select Administration Service** dialog box lists the Administration Services that are available in the specified forest. You can choose a different forest by clicking **Change**. The list items are sorted according to priority, considering site location and service load (less

loaded Administration Services are displayed at the top of the list). To add a certain Service to the **Connect to Administration Service** dialog box, click that Service and then click **OK**.

If you have connected to a specific Service, the console will attempt to automatically connect to that Service on every subsequent start. If you have selected **<Any available Administration Service>**, the console will attempt to connect to the nearest, least loaded Service in the specified forest, giving preference to the Services that belong to the same replication group as the Service to which the console was connected in the previous session.

By default, the console connects to the Administration Service in the security context of your logon account (that is, the user account to which you have logged on). This means that you can only use the console to perform the tasks that are delegated to your user account. You have the option to establish a connection using a different account, in order to change the scope of the allowed tasks. Click **Options** to expand the **Connect to Administration Service** dialog box, as shown in the following figure.

Figure 146: Connect to Administration Service



Click **The following user** and specify the user logon name and password of the account to be used for connection. By selecting the **Remember password** check box you can have the console automatically use the specified user name and password in the future connection sessions. Otherwise, on a subsequent start, the console will prompt you for a password.

Delegating control to users for accessing MMC interface

By default, on installing Active Roles, all users are allowed to log in to the MMC interface. To manage the MMC interface access for a user, you must configure the options using **Configuration Center** | **MMC Interface Access**| **Manage settings**. Selecting this

option restricts all non Active Roles Administrators from using the console. All delegated users are affected, however, it does not apply to Active Roles Administrators.

To be able to log in to the MMC interface, the user must be delegated with the **User Interfaces** access rights on the **User Interfaces** container under **Server Configuration**. User Interfaces Access templates that provide the access rights are available as part of the Active Roles built-in Access templates in the **User Interfaces** container.

To delegate the control to users in the User Interfaces container you must apply the User Interface Access Template

1. In the console tree, expand **Active Roles | Configuration | Server Configuration**.
2. Under **Server Configuration**, locate the **User Interfaces** container, right-click it, and click **Delegate Control**.
3. On the **Users or Groups** page, click **Add**, and then select the users or groups to which you want to delegate the control. Click **Next**.
4. On the **Access Templates** page, expand the **Active Directory** | **User Interfaces** folder, and select the check box next to **User Interface Management-MMC Full control**.
5. Click **Next** and follow the instructions in the wizard, accepting the default settings.
6. After you complete these steps, the users and groups you selected in Step 3 are authorized to log in to the MMC interface.
7. Click **OK** to close the **Active Roles Security** dialog box.

Steps for connecting to the Administration Service

When you start the Active Roles console, it automatically selects the appropriate Administration Service, and establishes a connection to that Service. However, you can connect to a specific Administration Service at any time.

To select the Administration Service to connect to

1. In the console tree, right-click **Active Roles**, and then click **Connect**.
2. In the **Service** box, type or select the name of the computer running the Administration Service to connect to, and then click the **Connect** button.

NOTE:

- If you do not see the Administration Service you want in the **Service** list, click the **Select** button next to the **Service** box. This displays a list of the Administration Services that are available in the specified forest. You can choose a different forest by clicking **Change**. To add a certain Service to the **Service** list, click that Service in the list of available Administration Services, and then click **OK**.
- If you need to establish a connection under a user account other than your logon account, click **Options** to display the **Connect as** area, and then click **The following user** and specify the user logon name and password of the account to be used for connection. By selecting the **Remember password** check box, you can have the console automatically use the specified user name and password in the future connection sessions. Otherwise, on a subsequent start, the console will prompt you for a password.

Adding and removing managed domains

Active Directory domains registered with Active Roles are referred to as *managed domains*. Each Administration Service maintains a list of managed domains, and stores this list in the Administration Database as part of the service configuration.

In the Active Roles console, the Add Managed Domain wizard is used to register domains for management. You can access the wizard as follows: Click the console tree root; then, in the details pane, in the **Domains** area, click **Add Domain**.

The Add Managed Domain wizard prompts you for the following information:

- The name of the domain you want to register.
- The credentials that Active Roles will use to access the domain.

You have the option to use the default credentials (the service account of the Administration Service) or enter the user name and password of a different account (override account). In both cases, the account must have adequate rights in the managed domain. For more information, refer to the "Access to Managed Domains" section in the Active Roles Quick Start Guide.

If you choose the option to access the managed domain using the service account information, consider the following. This option applies to all Administration Services in your environment. Each Administration Service in your environment will use its own service account to access the domain. Since different service accounts may have different levels of access to the domain, Active Roles may have different access rights to the domain, depending on which Administration Service is being used to manage the domain. The result is that the behavior of Active Roles may vary when you switch to a different Administration Service.

After you add a managed domain, the Administration Service retrieves the domain information, such as the Active Directory schema and the hierarchy of containers. This process is referred to as *loading domain information*.

It may take a few minutes for the Administration Service to load the domain information. Once this process is completed, the domain is available for management. Select the **Active Directory** item in the console tree and press F5 to refresh the details pane and display the new domain. To start managing the domain, select it in the details pane and press ENTER; or expand the domain item in the console tree.

It is possible to remove a domain from the list of managed domains. Once removed, the domain and all directory objects contained in the domain can no longer be managed with Active Roles. To remove a managed domain, select the console tree root and click **Go to Managed Domains** in the details pane, in the **Domains** area. This causes the details pane to display a list of managed domains. In the list, right-click the domain you want to remove, and click **Delete**.

Steps for adding or removing a managed domain

The operation of adding a managed domain results in the creation of an object that holds the registration information about the domain. For this reason, it is also referred to as registering a domain with Active Roles.

To add a managed domain

1. In the console tree, expand **Configuration | Server Configuration**.
2. Under **Server Configuration**, right-click **Managed Domains**, and select **New | Managed Domain** to start the Add Managed Domain wizard.
3. On the Welcome page of the wizard, click **Next**.
4. On the **Domain Selection** page, do one of the following, and then click **Next**.
   - Type the name of the domain you want to add.
   - Click **Browse**, and select the domain from the list.
5. On the **Active Roles Credentials** page, click one of these options that determine the logon information that Active Roles will use to access the domain:
   - **The service account information the Administration Service uses to log on**
   - **The Windows user account information specified below**

   If you choose the second option, type the user name and password of the user account you want Active Roles to use when accessing the domain.
6. Click **Next**, and then click **Finish**.

To remove a managed domain

1. In the console tree, expand **Configuration | Server Configuration**.
2. Under **Server Configuration**, click **Managed Domains**.

3. In the details pane, right-click the domain you want to remove, and then click **Delete**.

NOTE:

- You can use the **Properties** command on an object held in the **Managed Domains** container to view or modify the registration information for the respective managed domain. For example, it is possible to change the logon information that is used to access the domain: on the **General** tab in the **Properties** dialog box, choose the appropriate option and click **Apply**. You can choose one of the two options that are listed in Step 5 of the procedure above.
- The **Managed Domains** container holds the registration objects for all domains that are registered with Active Roles. You can un-register domains by deleting objects from that container.
- By default, no domains are registered with Active Roles. When you register a domain, the domain registration is saved as part of the Active Roles configuration.

Using unmanaged domains

After you've registered an Active Directory domain with Active Roles, you have the option to use the domain as an unmanaged domain. An unmanaged domain is basically a domain that is registered with Active Roles for read-only access. The use of the unmanaged domain option allows you to reduce licensing costs since the user count that corresponds to the unmanaged domains is not added to product usage statistics (see Evaluating product usage).

Unmanaged domains are instrumental in the following scenarios:

- **Group membership management** When used to add members to a group, by selecting the new members from a list of objects, Active Roles requires the domain that holds the objects to be registered. If you only use Active Roles for selecting member objects when managing group membership, you can configure the domain that holds the member objects as an unmanaged domain.
- **Exchange resource forest** When used to create Exchange mailboxes in a forest that is different from the forest that holds the accounts of the mailbox users, Active Roles requires the domain of the mailbox users (account domain) to be registered. If you do not use Active Roles for user management in the account domain, you can make that domain an unmanaged domain.

As applied to a registered unmanaged domain, the features and functions of Active Roles are limited to those that do not require write access to the objects held in that domain (including write access to the object data that is stored by Active Roles as virtual attributes). Thus, you can use Active Roles to:

- Search for, list and select objects from unmanaged domains
- Populate groups in regular managed domains with objects from unmanaged domains
- Retrieve and view properties of objects held in unmanaged domains
- Assign users or groups from unmanaged domains to the role of manager, primary owner, or secondary owner for objects held in regular managed domains
- Delegate management tasks and approval tasks to users or groups held in unmanaged domains
- Run Active Roles policies against objects held in unmanaged domains, provided that the policies require only read access to those objects
- Provision users from unmanaged domains with linked Exchange mailboxes held in a separate managed forest
- Populate Managed Units with objects from unmanaged domains

Since Active Roles has read-only access to unmanaged domains, it cannot:

- Create, move, or delete objects in unmanaged domains
- Change any properties of objects held in unmanaged domains
- Run any group membership related policies against the groups in unmanaged domains, including the Group Family and Dynamic Group policies
- Run any auto-provisioning or deprovisioning policies against the users or groups held in unmanaged domains
- Run any workflow that makes changes to objects in unmanaged domains
- Restore objects from Active Directory Recycle Bin in unmanaged domains

Configuring an unmanaged domain

You can configure an unmanaged domain by applying the **Built-in Policy - Exclude from Managed Scope** Policy Object in the Active Roles console.

To configure an unmanaged domain

1. In the console tree, under the **Active Directory** node, right-click the domain you want to configure, and click **Enforce Policy**.
2. Click **Add** in the dialog box that appears, and then select the **Built-in Policy - Exclude from Managed Scope** Policy Object.
3. Click **OK** to close the dialog boxes.

Once applied to a domain, the **Built-in Policy - Exclude from Managed Scope** Policy Object stops product usage statistics from counting objects in the domain and prevents any changes to the objects held in that domain, making the objects available for read access only. For more information, see Managed scope to control product usage.

Evaluating product usage

Active Roles provides a predefined collection of statistics that helps you understand how many Active Directory domain users, AD LDS, Azure, and SaaS users are managed by this product over time. By analyzing this statistical data, you can establish a baseline of product usage, verify your current Active Roles licensing compliance, and plan for future licensing needs. Since Active Roles' license fee is calculated based on the number of managed users, product usage statistics enables you to justify and predict your Active Roles licensing expenditures. For instructions on how to examine product usage, see Viewing product usage statistics.

For each Active Directory domain, AD LDS instance, Azure tenants, and SaaS applications registered with Active Roles, product usage data is collected on a scheduled basis by counting the number of enabled users in that domain, instance, registered Azure tenants, and connected SaaS applications with the resulting counts stored in the Active Roles database. For further details, see Scheduled task to count managed objects.

By default, Active Roles counts users in the entire domain or instance. It is possible to have Active Roles count users within a part of a domain or instance by changing managed scope—a tunable collection of containers assumed to hold the managed users. For further details, see Managed scope to control product usage.

Active Roles counts the managed objects on a scheduled basis, and provides a report of managed object statistics. This does not impose any restrictions on the number of objects managed by Active Roles. However, as the number of the managed objects is a key factor in determining the license fee, you may need to ensure that your managed object count does not exceed a certain limit. For this purpose, you can configure Active Roles to check the number of managed objects and send an e-mail notification if the total number of managed objects exceeds a given threshold value. For further details, see Voluntary thresholds for the managed object count.

Viewing product usage statistics

You can view the current total number of managed users on the root page in the Active Roles console. Select the console tree root to open the root page in the details pane, and then expand the **Product Usage Statistics** area on that page. The count of objects under **Active Directory Domains, AD LDS Directory Partitions**, **Azure tenants**, and **SaaS application** represents the current number of managed domain users , managed AD LDS users, Azure hybrid users, Azure cloud only users, Azure guest users, and SaaS users respectively.

NOTE: The count can be derived using the LDAP query "(&(objectCategory=person)(objectClass=user))".

It is possible to view the average or maximum number of managed users in each domain or instance for a certain reporting period. Click **Product Usage Statistics** to open a page allowing you to:

- Choose the reporting period.

  The page displays options to export data in HTML format and as raw counters for the period you choose from the Reporting period options, such as past month, past half-year, past year, or a custom date range.

- Examine the managed user counts for the reporting period you've chosen.

  The page displays the current number of managed users per Active Directory domain, AD LDS directory partition, Azure tenant, and SaaS application in the tables under **Total accounts**. The Average and the maximum values along with the total number of managed users can be viewed in the HTML file.

  License type and **Total estimated licenses**, display the type of license in use and the number of estimated license required, respectively.

- View the information about the license.

  Click **License description** to view a detailed information about the license.

- Save the contents of the page as an HTML file.

  Click **Save as HTML** at the bottom of the page and specify the desired file name and location.

- Export the raw statistical data to a file.

  Click **Export raw counters** at the bottom of the page and specify the desired file name and location. The data is exported in the comma-delimited (CSV) format, representing the daily counts of managed users over the reporting period.

Delegating access to the managed object statistics

By default, only Active Roles Admin role holders have permission to view managed object statistics. Active Roles provides the following Access Templates for delegating that task:

- **Managed Object Statistics - View Report**

  To delegate the task of viewing managed object statistics, apply this Access Template to the **Configuration/Server Configuration/Managed Object Statistics** container.

- **Managed Object Statistics - Read Detailed Data**

  To delegate the task of exporting raw statistical data, apply this Access Template to the **Configuration/Server Configuration/Managed Object Statistics** container.

You can find these two Access Templates in the **Configuration/Access Templates/Configuration** container in the Active Roles console.

Scheduled task to count managed objects

Active Roles uses a scheduled task to count the number of managed users in each Active Directory domain, AD LDS instance, Azure tenants, and SaaS applications registered with this product. Every Administration Service in your Active Roles environment runs that task on a daily basis, saving the obtained results in the Active Roles database. The statistical data collected by running that task over time is used to calculate managed object statistics, and can be exported by clicking **Export raw counters**.

The scheduled task in question is located in the **Configuration/Server Configuration/Scheduled Tasks/Builtin** container in the Active Roles console, and has the name **Export raw counters**. Changes to this task are not allowed, except for changing the start time. You can change the start time on the **Schedule** tab in the task's **Properties** dialog box in the Active Roles console.

Managed scope to control product usage

The area where Active Roles collects product usage statistics is referred to as *managed scope*. By default, managed scope comprises all Active Directory domains and AD LDS instances registered with Active Roles. This means that by default product usage statistics includes all enabled user accounts in all managed domains and instances. However, if you don't use Active Roles to manage a particular domain or instance, or a part of a domain or instance (for example, individual Organizational Units), then you can exclude the entire domain or instance, or a part of a domain or instance, from managed scope.

Active Roles provides a built-in Policy Object allowing you to exclude entire AD domains, AD LDS directory partitions, individual Organizational Units (OUs), or even Managed Units (MUs) from managed scope. This Policy Object is located in the **Configuration/Policies/Administration/Builtin** container in the Active Roles console, and has the name **Built-in Policy - Exclude from Managed Scope**. When applied to a container such as an AD domain, AD LDS directory partition, OU or MU, this Policy Object:

- Stops product usage statistics from counting objects held in that container, and
- Prevents any changes to the objects held in that container, making the objects available for read access only.

Thus, you can exclude a certain domain from managed scope by applying a Policy Object: Choose the **Enforce Policy** command on the domain object under the **Active Directory** node in the Active Roles console, click **Add**, and select the **Built-in Policy - Exclude from Managed Scope** Policy Object. This stops product usage statistics from counting objects in that domain, and makes all objects in that domain available for read access only. You will not be able to create new objects (users, groups, computers, and so forth) or make changes to existing objects in that domain by using Active Roles.

After you have excluded a domain from managed scope, you may need to make a particular OU in that domain available for read/write access. You can accomplish this by blocking policy inheritance: In the Active Roles console, choose the **Enforce Policy** command on the OU and then select the **Blocked** option next to **Built-in Policy - Exclude**

from Managed Scope. Doing so removes the read-only restriction from the OU and objects it contains, while causing product usage statistics to start counting objects held in that OU.

When you apply the **Built-in Policy - Exclude from Managed Scope** Policy Object to a Managed Unit, all objects that match the membership rules of that Managed Unit are excluded from managed scope. You can use this option to prevent product usage statistics from counting objects that satisfy certain conditions (for example, user accounts that have a particular country or department setting): Create a Managed Unit with the appropriate membership rules and then apply the **Built-in Policy - Exclude from Managed Scope** Policy Object to that Managed Unit. Doing so stops product usage statistics from counting objects that match the Managed Unit's membership rules, while making those objects read-only.

You can determine whether a given object is excluded from managed scope by looking at the **Managed** field on the **Object** tab in the **Properties** dialog box for that object in the Active Roles console or on the **General Properties** page in the Active Roles Web Interface. If the object is excluded from managed scope, the **Managed** field reads **No**; otherwise, the field reads **Yes**.

Voluntary thresholds for the managed object count

By default, Active Roles does not limit the number of managed objects. However, as Active Roles' license fee is based on the managed object count, you may need to verify if the object count is under a certain threshold. You can perform this task by specifying a threshold value for the number of managed objects. The scheduled task that counts managed objects then raises an alert each time it detects that the current number of managed objects exceeds the threshold value. The alert makes the **Product Usage Statistics** section red on the root page in the Active Roles console, and can send a notification over e-mail.

To configure thresholds and notification for the managed object count

1. Log on as Active Roles Admin, and open the Active Roles console.

   Only members of the Active Roles Admin account are authorized to configure thresholds and notification for the managed object count.

2. In the console tree, select the root node: **Active Roles**.
3. On the page in the details pane, expand the **Product Usage Statistics** section, and then click **Set License threshold value** to update the threshold.
4. In the **Threshold Value** dialog box that appears, specify the desired threshold value for Active Directory domains (AD DS), AD LDS directory partitions (AD LDS), Azure tenants, or SaaS applications.

   You can specify an AD DS threshold value, AD LDS threshold value, Azure tenant threshold value, and SaaS threshold value independently from each other. Active Roles raises an alert if the total number of managed objects in AD DS, AD LDS

directory partitions, Azure tenant, or SaaS application exceeds the corresponding threshold value. If the threshold value is specified for any of these, then Active Roles does not evaluate the managed object counts at all.

5. If you want Active Roles to notify of the threshold violation alert over e-mail, then, in the **Threshold Value** dialog box, configure the notification settings as follows:
   a. Select the **Notify of threshold violations by e-mail** check box.
   b. Click the button next to the **Recipients** field, and specify who you want to receive the notification messages. You can select recipients from an address book (requires Microsoft Outlook to be configured), or supply individual e-mail addresses.
   c. Click the button next to the **E-mail server settings** field. Then, on the **Mail Setup** tab in the dialog box that appears, supply the server name and other settings specific to your outgoing SMTP server.

   If multiple mail configuration objects exist in your Active Roles environment, then you may first need to select the appropriate object from the **E-mail server settings** list. Mail configuration objects can be created in the **Configuration/Server Configuration/Mail Configuration** container in the **Active Roles** console.
6. When finished, click **OK** to close the **Threshold Value** dialog box.

Installation label

The Active Roles console allows you to set a text label that helps you identify your Active Roles installation in the Managed Object Statistics report—a report that lists the managed object counts (see Viewing product usage statistics). You can use the installation label to distinguish, for example, between production and non-production or pilot installations. The label text is displayed in the title of the Managed Object Statistics report.

To set or change the installation label

1. Log on as Active Roles Admin, and open the Active Roles console.

   Only members of the Active Roles Admin account are authorized to set or change the installation label.
2. In the console tree, select the root node: **Active Roles**.
3. On the page in the details pane, expand the **Product Usage Statistics** section, and then click the **Change** link next to the **Installation label** field.

   The console does not display the **Change** link unless you are logged on as Active Roles Admin.
4. In the **Installation Label** dialog box that appears, type the label text you want, and then click **OK**.

Creating and using virtual attributes

Active Roles provides the facility to define custom (virtual) attributes for any existing object type. This allows additional object properties to be specified without extending the Active Directory schema. For example, custom attributes can be used to store specific user data.

You can configure a virtual attribute to store the attribute value in the Active Roles database. Otherwise, to use the virtual attribute, you need to implement a script policy to handle the attribute value.

To create a virtual attribute

1. In the console tree, expand **Configuration | Server Configuration**.
2. Under **Server Configuration**, right-click **Virtual Attributes** and select **New | Virtual Attribute**.
3. Follow the instructions in the Add Virtual Attribute wizard.
4. In the **Common-Name** and the **LDAP Display Name** boxes, type a common name and an LDAP display name for the new attribute.

   In the **Unique X.500 object ID** box, you can optionally change the default value of the attributeID property (OID) for the new attribute. The default value is generated automatically. If you want to generate your own value, you can use the Oidgen (oidgen.exe) tool, included with the Windows Server Resource Kit.

   In the **Schema ID GUID** box you can optionally change the default value of the schemaIDGUID property. The default value is generated automatically. If you want the new attribute to have the fixed schemaIDGUID property, replace the default value with your own value. For example, you can generate GUID with the Uuidgen tool, included with the Microsoft Platform SDK.
5. Optionally, in the **Description** box, type a description for the new virtual attribute. Click **Next**.
6. In the **Syntax** list, click the syntax you want for the new virtual attribute. If you want the new attribute to be multi-valued, select the **Multi-valued** check box. Click **Next**.
7. Select the check boxes next to the object classes with which you want the virtual attribute to be associated. Click **Next**.

   If you need the new attribute to be associated with object classes that are not listed by default, select the **Show all possible classes** check box.
8. If you want to store the values of the attribute in the Active Roles database, select the check box on the **Attribute Storage** page.

   If you choose not to store the attribute values in the database, a script policy is required to supply the attribute value when retrieving the attribute and to save the attribute value when updating the attribute.

   However, you should use this option carefully. Storing attribute values in the Active Roles configuration database may considerably increase the database size.

This option can be modified after the attribute is created, by managing properties of the virtual attribute.

9. Click **Next**, and then click **Finish** to complete the wizard.

After the new virtual attribute has been added, reconnect to the Administration Service. The new virtual attribute appears in the **Virtual Attributes** container under **Configuration/Server Configuration**.

To view or modify the value of a virtual attribute on an object

1. Right-click the object, and select **All Tasks | Advanced Properties**.
2. Select the **Show all possible attributes** and the **Include attributes with empty values** check boxes, for the list in the **Advanced Properties** dialog box to display all attributes of the object.
3. Click the attribute in the list, and then click the button beneath the list.
4. In the dialog box that opens, view or modify the value of the attribute.

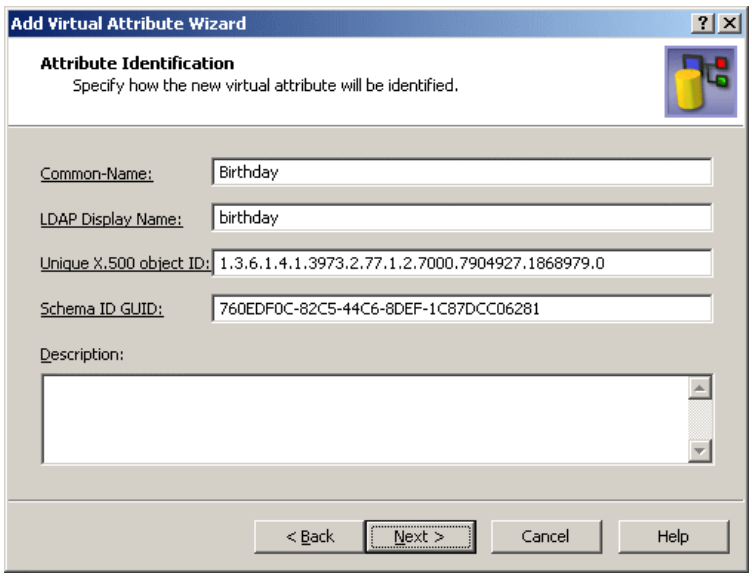
Scenario: Implementing a Birthday attribute

This scenario illustrates how to create and use a virtual attribute to store information on the birthdays of users.

To create the Birthday attribute

1. In the console tree, expand **Configuration | Server Configuration**.
2. Under **Server Configuration**, right-click **Virtual Attributes**, and select **New | Virtual Attribute**.
3. Click **Next**.
4. In the **Common-Name** and **LDAP Display Name** boxes, type **Birthday**, as shown in the following figure.

