Active Roles 8.1.3

Administration Guide

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/trademark-information.aspx. 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 - 15 November 2023, 22:43

For the most recent documents and product information, see Online product documentation.

Contents

1

Introduction

The Active Roles Administration Guide provides detailed information about how to configure and maintain an installed Active Roles deployment for day-to-day administrative operations.

The document describes how to:

- Configure rule-based and role-based administration settings.
- Configure automatic resource provisioning and deprovisioning.
- Set up automation and approval workflows for administrators or helpdesk personnel.
- Manage groups via temporal group memberships, group families or dynamic groups.
- Configure and monitor Active Roles reporting and Management History settings.
- Configure entitlement profiles to give access to specific information resources.
- Use the Active Directory Recycle Bin with Active Roles.
- Integrate Active Roles with One Identity Starling.
- Configure linked and remote Exchange mailboxes.
- Register Azure AD tenants with Active Roles to manage Azure AD objects and resources.
- Configure SQL Server replication.
- Use Administrative Templates to set the behavior and appearance of the Active Roles Console with Group Policies.
- Integrate Active Roles with other One Identity, Quest or third-party products and services.
- Use optional utilities (the Configuration Transfer Wizard, Diagnostic Tools, Add-on Manager or the Active Roles Language Pack) to enhance and maintain your Active Roles deployment.

> NOTE: For information about how to perform day-to-day administrative tasks, see the following documents:
>
> - For information about how to administer Active Directory resources in the Active Roles Console, see the *Active Roles Console User Guide*.

- For information about how to administer Active Directory and Azure AD resources with the Active Roles Web Interface, see the *Active Roles Web Interface User Guide*.

In addition, for information about how to configure and customize the Active Roles Web Interface component, see the *Active Roles Web Interface Configuration Guide*.

Getting started with Active Roles

This section describes how to start using Active Roles to prepare it for day-to-day administration operations.

> NOTE: The Active Roles Administration Guide only describes product configuration procedures. For the in-depth description of its features and user interfaces, see the following documents:
>
> - For more information on the product features, see the *Active Roles Feature Guide*.
> - For more information on the Active Roles Console and the day-to-day operations you can perform with it, see the *Active Roles Console User Guide*.
> - For more information on the Active Roles Web Interface and the day-to-day operations you can perform with it, see the *Active Roles Web Interface User Guide*.
> - For more information on customizing and configuring the Web Interface and its sites, see the *Active Roles Web Interface Configuration Guide*.

Starting the Active Roles Console

The Active Roles Console, also referred to as MMC Interface, is a comprehensive administrative tool that you can use to:

- Manage Active Directory and Microsoft Exchange resources.
- Configure organization-level access and administration policies.
- Set up automation or approval workflows for your administrators or helpdesk personnel.

To start the Active Roles Console

1. Log in to the system where Active RolesConsole is installed.
2. Depending on the version of your operating system:

- In the **Apps** page, click **Active Roles 8.1.3 Console**.
- From the **Start** menu, select **All Programs** > **One Identity Active Roles 8.1.3** > **Active Roles 8.1.3 Console**.

> NOTE: By default, the Active Roles Console automatically chooses an Administration Service instance 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.

Restricting access to the Active Roles Console

By default, after installing Active Roles, every user can log in to the Active Roles Console. To restrict access:

1. Use the **MMC Interface Access** setting of the Active Roles Configuration Center. This setting lets you restrict Console access only to Active Roles Admin users (or allow Console access again for all users, if the access is restricted). For details, see Restricting access to the Active Roles Console.
2. If Console access is already restricted to Active Roles Admin users, you can give Console access to individual users by assigning them to the **User Interface Management - MMC Full control**Access Template (AT). This AT gives access permission to the **Server Configuration** > **User Interfaces** > **MMC Interface** object. For details, see Restricting access to the Active Roles Console.

For more information, see *Restricting access to the Active Roles Console* in the *Active Roles Installation Guide*.

Minimum required permissions of the Active Roles service account

Active Roles performs operations on directory objects on behalf of delegated users. Because of this, the Active Roles service account that is used to manage the Active Directory domain requires adequate permissions.

> NOTE: One Identity strongly recommends to manage the Active Directory domain using an account that is a member of the **Domain Admins** role group. If this condition is not met, the information and instructions provided in the official One Identity product documentation may not be applicable to your Active Roles installation.