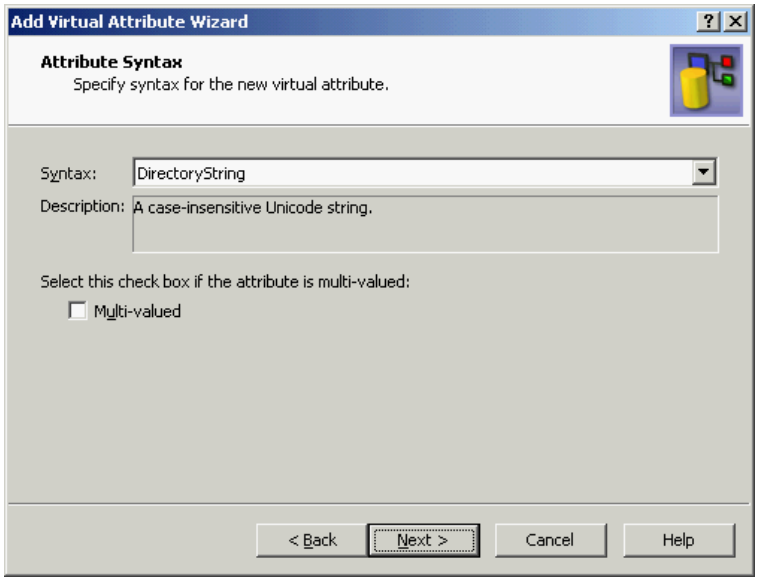
Figure 147: Attribute identification



5. Click **Next**.

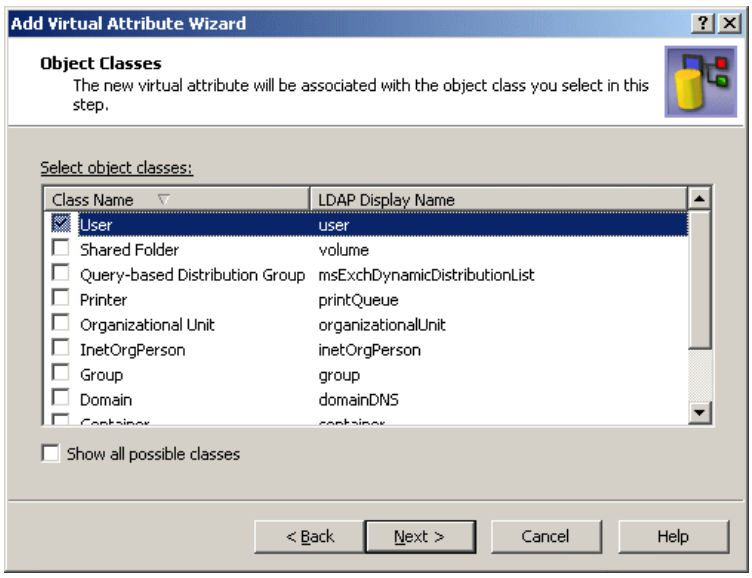
   The **Attribute Syntax** page should look as shown in the following figure.

Figure 148: Attribute syntax



6. Click **Next**.
7. On the **Object Classes** page, select the check box next to **User**, as shown in the following figure.

Figure 149: Object classes



8. Click **Next**.
9. On the Attribute Storage window, select the **Store values of this virtual attribute in the Active Roles Administration database** check box.
10. Click **Next**, and then click **Finish** to complete the wizard.

To enable the new attribute, reconnect to the Administration Service: right-click the console tree root and click **Reconnect**.

In the Active Roles console, you can manage the **Birthday** attribute on a user account as follows:

1. Right-click the user account and select **All Tasks | Advanced Properties**.
2. In the **Advanced Properties** dialog box, select both the **Show all possible attributes** and **Include attributes with empty values** check boxes.
3. Click **Birthday** in the list of properties, and then click **Edit**.
4. In the **Value** box, type a birthday date.
5. Click **OK**.

You can also manage the **Birthday** attribute via the Active Roles Web Interface.

First, you need to add the **Birthday** field to a form that displays user properties, and associate that field with the **Birthday** attribute. You can accomplish this by customizing the form. For instructions on how to add a field to a form, refer to the Active Roles Web Interface Administration Guide.

Then, the **Birthday** attribute can be managed by accessing user properties in a Web Interface site. For example, users can view and modify this attribute via Site for Self-Administration, provided that you have self-administration implemented (see Scenario 2: Implementing Self-administration in the Role-based Administration chapter earlier in this document).

Examining client sessions

The Active Roles console displays comprehensive information about client sessions. With the console connected to a given Administration Service, you can examine which clients are using that Service. Session information provided by the console includes the following:

- **User** Logon name of the account used by the session to connect to the Administration Service.
- **Active Roles Admin** Whether or not the client is logged on as a member of the Active Roles Admin account, and thus has administrator rights on the Administration Service.
- **Client Version** Client application, such as MMC Interface or Web Interface, and its version.
- **Last Access Time** Date and time that the Administration Service was last accessed within this session.
- **Logon Time** Date and time that the session was opened.
- **Client Host** DNS name of the computer running the client application.
- **Client Site** Network site of the computer running the client application.

To display a list of client sessions on the Administration Service

1. Connect to the Administration Service you want to examine for the client sessions.
2. In the console tree, expand **Configuration | Server Configuration**, and select **Client Sessions**.

   As a result, the details pane lists the client sessions for the Administration Service to which the console is connected.

By using the shortcut menu on a client session, you can also perform the following tasks:

- Send e-mail to the session user.
- Disconnect the session from the Administration Service.
- View additional information about the session.

For example, to view additional information about a session, right-click the session in the details pane and click **Properties**.

The **Properties** dialog box for a client session includes the following tabs:

- **General** Information about the session user, client version, client host, and client site.
- **Client Activity** Information about logon time, last access time, and the number of operations performed within the session, grouped by operation type.
- **Member Of** List of all security groups computed due to a transitive group membership expansion operation on the session user at the moment of session start.
- **Domain Controllers** Information about the domain controllers used to retrieve and update directory data within the session.

Monitoring performance

Active Roles includes a set of performance counters to monitor various aspects of the Administration Service's performance. Counters are grouped into performance objects that include the following:

- **Requests** Counts data management requests submitted to the Administration Service.
- **LDAP operations** Counts LDAP requests issued by the Administration Service.
- **Permissions propagation** Counts changes to Active Directory security made by the Administration Service.
- **External changes** Counts data changes polled by the Administration Service from Active Directory, and changes made to the Administration Database.
- **Script modules** Counts the average execution time of Active Roles script modules, the number of times a particular script module was executed, and number of script module instances being currently executed.
- **Miscellaneous** Counts the number of clients connected to the Administration Service and the number of queued post-policy processing operations.

To examine Administration Service performance counters, you can use the Performance tool on the computer running the Administration Service:

1. Start the Performance tool: click **Start** and select **All Programs | Administrative Tools | Performance**.
2. In the console tree, select **System Monitor**.
3. Click in the details pane, and then press CTRL+I to display the **Add Counters** dialog box.
4. From the list in the **Performance object** box, select any name that begins with the prefix **AR Server**. For example, you might select **AR Server:Requests**.
5. Select an item from the list of counters. For example, you might select **Requests/sec**.
6. Click **Add** and then click **Close**.

As a result, the Performance tool displays the output of the counter you have selected.

Customizing the console

The Active Roles console provides a convenient way to customize object creation wizards and property pages found in the console, and to customize display names for object types and object properties. Customization is performed through the use of Active Directory objects called *display specifiers*.

Each display specifier object holds information describing the various user interface elements for a particular object type. These elements include (but not limited to) creation wizard pages, property pages, and names to use for object types and properties in user interfaces.

The following sections summarize the customization-related features that are based on the use of display specifiers:

- **Other Properties** page in the object creation wizard
- **Other Properties** tab in the **Properties** dialog box
- Customizing display names

"Other Properties" page in object creation wizard

In the Active Roles console, directory objects are created using creation wizards. Thus, creating a user account starts the New Object - User wizard. The Active Roles console makes it possible to extend creation wizards with an extra page allowing additional properties to be populated in the course of the object creation process.

The Active Roles console makes it easy to view or modify the set of properties on the wizard extension page by using a separate tab in the **Properties** dialog box for display specifier objects. The **Other Properties to Display** tab provides a way to customize the set of properties included on the extension page of object creation wizards. If there are no properties to include on the extension page, the page is not displayed.

The **Other Properties to Display** tab can be used to add or remove properties from the extension page of the creation wizard for the object type that the display specifier is associated with. The tab lists the object properties included on the extension page, and allows you to make changes to that list.

You can use the following instructions to add the **Other Properties** page to the New Object - User wizard. Similarly, you can extend the creation wizard for a different object type by creating and configuring a custom display specifier for that object type. For example, to extend the wizard for Group, Computer, or Organizational Unit, create and configure a custom display specifier named **group-Display**, **computer-Display**, or **organizationalUnit-Display**, respectively.

Note that the names of display specifiers are case-sensitive, so you must type the name exactly as specified in the Active Directory schema. To view the names of display specifiers, you can use the console to examine the "Active Directory | Configuration Container | Display Specifiers | 409" container in Raw view mode.

To extend the New Object - User wizard

1. Open the Active Roles console and switch into Raw view mode (select **View | Mode**, then click **Raw Mode** and click **OK**).
2. In the console tree, expand **Configuration | Application Configuration**, and select the **Active Roles Display Specifiers (Custom)** container.

3. Use the **All Tasks | Advanced Create** command to create the appropriate locale container.

   The custom display specifier must be created in the locale container matching the locale of your environment. These locale containers are named using the hex representation of that locale's LCID. Thus the US/English locale's container is named **409**, the German locale's container is named **407**, the Japanese locale's container is named **411**, and so forth.

   You may need to first create the appropriate locale container. You can do this by using the **All Tasks | Advanced Create** command to create an object of the class **EDS-Display-Specifier-Container**.

4. In the locale container, create the custom display specifier named **user-Display**.

   You can do this by using the **All Tasks | Advanced Create** command on the locale container to create an object of the class **Display-Specifier**. Note that the name of the display specifier is case-sensitive, so you should type the name for the new display specifier exactly **user-Display**, not user-display or User-display.

5. In the details pane, right-click **user-Display** and click **Properties**.
6. Go to the **Other Properties to Display** tab.
7. Add one or more properties to the **Other properties in the object creation wizard** list. Then, click **OK**.
8. Restart the Administration Service and then reconnect the Console to the Service, for your changes to take effect.

As a result of these steps, the New Object - User wizard includes an extra page where you can specify values for the properties you selected in Step 7. You can start the wizard in the Active Roles console by right-clicking an organizational unit in the console tree, and then selecting **New | User**. Follow the wizard steps to reach the page containing the list of "other" properties.

"Other Properties" tab in the Properties dialog box

The Active Roles console also makes it possible to extend the **Properties** dialog box for directory objects with an extra tab named **Other Properties**, allowing the management of a custom set of object properties through the use of the **Properties** command.

The Active Roles console makes it easy to view or modify the set of properties on the **Other Properties** tab by using a separate tab in the **Properties** dialog box for display specifier objects. In this way, you can customize the set of properties included on the **Other Properties** tab. Note that the **Properties** dialog box only includes the **Other Properties** tab if there are any properties to display on that tab.

The **Other Properties to Display** tab can be used to add or remove properties from the **Other Properties** tab, only affecting the object type that the display specifier is associated with. The **Other Properties to Display** tab lists the object properties included

on the **Other Properties** tab for that object type, and allows you to make changes to the list.

You can use the following instructions to add the **Other Properties** tab to the **Properties** dialog box for user objects. Similarly, you can extend the property pages for a different object type by creating and configuring a custom display specifier for that object type. For example, to extend the **Properties** dialog box for Group, Computer, or Organizational Unit, create and configure a custom display specifier named **group-Display**, **computer-Display**, or **organizationalUnit-Display**, respectively.

Note that the names of display specifiers are case-sensitive, so you must type the name exactly as specified in the Active Directory schema. To view the names of display specifiers, you can use the console to examine the "Active Directory | Configuration Container | Display Specifiers | 409" container in the Raw view mode.

To extend the Properties dialog box for User objects

1. Open the Active Roles console and switch into Raw view mode (select **View | Mode**, then click **Raw Mode** and click **OK**).
2. In the console tree, expand **Configuration | Application Configuration**, and select the **Active Roles Display Specifiers (Custom)** container.
3. Use the **All Tasks | Advanced Create** command to create the appropriate locale container.

   The custom display specifier must be created in the locale container matching the locale of your environment. These locale containers are named using the hex representation of that locale's LCID. Thus the US/English locale's container is named **409**, the German locale's container is named **407**, the Japanese locale's container is named **411**, and so forth.

   You may need to first create the appropriate locale container. You can do this by using the **All Tasks | Advanced Create** command to create an object of the class **EDS-Display-Specifier-Container**.
4. In the locale container, create the custom display specifier named **user-Display**.

   You can do this by using the **All Tasks | Advanced Create** command on the locale container to create an object of the class **Display-Specifier**. Note that the name of the display specifier is case-sensitive, so you should type the name for the new display specifier exactly **user-Display**, not user-display or User-display.
5. In the details pane, right-click **user-Display** and click **Properties**.
6. Go to the **Other Properties to Display** tab.
7. Add one or more properties to the **Other properties on the object property pages** list. Then, click **OK**.
8. Restart the Administration Service and then reconnect the Console to the Service, for your changes to take effect.

As a result of these steps, the **Properties** dialog box includes the **Other Properties** tab where you can view or modify values of the properties you selected in Step 7. You can access that tab in the Active Roles console by right-clicking a user account and then clicking **Properties**.

Customizing object display names

In Active Directory, each object type may have a display name, and each property of objects may have a display name. In user interfaces, display names are used as friendly names to identify object types and properties. The display names specific to a given object type are stored in the display specifier objects for that object type.

The Active Roles console makes it easy to view or modify display names by using a separate tab in the **Properties** dialog box for display specifier objects. The **Display Names** tab provides a convenient way to customize display names for object types and properties.

The **Display Names** tab can be used to specify or change the display name for the object type that the display specifier is associated with, and to add, modify or remove display names for properties of objects of that type. The property display names are managed using a list of name pairs, with the first name being the LDAP display name of a property and the display name of that property following the LDAP display name.

To customize the English-language display name for the User object class within a forest

1. Open the Active Roles console and switch into Raw view mode (select **View | Mode**, then click **Raw Mode** and click **OK**).
2. In the console tree, expand **Active Directory | Configuration Container | Display Specifiers**, and select the **409** container.
3. In the details pane, right-click **user-Display** and click **Properties**.
4. On the **Display Names** tab, in **Display name for object type**, modify the display name as appropriate, and then click **OK**.
5. Restart the Administration Service and then reconnect the Console to the Service, for your changes to take effect.

By using these steps, you make changes to the display specifier held in Active Directory, so your changes affect not only Active Roles but also any client application intended to manage user objects in Active Directory, such as Active Directory Users and Computers. If you only want the display names to be customized within the Active Roles client interfaces, make changes to the custom display specifiers held in the **Active Roles Display Specifiers (Custom)** container. The **Properties** dialog box for custom display specifiers also includes the **Display Names** tab, allowing you to customize display names so that your changes only affect the Active Roles environment.

Using Configuration Center

Configuration Center provides a single solution for configuring Administration Service instances and Web Interface sites, allowing you to perform the core configuration tasks from a single location. Highlights include:

- Initial configuration tasks such as creation of Administration Service instances and default Web Interface sites
- Import of configuration and management history from earlier Active Roles versions
- Management of core Administration Service settings such as the Active Roles Admin account, service account, and database connection
- Creation of Web Interface sites based on site configuration objects of the current Active Roles version or by importing site configuration objects of earlier Active Roles versions
- Management of core Web Interface site settings such as the site's address on the Web server and configuration object on the Administration Service
- Configuration of One Identity Starling Join for Active Roles
- Management of MMC interface user access

The Configuration Center operations are fully scriptable using Windows PowerShell command-line tools provided by the Active Roles Management Shell.

Configuration Center design elements

Configuration Center is composed of the following elements:

- **Initial configuration wizards** After completing Active Roles Setup, the administrator uses the initial configuration wizards to create a new Active Roles instance, including the Administration Service and Web Interface. The wizards allow you to specify, in a logical manner, all the required configuration settings.
- **Hub pages and management wizards** Once initial configuration has been completed, Configuration Center provides a consolidated view of the core Active Roles configuration settings, and offers tools for changing those settings. Hub pages in the Configuration Center main window display the current settings specific to the Administration Service and Web Interface, and include commands to start management wizards for changing those settings.
- From the **Administration Service** page, you can view or change the service account, Active Roles and Admin account; configure the Active Roles Configuration Database and the Management history database; import configuration data or management history data from an Active Roles database of an earlier version or the current version; view status information, such as whether the Administration Service is started and ready for use; start, stop or restart the Administration Service.

  By allowing configuration data to be imported at any convenient time, Configuration Center makes Active Roles much easier to upgrade. You can install the new Administration Service version side-by-side with an earlier version and then import configuration data to the new version as needed.
- From the **Web Interface** page, you can view, create, modify, delete Web Interface sites, enable force SSL redirection, and configure authentication settings; export configuration of any existing Web Interface site to a file; open each site in a Web

browser. The site parameters available for setting, viewing and changing include the site's address (URL, which is based on the Web site and alias of the Web application that implements the Web Interface site on the Web server) and the configuration object that stores the site's configuration data on the Administration Service. When creating or modifying a Web Interface site, you can reuse an existing configuration object, or create a new configuration object based on a template or by importing data from another configuration object or from an export file.

Wizards that start from hub pages help you manage configuration settings. Management wizards streamline the core configuration tasks by reducing time it takes to change the service account, Active Roles Admin account and database; import configuration and management history; and configure Web Interface sites on the Web server.

- From the **Join to One Identity Starling** wizard, you can enable Active Roles to connect to One Identity Starling, the Software as a Service (SaaS) solution of One Identity.
- From the **MMC Interface Access** wizard, you can manage the settings for enabling or disabling user login to MMC interface.
- **Configuration Shell** Active Roles Management Shell enables access to all Configuration Center features and functions from a command line or from a script, allowing for unattended configuration of Active Roles components. The Windows PowerShell module named `ActiveRolesConfiguration` provides cmdlets for the key set of configuration tasks, such as creation of the Active Roles database, creation or modification of Administration Service instances and Web Interface sites, data exchange between Active Roles databases and between site configuration objects, querying the current state of the Administration Service, and starting, stopping or restarting the Administration Service. The cmdlets provided by the `ActiveRolesConfiguration` module have their noun prefixed with AR, such as `New-ARDatabase`, `Set-ARService`, or `Set-ARWebSite`.

Configuring a local or remote Active Roles instance

Configuration Center is installed as part of the Management Tools component when you install Active Roles on a 64-bit (x64) system. You can use this tool to perform configuration tasks on the local or remote computer that has the current version of the Administration Service or Web Interface installed. Configuration Center looks for these components on the local computer and, if no components has been found, prompts you to connect to a remote computer. Another way to connect to a remote computer is by using the menu on the heading bar at the top of the Configuration Center main window.

When connecting to a remote computer, Configuration Center prompts you for a user name and password. This must be the name and password of a domain user account that belongs to the Administrators group on the remote computer. In addition, whether you are going to perform configuration tasks on the local computer or on a remote computer, your logon account must be a member of the Administrators group on the computer running Configuration Center.

To perform configuration tasks on a remote computer, Configuration Center requires Windows PowerShell remoting to be enabled on that computer. Run the `Enable-PSRemoting` command in the PowerShell console to enable remoting (see the Enable-PSRemoting help topic at http://go.microsoft.com/fwlink/?LinkID=144300 for further details). On Windows Server 2012 or later, remoting is enabled by default.

Running Configuration Center

Configuration Center is installed and, by default, automatically started after you install the Administration Service or Web Interface, allowing you to perform initial configuration tasks on the computer on which you have installed those components. If you close Configuration Center and want to start it again, you can start Configuration Center from the following locations:

- On Windows Server 2012 or later, click the **Active Roles 7.4 Configuration Center** tile on the **Apps** page.

As Configuration Center can manage Active Roles not only on the local computer but also on remote computers, it is possible to use it on a client operating system as well as on server operating systems. You can install Configuration Center by installing Active Roles Management Tools on a 64-bit (x64) server or client operating system, and then connect it to a remote computer on which the Administration Service or Web Interface is installed. To start Configuration Center on a client operating system:

- On Windows 7, select **Start | All Programs | One Identity Active Roles 7.4 | Active Roles 7.4 Configuration Center**.
- On Windows 8 or later, click the **Active Roles 7.4 Configuration Center** tile on the **Apps** page.

Pre-requisites to run the Configuration Center

To run Configuration Center on a given computer, you must be logged on with a user account that has administrator rights on that computer.

If neither the Administration Service nor the Web Interface is installed on the local computer, then Configuration Center prompts you to select a remote computer. In the **Select Server** dialog box that appears, supply the fully qualified domain name of a server, on which the Administration Service or the Web Interface (or both) is installed, and type the logon name and password of a domain user account that has administrator rights on that server. You can connect to a remote server at any time by selecting the **Connect to another server** command from the menu on the heading bar at the top of the Configuration Center main window, which also displays the **Select Server** dialog box.

Before launching Configuration Center, it is recommended to perform the following steps:

1. On the system where Active Roles is installed, browse to *C:\Program Files\One Identity\Active Roles\7.4\Shell*.

2. Right click on the **ActiveRolesServiceConfiguration.psm1** file and select **Properties**.
3. On the **ActiveRolesServiceConfiguration Properties** dialog box, click **Digital Signatures->Details**.
4. On the **Digital Signatures Details** dialog box, click **View Certificate**.
5. On the **Certificate** dialog box, click **Install Certificate...**.
6. On the **Certificate Import Wizard** dialog box, from the **Store Location** select **Local Machine** and click **Next**.
7. On the **Certificate Store** section, select **Place all certificates in the following store** and click **Browse**.
8. On the **Select Certificate Store** dialog box, select **Trusted Publishers** and click **OK**.

   The **Certificate store** field is populated with the selected store name.
9. Click **Next**.

   The **Certificate Import Wizard** displays the selected certificate store.
10. Click **Finish**.

    The **Certificate Import Wizard** displays a message indicating that the import was successful.

> NOTE: If the Certificates from Trusted Publishers are not installed on the system on which Active Roles is installed, then the Configuration Center may not launch successfully.

Tasks you can perform in Configuration Center

Configuration Center enables you to perform:

- Initial configuration tasks, creating the Administration Service instance and the default Web Interface sites.
- Configuration management tasks, letting you manage the existing instance of the Administration Service or Web Interface.
- Logging management tasks, enabling or disabling, and viewing AppInsights and diagnostic logs for Active Roles components that are installed on the computer running Configuration Center.
- Configuration task to join Active Roles to One Identity Starling.
- Management of MMC interface user login settings.

To perform configuration tasks, you need administrator rights on computer on which the Administration Service or Web Interface is installed. In addition, if you are going to create a new Active Roles database, then you need SQL Server rights sufficient to create

databases. If you don't plan to create a new database, then you only need to be a member of the db_owner fixed database role in the Active Roles database used by the Administration Service.

To perform logging management tasks, you need administrator rights on the computer running Configuration Center.

Initial configuration tasks

Active Roles Setup only installs and registers the Active Roles files, without performing any configuration. Upon completion of Active Roles Setup, Configuration Center is used to create an instance of the Administration Service and deploy the default Web Interface sites. Here you can find an overview of these initial configuration tasks.

Configure the Administration Service

The Configure Administration Service wizard creates the Administration Service instance, getting the Administration Service ready for use. The wizard prompts you to supply the following settings:

- The logon name and password of the account in which this Administration Service instance will be running (service account)
- The name of the group or user account that will have full access to all Active Roles features and functions through this Administration Service instance (Active Roles Admin)
- The database in which this Administration Service instance will store the configuration data and management history data

  You have the option to create a new database, or use an existing database of the current Active Roles version. It is possible to have multiple Administration Service instances use the same database.
- The authentication mode that this Administration Service instance will use when connecting to the database

  With the Windows authentication option, the Administration Service will use the credentials of the service account; with the SQL Server authentication option, the Administration Service will use the SQL login name and password you supply in the wizard.
- With Azure AD authentication option, the Administration Service will use username and password of the AD User.
- Azure Databases can be connected using SQL Server authentication or Azure AD authentication.

To start the wizard, click **Configure** in the **Administration Service** area on the **Dashboard** page in the Configuration Center main window. For further information and step-by-step instructions, see "Steps to deploy the Administration Service" in the *Active Roles Quick Start Guide*.

Configure the Web Interface

The Configure Web Interface wizard creates the default Web Interface sites, getting the Web Interface ready for use. The wizard prompts you to choose which Administration Service will be used by the Web Interface you are configuring. The following options are available:

- Use the Administration Service instance running on the same computer as the Web Interface
- Use the Administration Service instance running on a different computer

  This option requires you to supply the fully qualified domain name of the computer running the desired instance of the Administration Service.
- Let the Web Interface choose any Administration Service instance that has the same configuration as the given one

  This option requires you to supply the fully qualified domain name of the computer running the Administration Service instance of the desired configuration. If your environment employs Active Roles replication, this must be the computer running the Administration Service instance whose database server acts as the Publisher for the Active Roles configuration database.

To start the wizard, click **Configure** in the **Web Interface** area on the **Dashboard** page in the Configuration Center main window. For further information and step-by-step instructions, see the "Initial configuration" topic in the "Installing and configuring the Web Interface" section in the Active Roles Quick Start Guide.

Administration Service management tasks

After installing Active Roles, you perform the initial configuration task to create the Administration Service instance, getting it ready for use. Then, you can use Configuration Center to:

- View or change the core Administration Service settings such as the service account, the Active Roles Admin account, and the database
- Import configuration data from an Active Roles database of the current version or an earlier version to the current database of the Administration Service
- Import management history data from an Active Roles database of the current version or an earlier version to the current database of the Administration Service
- View the state of the Administration Service
- Start, stop or restart the Administration Service

Here you can find an overview of these tasks.

View the core Administration Service settings

On the **Administration Service** page in the Configuration Center main window, you can view:

- The logon name of the service account
- The name of the group or user account that has the Active Roles Admin rights
- The SQL Server instance that hosts the Active Roles Configuration database
- The name of the Active Roles Configuration database
- The Configuration database connection authentication mode (Windows authentication or SQL Server login)
- The SQL Server instance that hosts the Active Roles Management History database
- The name of the Active Roles Management History database
- The Management History database connection authentication mode (Windows authentication or SQL Server login)

Change the core Administration Service settings

From the **Administration Service** page in the Configuration Center main window, you can change:

- The service account

  Click **Change** in the **Service account** area. In the wizard that appears, supply the logon name and password of the domain user account in which you want the Administration Service to run.

- The Active Roles Admin account

  Click **Change** in the **Active Roles Admin** area. In the wizard that appears, specify the group or user account you want to have the Active Roles Admin rights.

- The Active Roles database

  Click **Change** in the **Active Roles database** area. In the wizard that appears, specify the database type and the database server instance and the database you want the Administration Service to use, and choose the database connection authentication mode (Windows authentication or SQL Server login). You have the option to specify a separate database for storing management history data.

  NOTE: Azure Databases can be connected only using SQL Server authentication.

Import configuration data

IMPORTANT:
During in-place upgrade, when importing from the source database (Configuration and Management History database), the following database permissions are automatically migrated from the previously used (source) SQL database to the new (destination) SQL

database:

- ARS database users with associated permissions.
- SQL logins mapped to ARS database users.
- Roles.

The service account that is used for performing the in-place upgrade or the import or migration operation should have the following permissions in the SQL Server to perform the operation:

- **db_datareader** fixed database role in the source database.
- **db_owner** fixed database role and the default schema of **dbo** in the destination database.
- **sysadmin** fixed server role in the destination database.

By default, the database users, permissions, logins, and roles are imported to the destination database. You can clear the **Copy database users, permissions, logins, and roles** check box in the following locations depending on the operation:

- During in-place upgrade: in the **Upgrade configuration** window.
- Importing configuration: **Import Configuration > Source Database > Configure advanced database properties**.
- Importing management history: **Import Management History > Source database > Configure advanced database properties**.

NOTE: Depending on the infrastructure, the import operation may take several minutes to complete.

The task of importing configuration data arises when you upgrade the Administration Service. In this case, you need to transfer the Active Roles configuration data from the database used by your Administration Service of the earlier version to the database used by your Administration Service of the new version. To perform this task, click **Import configuration** on the **Administration Service** page in the Configuration Center main window, and follow the steps in the Import configuration wizard that appears.

The Import configuration wizard prompts you to specify the Active Roles database from which you want to import the configuration data (source database) and identifies the database of the current Administration Service to which the configuration data will be imported (destination database), letting you choose the connection authentication mode (Windows authentication, SQL Server login or Azure AD login) for each database.

The **Add-on advisor** page displays all the pre-installed add-ons for the earlier version of Active Roles. These Add-ons must be uninstalled manually from the earlier version using the Active Roles Add-on Manager and from the system where ever applicable, before continuing configuration import.

The **Azure Tenant association** page displays the lists of configured Azure tenants in the source database and options for association. The Azure Tenant association section notifies you to select an Azure tenant from the drop-down list of Azure tenants configured in the source database, and the selected Azure tenant is associated with all Azure objects in the destination database. You can also choose to **Run Azure Tenant association**

, where you select the date and time from the Calendar to run the Azure tenant association.

NOTE:

- If Azure Tenant association is scheduled at a certain time and the upgrade/import operation is still in progress or completes after the Azure Tenant association scheduled time, the tenants are not associated. You have to run the built-in scheduled task **Update Azure Objects Associated Tenant Id** from the Active Roles console to manually associate the Azure Tenants.
- Alternatively, Azure Tenant association can be run at any time using the template workflow **Update Azure Objects Associated Tenant Id** available in the Built-in Workflow Container. The parameter in the script used by the workflow can be configured with the required tenant ID. You can use the drop-down to select a default Azure Tenant from the list of available Azure Tenants. The script used by the workflow can be modified to Search Azure objects based on the requirement.

The **Services association** page displays options to configure the Administration services for executing Dynamic Groups, Group Families, and Scheduled tasks. You can choose to run the Services association immediately or Schedule Service association.

NOTE: If Services association is scheduled at a certain time and the upgrade/import operation is still in progress or completes after the Services association scheduled time, the services are not associated. You have to run the built-in scheduled task **Update Services To ExecuteOn** from the Active Roles console to manually associate the Services.

To ensure Dynamic Groups, Group Families, and Scheduled tasks continue to function after an import the installation configures the new Active Roles server as the executing server for the tasks mentioned above. The configuration mentioned in the **Services association** page runs after an upgrade.

NOTE:

- Alternatively, Services association can be performed any time using the template workflow **Update Services To Execute On** available in the built-in Workflow Container. The parameters in the script used by the workflow can be configured to the required administration services, such as, **Dynamic Group Service**, **Group Family Service**, **Scheduled Task Service**. You can select the administration service from the drop-down list. The drop-down list displays all the currently running administration services that are connected to the current configuration database. If the parameter value is not selected, then the current administration service is used.
- Services association does not update certain scheduled tasks, For example, scheduled tasks that cannot be edited (Managed Object Counter) or scheduled tasks that are set to **All servers** option.

After successfully uninstalling the add-ons, the wizard performs the import operation. During the import operation, the wizard retrieves and upgrades the data from the source database, and replaces the data in the destination database with the upgraded data from the source database.

For further information and step-by-step instructions, see “Importing configuration data” in the Active Roles Quick Start Guide.

Import management history data

IMPORTANT:
During in-place upgrade, when importing from the source database (Configuration and Management History database), the following database permissions are automatically migrated from the previously used (source) SQL database to the new (destination) SQL database:

- ARS database users with associated permissions.
- SQL logins mapped to ARS database users.
- Roles.

The service account that is used for performing the in-place upgrade or the import or migration operation should have the following permissions in the SQL Server to perform the operation:

- **db_datareader** fixed database role in the source database.
- **db_owner** fixed database role and the default schema of **dbo** in the destination database.
- **sysadmin** fixed server role in the destination database.

By default, the database users, permissions, logins, and roles are imported to the destination database. You can clear the **Copy database users, permissions, logins, and roles** check box in the following locations depending on the operation:

- During in-place upgrade: in the **Upgrade configuration** window.
- Importing configuration: **Import Configuration > Source Database > Configure advanced database properties**.
- Importing management history: **Import Management History > Source database > Configure advanced database properties**.

Although this task looks similar to the task of importing configuration data, there are important differences:

- Due to a much larger volume of management history data compared to configuration data, importing management history data takes much longer than importing configuration data.
- As management history data has dependencies on configuration data (but not vice versa), configuration data must be imported first, and then management history data can be imported as needed.

Because of these considerations, Configuration Center provides a different wizard for importing management history. The distinctive features of the Import Management History wizard are as follows:

- The wizard does not replace the existing data in the destination database. It only retrieves and upgrades management history records from the source database, and then adds the upgraded records to the destination database.

- The wizard allows you to specify the date range for the management history records you want to import, so you can import only records that occurred within a particular time frame instead of importing all records at a time.
- Canceling the wizard while the import operation is in progress does not cause you to lose the import results, so you can stop the import operation at any time. The records imported by the time that you cancel the wizard are retained in the destination database. If you start the wizard again, the wizard imports only records that were not imported earlier.

To start the Management History Import wizard, click **Import Management History** on the **Administration Service** page in the Configuration Center main window. The wizard prompts you to specify the Active Roles database from which you want to import the management history data (source database) and identifies the database of the current Administration Service to which the management history data will be imported (destination database), letting you choose the connection authentication mode (Windows authentication, SQL Server login, or Azure AD login) for each database. Then, the wizard lets you choose whether you want to import all management history records or only records within a certain date range, and performs the import operation. During the import operation, the wizard retrieves and upgrades management history records from the source database, and adds the upgraded records to the destination database.

For further information and step-by-step instructions, see "Importing management history data" in the *Active Roles Quick Start Guide*.

View the state of the Administration Service

On the **Administration Service** page in the Configuration Center main window, you can view the state of the Administration Service, such as:

- **Ready for use** Administration Service is running and ready to process client requests.
- **Getting ready** Administration Service has just started and is preparing to process client requests.
- **Stopping** Administration Service is preparing to stop.
- **Stopped** Administration Service is stopped.
- **Unknown** Unable to retrieve the state information.

Start, stop or restart the Administration Service

You can start, stop or restart the Administration Service by clicking the **Start**, **Stop** or **Restart** button at the top of the **Administration Service** page in the Configuration Center main window. If the function of a given button is not applicable to the current state of the Administration Service, the button is unavailable.

Web Interface management tasks

After installing Active Roles, you perform the initial configuration task to create the default Web Interface sites, getting the Web Interface ready for use. Then, you can use Configuration Center to:

- Identify the Web Interface sites that are currently deployed on the Web server running the Web Interface
- Create, modify or delete Web Interface sites
- Export a Web Interface site's configuration object to a file

Here you can find an overview of these tasks.

Identify Web Interface sites

The **Web Interface** page in the Configuration Center main window lists all Web Interface sites of the current version that are deployed on the Web server running the Web Interface. For each Web Interface site, the list provides the following information:

- **IIS Web site** The name of the Web site that holds the Web application implementing the Web Interface site
- **Web app alias** The alias of the Web application that implements the Web Interface site, which defines the virtual path of that application on the Web server
- **Configuration** Identifies the object that holds the Web Interface site's configuration and customization data on the Active Roles Administration Service

From the **Web Interface** page, you can open Web Interface sites in your Web browser: Click an entry in the list of Web Interface sites and then click **Open in Browser** on toolbar.

Create a Web Interface site

You can create a Web Interface site by clicking **Create** on the **Web Interface** page in the Configuration Center main window. The Create Web Interface Site wizard appears, prompting you to:

- Choose the Web site to contain the Web application that implements the new Web Interface site
- Supply the desired alias for that Web application. The alias defines the virtual path that becomes part of the Web Interface site's address (URL).

Then, the wizard lets you specify the object to hold the configuration and customization data of the new Web Interface site on the Active Roles Administration Service. You can choose from the following options:

- Create the object from a template

  The new site will have the default configuration and customization based on the template you select.

- Use an existing object

  The new site will have the same configuration and customization as any existing Web Interface site that also uses the object you select. This option is intended for the scenario where you create an additional instance of one of your existing Web Interface sites on a different Web server.

- Create the object by importing data from another object

  The new site will inherit the configuration and customization of the site that used the object you select for data import. This option is mainly intended for the upgrade scenario where you create Web Interface sites of the new Active Roles version that have the same configuration and customization as your Web Interface sites of an earlier Active Roles version. In this scenario, you import the configuration data of the earlier version to the Administration Service of the new version (which also imports the site configuration objects of the earlier version), and then create configuration objects for Web Interface sites of the new version by importing data from site configuration objects of the earlier version.

- Create the object by importing data from an export file

  The new site will inherit the configuration and customization of the site whose configuration data was saved to the export file you specify. You can choose an export file of any supported Active Roles version.

For further information and step-by-step instructions, see the "Additional configuration" topic in the "Installing and configuring the Web Interface" section in the Active Roles Quick Start Guide.

Modify a Web Interface site

From the **Web Interface** page in the Configuration Center main window, you can make changes to existing Web Interface sites: Click an entry in the list of sites and then click **Modify** on the toolbar. The Modify Web Interface Site wizard starts, allowing you to:

- Choose the Web site to contain the Web application that implements the Web Interface site
- Supply the desired alias for that Web application. The alias defines the virtual path that becomes part of the Web Interface site's address (URL).

Then, the wizard lets you specify the object to hold the site's configuration and customization data on the Active Roles Administration Service. You can choose from the following options:

- Keep on using the current object (default option)

  The site's configuration will remain intact. The wizard displays the name and version of the current configuration object.

- Create the object from a template

  The site will have the default configuration and customization based on the template you select.

- Use an existing object

  The site will have the same configuration and customization as any existing Web Interface site that also uses the object you select. You could use this option to deploy an additional instance of one of your existing Web Interface sites on a different Web server.

- Create the object by importing data from another object

  The site will inherit the configuration and customization of the site that used the object you select for data import. You could use this option to deploy a Web Interface site of the new Active Roles version with the same configuration and customization as one of your Web Interface sites of an earlier Active Roles version. In this case, you import the configuration data of the earlier version to the Administration Service of the current version (which also imports the site configuration objects of the earlier version), and then create the site configuration object by importing data from the appropriate site configuration object of the earlier version.

- Create the object by importing data from an export file

  The site will inherit the configuration and customization of the site whose configuration data was saved to the export file you specify. You can choose an export file of any supported Active Roles version.

For further information and step-by-step instructions, see the "Additional configuration" topic in the "Installing and configuring the Web Interface" section in the Active Roles Quick Start Guide.

Delete a Web Interface site

On the **Web Interface** page in the Configuration Center main window, you can delete Web Interface sites: Click an entry in the list of sites and then click **Delete** on the toolbar. This operation only deletes the Web Interface site from the Web server, without deleting the site's configuration object from the Administration Service.

When you delete a site, the site's configuration object remains intact on the Administration Service. You can set up a Web Interface site with the same configuration as the site you have deleted, by choosing the option to use that object on the **Configuration** step in the wizard for creating or modifying Web Interface sites.

Export a Web Interface site's configuration object to a file

From the **Web Interface** page in the Configuration Center main window, you can export site configuration objects: Click an entry in the list of sites and then click **Export Configuration** on the toolbar. A wizard starts, prompting you to specify the export file. The wizard then retrieves the site's configuration object from the Administration Service, and saves the data from that object to the export file.

The export file could be considered a backup of the site's configuration. You can set up a Web Interface site with the configuration restored from an export file, by importing that file on the **Configuration** step in the wizard for creating or modifying Web Interface sites.

Configure Web interface for secure communication

By default, Active Roles users connect to the Web interface using a HTTP protocol, which does not encrypt the data during communication. However, it is recommended to use a HTTPS protocol to transfer data securely over the web. You can use the **Force SSL Redirection** option in the Configuration Center to enable secure communication over HTTPS for the Web interface on local or remote servers.

To configure the Web interface for secure communication for the first time

1. In the Configuration Center main window, click **Web Interface**.

   The Web Interface page lists all the Web interface sites that are deployed on the Web server running the Web interface.

2. To modify the secure communication settings for the sites, click **Force SSL Redirection**.

   The **Manage Force SSL Redirection Settings** for sites window is displayed.

3. In the **Available Websites** field, select the required web site from the drop-down list.

   The configuration status of the website is displayed.

4. To enable the force SSL redirection, switch between the **Enable Force SSL Redirection** states. Turn it on.

   NOTE:

   - If the website is not configured earlier for secure communication, the **Enable Force SSL Redirection** option is not selected by default and the HTTPS configuration status is shown as **Not configured**.
   - If the website is configured earlier for secure communication, then the **Enable Force SSL Redirection** option is selected by default and the HTTPS configuration status shows as **Configured**.
   - If the website is configured earlier for secure communication, and the SSL bindings was deleted in the IIS site, the **Enable Force SSL Redirection** option is selected by default. The status **Binding Deleted** is displayed. In this case, the secure communication must be configured again for the web site.

5. In the **Available HTTPS Bindings** field, click the drop-down list and select the required binding for the web site.

6. Click **Modify**.

   After successful completion of configuration changes, in the Web Interface window, the Force SSL Redirection configuration state for the selected web site is displayed as green and enabled.

7. Click **Finish**.

   NOTE: The browser cache must be cleared after any changes are made to SSL settings.

   For the configured web site, any HTTP communication is now redirected to HTTPS automatically.

Disabling secure communication for Web interface sites

By default, Active Roles users connect to the Web interface using a HTTP protocol, which does not encrypt the data during communication. However, it is recommended to use a HTTPS protocol to transfer data securely over the web. You can use the **Force SSL Redirection** option in the Configuration Center to enable secure communication over HTTPS for Web interface on local or remote servers.

In case you do not want a secure communication enabled for transferring data over the web, you can disable the HTTPS option using the **Force SSL Redirection** option in the Configuration Center.

To disable the secure communication for Web interface sites

1. In the Configuration Center main window, click **Web Interface**.

   The Web Interface page displays all the Web interface sites that are deployed on the Web server running the Web interface.

2. To modify the secure communication settings for the sites, click **Force SSL Redirection**.

   The Manage Force SSL Redirection Settings for sites window is displayed. The **Enable Force SSL Redirection** option is enabled after HTTPS configuration.

3. In the **IIS Web site** field, select the required web site from the drop-down list.
4. To disable the force SSL redirection, switch between the **Enable Force SSL Redirection** states. Turn it off.
5. Click **Modify** , and then **Finish**.

   NOTE: The browser cache must be cleared after any changes are made to the SSL settings.

   After successful completion of the configuration changes, in the Web Interface window, the Force SSL Redirection configuration state for the selected web site is displayed as not configured.

   After disabling the Force SSL Redirection, all communication is now redirected to HTTP.

For more information on secure communication and Federated Authentication, see Working with Federated Authentication.

Configuring Federated authentication

You can access an application or web sites by authenticating them against a certain set of rules known as claims, by using the **Federated authentication** feature. The **Federated authentication** feature uses the Security Assertion Markup Language (SAML), through which you can sign in to an application once using the single sign-on option and you are authenticated to access websites. For more information, see Working with Federated Authentication.

Starling join configuration task

Active Roles version 7.4 supports integration with One Identity Starling services. The Starling Join feature in Active Roles now enables you to connect to One Identity Starling, the Software as a Service (SaaS) solution of One Identity. The Starling Join feature enables access to the Starling services through Active Roles thus allowing to benefit from the Starling services such as Two-factor Authentication and Identity Analytics and Risk Intelligence.

You can use the Active Roles Configuration Center to join One Identity Starling to Active Roles on the Starling wizard.

To start the wizard, click **Configure** in the **Starling** area on the **Dashboard** page in the Configuration Center main window. The Starling wizard enables you to perform the Starling join operation.

For more information on configuring Starling join for Active Roles, see Configuring Active Roles to join One Identity Starling

MMC interface access management

On installing Active Roles on a computer, the MMC interface user access setting is not enabled by default, and any user is enabled to log in to the MMC interface. You can use Configuration Center, to set the Active Roles MMC interface user access.

To manage the MMC interface access

1. On the **Dashboard** page in the **Configuration Settings** main window, in the **MMC Interface Access** area, click **Manage Settings** .
2. On the MMC Interface Access page that opens, in the **Settings** area, click on the **Component** item, and then click **Modify** or double-click on the **Component** item.
3. On the MMC Interface Access wizard that is displayed, select one of the following options:
    - **Allow Console (MMC Interface) access for all users**: Enables user to log in to MMC interface.
    - **Restrict Console (MMC Interface) access for all users**: Selecting this option restricts all non Active Roles Administrators from using the console. All

delegated users are affected, however, it does not apply to Active Roles Administrators.

4. Click **OK**.

   The MMC Interface Access settings get configured successfully. A message is displayed prompting you to restart the Administrative Service to disconnect the current MMC interface user sessions and for the updated settings to be reflected on the MMC interface.

NOTE:

- The user must be delegated with the **User Interfaces** access rights on the **User Interfaces** container under **Server Configuration** to obtain access to the MMC interface. User Interfaces Access templates that provide the access rights are available as part of the Active Roles built-in Access templates in the **User Interfaces** container.
- For information on delegating Console access to specified users, see Delegating control to users for accessing MMC interface

Logging management tasks

You can use Configuration Center to enable or disable, and view diagnostic logs for the Active Roles components that are installed on the computer running Configuration Center. On the **Logging** page, Configuration Center lists the following information:

- **Component** Name of the component, such as Administration Service, Web Interface or Console (MMC Interface)
- **Logging** Indicates whether logging is enabled or disabled for the given component, and the logging level, such as Basic or Verbose
- **Log location** Depending upon the component, identifies either the folder containing the log files or the log file for that component

The toolbar on the **Logging** page allows you to perform the following tasks:

- To enable or disable logging for a given component, select the component in the list, and then click **Modify** on the toolbar.
- To open the folder that contains the log file or files for a given component, select the component in the list, and then click **Browse with Explorer** on the toolbar.
- To examine the Administration Service log file in Log Viewer, select Administration Service in the list of components and then click **Open in Log Viewer** on the toolbar. For information about Log Viewer, see Active Roles Log Viewer later in this document.

Solution Intelligence

Active Roles supports Solution Intelligence to monitor the web application and detect performance issues. Active Roles administrators can enable or disable the Solution Intelligence feature that supports intelligent collection for Active Roles solution usage data.

The telemetry data that is captured for Active Role is sent to the Azure portal and can be accessed by the development team for analysis. In addition to the general telemetry data that is collected by Microsoft Azure, Solution Intelligence in Active Roles helps captures data about the Active Roles language pack usage by customers, referred to as Language pack telemetry and the area of bugs and issues referred to as the diagnostic telemetry.

The Language pack telemetry provides insights for the following:

- Product version
- Language name
- Language display name
- Language code identifier
- Installation of language pack

You can enable or disable Solution Intelligence by using Configuration Center. For information on managing Solution Intelligence for Active Roles, see Enabling or disabling Solution Intelligence.

Enabling or disabling Solution Intelligence

On installing Active Roles on a computer, the Solution Intelligence setting is not enabled by default. To allow the Solution Intelligence to retrieve telemetry data of Active Roles, you can use Configuration Center, to enable the Active Roles Solution Intelligence.

NOTE: Active Roles Service must be installed and running on the system for the **Solution Intelligence** feature to be .

To manage the Solution Intelligence setting

1. On the **Dashboard** page in the **Configuration Settings** main window, click **Solution Intelligence**.
2. On the **Solution Intelligence** page, select the **Enable Solution Intelligence** option.
3. Click **Save**.

   The Solution Intelligence settings are configured successfully and a success message is displayed.

   NOTE: The changed status may take approximately up to 30 minutes to reflect during which, the telemetry may still be sent until new setting is applied to the website. You may Reset IIS if you want the settings to be applied immediately.

Configuring gMSA as an Active Roles Service account

Active Roles Configuration Center enables you to configure the gMSA account as a service account . Before you configure a gMSA account as an Active Roles Service account, the following pre-requisites must be met:

- The Key Distribution Services (KDS) Root Key must be available in the KDS service on the Domain controller.
- The computers and groups that have servers with Active Roles Service installed on them, must be added to the gMSA account.
- The gMSA account must be available in the Local Administrators group where the Active Roles service is installed and in the built-in Administrators group of the domain.
- The gMSA account must have an SQL login with **db_Owner** permission for Active Roles database.
- The gMSA account name must be unique across domains.

NOTE: Exchange operations cannot be performed on the on-premises Exchange Server environment using the gMSA account. For example, Remote mailbox, User mailbox, or Contact.

For information on creating a new database see Configuring the Active Roles Service account to use a gMSA account.

For more information on managing gMSA accounts see Management of Group Managed Service Accounts in the *Active Roles User's Guide*.

Configuring the Active Roles Service account to use a gMSA account

After completion of Active Roles Setup, the Configuration Center enables you to create an instance of the Administration Service to get the Administration Service ready for use.

To configure the Administration Service account to use a gMSA account as the service account during initial configuration

1. Start Configuration Center on the computer running the Administration Service.

   You can start Configuration Center by selecting **Configuration Center** on the **Apps** page or **Start** menu, depending upon the version of your Windows operating system. For detailed instructions, see Running Configuration Center.
2. In the Configuration Center main window, under **Administration Service**, click **Configure**
3. On the **Administration Service** page, in the **Service Account** area, click **Browse**.
4. In the **Select User** or **Service Account** dialog box, click **Object Types**.

5. In the **Object Types** dialog box, select the **Service Accounts** object type along with the Users object type and click OK.
6. In the **Service User** or **Service Account** dialog box, click **Check Names** to select the required gMSA account, and click **OK**.

   The Configure Administration Service dialog box displays the new login name for the gMSA account. The **Password** field is disabled.
7. Click **Next** to complete the service account configuration.

   If the system running the Active Roles Service is not linked to the gMSA account, then an error is displayed prompting you to check if the system is permitted to use the provided gMSA.

   If the gMSA account is not part of the Local Administrators group, then an error is displayed prompting you to check if the gMSA account is a member of the Local Administrator's group on the system.
8. If all the pre-requisites are met, you can proceed to the next step. Provide the name of the group or user account that will have full access to all Active Roles features and functions through this Administration Service instance (Active Roles Admin). Click **Next**.
9. Provide the details for the database in which this Administration Service instance will store the configuration data and management history data.

   You have the option to create a new database or use an existing database of the current Active Roles version. It is possible to have multiple Administration Service instances use the same database.

   > NOTE: When you create a new database, you can add the DB_owner permission to gMSA account for the new database only after the Administration Service is configured.

   Based on the authentication mode that the Administration Service instance uses when connecting to the database, the Administrative Service uses the relevant credentials:

   - With the Windows authentication option, the Administration Service will use the credentials of the service account.
   - With the SQL Server authentication option, the Administration Service will use the SQL login name and password you supply in the wizard.
10. After all steps are complete, review the settings on the **Ready to Configure** summary page and click **Configure** to complete the configuration.

The Active Roles Admin setting is specific to the instance of the Administration Service. If you have multiple Administration Service instances deployed in your environment, then you need to apply the changes on each computer running the Administration Service.

Changing the Active Roles Service account to use a gMSA account

Active Roles provides support to change an Active Roles account to use a gMSA account.

To change the Administration Service account to use a gMSA account as the service account

1. Start Configuration Center on the computer running the Administration Service.

   You can start Configuration Center by selecting **Configuration Center** on the **Apps** page or **Start** menu, depending upon the version of your Windows operating system. For detailed instructions, see Running Configuration Center.

2. In the Configuration Center main window, under **Administration Service** | **Service account**, click **Change**.
3. In the Change Service Account dialog box, under **Service Account**, click **Browse**.
4. In the **Select User** or **Service Account** dialog box, click **Object Types**.
5. In the **Object Type**s dialog box, select the **Service Accounts** object type along with the Users object type and click **OK**.
6. In the **Service User** or **Service Account** dialog box, click **Check Names** to select the required gMSA account, and click **OK**.

   The **Change Service Account** dialog box displays the new login name for the gMSA account. The **Password** field is disabled.

7. Click **Change** to complete the changes to the service account.

   If the system running the Active Roles Service is not linked to the gMSA account, then an error is displayed prompting you to check if the system is permitted to use the provided gMSA.

   If the gMSA account is not part of the Local Administrators group, then an error is displayed prompting you to check if the gMSA account is a member of the Local Administrator's group on the system.

   If all the pre-requisites are met, the service account is changed to GMSA account successfully and the success message is displayed.

Changing the Active Roles Admin account

When you configure the Active Roles Administration Service, you are prompted to specify the group or user account that will have unrestricted access to all Active Roles features and functions. This account is referred to as Active Roles Admin. By default, Active Roles Admin is the Administrators local group on the computer running the Administration Service. You can change this setting in the Configure Administration Service wizard when initially configuring the Administration Service.

After you have configured the Administration Service, you can choose a different Active Roles Admin account by using Active Roles Configuration Center on the computer running the Administration Service.

To change the Active Roles Admin Account

1. Start Configuration Center on the computer running the Administration Service.

You can start Configuration Center by selecting **Configuration Center** on the **Apps** page or **Start** menu, depending upon the version of your Windows operating system. For detailed instructions, see Running Configuration Center.

2. In the Configuration Center main window, under **Administration Service**, click **Manage Settings**.
3. On the **Administration Service** page, in the **Active Roles Admin** area, click **Change**.
4. On the **Active Roles Admin** page in the Change Active Roles Admin wizard that appears, click **Browse** and select the group or user account you want to be designated as Active Roles Admin.

   If you select a group, any member of that group will have the Active Roles Admin rights. If you select a user account, then only that account will have the Active Roles Admin rights.

5. Click the **Change** button on the **Active Roles Admin** page.

The Active Roles Admin setting is specific to the instance of the Administration Service. If you have multiple Administration Service instances deployed in your environment, then you need to apply the changes on each computer running the Administration Service.

Enabling or disabling diagnostic logs

Active Roles administrators can enable diagnostic logging at the request of support personnel to assist them in finding root causes of issues that occur during Active Roles operations. The diagnostic information includes the Active Roles configuration statistics (referred to as Active Roles system summary), the Active Roles Administration Service diagnostic log and the Active Roles Console diagnostic log.

The Active Roles Administration Service's diagnostic log (ds.log) contains tracing information, such as API calls, internal function calls and state transitions performed by the Administration Service. This information is stored in the ds.log file that you can send to the support team for issue diagnostic purposes. Two logging levels are available: Basic and Verbose. The Verbose option writes much more information to the log, which can aid in the process of isolating an issue. However, with the increase in verbosity comes a corresponding decrease in performance and increase in the size of the log file.

The Active Roles console's diagnostic log (EDMSnap.txt) contains debugging information specific to the Active Roles console, and can be helpful in isolating console-related issues.

You can use the Active Roles console to perform the following tasks:

- Export Active Roles system summary.

  This option allows you to save the Active Roles configuration statistics to a file that you can later send to the support team for issue diagnostic purposes.

- Turn the Administration Service's diagnostic log on or off.

  The console shows the path to the log file located on the computer running the Administration Service.

- Choose the level of verbosity for the Administration Service: **Basic** or **Verbose**.

  The **Verbose** option results in a more detailed log, but considerably increases the size of the log file.

- Turn the console's diagnostic log on or off.

  The console shows the path to the console's log file on the local computer.

It is also possible to enable or disable diagnostic logs by using Configuration Center (see Logging management tasks earlier in this document). The following instructions apply to the Active Roles console.

To view or change the diagnostic settings

1. Log on as an Active Roles Admin, and open the Active Roles console.
2. In the Active Roles console tree, click the root node to display the Active Roles summary page in the details pane.
3. On the summary page, expand the **Diagnostics** area.

   In the **Diagnostics** area, you can view whether the Active Roles Administration Service's diagnostic logging is currently enabled (turned on) or disabled (turned off).

4. In the **Diagnostics** area, click **View or change diagnostic settings**.

   This opens the **Diagnostics** page in the **Properties** dialog box for the Administration Service instance to which the console is currently connected. Another way to open that page is by directly opening the **Properties** dialog box from the Administration Service object in the **Configuration/Server Configuration/Administration Services** container.

5. Use the **Diagnostics** page to perform the following tasks:
6. Click **Export Active Roles system summary** to save the Active Roles configuration statistics to a file that you can later send to the support team for issue diagnostic purposes.
   - Click the appropriate option to turn on or off the Administration Service's log. This option enables or disables the Administration Service diagnostic logging on the computer running the Administration Service instance to which the console is currently connected.
   - Choose the level of verbosity from the **Logging level** list, if you have selected the option to turn on the Administration Service's log.
   - View the path and name of the Administration Service's log file, along with the name of the computer that holds the log file.
   - Click the appropriate option to turn on or off the console's log. This option enables or disables the console diagnostic logging on the local computer.

- View the path and name of the console's log file, along with the name of the computer that holds the log file.

7. When finished, click **OK** or **Apply** for your changes to take effect.

Active Roles Log Viewer

The Log Viewer tool enables you to browse and analyze diagnostic log files created by the Active Roles Administration Service as well as event log files created by saving the Active Roles event log in Event Viewer on the computer running the Administration Service. Log Viewer can help you drill down through the sequence or hierarchy of requests processed by the Administration Service, identify error conditions that the Administration Service encountered during request processing, and find Knowledge Articles that apply to a given error condition.

With Log Viewer, you can open an Active Roles diagnostic log file (ds.log) or saved event log file (.evtx), and view a list of:

- Errors encountered by the Administration Service and recorded in the log file
- Requests processed by the Administration Service and traced in the log file
- All trace records found in the diagnostic log file
- All events found in the event log file

When you select an error in the list, you can choose a command to look for solution in Knowledge Base. The command performs a search in One Identity Software Knowledge Base to list the Knowledge Articles that can provide helpful information on how to troubleshoot the error you selected.

Log Viewer also enables you to:

- Search the list for a particular text string, such as an error message
- Filter the list by various conditions, to narrow the set of list items to those you are interested in
- View detailed information about each list item, such as error details, request details or stack trace

Using Log Viewer

To start Log Viewer, click **Start Log Viewer** in the Configuration Center main window.

Once you have started Log Viewer, open your Active Roles diagnostic log file or saved event log file: Click **Open** on the Log Viewer toolbar, and supply the path and name of the log file.

By default, Log Viewer displays a list of errors encountered by the Administration Service and recorded in the log file. You can use Log Viewer to look for information on how to troubleshoot a given error: Right-click the error in the list and then click **Look for**

solution in Knowledge Base. Log Viewer performs a search in One Identity Software Knowledge Base to list the Knowledge Articles that apply to the error you selected.

Other tasks you can perform:

- To view a list of requests processed by the Administration Service and traced in the log file, click **Requests** in the **View** area on the Log Viewer toolbar.
- To view all trace records found in the diagnostic log file or all events found in the event log file, click **Raw log records** in the **View** area on the Log Viewer toolbar.
- To search the list for a particular text string, such as an error message, type the text string in the **Search** box on the Log Viewer toolbar and press Enter.
- To narrow the set of list items to those you are interested in, click **Filter** on the Log Viewer toolbar and specify the desired filter conditions.
- To view detailed information about an error, request, trace record or event, right-click the corresponding list item, and click **Details**.
- To view all trace records that apply to a given request, right-click the corresponding item in the **Requests** list and click **Stack trace**. This task is unavailable in case of an event log file.
- To view the request that caused a given error, right-click the error in the **Errors** list and click **Related request**. This task is unavailable in case of an event log file.

To view all trace records that apply to the request that caused a given error, right-click the error in the **Errors** list and click **Stack trace for related request**. This task is unavailable in case of an event log file.

21

SQL Server Replication

SQL Server database replication feature enables copying and distribution of data between different nodes, which provide the functionality for maintaining replicated data.

Active Roles uses the replication functionality of Microsoft SQL Server to copy and distribute configuration data from one Administration Service database to another, and to synchronize data among the databases for consistency.

To understand the replication terminologies, the type of SQL Server Replication model that Active Roles uses, and the steps required to configure replication see the following topics:

- Replication terminology
- Understanding the Replication model
- SQL Server-related permissions
- Configuring SQL Server
- Configuring replication
- The replication group
- Monitoring replication
- Using AlwaysOn Availability Groups
- Using database mirroring
- Best practices
- Troubleshooting Replication failures

NOTE: SQL Server Books Online should be the primary resource you use for SQL Server replication questions. All contents relevant to Microsoft SQL Server replication is indexed under the "SQL Server Replication" topic in SQL Server Books Online at http://technet.microsoft.com/library/ms151198.aspx.

Replication terminology

This section explains the basic terms and concepts used in replication.

Replication

To replicate its configuration data, Active Roles employs the replication capabilities of Microsoft SQL Server. In SQL Server, the term *replication* refers to a process that copies and distributes data and database objects from one database to another and then synchronizes information between databases for consistency.

Publisher

The *Publisher* is a database server that makes data available for replication to other database servers. The Publisher can have one or more publications, each representing a logically related set of data. In the Active Roles replication model, the Publisher has only one publication.

Subscribers

Subscribers are database servers that receive replicated data. Depending on the type of replication, the Subscriber can propagate data changes back to the Publisher or republish the data to other Subscribers. In the Active Roles replication model, a Subscriber can propagate data changes to the Publisher and receive replicated data from the Publisher.

Distributor

The *Distributor* is a server that hosts the distribution database and stores history data, transactions, and metadata. In the Active Roles replication model, the same server is used as both the Publisher and Distributor.

Replication group

In the Active Roles replication model, the Publisher and its Subscribers are collectively referred to as *replication group*, with each server in the replication group being referred to as *replication partner*.

Replication group is comprised of *replication partners* that include a single Publisher and may include any number of Subscribers. When data in a replication partner's database changes, replication ensures that the data changes are propagated to the databases maintained by all the other replication partners.

NOTE: In the SQL Server documentation, replication partners are referred to as *synchronization partners*.

Standalone database server

When initially set up, the Administration Service's database server is configured as a *standalone server* that does not belong to any replication group.

Articles and publications

Articles are tables of data, partitions of data, or database objects that are specified for replication. Each publication is a collection of articles from one database. This grouping of multiple articles makes it easier to specify a logically related set of data that is to be replicated together. In the Active Roles replication model, each article is a table of data.

SQL Server Agent

SQL Server Agent hosts and schedules the agents used in replication, and provides a way to run replication agents. SQL Server Agent also controls and monitors several other operations outside of replication, including monitoring the SQL Server Agent service, maintaining error logs, running jobs, and starting other processes.

Replication Agents

Replication Agents used with Microsoft SQL Server replication carry out the tasks associated with copying and distributing data. The Active Roles replication model employs the Snapshot Agent and Merge Agents.

Snapshot Agent

The Snapshot Agent prepares schema and initial data files of published tables and stored procedures, stores the snapshot files, and records information about synchronization in the distribution database. In the Active Roles replication model, the Snapshot Agent runs at the Publisher.

Merge Agent

The Merge Agent applies the initial snapshot to the Subscriber, and moves and reconciles incremental data changes that occur. Each Subscriber has its own Merge Agent that connects to both the Publisher and the Subscriber and updates both.

In the Active Roles replication model, the Merge Agents run continuously at the Publisher. Each Merge Agent uploads data changes from its Subscriber to the Publisher, and downloads data changes from the Publisher to the Subscriber.

Understanding the Replication model

NOTE: Operations related to replication are not supported by the Azure SQL databases.

Active Roles replication propagates the changes to configuration data to all replication partners whenever the data is modified on any one of replication partners. To achieve this goal, Active Roles relies on the *merge replication* provided by Microsoft SQL Server. For details on merge replication, refer to the content indexed under the Merge Replication topic in SQL Server Books Online.

In the Active Roles environment, the SQL Server replication function is used to propagate changes to configuration data to all the replication partners, as soon as data is modified on one of the replication partners. The replication process is initiated immediately after changes are committed to a replication partner. Active Roles does not offer the facility to change this behavior.

As there is usually a moderate volume of changes, and since replication only propagates modified data (merge replication model), the amount of replication traffic is manageable. Therefore, you do not need to schedule or manually force replication in Active Roles.

A merge replication model normally requires a means of resolving conflicts that could result from changing the same data on different replication partners. In the Active Roles replication model, the outcome of the conflict is decided on a "later wins" basis, that is, the last to modify the data wins the conflict.

In the Active Roles replication model, each Administration Service database server can have one of the following roles:

- **Publisher** The Publisher is the database server that makes data available for replication to other replication partners.

  The Administration Service that uses the Publisher database server is referred to as the *Publisher Administration Service*.

- **Subscriber** Subscribers are database servers that receive replicated data. Subscribers can receive data changes from the Publisher and propagate data changes back to the Publisher.

  The Administration Service that uses a Subscriber database server is referred to as the *Subscriber Administration Service*.

This section briefly discusses the following elements of the Active Roles replication model:

- Replication group management
- Data synchronization and conflict resolution

Replication group management

The tasks performed when managing a replication group include the Publisher-related tasks, such as **Promote** or **Demote**, and the Subscriber-related tasks, such as **Add** or **Delete**.

Promote

This task assigns the Publisher role to the Administration Service database server, thereby creating a replication group. When performing the **Promote** task, SQL Server creates the **AelitaReplica** publication, and starts the Snapshot Agent. The Agent creates an initial snapshot of schema and data, and saves it to the snapshot folder.

Active Roles automatically specifies and passes to SQL Server all replication settings, such as filters, type of replication, and retention period for subscriptions. For details, see Viewing replication settings later in this document.

Add

This task adds the Administration Service database server to the replication group, thus assigning the Subscriber role to the database server. When performing the **Add** task, SQL Server starts the Merge Agent. The Agent copies data from the Publisher's snapshot folder to the Subscriber SQL Server. This process is referred to as applying the initial snapshot (see "Create and Apply the Snapshot" in SQL Server Books Online at http://msdn.microsoft.com/en-us/library/ms151785.aspx).

Delete

This task removes the Subscriber from the replication group, causing the database server to revert to the standalone state. When performing the **Delete** task, SQL Server deletes the subscription at the Publisher. The database of the former Subscriber retains the replicated data.

Demote

This task removes the Publisher from the replication group, causing the database server to revert to the standalone state. The Publisher can only be demoted after all of its Subscribers are deleted. When performing the **Demote** task, SQL Server deletes the **AelitaReplica** publication, and erases data in the snapshot folder.

Data synchronization and conflict resolution

After applying the initial snapshot to Subscribers, SQL Server tracks changes to published data at the Publisher and at the Subscribers:

- When data is modified at a Subscriber, the data changes are sent to the Publisher. Then, the Publisher propagates the data changes to the other Subscribers.
- When data is modified at the Publisher, the data changes are propagated to the Subscribers.

These operations are performed by the Merge Agents running on the Publisher SQL Server.

The Merge Agents are configured so that once data changes are made at a given replication partner, it normally takes two minutes or less for SQL Server to start synchronizing the data changes with other replication partners. The time required for the synchronization process to be completed depends on SQL Server load and on the bandwidth of network connections. As there is normally a moderate volume of data changes, the replication traffic is manageable.

The synchronization process tracks data changes on both the Subscribers and the Publisher. At the Publisher, the changes are merged to form a single version of the data. During the merge, some conflicts may be found where multiple Subscribers modified the same data.

.Any conflict between the arrived values is automatically resolved based on the Microsoft SQL Server DATETIME (Later Wins) Conflict Resolver: The winner of the conflict is chosen according to a "later wins" solution, with the last to modify the data winning the conflict. For information about conflict resolvers, see Microsoft COM-Based Resolvers in SQL Server Books Online at http://msdn.microsoft.com/en-us/library/ms152573.aspx.

SQL Server-related permissions

The health of Active Roles replication heavily depends on the access permissions that the Administration Service and SQL Server Agent has on SQL Server. The required permissions are listed in the "SQL Server permissions" section in the Active Roles Quick Start Guide.

Configuring SQL Server

To ensure that SQL Server is properly configured for Administration Service replication, ensure that the SQL Server Agent service is started and configured properly.

The SQL Server Agent service must be up and running on SQL Server that holds the role of the Publisher database server (Publisher SQL Server). It is recommended that the startup type for this service be set to **Automatic**.

The SQL Server Agent service should be configured to log on with a domain user account. The service logon account must have sufficient rights to connect to the Publisher SQL Server and to the Subscriber SQL Server (see "Replication agent permissions" in the Active Roles Quick Start Guide).

Configuring replication

Active Roles uses the replication functionality of Microsoft SQL Server to copy and distribute configuration data from one Administration Service database to another, and to synchronize data among the databases for consistency.

Administration Service database servers synchronized by using the SQL Server replication function are referred to as *replication partners*. Each replication partner maintains a writable copy of the Service's configuration and Management history data. Whenever changes are made to one replication partner, the changes are propagated to the other replication partners.

The replication group

The Publisher and its Subscribers constitute a replication group. Every replication group must include a single Publisher and may include any number of Subscribers. The members of a replication group are referred to as *replication partners*.

Each member of a replication groups (replication partner) maintains a separate, writable copy of the Administration Service's configuration and management history data. Replication copies and distributes data from one member database to another, and synchronizes data between the databases for consistency. When changes are made on the Publisher, the Publisher replicates these changes to each Subscriber. When data changes are made on a Subscriber, the Subscriber propagates the changes to the Publisher, which in turn replicates them to the other Subscribers.

This replication process ensures the same configuration for all Administration Services that use the database servers belonging to the replication group.

When initially set up, the Administration Service database server is configured as a *standalone database*, that is, it does not have replication partners and does not belong to any replication group. The Administration Service that uses a standalone database server is referred to as *standalone Administration Service*.

It is possible to add a standalone database server to any replication group that already exists. When you do that, the database server becomes a Subscriber. Each Administration Service database server may belong to only one replication group. Once removed from a replication group, it can be added to a different group.

To create a new replication group, a standalone database server must be designated as the Publisher. The new replication group will then have a single member—the Publisher. Later, you may add Subscribers to the group.

If there are any replication failures in Active Roles, the Active Roles console provides a visual indication of this issue by modifying the icon of the **Server Configuration** and **Configuration Databases** containers in the console tree: a label with the exclamation point appears next to each of the containers. This allows the administrator to detect a replication failure without examining individual replication partners.

Creating a replication group

To create a replication group, designate a standalone Administration Service database server as the Publisher. You can do that by using the Active Roles console:

1. Connect to a standalone Administration Service.
2. Promote the Administration Service database server to Publisher.

To connect to the Administration Service, use the instructions provided earlier in this chapter (see Connecting to the Administration Service).

Once connected to the Administration Service, perform the following steps to promote the Administration Server database server to Publisher:

1. In the console tree, navigate to the **Configuration/Server Configuration/Configuration Databases** container.
2. In the details pane, right-click the database and click **Promote**.

> NOTE: The **Promote** command is only displayed if the Administration Service uses a standalone database server, that is, a database server that does not belong to any replication group.

After you click **Promote**, it takes several minutes to complete the operation. When the operation is completed, the new replication group has a single member—the Publisher. Once the replication group has been created, you can add replication partners—Subscribers.

After the Promote operation is completed, both the configuration and management history databases are replicated.

If Active Roles does not have sufficient rights to perform the **Promote** operation on SQL Server, then the Active Roles console prompts you to supply an alternative account for that operation (see "Permissions for creating or removing the Publisher" in the *Active Roles Quick Start Guide)*.

Adding members to a replication group

To add a member to a replication group, designate a standalone database server as a Subscriber of the group's Publisher. You can do that by using the Active Roles console:

1. Connect to the Publisher Administration Service.
2. Start and complete the New Replication Partner wizard.

Once connected to the Publisher Administration Service, display the contents of the **Configuration Databases** container. The details pane lists the names of the Publisher and its Subscribers. Right-click the Publisher and click **Add Replication Partner**. Then, follow the instructions in the New Replication Partner wizard.

On the **Database Selection** page, click **Browse** to select the Administration Service that uses the database server you want to designate as a Subscriber. Clicking **Browse** displays the dialog box similar to that described in the Connecting to the Administration Service section earlier in this chapter. Type the name of the computer running the Administration Service, or select it from the list.

The wizard automatically selects the database server that hosts the database of the Administration Service you specify. If Active Roles does not have sufficient rights to perform the **Add Replication Partner** operation on SQL Server, then the wizard prompts you to supply an alternative account for that operation (see "Permissions for adding or removing a Subscriber" in the Active Roles Quick Start Guide).

The next page of the wizard displays the database name and location retrieved from the specified Administration Service, and prompts you to select one of the following options that determine how the replication agent running on the Publisher SQL Server will connect to the Subscriber SQL Server:

- **Impersonate SQL Server Agent service account** Use this option if the SQL Server Agent service on the Publisher SQL Server is configured to log on as a Windows user account that has sufficient rights on the Subscriber SQL Server. If this option is selected, the replication agent connects to the Subscriber SQL Server under the logon account of the SQL Server Agent service running on the Publisher SQL Server.
- **Use SQL Server Authentication with the following login and password** Use this option if the SQL Server Agent service logon account cannot be configured to have sufficient rights on the Subscriber SQL Server. You are prompted to specify the SQL Server login and password that the replication agent running on the Publisher SQL Server will use to connect to the Subscriber SQL Server.

The account that the replication agent uses to connect to the Subscriber SQL Server must at minimum be a member of the **db_owner** fixed database role in the subscription database (Active Roles' database on the Subscriber). For further details, see "Replication agent permissions" in the Active Roles Quick Start Guide.

The completion page of the wizard allows you to review summary information about the database server you are going to make a Subscriber. After you click **Finish**, the database server is added to the replication group. The replication process updates the database of the new Subscriber with the data retrieved from the Publisher.

IMPORTANT: The Publisher copies new data to the database, overwriting the existing data. If the database contains valuable information, such as custom Access Templates or Policy Objects, you should export those objects before designating the database server as a Subscriber, and import them back after the operation is completed.

Steps for adding members to a replication group

To add a member to a replication group, designate a standalone database server as a Subscriber of the group's Publisher.

To add a replication partner to a replication group

1. Connect to the Administration Service whose database server holds the Publisher role.

   For instructions on how to connect to the Administration Service, see Connecting to the Administration Service earlier in this chapter.

2. In the console tree, expand **Configuration | Server Configuration**, and click **Configuration Databases**.
3. In the details pane, right-click the Publisher, and click **Add Replication Partner**.
4. Follow the instructions in the New Replication Partner wizard.
5. On the **Database Selection** page, click **Browse**.
6. Use the **Connect to Administration Service** dialog box to select the Administration Service whose SQL Server is to be configured as a Subscriber to this Publisher.

   If Active Roles does not have sufficient rights to perform the **Add Replication Partner** operation on SQL Server, then the wizard prompts you to supply an alternative account for that operation (see "Permissions for adding or removing a Subscriber" in the Active Roles Quick Start Guide).

   The next page of the wizard displays the database name and location retrieved from the specified Administration Service, and prompts you to select one of the following options that determine how the replication agent running on the Publisher SQL Server will connect to the Subscriber SQL Server

7. Choose one of these options:

   - **Impersonate the SQL Server Agent service account** Select this option if the SQL Server Agent service on the Publisher SQL Server is configured to log on as a Windows user account that has sufficient rights on the Subscriber SQL Server. If this option is selected, the replication agent connects to the Subscriber SQL Server under the logon account of the SQL Server Agent service running on the Publisher SQL Server.
   - **Use SQL Server Authentication with the following login and password** Select this option if the SQL Server Agent service logon account cannot be configured to have sufficient rights on the Subscriber SQL Server. You are prompted to specify the SQL Server login and password that the replication agent running on the Publisher SQL Server will use to connect to the Subscriber SQL Server.

   The account that the replication agent uses to connect to the Subscriber SQL Server must at minimum be a member of the **db_owner** fixed database role in the subscription database (Active Roles' database on the Subscriber). For further details, see "Replication agent permissions" in the Active Roles Quick Start Guide.

8. Click **Next**, and the click **Finish**.

NOTE:

- After you click **Finish**, the database server is added to the replication group. The replication process updates the database of the new Subscriber with the data retrieved from the Publisher.
- The Publisher copies new data to the database, overwriting the existing data. If the database contains valuable information, such as custom Access Templates or Policy Objects, you should export those objects before designating the database as a Subscriber, and import them back after the operation is completed.
- A database cannot be added to a replication group if it already belongs to another replication group. To add the database to another replication group, you must first remove it from its current replication group, and then add it to the other one.

Removing members from a replication group

You can remove Subscribers from a replication group by using the Active Roles console:

1. Connect to the Publisher Administration Service.
2. Select a Subscriber from the **Configuration Databases** container, right-click the Subscriber, and click **Delete**.

If Active Roles does not have sufficient rights to perform the operation on SQL Server, then the Active Roles console prompts you to supply an alternative account for that operation (see "Permissions for adding or removing a Subscriber" in the Active Roles Quick Start Guide).

Using this method, you can remove only Subscribers. The Publisher cannot be removed from its replication group when the group includes Subscribers.

To remove the Publisher, you must first remove all Subscribers, and then demote the Publisher. This action deletes the entire replication group.

After you remove all Subscribers, you can demote the Publisher: in the **Configuration Databases** container, right-click the Publisher and click **Demote**.

If Active Roles does not have sufficient rights to perform the **Demote** operation on SQL Server, then the Active Roles console prompts you to supply an alternative account for that operation (see "Permissions for creating or removing the Publisher" in the Active Roles Quick Start Guide).

Steps for removing members from a replication group

To remove Subscribers from a replication group

1. Connect to the Publisher Administration Service.
2. In the console tree, expand **Configuration | Server Configuration**, and click **Configuration Databases**.
3. In the details pane, right-click the Subscriber, and then click **Delete**.

To remove the Publisher from a replication group

1. Connect to the Publisher Administration Service.
2. In the console tree, expand **Configuration | Server Configuration**, and click **Configuration Databases**.
3. In the details pane, right-click the Publisher, and then click **Demote**.

NOTE:

- For information on how to connect to an Administration Service, see Connecting to the Administration Service earlier in this chapter.
- The Publisher cannot be removed from its replication group when the group includes Subscribers. To remove the Publisher, you must first remove all Subscribers, and then demote the Publisher. This action deletes the replication group. After you remove all Subscribers, you can demote the Publisher.
- The **Demote** command is not displayed unless the Publisher is the only member of the replication group.
- If Active Roles does not have sufficient rights to perform the operation on SQL Server, then the Active Roles console prompts you to supply an alternative account for that operation (see "Replication configuration permissions" in the Active Roles Quick Start Guide).

Monitoring replication

Active Roles makes it possible to monitor the status of replication partners. Monitoring allows you to determine whether Active Roles replication is working efficiently and correctly. You can view the status of a replication partner via the Active Roles console:

1. Connect to any Administration Service within the replication group.
2. Open the **Properties** dialog box for the replication partner and go to the **Replication Status** tab.

To connect to the Administration Service, use the instructions provided earlier in this chapter (see Connecting to the Administration Service).

Once connected to the Administration Service, perform the following steps to open the **Properties** dialog box for a replication partner:

1. In the console tree, expand **Configuration | Server Configuration**, and select **Configuration Databases**.
2. In the details pane, right-click the replication partner, and click **Properties**.

The **Replication Status** tab in the **Properties** dialog box provides information about the last replication action of the partner and indicates whether the action completed successfully, failed, or is in progress.

If there are any replication failures in Active Roles, the Active Roles console provides a visual indication of this issue by modifying the icon of the **Server Configuration** and **Configuration Databases** containers in the console tree. This allows you to detect a replication failure without examining individual databases.

For more information on how to monitor the health of Active Roles replication, refer to the Active Roles Replication: Best Practices and Troubleshooting document.

Using AlwaysOn Availability Groups

You can improve the availability of the Active Roles Administration Service by using the AlwaysOn Availability Groups feature introduced in Microsoft SQL Server 2012. With the AlwaysOn Availability Groups feature, SQL Server provides a failover environment, known as an *availability group*, for a set of availability databases that fail over together from one SQL Server instance to another. You can add the Active Roles database to an availability group, and have the Administration Service automatically reconnect to the database when the availability group fails over to another SQL Server instance.

An availability group defines a set of availability replicas to host copies of each availability database. Each availability group has at least two availability replicas: the *primary replica* and a *secondary replica*. The primary replica hosts the read-write copy of each availability database held in the availability group; a secondary replica hosts a read-only copy of each availability database, and serves as a potential failover target for the availability group. During a failover, a secondary replica transitions to the primary role, becoming the new primary replica. The new primary replica brings its databases online as the primary databases for read-write access.

Adding the Active Roles database to an availability group helps ensure uninterrupted operation of the Active Roles Administration Service. If a server or software failure occurs on the SQL Server side, the availability group can instantly switch the database to a secondary replica, enabling the Administration Service to reconnect seamlessly to the database in the new location.

For detailed information about the AlwaysOn Availability Groups feature, see "AlwaysOn Availability Groups (SQL Server)" at http://go.microsoft.com/fwlink/p/?LinkId=245660.

Availability Group setup in Active Roles

Suppose you have the Active Roles Administration Service installed and configured as described in the Active Roles Quick Start Guide. This section provides instructions on how to configure the Administration Service to use the database that belongs to an availability group (an availability database).

Note that the Administration Service whose database belongs to an availability group cannot participate in Active Roles replication. Active Roles does not support the replication function for availability databases. If you attempt to perform the "Promote to Publisher" or "Add Subscriber" operation with the Administration Service connected to an availability database, you receive an error.

Here we assume that the Active Roles database is already added to an availability group on SQL Server. For instructions on how to configure an availability group, and how to add a database to an availability group, refer to Microsoft's documentation (see "Getting Started with AlwaysOn Availability Groups (SQL Server)" at http://technet.microsoft.com/en-us/library/gg509118.aspx). We also assume that Active Roles replication is not configured, neither for Configuration data nor for Management History data.

Under these conditions, you can configure the Administration Service to connect to the database via the availability group listener. By using the listener, the Administration Service can connect to the current primary replica of the availability group that holds the Active Roles database without knowing the name of the physical instance of SQL Server that hosts the primary replica. The listener enables support for failover redirection. In case of a failover, the listener automatically redirects the Administration Service's connection to the new primary replica.

Configuring the database connection to use the listener

To enable the use of the availability group listener, you need to modify the database connection setting of the Administration Service. You can specify the availability group listener in the **SQL Server** field on the Change Active Roles Database wizard pages provided by Active Roles Configuration Center.

Depending upon the location of the Management History database in your Active Roles environment, you need to specify the listener in the **SQL Server** field on the **Connection to Database** page, on the **Connection to Management History Database** page, or on both pages. The value in the **SQL Server** field must identify the DNS host name and, optionally, the TCP port of the listener of the availability group to which the database belongs. For example, if the DNS host name of the listener is `AGListener` and the TCP port used by this listener is `1234`, the value is `AGListener,1234`. You may omit the port number in case of the default port, `1433`.

Management History data stored in a separate database

Using Active Roles, it is mandatory to store the Management History data in a separate database. If you do this, then you have two databases, the Configuration database and the Management History database, each of which (or both) can belong an availability group. In this case:

- If the Configuration database belongs to an availability group, specify the listener of that availability group in the **SQL Server** field on the **Connection to Database** page in the Change Active Roles Database wizard. Otherwise, leave the value data intact.
- If the Management History database belongs to an availability group, specify the listener of that availability group in the **SQL Server** field on the **Connection to Management History Database** page in the Change Active Roles Database wizard. Otherwise, leave the value data intact.

To specify the listener

1. Start Configuration Center on the computer running the Administration Service, or connect Configuration Center to that computer.

   You can start Configuration Center by selecting **Active Roles Configuration Center** on the **Apps** page or **Start** menu, depending upon the version of your Windows operating system. For detailed instructions, see Running Configuration Center.

2. On the **Dashboard** page in the Configuration Settings main window, click **Manage Settings** in the **Administration Service** area.
3. On the **Administration Service** page that opens, click **Change** in the **Active Roles databases** area.
4. On the **Connection to Database** page in the Change Active Roles Database wizard that appears, do the following:
   a. If the Configuration database belongs to an availability group, then, in the **SQL Server** field, supply the DNS host name and, optionally, the TCP port of the listener of that availability group. Otherwise, don't change the value in the **SQL Server** field.
   b. If the Administration Service uses the SQL Server authentication option, type the password of the SQL login used for connection to the Configuration database.
   c. Click **Next**.
5. On the **Management History Database Options** page, select the **Existing Active Roles database** option (if not already selected), and then click **Next**.
6. On the **Connection to Management History Database** page, do the following:
   a. If the Management History database belongs to an availability group, then, in the **SQL Server** field, supply the DNS host name and, optionally, the TCP port of the listener of that availability group. Otherwise, don't change the value in the **SQL Server** field.

b. If the Administration Service uses the SQL Server authentication option, type the password of the SQL login used for connection to the Management History database.

c. Click **Next**.

7. Follow the instructions in the wizard to complete the configuration.

Using database mirroring

Active Roles can use the Microsoft SQL Server database mirroring technology to improve the availability of the Administration Service. Database mirroring provides a standby database server that supports failover. Once the current database server fails, the Administration Service can recover quickly by automatically reconnecting to the standby server.

Database mirroring increases database availability by supporting rapid failover. This technology can be used to maintain two copies of a single Active Roles database on different server instances of SQL Server Database Engine. One server instance serves the database to the Administration Service; this instance is referred to as the *principal server*. The other instance acts as a standby server; this instance is referred to as the *mirror server*.

Role switching

Within the context of database mirroring, the mirror server acts as the failover partner for the principal server. In the event of a disaster, the mirror server takes over the role of the principal server, bringing the mirror copy of the database online as the new principal database. The former principal server, if available, then assumes the role of the mirror server. This process, known as *role switching*, can take the form of:

- **Automatic failover** If the principal server becomes unavailable, quickly brings the mirror copy of the database online as the new principal database.
- **Manual failover** Allows the database owner to reverse the roles of the failover partners, if necessary.
- **Forced service** If the principal server becomes unavailable, allows the database owner to restore access to the database by forcing the mirror server to take over the role of the principal server.

In any role-switching scenario, as soon as the new principal database comes online, the Administration Service can recover by automatically reconnecting to the database.

For more information about the database mirroring technology, and instructions on how to set up and administer database mirroring on SQL Server, see the "Database Mirroring" topics in the SQL Server product documentation at http://msdn.microsoft.com/en-us/library/bb934127.aspx.

NOTE: The Active Roles replication function is not supported for the databases that have mirroring set up. If you attempt to perform the "Promote to Publisher" or "Add Subscriber" operation on such a database, you receive an error.

Database Mirroring setup in Active Roles

Here we assume that mirroring for the database of Active Roles is already set up on the SQL Server side in accord with the recommendations and instructions found in Microsoft's documentation, so that the following conditions are fulfilled:

- The Administration Service is connected to the Configuration database on the principal database server.
- Replication is not configured for the Configuration database (the database server acts as a stand-alone server as applied to Active Roles replication).
- The Administration Service is connected to the Management History database on the principal database server (by default, the Management History database is the same as the Configuration database).
- Replication is not configured for the Management History database (the database server acts as a stand-alone server as applied to Active Roles replication).

Under these conditions, the Administration Service can be instructed to automatically connect to the new principal database in the event of database server role switching. On the computer running the Administration Service, add a string value to each of these two registry keys, and then restart the Administration Service:

- **Key:** HKLM\SOFTWARE\One Identity\Active Roles\7.4\Service\DatabaseConnectionString\
  Value Name: Failover Partner
  Value Data: *<Identifies the SQL Server instance that currently owns the mirror server role for the Configuration database>*
- **Key:** HKLM\SOFTWARE\One Identity\Active Roles\7.4\Service\CHDatabaseConnectionString\
  Value Name: Failover Partner
  Value Data: *<Identifies the SQL Server instance that currently owns the mirror server role for the Management History database>*

If the default instance is used, the value data is the short name of the computer running SQL Server. Otherwise, the value data is the short name of the computer, followed by a backslash, followed by the name of the instance (such as *ComputerName\InstanceName*).

By default, the same database is used for the Configuration and Management History data; therefore, the value data would be the same in the DatabaseConnectionString and CHDatabaseConnectionString keys.

To restart the Administration Service, open Configuration Center and click the **Restart** button at the top of the **Administration Service** page in the Configuration Center main window. For instructions on how to run Configuration Center, see Running Configuration Center later in this document.

In the Active Roles console, you can view the mirroring status of the Configuration or Management History database that is used by a particular instance of the Administration Service:

1. In the console tree, select **Configuration | Server Configuration | Administration Services**.
2. In the details pane, double-click the name of the Administration Service whose database you want to examine.
3. In the Properties dialog box, click the **Configuration Database** or **Management History Database** tab, and observe the information in the **Database mirroring** area:
   - **Role** Current role of the database in the database mirroring session (Principal or Mirror).
   - **Partner** The instance name and computer name for the other partner in the database mirroring session.
   - **State** Current state of the mirrored database and of the database mirroring session. For more information about this field, see the "Mirroring States" topic at http://msdn.microsoft.com/en-us/library/ms189284.aspx

   If no information is displayed in the **Database Mirroring** area, then database mirroring is not configured.

You can also view the mirroring status of a Configuration database or a Management History database on the **General** tab in the **Properties** dialog box for the object representing that database in the **Configuration/Server Configuration/Configuration Databases** or **Configuration/Server Configuration/Management History Databases** container, respectively.

Best practices

This section provides instructions on how to monitor replication and perform administrative tasks to resolve replication-related problems. The following topics are covered:

- Viewing replication settings
- Monitoring replication
- Viewing database connection settings
- Modifying database connection settings
- Changing the service account
- Changing the SQL server logon account
- Modifying replication agent credentials
- Moving the publisher role
- Recovering replication if publisher is not available

Viewing replication settings

When configuring replication, Active Roles automatically sets replication parameters to the appropriate values. This ensures that replication is functioning properly. Normally, there is no need to modify the replication settings except for some error situations outlined the Troubleshooting Replication failures section later in this document.

The following table lists the values that Active Roles assigns to certain replication parameters.

Table 111: Values assigned to Replication parameters

Replication Parameter	Value
Publication name	AelitaReplica
Replication type	Merge
Subscription type	Push
Subscription expiration	Subscriptions expire and may be dropped if not synchronized in 60 days.
Schedule	The Merge Agents are running continuously at the Publisher. The Snapshot Agent starts daily at 12:00 a.m. at the Publisher.

You can use the following instructions to examine these settings using SQL Server Management Studio.

It is advisable not to change these settings. Replication may not be functioning correctly if you manually modify replication settings with the use of SQL Server tools.

Start Management Studio and connect to the Publisher SQL Server:

1. In Object Explorer, click **Connect**, and then click **Database Engine**.
2. Complete the **Connect to Server** dialog box to connect to the instance of the SQL Server Database Engine that holds the Publisher role.

Open the **Publication Properties** dialog box:

1. In Object Explorer, under the Publisher SQL Server, expand **Replication | Local Publications**.
2. In Object Explorer, under **Local Publications**, right-click **AelitaReplica**, and click **Properties**.

In the **Publication Properties** dialog box, you can review the Active Roles publication settings.

Open the **Subscription Properties** dialog box:

1. In Object Explorer, under **Local Publications**, expand **AelitaReplica**.
2. In Object Explorer, under **AelitaReplica**, right-click a Subscription, and click **Properties**.

In the **Subscription Properties** dialog box, you can review the Active Roles subscription settings.

Replication Agent schedule

By default, Active Roles schedules the Replication Agents to run as follows:

- The Snapshot Agent starts every day at 12:00 a.m. at the Publisher.
- The Merge Agents start automatically when SQL Server Agent starts, and runs continuously at the Publisher.

To verify the Snapshot Agent schedule

1. Open SQL Server Management Studio.
2. In Object Explorer, connect to the instance of the SQL Server Database Engine that holds the Publisher role, and then expand that instance.
3. Right-click the **Replication** folder, and click **Launch Replication Monitor**.
4. In the left pane of the **Replication Monitor** window, expand your Publisher SQL Server, and click **AelitaReplica**.
5. In the right pane of the **Replication Monitor** window, on the **Warnings and Agents** tab, right-click the Snapshot Agent in the **Agents and jobs related to this publication** list, and click **Properties**.
6. In the left pane of the **Job Properties** window, click **Schedules**.
7. Review the replication agent schedule settings in the right pane of the **Job Properties** window.
8. Click the **Edit** button to examine the replication agent schedule settings in detail.

To verify the Merge Agent schedule

1. Open SQL Server Management Studio.
2. In Object Explorer, connect to the instance of the SQL Server Database Engine that holds the Publisher role, and then expand that instance.
3. Right-click the **Replication** folder, and click **Launch Replication Monitor**.
4. In the left pane of the **Replication Monitor** window, expand your Publisher SQL Server, and click **AelitaReplica**.
5. In the right pane of the **Replication Monitor** window, on the **All Subscriptions** tab, right-click the subscription whose Merge Agent you want to examine, and click **View Details**.
6. In the **Subscription** window, on the **Action** menu, click **Merge Agent Job Properties**.

7. In the left pane of the **Job Properties** window, click **Schedules**.
8. Review the replication agent schedule settings in the right pane of the **Job Properties** window.
9. Click the **Edit** button to examine the replication agent schedule settings in detail.

Monitoring replication

In order to identify replication-related problems, you can use the Active Roles console connected to the Publisher Administration Service. If there are any replication failures, a red triangle is displayed on the **Server Configuration** and **Configuration Databases** containers in the console tree. In the details pane, the same icon is used to highlight the database affected by a replication failure.

If you have encountered a replication failure, you should ensure that the SQL Server Agent service is started on the computer running the Publisher SQL Server, and then use SQL Server Management Studio to get more information on that failure:

1. In Object Explorer, connect to the instance of the SQL Server Database Engine that holds the Publisher role, and then expand that instance.
2. Right-click the **Replication** folder, and click **Launch Replication Monitor**.
3. In the left pane of the **Replication Monitor** window, expand your Publisher SQL Server, and click **AelitaReplica**.
4. In the right pane of the **Replication Monitor** window, on the **Warnings and Agents** tab, look for a red icon under **Agents and jobs related to this publication**. This icon indicates a Snapshot Agent error:
5. Right-click the agent that has encountered an error and then click **View Details**.
6. In the **Snapshot Agent** window, view the error description under **Error details or message of the selected session**.
7. In the right pane of the **Replication Monitor** window, on the **All Subscriptions** tab, look for a red icon in the list of subscriptions. This icon indicates a Merge Agent error:
8. On the **All Subscriptions** tab, right-click the subscription that has encountered an error and then click **View Details**.
9. In the **Subscription** window, view the error description under **Last message of the selected session**.

Some typical errors are discussed later in this document (see the Troubleshooting Replication failures section). The Troubleshooting Replication failures section also provides information on how to resolve such errors.

Viewing database connection settings

The most common reasons for replication problems are access failures that Replication Agents encounter when attempting to connect to the Publisher or Subscriber SQL Server. Given that security credentials of Replication Agents depend on authentication mode of the Administration Service, you may need to examine Administration Service database connection settings in order to see which mode is actually used—Windows authentication or SQL Server authentication.

You can view connection settings in the Active Roles console:

1. In the console tree, select **Configuration | Server Configuration | Administration Services**.
2. In the details pane, right-click the Administration Service you want to examine, and click **Properties**.
3. In the **Properties** dialog box, go to the **Configuration Database** tab.

The **Configuration Database** tab displays the following information:

- **SQL Server** Identifies the SQL Server instance used by the Administration Service.
- **Database** The name of the Administration Service database.
- **Use Windows authentication** When selected, indicates that the Administration Service uses Windows authentication mode when connecting to SQL Server.
- **Use SQL Server authentication** When selected, indicates that the Administration Service uses SQL Server authentication mode when connecting to SQL Server.
- **Login name** The name of the SQL Server login that the Administration Service uses to access SQL Server; only applies to the **Use SQL Server authentication** option.

Modifying database connection settings

You may need to modify Administration Service database connection settings if the login of the Administration Service for SQL Server authentication is no longer valid, or has the password changed. If you change the login, you also need to change it for Replication Agents, as described in the Modifying Replication Agent credentials section later in this document.

You can modify connection settings by using Active Roles Configuration Center:

1. Start Configuration Center on the computer running the Administration Service, or connect Configuration Center to that computer.

   You can start Configuration Center by selecting **Active Roles 7.5 Configuration Center** on the **Apps** page or **Start** menu, depending upon the version of your Windows operating system. For detailed instructions, see "Running Configuration Center" in the Active Roles Administrator Guide.

2. On the **Dashboard** page in the Configuration Center main window, click **Manage Settings** in the **Administration Service** area.
3. On the **Administration Service** page that opens, click **Change** in the **Active Roles database** area.
4. Use the Change Active Roles Database wizard that appears to view or change the login or password of the Administration Service for SQL Server authentication: Type the appropriate login name and password in the fields under the **SQL Server authentication** option on the **Connection to Database** page.

Changing the service account

With the Windows authentication option selected for database connection, the Administration Service uses its service account to authenticate with SQL Server. Additionally, if the Administration Service's database server holds the Publisher role, and has a Subscriber with Windows authentication, the service account requires the appropriate permissions on the Subscriber SQL Server. For details, see the "SQL Server permissions" section in the Active Roles Quick Start Guide.

Given this role of the service account, you may need to specify a different service account with sufficient SQL Server permissions. Also, you may need to change the service account's password. You can view or change the service account by using Active Roles Configuration Center as follows.

1. Start Configuration Center on the computer running the Administration Service, or connect Configuration Center to that computer.

   You can start Configuration Center by selecting **Active Roles 7.5 Configuration Center** on the **Apps** page or **Start** menu, depending upon the version of your Windows operating system. For detailed instructions, see "Running Configuration Center" in the Active Roles Administrator Guide.
2. On the **Dashboard** page in the Configuration Center main window, click **Manage Settings** in the **Administration Service** area.
3. On the **Administration Service** page that opens, click **Change** in the **Service account** area.
4. On the **Change Service Account** page that appears, type the logon name and password of the service account, and then click **Change**.

Changing the SQL Server Agent logon account

If the Publisher has a Subscriber that uses Windows authentication, it is required that the SQL Server Agent logon account on the Publisher SQL Server have appropriate access

permissions on the Subscriber SQL Server. For details, see the "SQL Server permissions" section in the Active Roles Quick Start Guide.

Given these requirements of the SQL Server Agent logon account, you may encounter a situation where you need to specify a different logon account with sufficient access permissions. You may also need to change password for the logon account. This section provides instructions on how to change the SQL Server Agent logon account.

You can specify the name and password of the SQL Server Agent logon account by using SQL Server Configuration Manager:

1. On the computer running the Publisher SQL Server, open SQL Server Configuration Manager.
2. In the console tree, select **SQL Server Services**.
3. In the details pane, right-click the SQL Server Agent to modify, and then click **Properties**.
4. On the **Log On** tab, click **This account**, and specify the account name and password.
5. Click **OK**.
6. For the changes to take effect, click **Yes** in the confirmation message box.

Modifying Replication Agent credentials

This section provides information on how to repair Active Roles replication if it fails due to insufficient permissions of Replication Agents. The credentials used by Replication Agents to access a given SQL Server depend on authentication mode of the Administration Service connection to that SQL Server:

- **Windows authentication** In this mode, Replication Agents use the credentials of the SQL Server Agent service running on the Publisher SQL Server computer
- **SQL Server authentication** In this mode, Replication Agents use the credentials of the SQL Server login specified for the Administration Service connection to SQL Server

The following sub-sections elaborate on each of these two options.

Windows authentication

If the Administration Service uses Windows authentication, Replication Agents connect to SQL Server in the security context of the SQL Server Agent service. Therefore, the SQL Server Agent logon account must have sufficient permissions for replication to work properly (see the "SQL Server permissions" section in the Active Roles Quick Start Guide.

If the SQL Server Agent logon account does not have the appropriate permissions, is deleted, or has the password changed, Active Roles replication fails. To resolve this problem, give the required permissions to the logon account, or configure the SQL Server

Agent service to log on with a different account that has the appropriate permissions. For instructions on how to configure the SQL Server Agent service to log on with a given account, see Changing the SQL Server Agent logon account earlier in this document.

You can use the following instructions to verify that the Replication Agents are configured properly. The instructions vary depending on whether the SQL Server holds the Publisher or Subscriber role. In both cases, you should connect to the Publisher SQL Server using SQL Server Management Studio.

Replication Agent connection to Publisher

If the Administration Service connects to the Publisher SQL Server using Windows authentication, follow these steps to verify that the Replication Agents are configured properly:

1. With SQL Server Management Studio, connect to the Publisher SQL Server.
2. In the Object Browser, under the Publisher SQL Server, right-click the **Replication** folder, and then click **Distributor Properties**.
3. In the left pane of the **Distributor Properties** window, click **Publishers**.
4. In the **Publishers** list, select the entry representing the Publisher SQL Server, and click the button in that entry to display the **Publisher Properties** dialog box.
5. In the **Publisher Properties** dialog box, under **Agent Connection to the Publisher**, verify that the Agent Connection Mode property is set to **Impersonate the agent process account**.

Replication Agent connection to Subscriber

If the Administration Service connects to the Subscriber SQL Server using Windows authentication, follow these steps to verify that the Replication Agents are configured properly:

1. With SQL Server Management Studio, connect to the Publisher SQL Server.

   NOTE: You must have Management Studio connected to the Publisher SQL Server, regardless of whether you are managing Replication Agents for the Publisher or for a Subscriber.

2. In the Object Browser, under the Publisher SQL Server, expand **Replication | Local Publications | AelitaReplica**.
3. In the list under **AelitaReplica**, right-click the entry corresponding to the Subscriber SQL Server and click **Properties**.
4. In the **Subscription Properties** window, in the **Security** section, expand the **Subscriber connection** entry.
5. Verify that the **Subscriber connection** property is set to **Impersonate agent process account (Windows Authentication)**.

SQL Server authentication

If the Administration Service uses SQL Server authentication, the Replication Agents connect to SQL Server in the security context of the SQL Server login specified for the Administration Server connection to SQL Server.

If the login does not have sufficient rights, is deleted, or has the password changed, Active Roles replication fails. To resolve this problem, do the following:

1. Choose a SQL Server login with sufficient rights (see the "SQL Server permissions" section in the Active Roles Quick Start Guide).
2. Configure the Administration Service to use that login (see Viewing database connection settings earlier in this document).
3. Configure the Replication Agents to use that login.

The following sections elaborate on how to configure the Replication Agents to use a given SQL Server login. The instructions vary depending on whether SQL Server in question is the Publisher or a Subscriber.

Replication Agent connection to Publisher

If you have changed the SQL Server login for the Administration Service connection to the Publisher, use the following steps to configure the Replication Agents with that login:

1. With SQL Server Management Studio, connect to the Publisher SQL Server.
2. In the Object Browser, under the Publisher SQL Server, right-click the **Replication** folder, and then click **Distributor Properties**.
3. In the left pane of the **Distributor Properties** window, click **Publishers**.
4. In the **Publishers** list, select the entry representing the Publisher SQL Server, and click the button in that entry to display the **Publisher Properties** dialog box.
5. In the **Agent Connection to the Publisher** area, click **Login**, and type the login name.
6. Click **Password**, and then click the button in the **Password** entry.
7. In the **Enter Password** dialog box, type and confirm by retyping the password of that login.
8. Click **OK** to close the **Enter Password** dialog box.
9. Click **OK** to close the **Publisher Properties** dialog box.

Replication Agent connection to Subscriber

If you have changed the SQL Server login for the Administration Service connection to a Subscriber, use the following steps to configure the Replication Agents with that login:

1. With SQL Server Management Studio, connect to the Publisher SQL Server.

   NOTE: You must have Management Studio connected to the Publisher SQL Server, regardless of whether you are managing Replication Agents for the Publisher or for a Subscriber.

2. In the Object Browser, under the Publisher SQL Server, expand **Replication | Local Publications | AelitaReplica**.
3. In the list under **AelitaReplica**, right-click the entry corresponding to the Subscriber SQL Server and click **Properties**.
4. In the **Subscription Properties** window, in the **Security** section, expand the **Subscriber connection** entry.
5. Click the button in the **Subscriber Connection** entry.

   This displays the **Enter Connection Information** dialog box.

6. In the **Login** box, type the login name.
7. In the **Password** and **Confirm password** boxes, type and confirm by retyping the password of that login.
8. Click **OK** to close the **Enter Connection Information** dialog box.
9. Click **OK** to close the **Subscription Properties** dialog box.

Moving the Publisher role

In the Active Roles replication model, a replication group includes the Publisher and may include several Subscribers. The Publisher plays a special role in the replication group: it synchronizes data changes between Subscribers. In some scenarios, you may want to move the Publisher role to another SQL Server.

For example, you might need to move the Publisher role to a different SQL Server if the service level becomes insufficient. Given that the Publisher receives and synchronizes data changes from all Subscribers, the volume of requests being serviced by the Publisher increases as the number of Subscribers grows. Respectively increases the workload for SQL Server that holds the Publisher role so its performance can suffer. To resolve this problem, you can transfer the Publisher role to another, more powerful server.

This section provides instructions on how to reconfigure the existing replication group so that the Publisher role is assigned to SQL Server other than the current Publisher. You can perform this task using the Active Roles console connected to the Administration Service whose database server currently holds the Publisher role (Publisher Administration Service).

NOTE: The Publisher Administration Service must be up and running. If the Publisher is unavailable, you can move the Publisher role using the instructions outlined in the next section of this document.

Open the Active Roles console and connect to the Publisher Administration Service:

1. Look for the **Active Roles Console** application, and then click to start that application.
2. Right-click the console tree root, click **Connect**, and then select the Administration Service whose database server currently holds the Publisher role.

Use the Active Roles console to remove all Subscribers and to demote the Publisher as follows:

1. In the console tree, expand **Configuration | Server Configuration**, and select **Configuration Databases**.
2. In the details pane, right-click a Subscriber, and click **Delete**.
3. In the confirmation message box, click **Yes**.
4. Repeat Steps 2–3 for each Subscriber.
5. In the details pane, right-click the Publisher, and click **Demote**.
6. In the confirmation message box, click **Yes**.
7. Wait while Active Roles demotes the Publisher.

After these steps, you can promote the appropriate SQL Server to Publisher and designate the other SQL Servers as Subscribers to the new Publisher, thus configuring the new replication group.

TIP: After you add a Subscriber, the configuration data stored on the Publisher is replicated to the Subscriber, overriding the data on that Subscriber. Therefore, in order to retain your existing Active Roles configuration, it is advisable to assign the Publisher role to SQL Server that belonged to the old replication group. This ensures that each Administration Service in the new replication group inherits the configuration that was in place when you removed the Subscribers and demoted the Publisher.

To configure the new replication group, perform the following steps using the Active Roles console:

1. Right-click the console tree root, click **Connect**, and then select the Administration Service whose SQL Server you want to hold the Publisher role.
2. In the console tree, expand **Configuration | Server Configuration**, and select the **Configuration Databases** container.
3. In the details pane, right-click the database and click **Promote**.
4. In the confirmation message box, click **Yes**.
5. Wait while Active Roles performs the operation.
6. In the details pane, right-click the Publisher, and click **Add Replication Partner**.
7. On the **Welcome** page in the New Replication Partner wizard, click **Next**.
8. On the **Database Selection** page, click **Browse**.
9. In the **Connect to Administration Service** dialog box, select the Administration Service whose SQL Server is to be configured as a Subscriber to this Publisher. Click **OK**.

10. In the New Replication Partner wizard, click **Next**, click **Next**, and then click **Finish**.
11. Repeat Steps 6–10 for each SQL Server you want to make a Subscriber.

Recovering replication if Publisher is not available

Once the Publisher becomes unavailable, Subscribers cannot synchronize configuration data. The only way that replication can be recovered is by restoring the current Publisher or making another SQL Server the Publisher.

If the current Publisher cannot be restored, you need to transfer the Publisher role to SQL Server that holds the Subscriber role, and reconfigure the other Subscribers to use the new Publisher. This requires that you first remove all Subscribers from the replication group.

Given that the Publisher is unavailable, you can remove a Subscriber from the replication group by using the Active Roles console as follows:

1. Right-click the console tree root, click **Connect**, and then select the Administration Service that uses the Subscriber SQL Server.
2. In the console tree, expand **Configuration | Server Configuration**, and select **Configuration Databases**.
3. In the details pane, right-click the Subscriber and select **All Tasks | Advanced Properties**.
4. In the **Advanced Properties** window, select both the **Show all possible attributes** and **Include attributes with empty values** check boxes.
5. In the list of attributes, double-click the attribute **edsvaReplicationForceStandalone**.
6. In the **Edit Attribute** window, type **TRUE** in the **Value** box. Click **OK**.
7. In the **Advanced Properties** window, click **OK**.

Once you have removed all Subscribers from the replication group, you can promote one of the former Subscribers to Publisher and add Subscribers to the new Publisher by using the Active Roles console as follows:

1. Right-click the console tree root, click **Connect**, and then select the Administration Service whose SQL Server you want to hold the Publisher role.

   TIP: After you add a Subscriber, the configuration data stored on the Publisher will be replicated to the Subscriber, overriding the data on that Subscriber. Therefore, in order to retain your existing Active Roles configuration, assign the Publisher role to one of the former Subscribers. This ensures that each Administration Service in the new replication group inherits the configuration that was in place when you removed the Subscribers from the replication group.

2. In the console tree, expand **Configuration | Server Configuration**, and select **Configuration Databases**.
3. In the details pane, right-click the database and click **Promote**.
4. In the confirmation message box, click **Yes**.
5. Wait while Active Roles performs the operation.
6. In the details pane, right-click the Publisher, and click **Add Replication Partner**.
7. On the **Welcome** page in the New Replication Partner wizard, click **Next**.
8. On the **Database Selection** page, click **Browse**.
9. Use the **Connect to Administration Service** dialog box to specify the Administration Service whose SQL Server is to be configured as a Subscriber to this Publisher. Click **OK**.
10. In the New Replication Partner wizard, click **Next**, click **Next**, and then click **Finish**.
11. Repeat Steps 6–10 for each SQL Server you want to make a Subscriber.

Troubleshooting Replication failures

If there are any replication failures in Active Roles, the Active Roles console provides a visual indication of this issue by placing a red triangle on the **Server Configuration** and **Configuration Databases** containers in the console tree. To get more information on a replication failure, you can use SQL Server Management Studio (see Monitoring replication earlier in this document).

The following sections discuss specific actions to take if you encounter a replication problem in Active Roles.

Replication Agent malfunction

Symptoms

Replication stops synchronizing changes to configuration data, that is, changes made on a replication partner are not propagated to other replication partners. Replication Monitor in SQL Server Enterprise Manager or SQL Server Management Studio does not indicate any error.

Solution

Verify that the SQL Server Agent service is started on the Publisher SQL Server:

1. With SQL Server Management Studio, connect to the Publisher SQL Server.
2. In the console tree, right-click **SQL Server Agent**, and then click **Start**.

   If the **Start** command is unavailable, the SQL Server Agent service is already started.

Ensure that the Merge Agents are started on the Publisher SQL Server:

1. With SQL Server Management Studio, connect to the Publisher SQL Server.
2. In the console tree, right-click **Replication**, and click **Launch Replication Monitor**.
3. In Replication Monitor, in the left pane, browse the **My Publishers** branch to select the **AelitaReplica** publication.
4. In Replication Monitor, in the right pane, right-click a subscription and click **Start Synchronizing**. Perform this step for each subscription of the **AelitaReplica** publication.

   If the **Start Synchronizing** command is unavailable, the agent is already started.

Verify that the replication agent schedule is correct. The Merge Agents must be configured to run continuously at the Publisher. The Snapshot Agent must be configured to start daily at 12:00 a.m. at the Publisher. For details, see Replication Agent schedule earlier in this document.

Replication Agent authentication problems

Symptoms

Replication fails with one of the following errors on the Snapshot Agent or Merge Agent (see Monitoring replication earlier in this document):

- The process could not connect to Publisher '*<Server_name>*'. Login failed for user '*<User_name>*'.
- The process could not connect to Subscriber '*<Server_name>*'. Login failed for user '*<User_name>*'.

Solution

By using SQL Server Enterprise Manager or SQL Server Management Studio, verify that the Replication Agent credentials are set properly. The following conditions must be met:

Table 112: Conditions for Replication Agent credentials

Server role	Authentication mode	Replication Agent credentials
Publisher	Windows Authentication	Impersonate the SQL Server Agent account on the computer running the Publisher SQL Server (trusted connection)
	SQL Server Authentication	SQL Server login and password that the Publisher Administration Service uses to connect to its SQL Server
Subscriber	Windows Authentication	Impersonate the SQL Server Agent account on the computer running the Publisher SQL Server (trusted connection)
	SQL Server Authentication	SQL Server login and password that the Subscriber Administration Service uses to connect to its SQL Server

For information on how to view or modify the credentials that the Snapshot Agent and Merge Agents use to connect to the Publisher and Subscribers, see Modifying Replication Agent credentials earlier in this document.

SQL Server identification problems

Symptoms

When promoting SQL Server to Publisher, or adding it as a Subscriber to the existing Publisher, the operation fails with the following error: "An alias cannot be used for replication. Use the name of the SQL Server instance."

Solution

This error may be due to one of the following reasons:

- Incorrect server name. The computer running SQL Server is renamed, or SQL Server has lost its name.

- Administration Service identifies SQL Server by alias. An alias was used to specify SQL Server when installing the Administration Service.

Incorrect server name

To isolate and resolve this problem, run the following two queries on the SQL Server instance affected by this issue. Copy these queries "as is," without making any substitutions for the **servername** parameter:

```
select @@servername
```

```
select serverproperty('servername')
```

If **select @@servername** returns a non-null value that is different from the value returned by the second query, execute the following SQL script:

```
exec sp_dropserver 'oldname', 'droplogins'
```

```
exec sp_addserver 'newname', 'local'
```

In this script, replace:

- *oldname* with the value returned by **select @@servername**
- *newname* with the value returned by **select serverproperty('servername')**

If **select @@servername** returns NULL, execute the following SQL script:

```
exec sp_addserver 'newname', 'local'
```

In this script, replace *newname* with the value returned by **select serverproperty ('servername')**.

For these changes to take effect, you must restart SQL Server. You can restart SQL Server by using SQL Server Configuration Manager:

1. In the console tree, select **SQL Server Services**.
2. In the details pane, right-click the SQL Server instance to restart, and then click **Restart**.

Administration Service identifies SQL Server by alias

The Administration Service must be configured to identify SQL Server by computer name, rather than using a client alias. Otherwise, when attempting to make SQL Server the Publisher or a Subscriber, you encounter the error "An alias cannot be used for replication. Use the name of the SQL Server instance."

To avoid this problem, you may need to reinstall the Administration Service. When installing the Administration Service, use the following syntax to identify SQL Server:

- *computername* — for the default instance
- *computername\instancename* — for a named instance

In this syntax:

- *computername* is the (short) NetBIOS name of the computer running SQL Server;
- *instancename* is the name of a SQL Server named instance.

Appendix A: Using regular expressions

When configuring search filter conditions or property validation criteria, you may need to use regular expressions. This section helps you learn about the syntax you must use in regular expressions.

A regular expression is a pattern of text that consists of ordinary characters (for example, letters a to z) and special characters, known as metacharacters. It serves as a template for matching a character pattern to the string value being validated.

The following table contains a list of metacharacters and their behavior in the context of regular expressions that can be used to create search filter conditions and property validation criteria in Active Roles. To match an exact metacharacter, precede the character with a backslash (\).

Table 113: Metacharacters in the context of regular expressions

Character	Definition
****	Marks the next character as a special character, a literal, or an octal escape. For example, n matches the character n; \n matches a new line character. The sequence \\ matches \ and \(matches (.
^	Matches the position at the beginning of the input string.
$	Matches the position at the end of the input string.
*****	Matches the preceding sub-expression zero or more times. For example, zo* matches z and zoo. * is equivalent to {0,}.
+	Matches the preceding sub-expression one or more times. For example, zo+ matches zo and zoo, but not z. + is equivalent to {1,}.
?	Matches the preceding sub-expression zero or one time. For example, do(es)? matches the do in do and does. ? is equivalent to {0,1}.
{n}	n is a nonnegative integer. Matches the preceding sub-expression exactly n times. For example, o{2} does not match the o in Bob, but matches the two o's in food.

Character	Definition
{n,}	n is a nonnegative integer. Matches the preceding sub-expression at least n times. For example, o{2,} does not match the o in Bob, but matches all the o's in foooood. o{1,} is equivalent to o+. o{0,} is equivalent to o*.
{n,m}	m and n are nonnegative integers, where n <= m. Matches the preceding sub-expression at least n and at most m times. For example, o{1,3} matches the first three o's in fooooood. o{0,1} is equivalent to o?. Note that there cannot be spaces between the comma and the numbers.
?	When this character immediately follows any of the other quantifiers (*, +, ?, {n}, {n,}, {n,m}), the matching pattern is non-greedy. A non-greedy pattern matches as little of the searched string as possible, whereas the default greedy pattern matches as much of the searched string as possible. For example, in the string oooo, o+? matches a single o, while o+ matches all o's.
.	Matches any single character except \n. To match any character including the \n, use a pattern such as [.\n].
( )	Groups one or more regular expressions to establish a logical regular expression consisting of sub-expressions. Used to override the standard precedence of certain operators. To match parentheses characters ( ), use \(or \).
x|y	Matches either x or y. For example, z|food matches z or food. (z|f)ood matches zood or food.
[xyz]	A character set. Matches any one of the enclosed characters. For example, [abc] matches the a in plain.
[^xyz]	A negative character set. Matches any character not enclosed. For example, [^abc] matches the p in plain.
[a-z]	A range of characters. Matches any character in the specified range. For example, [a-z] matches any lowercase alphabetical character in the range a to z.
[^a-z]	A negative range of characters. Matches any character not in the specified range. For example, [^a-z] matches any character not in the range a to z.
\b	Matches a word boundary, that is, the position between a word and a space. For example, er\b matches the er in never but not the er in verb.
\B	Matches a non-word boundary. For example, er\B matches the er in verb but not the er in never.
\cx	Matches the control character indicated by x. For example, \cM matches a Control-M or carriage return character. The value of x must be in the

Character	Definition
	range of A-Z or a-z. If not, c is assumed to be a literal c character.
\d	Matches a digit character. Equivalent to [0-9].
\D	Matches a non-digit character. Equivalent to [^0-9].
\s	Matches any white space character including space, tab, form-feed, etc. Equivalent to [\f\n\r\t\v].
\S	Matches any non-white space character. Equivalent to [^ \f\n\r\t\v].
\w	Matches any word character including underscore. Equivalent to [A-Za-z0-9_].
\W	Matches any non-word character. Equivalent to [^A-Za-z0-9_].
\xn	Matches n, where n is a hexadecimal escape value. Hexadecimal escape values must be exactly two digits long. For example, \x41 matches A. Allows ASCII codes to be used in regular expressions.

Examples of regular expressions

The following table includes some examples of regular expressions and matches.

Table 114: Examples of regular expressions

Expression	Matches	Does not match
st.n	Austin and Boston	Webster
st[io]n	Austin and Boston	Stanton
st[^io]n	Stanton	Boston or Austin
^boston	Boston	South Boston or North Boston Harbor
ston$	Boston and Galveston	Stonewall
sea|side	Seattle and Seaside and Oceanside	Seoul or Sidney
dal(l|h)art	Dalhart	Dallas or Lockhart
il?e$	Etoile and Wylie	Beeville
il*e$	Etoile and Wylie and Beeville	Bellaire
il+e$	Etoile and Beeville	Wylie

Expression	Matches	Does not match
ad{2}	Addison and Caddo	Adkins
(la.*){2,}	Highland Village and Lake Dallas	Laredo

Order of precedence

Once you have constructed a regular expression, it is evaluated much like an arithmetic expression. It is evaluated from left to right and follows an order of precedence.

The following table shows the order of precedence for the various regular expression operators, starting with the highest:

Table 115: Order of precedence

Character	Description
****	Escape
(), **[]**	Parentheses and Brackets
*****, **+**, **?**, **{n}**, **{n,}**, **{n,m}**	Quantifiers
^, **$**, **\anymetacharacter**	Anchors and Sequences
|	Alteration

Appendix B: Administrative Template

The Active Roles Administrative Template allows you to control the behavior and appearance of the Active Roles console by using Group Policy (see Active Roles snap-in settings).

This Administrative Template also provides a number of policy settings allowing you to limit the list of Active Roles' Administration Service instances for auto-connect (see Administration Service auto-connect settings later in this document).

Active Roles snap-in settings

With the Active Roles Snap-in policy settings you can:

- Cause the console to hide some portions of the user interface.
- Specify default settings for some user interface elements.
- Specify settings to register extension snap-ins with the Active Roles console.

The Administrative Template provides the following policy settings to control the behavior and appearance of the Active Roles console:

Table 116: Policy settings to control the behavior and appearance of

Policy Setting	Explanation
Hide Exchange management	Removes all user interface elements (commands, wizards, and dialog boxes) intended to manage Exchange recipients. If you enable this policy, users cannot perform any Exchange tasks and manage any Exchange recipient settings with the Active Roles console. If you disable this policy or do not configure it, users with appropriate permissions can use the Active Roles console to perform Exchange tasks and manage Exchange recipient settings.
Set default view	Specifies view mode in which the Active Roles console will start. If

Policy Setting	Explanation
mode	**View** menu. If you want to enforce view mode, select the **User is not allowed to change view mode** policy option. This option ensures that the console user cannot change the view mode you have selected.
Hide Configuration node	Removes the **Configuration** node from the console tree when the Active Roles console is in Advanced view mode. If you enable this policy, in Advanced view mode, all objects and containers related to the Active Roles configuration are not displayed. The **Managed Units** node and its contents are displayed as well as all advanced Active Directory objects and containers.
Disable 'Remember password' option	Clears and disables the **Remember password** check box in the **Connect to Administration Service** dialog box. If you enable this policy, the **Connect as: The following user** option in the Active Roles console requires that the user enter his password every time when using that option, rather than encrypting and storing the password once it has been entered. Note that saving passwords may introduce a potential security risk.
Disable 'Connect as' options	Disables the **Connect as** options in the **Connect to Administration Service** dialog box, including the **Remember password** check box. If you enable this policy, the console users are only allowed to connect to the Administration Service under their logon accounts. With this policy, the **Current user** option is selected under **Connect as**, and cannot be changed.
Set controlled objects to be marked by default	Specifies whether to use a special icon for visual indication of the objects to which Access Templates or Policy Objects are applied (linked). If you enable this policy, you can choose the category of object to be marked with a special icon by default. Users can modify this setting using the **Mark Controlled Objects** command on the **View** menu.

In addition, the Administrative Template provides for policies allowing you to register extension snap-ins with the Active Roles console. These policies are located in the folder named **Extension Snap-ins**. Each policy in that folder is used to register one of the following:

Table 117: Policies allowing to register extension snap-ins with Active Roles Console

Policy Setting	Explanation
Namespace extensions	Allows you to register extension snap-ins to extend the namespace of the Active Roles console.
Context menu extensions	Allows you to register extension snap-ins to extend a context menu in the Active Roles console.

Policy Setting	Explanation
Toolbar extensions	Allows you to register extension snap-ins to extend the toolbar of the Active Roles console.
Property sheet extensions	Allows you to register extension snap-ins to extend property sheets in the Active Roles console.
Task pad extensions	Allows you to register extension snap-ins to extend a task pad in the Active Roles console.
View extensions	Allows you to register extension snap-ins to add user interface elements to an existing view or to create new views in the Active Roles console.

When configuring a policy from the **Extension Snap-ins** folder, you are prompted to specify the name and the value of the item to be added.

The name parameter determines the type of the node you want to extend. Each type is identified with a GUID. For example, if you want to extend user objects, the GUID is {D842D417-3A24-48e8-A97B-9A0C7B02FB17}. For information on other node types, refer to the Active Roles SDK.

The value parameter determines the extension snap-ins to be added. Each snap-in is identified with a GUID. You add multiple snap-ins by entering their GUIDs separated by semicolons. For example, value might look as follows:

```
{AD0269D8-27B9-4892-B027-9B01C8A011A1}"Description";{71B71FD3-0C9B-473a-B77B-12FD456FFFCB}"Description"
```

The entry `"Description"` is optional and may contain any text describing the extension snap-in, enclosed in double quotation marks.

Administration Service auto-connect settings

The Administrative Template provides the following settings that allow you to limit the list of Active Roles' Administration Service instances for auto-connect:

- 'Allowed Servers for Auto-connect' setting
- 'Disallowed Servers for Auto-connect' setting
- 'Additional Servers for Auto-connect' setting

When applied to a computer running an Active Roles client application, such as the Active Roles console, Web Interface or ADSI Provider, these settings make it possible to restrict auto-connection of the client application to a pre-defined set of computers running the Administration Service, with inclusions or exclusions of certain computers from the pool of the Administration Service instances to auto-connect.

You can enable all these settings or only some of these settings. For example, if you only want to allow the client application to auto-connect to specific instances of the Administration Service (and only to those instances), then you could only enable and configure the **Allowed Servers for Auto-connect** setting. If you only want to prevent the client application from auto-connecting to particular instances of the Administration Service, you could only enable and configure the **Disallowed Servers for Auto-connect** setting. If you want the client application to auto-connect to a server identified by a computer alias, enable the **Additional Servers for Auto-connect** setting and add the computer alias to that setting.

The following rules apply when two or more settings are enabled. If the name of a given computer is listed in both the **Allowed Servers for Auto-connect** and **Disallowed Servers for Auto-connect** settings, then the client application is allowed to auto-connect to the Administration Service on that computer. If the name or alias of a particular computer is listed in the **Additional Servers for Auto-connect** setting, then the client application auto-connects to the Administration Service on that computer regardless of the **Allowed Servers for Auto-connect** and **Disallowed Servers for Auto-connect** settings.

'Allowed Servers for Auto-connect' setting

When applied to a computer running an Active Roles client application, such as the Active Roles console, Web Interface or ADSI Provider, this setting determines the instances of the Active Roles Administration Service to which the client application is allowed to auto-connect. This setting only affects the Administration Service instances that are published by Active Roles for auto-discovery. To have the client application connect to the Administration Service on a computer whose name or alias is not published for Administration Service auto-discovery, use the **Additional Servers for Auto-connect** setting.

If you enable this setting, you can specify a list of computer names identifying the computers running the Administration Service to which the client application is allowed to auto-connect. In a computer name, you may use an asterisk wildcard character (*) to represent any string of characters. If a given computer is listed in this setting, then the client application is allowed to auto-connect to the Administration Service on that computer. If a given computer is not listed in this setting, then the client application is not allowed to auto-connect to the Administration Service on that computer unless the name or alias of that computer is listed in the **Additional Servers for Auto-connect** setting.

If this setting is disabled or not configured, the client application normally auto-connects to any available Administration Service that is published by Active Roles for auto-discovery. However, you can use the **Disallowed Servers for Auto-connect** setting to prevent the client application from auto-connecting to certain published instances of the Administration Service.

'Disallowed Servers for Auto-connect' setting

When applied to a computer running an Active Roles client application, such as the Active Roles console, Web Interface or ADSI Provider, this setting determines the instances of the Active Roles Administration Service to which the client application is not allowed to auto-connect. This setting only affects the Administration Service instances that are published by Active Roles for auto-discovery.

If you enable this setting, you can specify a list of computer names identifying the computers running the Administration Service to which the client application is not allowed to auto-connect. In a computer name, you may use an asterisk wildcard character (*) to represent any string of characters. If a given computer is listed in this setting, then the client application is not allowed to auto-connect to the Administration Service on that computer unless the name or alias of that computer is listed in the **Allowed Servers for Auto-connect** or **Additional Servers for Auto-connect** setting.

If this setting is disabled or not configured, the client application normally auto-connects to any available Administration Service that is published by Active Roles for auto-discovery. However, you can use the **Allowed Servers for Auto-connect** and **Additional Servers for Auto-connect** settings to specify explicitly the instances of the Administration Service to which the client application should auto-connect.

'Additional Servers for Auto-connect' setting

When applied to a computer running an Active Roles client application, such as the Active Roles console, Web Interface or ADSI Provider, this setting specifies the instances of the Active Roles Administration Service to which the client application auto-connects regardless of whether or not those instances are published by Active Roles for auto-discovery.

If you enable this setting, you can specify a list of computer names or aliases identifying the computers running the Administration Service to which the client application auto-connects even though it cannot discover the Administration Service on those computers by using Active Roles' service connection points in Active Directory. If a given computer is listed in this setting, then the client application auto-connects to the Administration Service on that computer regardless of the **Allowed Servers for Auto-connect** and **Disallowed Servers for Auto-connect** settings.

If this setting is disabled or not configured, the client application normally auto-connects to any available Administration Service that is published by Active Roles for auto-discovery. However, you can use the **Allowed Servers for Auto-connect** and **Disallowed Servers for Auto-connect** settings to restrict auto-connection of the client application to specific instances of the Administration Service published for auto-discovery.

Loading the Administrative Template

The Administrative Template consists of the **ActiveRoles.admx** (ADMX) and **ActiveRoles.adml** (ADML) files. The ADML file is a language-specific complement to the ADMX file.

To load the Administrative Template to a domain-wide Group Policy object, you need to copy the ADMX and ADML files to the central store in the **sysvol** folder on a domain controller:

1. Copy the ADMX file to the folder *%systemroot%***sysvol\domain\policies\PolicyDefinitions**
2. Copy the ADML file to the folder *%systemroot%***sysvol\domain\policies\PolicyDefinitions\en-US**

Create those folders if they do not exist. For more information about ADMX files, see Managing Group Policy ADMX Files Step-by-Step Guide at http://go.microsoft.com/fwlink/p/?LinkId=75124.

Group Policy Object Editor automatically reads all ADMX files found in the central store of the domain in which the Group Policy object is created. You can configure Active Roles policy settings in Group Policy Object Editor by selecting **User Configuration/Policies/Administrative Templates/Active Roles Snap-in Settings** or **Computer Configuration/Policies/Administrative Templates/Active Roles/Administration Service Auto-connect Settings**, and then apply the Group Policy object as appropriate.

Appendix C: Communication ports

This section provides a list of communication ports that need to be open in the firewall for Active Roles to function properly.

Access to the managed environment

If the environment managed by Active Roles is located behind a firewall, then the following ports must be open between the Active Roles Administration Service and managed environment.

For instance, if there is a firewall between Active Roles and DNS, then port **15172** must be open (Inbound/Outbound) on the Active Roles host (or the firewall between Active Roles and Exchange) and port **53** must be open on the DNS server (or the firewall between Active Roles and DNS).

Access to DNS servers

- Port **53** TCP/UDP Inbound/Outbound

Access to domain controllers

- Port **88** (Kerberos) TCP/UDP Inbound/Outbound
- Port **135** (RPC endpoint mapper) TCP Inbound/Outbound
- Port **139** (SMB/CIFS) TCP Inbound/Outbound
- Port **445** (SMB/CIFS) TCP Inbound/Outbound
- Port **389** (LDAP) TCP/UDP Outbound
- Port **3268** (Global Catalog LDAP) TCP Outbound
- Port **636** (LDAP SSL) TCP Outbound

This port is required if Active Roles is configured to access the domain by using SSL.

- Port **3269** (Global Catalog LDAP SSL) TCP Outbound

  This port is required if Active Roles is configured to access the domain by using SSL.

- The TCP port allocated by RPC endpoint mapper for communication with the domain controller

  You can configure Active Directory domain controllers to use specific port numbers for RPC communication. For instructions, see http://support.microsoft.com/kb/224196.

- The following ports must be open for the notifications specific to SaaS-based operations to work. The Web Interface machine should be able to resolve Service machine name for Notifications to work.
  - Port 7465 (HTTP) TCP Inbound/Outbound
  - Port 7466 (HTTPS) TCP Inbound/Outbound

Access to Exchange servers

- Port **135** (RPC endpoint mapper) TCP Inbound/Outbound
- The TCP port allocated by RPC endpoint mapper for communication with the Exchange server.

You can configure Exchange servers to use specific port numbers for RPC communication. For more information, contact Microsoft Support.

The following ports must be open for operations related to the WinRM service to work:

- Port **5985** (HTTP) TCP Inbound/Outbound
- Port **5986** (HTTPS) TCP Inbound/Outbound
- Port **80** TCP Inbound/Outbound

Computer resource management

- Port **139** (SMB/CIFS on the managed computers) TCP Inbound/Outbound
- Port **445** (SMB/CIFS on the managed computers) TCP Inbound/Outbound

Computer restart

- Port **139** (SMB/CIFS on the managed computers) TCP Inbound/Outbound
- Port **137** (WINS) UDP Outbound

- Port **138** (NetBIOS datagrams) UDP Outbound

Home folder provisioning and deprovisioning

- Port **139** (SMB/CIFS on the servers that host home folders) TCP Inbound/Outbound
- Port **445** (SMB/CIFS on the servers that host home folders) TCP Inbound/Outbound

Access to SMTP server for e-mail integration

- Port **25** (Default SMTP port) TCP Outbound
- Active Roles uses SMTP port 25 by default. The default port number can be changed in the properties of the Mail Configuration object in the Active Roles console. If Mail Configuration specifies a different port, open that port rather than port 25.

Access to AD LDS instances

- The TCP port specified when registering the AD LDS instance with Active Roles

Access to SMTP server for e-mail integration

- Port **25** (Default SMTP port) TCP Outbound
- Active Roles uses SMTP port 25 by default. The default port number can be changed in the properties of the Mail Configuration object in the Active Roles console. If Mail Configuration specifies a different port, open that port rather than port 25.

Access to Active Roles Administration Service

You can set up a firewall between Active Roles client components, such as the Active Roles Console (also known as the MMC Interface), Web Interface, ADSI Provider or Management Shell, and the Active Roles Administration Service.

To access the Active Roles Administration Service with the Active Roles client components through a firewall, you must open port **15172** and all high ports (**1024-65535**) on port **15172** in the firewall. The client machines randomly select high ports to use for outgoing traffic on port **15172** to access the Active Roles Administration Service.

To give access to the Active Roles Administration Service through a firewall

1. In the firewall, open port **15172** TCP Inbound/Outbound.

   NOTE: For more information about opening ports in your firewall, refer to the operating system's or the network device vendor's documentation.

2. In the firewall, open the high ports (port range **1024-65535**) on port **15172**.

   NOTE: To check the list of high ports being used on port **15172**, in the Active Roles Console of a client machine, use the **`netstat -an`** command.

Access to Web Interface

If you want to access the Active Roles Web Interface through a firewall, then you need to open the following ports:

- Port **80** (Default HTTP) TCP Inbound/Outbound
- Port **443** (Default HTTPS) TCP Inbound/Outbound

The Web Interface normally runs over port 80, or over port 443 if SSL is enabled (off by default).

Appendix D: Active Roles and supported Azure environments

Active Roles supports 3 different Azure environment configurations: Non-federated, Synchronized identity, and Federated.

Non-federated

In a non-federated environment, the on-premises domains are not registered in Azure AD, and neither Azure AD Connect nor any third-party synchronization tools are configured in the domain for synchronization. In non-federated environments, the changes made in Active Roles are immediately replicated to Azure or Office 365 using Graph API Calls or Command-let calls. Users or Guest Users are typically created in Azure with the **onmicrosoft.com** UPN suffix.

Example: Non-federated environment configuration

A non-federated environment may have the following settings:

- On-premises domain: `test.local`
- Azure AD Domain: `ARSAzure.onmicrosoft.com`
- Azure AD Connect is not configured for synchronization.

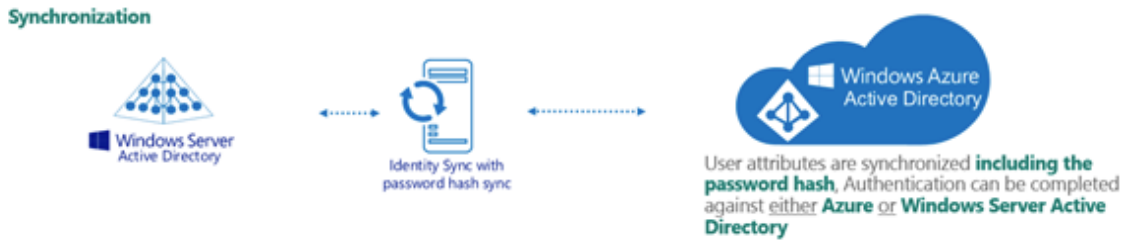
The on-premises domain is not registered in Azure. The user or guest user is created in Active Roles with the ID of **user001@test.local** and in Azure as **user001@ARSAzure.onmicrosoft.com**. The user is created in Azure simultaneously when it is created in Active Roles using a GRAPH API call.

NOTE: One Identity recommends using non-federated environments for testing purposes only, and does not recommend setting them up as a live production environment.

Synchronized identity

In a Synchronized identity environment, the on-premises domain is optionally registered in Azure AD, while Azure AD Connect is configured to synchronize the local AD objects to Azure. Azure Users or Guest Users are typically created either with the selected on-premises domain or with the **onmicrosoft.com** UPN suffix.

Figure 150: Synchronized identity configuration



Example: Synchronized identity configuration

A synchronized identity environment may have the following settings:

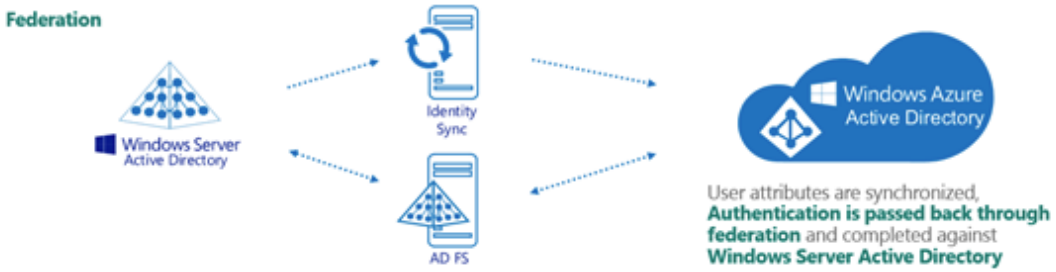
- On-premises domain: `test.local`
- Azure AD Domain: `rd4.qsftdemo.com`
- Azure AD Connect is configured for synchronization.

The on-premises domain is optionally registered in Azure. The user is created in Active Roles with the ID of **user001@test.local** and in Azure as **user001@rd4.qsftdemo.com**.

Federated

In a federated environment, the on-premises domain is always registered in Azure AD, while Azure AD Connect and Active Directory Federation Services (ADFS) are configured to facilitate synchronization. Users and Guest Users are typically created with the UPN suffix of the selected on-premises domain.

Figure 151: Federated configuration



Example: Federated configuration

A federated configuration may have the following settings:

- On-premises domain: `rd4.qsftdemo.com`
- Azure AD Domain: `rd4.qsftdemo.com`
- Azure AD Connect and ADFS are configured for synchronization.

The on-premises domain is registered and verified in Azure. The User is created in Active Roles and Azure AD with the same ID of **user001@rd4.qsftdemo.com**.

Azure Object Management supported in various Azure environments

This section provides information about the supported operations and methods for performing the operations for Azure objects in various Azure environments using Active Roles Web interface, such as Federated, Synchronized Identity, and Non-Federated environments.

In Active Roles Web interface, the required Azure environment configuration can be selected during the Azure tenant creation. The specified configuration can be modified later if needed by changing the Azure properties of the tenant.

Active Roles identifies the environment based on the Azure Tenant type and applies the changes to the Web interface.

Azure Object Management in Non-Federated environment

Non-federated environment is used generally for testing purposes. In non-federated environment, most of the Azure properties can be modified, other than attributes such as **UserPrincipalName** and **ObjectId** which identify the object uniquely.

The following table provides information about the operations and methods of operation that can be performed on Azure Objects in a non-federated environment.

Table 118: Supported Azure configurations comparison chart

Object	Operation	Non-Federated : Method
User	Create	Using GRAPH API
	Read	Using GRAPH API and Exchange Online Command-lets
	Update	Using GRAPH API and Exchange Online Command-lets
	Delete	Using GRAPH API
Guest User	Create	Using GRAPH API
	Read	Using GRAPH API
	Update	Using GRAPH API
	Delete	Using GRAPH API
Security Group	Create	Using GRAPH API
	Read	Using GRAPH API
	Update	Using GRAPH API
	Delete	Using GRAPH API
Mail Enabled Security Group	Create	Using Exchange Online Command-lets
	Read	Using GRAPH API
	Update	Using GRAPH API
	Delete	Using GRAPH API

Object	Operation	Non-Federated : Method
Distribution Group	Create	Using Exchange Online Command-lets
	Read	Using GRAPH API
	Update	Using GRAPH API
	Delete	Using GRAPH API
Native Office 365 Group (Cloud-only*)	Create	Using GRAPH API
	Read	Using GRAPH API
	Update	Using GRAPH API
	Delete	Using GRAPH API
Contacts	Create	Using Exchange Online Command-lets
	Read	Using GRAPH API
	Update	Using Exchange Online Command-lets
	Delete	Using GRAPH API

NOTE: *Active Roles provides cloud-only support only for Office 365 Groups management.

Azure Object Management in Federated and Synchronized Identity environments

Synchronization methods are applicable only in Synchronized and Federated environments and AAD Connect is used to perform the synchronization. Azure non-federated environment does not require synchronization and the direct GRAPH API calls are used to make the Azure or Office 365 object management.

The following table provides information about the operations and methods of operation that can be performed on Azure Objects in Federated and Synchronized Identity environments.

Table 119: Supported Azure configurations comparison chart

Object	Operation	Commands	Tabs	Federated/Synchronized : Method
User	Create			Created by GRAPH API
	Read			Using GRAPH API and Exchange Online Command-lets
	Update	**Azure properties**	**Identity**	Synced using AAD Connect
			Settings	Using GRAPH API
			Job Info	Synced using AAD Connect
			Contact Info	Synced using AAD Connect
			Licenses	Using GRAPH API
			O365 Admin Roles	Using GRAPH API
			OneDrive	Created by OneDrive Policy using PowerShell commands
		Exchange Online properties	**Mail Flow Settings**	Using Exchange Online cmdlets
			Delegation	Using Exchange Online cmdlets
			E-mail Address	Synced using AAD Connect
			Mailbox Features	Using Exchange Online cmdlets
			Mailbox Settings	Using Exchange Online cmdlets
	Delete			Using GRAPH API
Guest Users	Create	**Invite Guest**		Created by GRAPH API
	Read			Using GRAPH API

Object	Operation	Commands	Tabs	Federated/Synchronized : Method
	Update	**Azure properties**	**Identity**	Synced using AAD Connect
			Settings	Using GRAPH API
			Job Info	Synced using AAD Connect
			Contact Info	Synced using AAD Connect
			Licenses	Using GRAPH API
			O365 Admin Roles	Using GRAPH API
		Exchange Online properties	**Mail Flow Settings**	Using Exchange Online cmdlets
			Delegation	Using Exchange Online cmdlets
			E-mail Address	Synced using AAD Connect
			Mailbox Features	Using Exchange Online cmdlets
			Mailbox Settings	Using Exchange Online cmdlets
	Delete			Using GRAPH API
Security Group	Create			Created in Azure, Back Synced to Active Roles, Synced using AAD Connect
	Read			Using GRAPH API
	Update			Synced using AAD Connect
	Delete			Using GRAPH API
Mail Enabled Security Group	Create			Created in Azure, Back Synced to Active Roles, Synced using AAD Connect
	Read			Using GRAPH API
	Update			Synced using AAD Connect
	Delete			Using GRAPH API
Distribution	Create			Created in Azure, Back

Object	Operation	Commands	Tabs	Federated/Synchronized : Method
Group				Synced to Active Roles, Synced using AAD Connect
	Read			Using GRAPH API
	Update			Synced using AAD Connect
	Delete			Using GRAPH API
Native Office 365 Group (Cloud-only*)	Create			Using GRAPH API
	Read			Using GRAPH API
	Update			Using GRAPH API
	Delete			Using GRAPH API
Contacts	Create			Synced using AAD Connect
	Read			Using GRAPH API
	Update			Synced using AAD Connect
	Delete			Using GRAPH API

NOTE:

- *Active Roles provides cloud-only support only for Native Office 365 Group management.
- Synced using AAD Connect referenced in the table specifies that the object operation is initially performed on the on-premise object . After a Microsoft Azure AD Connect synchronization cycle, the object is updated in Azure AD or Office 365.
- For more information on how to perform Back Sync operation refer Active Roles Configuration to synchronize existing Azure AD objects to Active Roles in the Active Roles Administration Guide.

Appendix E: Enabling Federated Authentication

To enable Federated Authentication in Active Roles, perform the following procedures.

NOTE: To access the Active Roles Web Interface for Federated Authentication purposes, you can use any of the following supported web browsers: Google Chrome, Mozilla Firefox, or Microsoft Edge on Windows 10.

NOTE: While switching between the STS providers, restart IIS and clear the browser cache.

1. Configuring the domain service account for Federated Authentication
2. Updating Local Policies
3. Creating SPN entries for the domain service account
4. Enabling delegation for Federated Authentication

Configuring the domain service account for Federated Authentication

To create a dedicated Federated Authentication domain service account and configure its domain and local group memberships, follow the steps.

To configure the domain service account for Federated Authentication

1. Create a dedicated Federated Authentication domain service account, for example: **_arfed**.
2. Ensure that the domain service account has a password that does not expire and set the following:
   a. To configure domain group membership, add the domain service account to the **Domain Users** group.
   b. To configure local group membership on the Active Roles server with the Active Roles Web Interface, add the domain service account to both **Distributed COM Users** and **IIS_USRS** groups.

Updating Local Policies

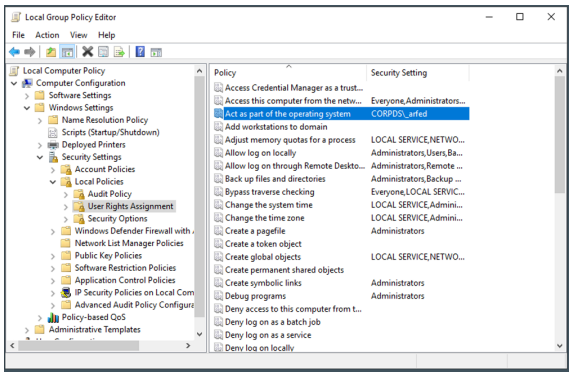
To grant the Federated Authentication domain service account the **Act as part of the operating system** User Rights Assignment on the local Active Roles server with the Local

Group Policy Editor and the Active Roles Web Interface, follow the steps.

To add the Federated Authentication domain service account to the Act as part of the operating system policy

1. Right-click the Windows Start button and select **Run**.
2. Enter `gpedit.msc` and click **OK**.
3. Navigate to **Computer Configuration** > **Windows Settings** > **Security Settings** > **Local Policies** > and select **User Rights Assignment**.
4. In the right pane, double-click **Act as part of the operating system**.
5. Click **Add User or Group**.
6. Enter the name of the Federated Authentication domain service account and click **OK**.
7. Click **OK**.
8. Close the Local Group Policy Editor. The account is now present for the **Act as part of the operating system** policy.

   Figure 152: Local Group Policy Editor

9. Open the Command Prompt with Administrator privileges and run `gpupdate /force` to refresh the local policy with the changes.
10. Close the Command Prompt.

Creating SPN entries for the domain service account

To create ServicePrincipalName (SPN) entries for the Federated Authentication domain service account, follow the steps.

To create SPN entries for the domain service account

1. On a Domain Controller in the domain where Active Roles is installed, log in with Domain Admin credentials to set the following SPNs for the Federated Authentication domain service account, and the delegation settings.
   a. Open the Command Prompt with Administrator privileges and enter each of the following commands one by one:

   ```
   setspn -U -S HTTP/ARWebServerName.YourDomain.com
   YourDomain\ARFederatedAccountName
   ```

   ```
   setspn -U -S HTTP/ARWebServerName YourDomain\ARFederatedAccountName
   ```

   ```
   setspn -U -S ArAdminSvc/ARServerName.YourDomain.com
   YourDomain\ARFederatedAccountName
   ```

   ```
   setspn -U -S ArAdminSvc/ARServerName YourDomain\ARFederatedAccountName
   ```

   b. To confirm that all SPNs are set, run:

   ```
   setspn -L YourDomain\ARFederatedAccountName
   ```

   c. Close the Command Prompt.

Enabling delegation for Federated Authentication

To enable delegation for Federated Authentication, follow the steps.

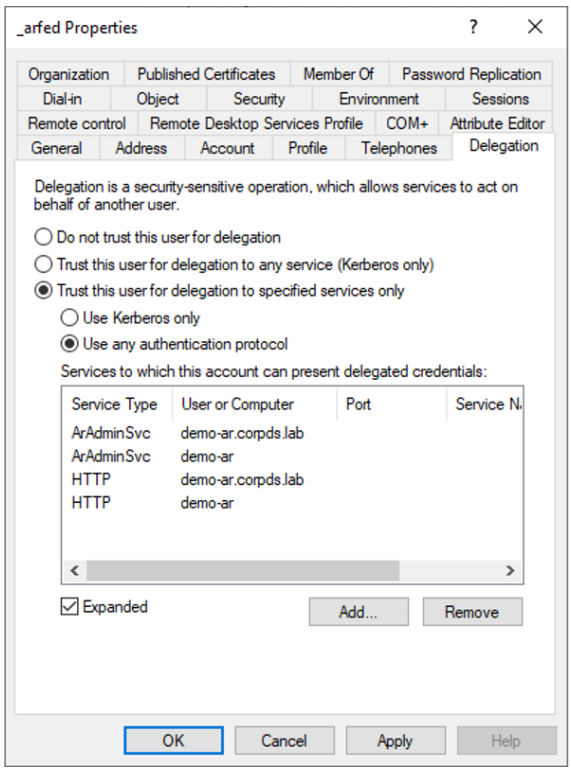
To enable delegation for Federated Authentication

1. Open the Active Directory Users and Computers tool.
2. Open the properties of the Federated Authentication domain service account and click on the **Delegation** tab.
3. Select **Trust this user for delegation to specified services only**.
4. Ensure **Use any authentication protocol** is selected.
5. Click **Add**.
6. Click **Users or Computers**.
7. Enter the name of the Federated Authentication domain service account and click **OK**.

   The **ARAdminSvc** and **HTTP** Service Types are displayed for the short name of the Active Roles server.

8. Click **Select All**.
9. Click **OK**. The Service Types are now listed.
10. To see the FQDN Service Types, select the **Expanded** check box. Click **OK**.

Figure 153: Delegation settings of the Federated Authentication domain service account



11. Close the **Active Directory Users and Computers** tool and log off the Domain Controller.

Redistributable STS

Redistributable STS (RSTS) is a third party authentication component that you can use as an alternative to the default IIS Windows authentication, when logging in to the Active Roles Web Interface.

- To install RSTS, see Installing Redistributable STS.
- To configure RSTS in in RStsApiAdmin, see Configuring Redistributable STS in RStsApiAdmin.
- To configure RSTS for Active Roles Web Interface in Configuration Center, see Configuring Redistributable STS in Configuration Center.

Installing Redistributable STS

Redistributable STS (RSTS) is a third-party authentication component that you can use as an alternative to the default IIS Windows authentication, when logging in to the Active Roles Web Interface.

To install the RSTS application, follow the steps.

Prerequisites

1. In Windows, open **Local Security Policy** and navigate to **Security Settings > Local Policies > User Rights Assignment** to add the domain user(s) to the **Act as part of the operating system** policy.
2. Make sure that the User Logon Name and the UPN Suffix is set for the domain user (s). Example: **`administrator@ars1.net`**.

Installing Redistributable STS (RSTS)

1. Open the Command Prompt with Administrator privileges.
2. Change the directory to the location of the Redistributable STS (RSTS). By default, RSTS is available in `C:\Program Files\One Identity\Active Roles\<version>\Web\RSTS`.
3. Type `RSTS /install` to install the RSTS application.

Configuring Redistributable STS in RStsApiAdmin

To configure Redistributable STS (RSTS) in the RStsApiAdmin application, follow the steps.

To configure Redistributable STS (RSTS) in RSts Api Admin

1. In `C:\Program Files\One Identity\Active Roles\<version>\Web\RSTS`, Open the **RStsApiAdmin** application.
2. In **Connection Settings**, enter **`localhost`** in the **Server** field and click **Login**.
3. In the **Directory Service Providers** pane, open the **Provider Specific Options** tab to configure Active Roles.
   - **Domain**: The name of the domain.
   - **Username**: The sAMAccountName of an Active Directory service account with read permissions.
   - **Password**: The password of the domain user.
4. In the **Claim Definitions** tab, right-click and select **New** to open the **Add Claim Definition** window.

5. Select the **RSts.Sts.Attribute.Claim.Definition** claim definition type and click **OK**.
6. In the **Settings** tab, configure the settings of the new attribute claim for User Principal Name (UPN).
   - **Claim ID**: Preferred name. Example: `AccountName`
   - **Claim Type**: `http://schemas.xmlsoap.org/ws/2005/05/identity/claims/userprincipalname`

     NOTE: For the claim type, you must provide `userprincipalname` instead of the abbreviation `upn`.
   - **Attribute ID**: `userPrincipalName`
7. Click **Save**.

Configuring Redistributable STS in Configuration Center

To configure the Redistributable STS (RSTS) authentication method in the Active Roles Configuration Center, follow the steps.

To configure Redistributable STS (RSTS) in Configuration Center

1. In the **Configuration Center**, navigate to **Web Interface > Authentication**.
2. In the **Site authentication settings** window, select the **Federated** tab.
3. In **Identity provider configuration**, configure the settings of the identity provider.
   a. From the **Identity provider** drop-down, select **RSTS**.
   b. **Federated metadata URL**: `http://localhost/rsts/wsfedmetadata`
   c. Click **Test metadata** to validate the metadata.
   d. **Realm**: The URL of the Web Interface: `https://<Web Server Name>/arwebadmin/`
   e. **Reply URL**: The URL of the Web Interface: `https://<Web Server Name>/arwebadmin/`

      NOTE: To check the URL of the Web Interface, navigate to **Web Interface > Sites** and click **Open in Browser**.
4. In **Claim editor**, click **Add** to open the **Add claim** window.
   a. Make sure that the **Claim type** is UPN.
   b. Set the following **Claim value**: `http://schemas.xmlsoap.org/ws/2005/05/identity/claims/userprincipalname`
   c. Set the following **Display name**: `IUser.UPN`
   d. Click **Save**.
5. Enter the **Domain user login credentials**.

a. **Username**: The name of the domain and the user in domain\user format, for example: **ars1\Administrator**

b. **Password**: The password of the domain user.

c. Click **Modify**.

6. In the list of services, check if the `RedistributableSecureTokenServer` service is running.

As a result of modifying the authentication settings in the Configuration Center, you can log in with the domain user to the Active Roles Web Interface using RSTS authentication.

Examples of configuring identity providers

Refer to the following examples of configuring the identity providers when using Federated Authentication.

Azure

- **Metadata url**: **https://login.microsoftonline.com/<AzureTenantID>/FederationMetadata/2007-06/FederationMetadata.xml**
- **realm**: **spn:<Azure Application ID>**
- **replyurl**: **https://<Web Server Name>/arwebadmin/**

Active Directory Federation Services (AD FS)

- **Metadata url**: **https://<ADFS Server name>/FederationMetadata/2007-06/FederationMetadata.xml**
- **realm**: **https://<Web Server Name>/arwebadmin/**
- **replyurl**: **https://<Web Server Name>/arwebadmin/**

Appendix F: Active Roles integration with other One Identity and Quest products

You can integrate Active Roles with the following One Identity products to complement and extend identity and access management in your organization.

Change Auditor

Quest Change Auditor for Active Directory is a security auditing solution providing real-time notifications for critical AD, Azure AD and ADFS configuration changes. The application tracks, audits, reports and alerts on all key configuration changes (for example, modifications in users, groups, nested groups, GPOs, computers, services, registry entries, local users or the DNS), and consolidates them in a single console without the overhead of native auditing.

In addition, you can lock down critical AD objects to protect them from unauthorized or accidental modifications and deletions. Correlating activity across the on-premises and cloud directories, Change Auditor provides a single pane of glass view of your hybrid environment and makes it easy to search all events regardless of where they occurred.

For more information on integrating Active Roles with Change Auditor, see *Active Roles Integration* in the Change Auditor Installation Guide, or Change Auditor Knowledge Base Article 309842.

Cloud Access Manager

One Identity Cloud Access Manager (CAM) delivers real productivity gains by minimizing the effort required to control access to the on-premises applications and cloud service accounts of your organization. When using CAM, users (such as employees or customers) require only a single username and password combination to gain secure access to their resources through a customizable application portal. However, if stronger authentication is preferred, you can also configure CAM to require one-time passwords (OTP) during login.

For more information on using CAM with Active Roles, see *Integrated Windows Authentication* in the Cloud Access Manager Configuration Guide.

Defender

One Identity Defender is a cost-effective security solution that authenticates users who access your network resources. When deployed in your organization, only users who successfully authenticate via Defender are granted access to the secured resources.

Defender uses the user identities stored in Microsoft Active Directory (AD) to enable two-factor authentication (2FA), taking advantage of its inherent scalability and security, and eliminating the costs and time required to set up and maintain proprietary databases. The web-based administration tool and the user self-service portal of Defender ease the

implementation of 2FA for both administrators and users. Defender also provides a comprehensive audit trail that enables compliance and forensics.

For more information on using Defender with Active Roles, see *Integration with Active Roles* in the Defender Administration Guide.

Enterprise Reporter

Quest Enterprise Reporter provides administrators, security officers, help desk staff, and other stakeholders insight into their network environment. Reporting on your network environment provides general visibility into the security and configuration of your environment, validation against your security policies to ensure objects are configured as expected, and an easy way to respond to inquiries from auditors requesting security and configuration information.

Enterprise Reporter provides a unified solution for data discovery and report generation. Using the Enterprise Reporter Configuration Manager, administrators can easily configure and deploy discoveries to collect and store data. Once the data has been collected, the Report Manager allows users to produce reports that help organizations ensure that they comply with industry regulations and standards, adhere to internal security policies, and fulfill hardware and software requirements.

For more information on using Enterprise Reporter with Active Roles, see the Enterprise Reporter Configuration Manager User Guide, or the Quest Enterprise Reporter Knowledge Base.

Identity Manager

One Identity Manager simplifies managing user identities, access permissions, and security policies. By delegating identity management and access decisions directly to the organization, Identity Manager can ease the workload of the company IT team, so they can focus on their core competences.

For more information on integrating Active Roles with Identity Manager, see *Working with One Identity Manager* in the Active Roles Synchronization Service Administration Guide and the Identity Manager Administration Guide for Active Roles Integration.

Recovery Manager for Active Directory

Quest Recovery Manager for Active Directory (RMAD) is an AD recovery tool that enables you to recover sections of the organization AD (for example, selected objects or object properties) without taking AD offline. RMAD minimizes potential AD downtimes that data corruption or improper directory modifications can cause by offering automatic backup options, and fast, remotely managed recovery operations.

Active Roles supports integration with RMAD through its Active Roles Add-on for RMAD extension. When installed, the Active Roles Web Interface receives a new **Restore Object** option, opening the Recovery Manager Portal of RMAD, and allowing you to restore modified or deleted directory objects.

For more information on RMAD, see the RMAD technical documentation. For more information on the Active Roles Add-on for RMAD extension, see the *Active Roles Add-on for Recovery Manager for Active Directory Release Notes*.

Safeguard

One Identity Safeguard is a privileged management software used to control, monitor, and govern privileged user accounts and activities to identify possible malicious activities, detect entitlement risks, and provide tamper proof evidence. Safeguard products also aid incident investigation, forensics work, and compliance efforts.

The One Identity Safeguard for Privileged Passwords (SPP) appliances are built specifically for use only with the SPP privileged management software, which is pre-installed and ready for use on the SPP appliance. The SPP appliance is hardened to ensure the system is secured at the hardware, operating system, and software levels as well. The hardened appliance approach protects the privileged management software from attacks while simplifying deployment and ongoing management, and also shortening the time frame to value period.

For more information on SPP, see the latest One Identity Safeguard for Privileged Passwords documentation.

Safeguard Authentication Services

One Identity Safeguard Authentication Services (SAS) extends the security and compliance of AD to Unix, Linux, and macOS platforms and enterprise applications with the following features:

- Addressing the compliance need for cross-platform access control.
- Addressing the operational need for centralized authentication and single sign-on.
- Unifying identities and directories for simplified identity and access management.

For more information on integrating Active Roles with SAS, see the Authentication Services Active Roles Integration Pack Release Notes or SAS Knowledge Base Article 253135.

Starling

Active Roles supports integration with the One Identity Starling Two-Factor Authentication and Starling Connect services.

- One Identity Starling Two-Factor Authentication is designed to support non-federated applications and applications acting as Identity Providers (IdP) to accept an OTP for 2FA via text message, phone call or the Starling 2FA app. Starling Two-Factor Authentication also supports push notifications, where users receive approval requests on their Starling 2FA app for 2FA. Applications using Starling Two-Factor Authentication can validate OTP and redirect all OTP and push notification management workflows directly to Starling Two-Factor Authentication, providing a single interface for all 2FA operations.

  For more information on integrating Active Roles with Starling Two-Factor Authentication, see Configuring Active Roles to join One Identity Starling.

- One Identity Starling Connect is a cloud-based service extending the provisioning capabilities of Active Roles to a growing collection of Software-as-a-Service (SaaS) applications, enabling organizations to streamline processes and secure hybrid

environments. This allows you to extend your on-premises Active Roles deployment to provision additional applications, regardless of whether they are on-premises or cloud-based.

For more information on integrating Active Roles with Starling Connect, see Starling

TPAM

The Privileged Appliance and Modules (TPAM) appliance is a robust collection of integrated modular technologies designed specifically to meet the complex and growing compliance and security requirements associated with privileged identity management and privileged access control. TPAM consists of two main modules:

- Privileged Password Manager ensures that when administrators require elevated access, that access is granted according to the established organization policy, with all appropriate approvals, with all actions fully audited and tracked, and in a way that the password used is immediately changed upon its return.
- Privileged Session Manager provides tools for session and proxy control, and for auditing, recording and replaying the activities of high-risk users, for example administrators or remote vendors. It provides a single point of control from which you can authorize connections, limit access to specific resources, view active connections, record all activities, or raise alerts and terminate connections if they exceed the pre-set time limits.

With its available modules, One Identity TPAM automates, controls and secures the entire process of granting administrators the credentials necessary to perform their duties.

For more information on the TPAM Appliance, see the One Identity TPAM technical documentation, or the TPAM Knowledge Base.

Appendix G: Active Roles integration with Duo MFA

Active Roles can be integrated with Duo Multi-Factor Authentication (MFA) to complement and extend identity and access management. For more information about Duo, see https://duo.com.

When used together with Duo (or another MFA provider), Active Roles directs all connections to the authentication tool, and upon successful authentication, it permits the user to access the information system.

Active Roles can interact with your Duo account and can automatically request strong MFA for your users who are accessing the servers and services protected by Active Roles. When used together with Duo, the Active Roles Web Interface prompts the user for a second factor authentication, and upon successful authentication, it allows the user to access the information system.

The integration adds an additional security layer to the gateway authentication performed on Active Roles. If the Duo Mobile app is installed on the user's device (smartphone, notebook, smartwatch, and so on), the user can generate a One-Time Password (OTP) using the device. The user then can use the OTP to authenticate to the One Identity platform. The one-time password is changed after 60 seconds.

Configuring Duo for Active Roles

To create and configure a new Active Roles application in Duo, see Duo Single Sign-On in the *Duo Documentation*.

Prerequisites:

- You must have a valid Duo subscription with the **Owner** role. Access the Duo Admin Panel where you can manage your Duo two-factor authentication (2FA) system. For more information, see Duo Administration - Admin Panel Overview in the *Duo Documentation*.
- The affected users must be enrolled and activated in Duo. For more information on enrollment types and activation, see Duo Administration - Enroll Users in the *Duo Documentation*.
- The affected users must install the Duo Mobile app.

Integrating Active Roles with Duo

To integrate Active Roles with Duo, configure the RSTS Api Admin application.

Prerequisites

- Enable Federated Authentication in Active Roles. For more information, see Appendix E: Enabling Federated Authentication.
- Install and configure RSTS. For more information, see Redistributable STS.
- Enroll and activate users in Duo. For more information on enrollment types and activation, see Duo Administration - Enroll Users in the *Duo Documentation*.

To configure RSTS Api Admin for Integrating Active Roles with Duo

1. In `C:\Program Files\One Identity\Active Roles\<version>\Web\RSTS`, Open the **RStsApiAdmin** application.
2. In **Connection Settings**, enter **`localhost`** in the **Server** field and click **Login**.
3. In the **Directory Service Providers** pane, right-click and select **New** to open the **Add Directory Service Provider** window.
4. Select **RSts.Sts.DuoWebConfiguration** and click **OK**.

   NOTE: You can only add the Duo Web 2FA provider for Two-Factor Authentication.
5. Obtain the integration key, secret key and API hostname.

   NOTE: For more information, see Protecting Applications in the *Duo Documentation*.

   a. Log in to the Duo Admin Panel and navigate to **Applications**.

   b. Click **Protect an Application** and in the applications list, locate the entry for **Auth API**.

   c. Click **Protect** to configure the application and get your integration key, secret key, and API hostname displayed in the **Details** section.
6. In **RSTS Api Admin**, in the **Directory Service Providers** pane, open the **Provider Specific Options** tab.
7. To configure the Duo Web 2FA provider, enter the following values based on step Obtain the integration key, secret key and API hostname.

   - **Integration key**
   - **Secret key**
   - **API hostname**
8. Enter the value for **Username Attribute**.

   The value passed in the Duo Web 2FA token must match the **Associated Directory ID** provider of the user in the **Standard Options** pane. The value of the attribute must be unique in the primary provider.
9. Click **Save**.

Validating Duo Multi-Factor Authentication

Follow the steps to validate Duo Multi-Factor Authentication (MFA) in the Active Roles Web Interface.

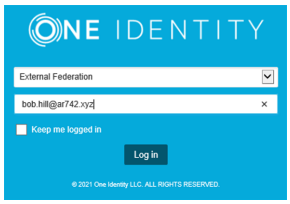
Prerequisites

- Install and configure Redistributable STS (RSTS). For more information, see Redistributable STS.
- Configure Duo. For more information, see Configuring Duo for Active Roles.
- Configure the RSTS Api Admin application. For more information, see Integrating Active Roles with Duo.
- Create a user account in Active Directory (on-premises or Office 365) with a valid email address identical to the user account created in Duo.

To validate Duo MFA in Active Roles Web Interface

1. In the Active Roles Web Interface, open the Administration site (.\ARWebAdmin).
2. In the RSTS login page, set the Authentication Provider to External Federation.

   Figure 154: Logging in to RSTS with an example user

3. Enter the credentials of the user created in Duo.
4. You are redirected to Duo. Log in with the same credentials.
5. To add and verify the identity of the user account on different mobile devices, see Enrollment Guide in the *Duo Documentation*.
6. Once you added the user account to the Duo Mobile app, the Active Roles Web Interface opens.
7. (Optional) Select **Remember Me** to reduce the amount of all subsequent logins.

Appendix H: Active Roles integration with Okta MFA

Active Roles can be integrated with Okta Multi-Factor Authentication (MFA) to complement and extend identity and access management. For more information about Okta, see https://www.okta.com/.

Okta is a cloud-based identity service offering identity, authentication, and access control functions as a service. To support MFA, Active Roles integrates with the Okta identity management service. This enables you to leverage an additional out-of-band factor (typically through the user's registered smartphone) when authenticating the user. The additional factor is processed in-line with the connection, so users do not have to switch to an external application to process the additional factor. This results in a seamless and efficient user experience that is readily accepted by the users.

Active Roles can interact with your Okta account and can automatically request strong MFA for your users who are accessing the servers and services protected by Active Roles. Okta supports a broad range of authentication methods, including software, hardware, and mobile-based solutions.

Configuring Okta for Active Roles

To create and configure a new Active Roles application in Okta, see Integrate your SAML app in the *Okta Admin Documentation*.

Prerequisites

- Download the Okta Verify app to your phone.
- Sign up for a free Okta Developer instance: https://developer.okta.com/signup/.

Integrating Active Roles with Okta

To integrate Active Roles with Okta, configure the RSTS Api Admin application.

Prerequisites

- Enable Federated Authentication in Active Roles. For more information, see Appendix E: Enabling Federated Authentication.
- Install and configure RSTS. For more information, see Redistributable STS.

- Configure Okta for Active Roles. For more information, see Configuring Okta for Active Roles.

To configure RSTS Api Admin for Integrating Active Roles with Okta

1. In `C:\Program Files\One Identity\Active Roles\<version>\Web\RSTS`, Open the **RStsApiAdmin** application.
2. In **Connection Settings**, enter **`localhost`** in the **Server** field and click **Login**.
3. In the **Directory Service Providers** pane, right-click and select **New** to open the **Add Directory Service Provider** window.
4. Select **RSts.Sts.RelyingPartyConfiguration** and click **OK**.
5. In the **Directory Service Providers** pane, open the **Standard Options** tab to enter the following values.
   - **ID**: Preferred name, for example **`Okta`**.
   - **Associated Directory ID**: This field is case sensitive, for example **`ad`**.
6. In the **Directory Service Providers** pane, open the **Provider Specific Options** tab to enter the following values.
   - **Realm**: Enter your valid email address domain (on-premises or Office 365).
   - **Federation metadata**: Insert the metadata obtained during the Okta configuration. For more information, see Integrate your SAML app in the *Okta Admin Documentation*.
7. Click **Save**.

Validating Okta Multi-Factor Authentication

Follow the steps to validate Okta Multi-Factor Authentication (MFA) in the Active Roles Web Interface.

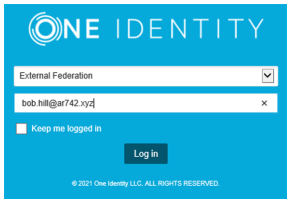
Prerequisites

- Install Redistributable STS (RSTS). For more information, see Redistributable STS.
- Configure Okta. For more information, see Configuring Okta for Active Roles
- Configure the RSTS Api Admin application. For more information, see Integrating Active Roles with Okta.
- Create a user account in Active Directory (on-premises or Office 365) with a valid email address identical to the user account created in Okta.

To validate Okta MFA in Active Roles Web Interface

1. In the Active Roles Web Interface, open the Administration site (.\ARWebAdmin).
2. In the RSTS login page, set the Authentication Provider to External Federation.

   Figure 155: Logging in to RSTS with an example user

3. Enter the credentials of the user created in Okta.
4. You are redirected to Okta. Log in with the same credentials.
5. To add and verify the identity of the user account on different mobile devices, see Get started with Okta Verify in the *Okta Documentation for end users*.
6. Once you added the user account to the Okta Verify app, the Active Roles Web Interface opens.
7. (Optional) Select **Remember Me** to reduce the amount of all subsequent logins.

About us

One Identity solutions eliminate the complexities and time-consuming processes often required to govern identities, manage privileged accounts and control access. Our solutions enhance business agility while addressing your IAM challenges with on-premises, cloud and hybrid environments.

Contacting us

For sales and other inquiries, such as licensing, support, and renewals, visit https://www.oneidentity.com/company/contact-us.aspx.

Technical support resources

Technical support is available to One Identity customers with a valid maintenance contract and customers who have trial versions. You can access the Support Portal at https://support.oneidentity.com/.

The Support Portal provides self-help tools you can use to solve problems quickly and independently, 24 hours a day, 365 days a year. The Support Portal enables you to:

- Submit and manage a Service Request
- View Knowledge Base articles
- Sign up for product notifications
- Download software and technical documentation
- View how-to videos at www.YouTube.com/OneIdentity
- Engage in community discussions
- Chat with support engineers online
- View services to assist you with your product