> TIP: One Identity recommends using separate service accounts for service tasks and for domain management duties. Doing so can ensure that you can use the service account with the minimum required permissions listed below. However, consider that the proxy

> account must still be a member of the **Domain Admins** role group to stay within the support model of Active Roles.

The service account credential has the following five main roles.

Accessing the Administration Service computer

To meet this requirement, the service account must be a member of the **Administrators** group on the computer running the Active Roles Administration Service.

Service publication in Active Directory

Once configured, the Administration Service attempts to publish itself in Active Directory, so that Active Roles clients can automatically discover the Administration Service instance.

> NOTE: While this functionality is not critical, if the service publication permissions are not granted, Active Roles clients will not be able to automatically discover the Active Roles Administration Service instance. However, they can still connect to the Administration Service if they specify in Active Roles Console either the service name or the IP address of the computer running the instance.

For more information, see *Service publication in Active Directory* in the *Active Roles Installation Guide*.

Running all Script Modules under the security context of the Active Roles Service Account

The permissions required by custom scripts vary according to the requirements of the individual scripts. As such, review them on a case-by-case basis as a Best Practice security model.

Connecting to the Microsoft SQL database

In some Active Roles configurations, assigning the SQL database connection permissions to the service account is optional, as you can also use an SQL Authentication credential (which then receives the required permissions instead of the service account).

For more information on the necessary SQL Server permissions, see *SQL Server Permissions* 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 you want to use the Active Roles security synchronization feature.

Configuring rule-based administrative views

To provide additional flexibility beyond the default Active Directory and Azure AD capabilities in managing directory resources, Active Roles supports creating, editing and deleting securable, flexible, rule-based administrative views, known as Managed Units (MUs).

With MUs, administrators can configure distributed administration units independent of the OU hierarchy. As such, MUs are dynamic virtual collections of AD or Azure AD directory objects, and may include them regardless of their location in the organization network.

> TIP: For more information on Managed Units and their main features, see *Managed Units* in the *Active Roles Feature Guide*.

Administering Managed Units

This section guides you through the Active Roles Console to administer Managed Units.

Creating a Managed Unit

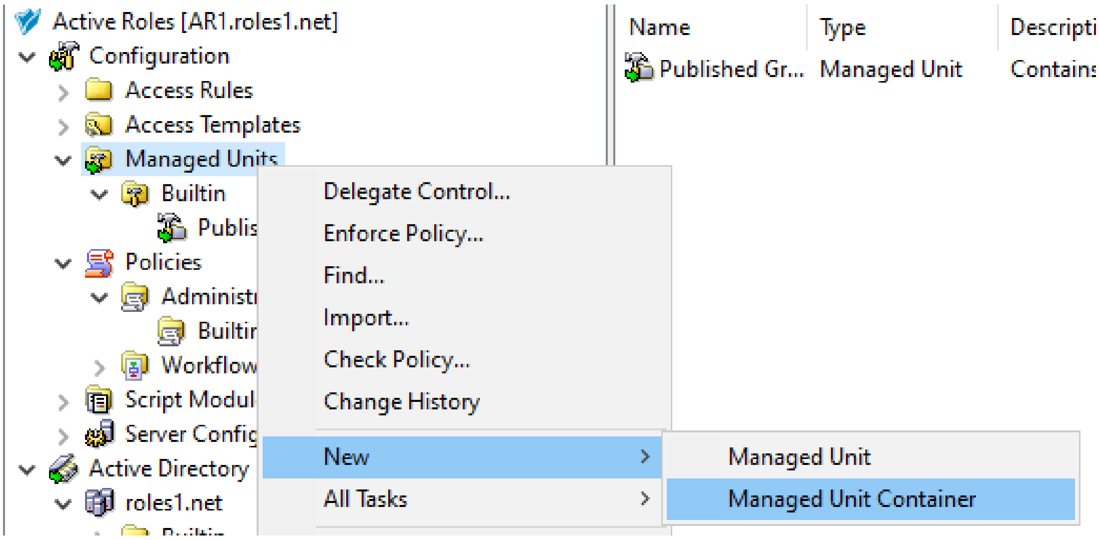
You can create a new Managed Unit (MU) in the Active Roles Console.

Prerequisites

To create MUs in the Active Roles Console, you must use an Active Roles Administration Service account. For more information, see *Configuring the Administration Service account* in the *Active Roles Quick Start Guide*.

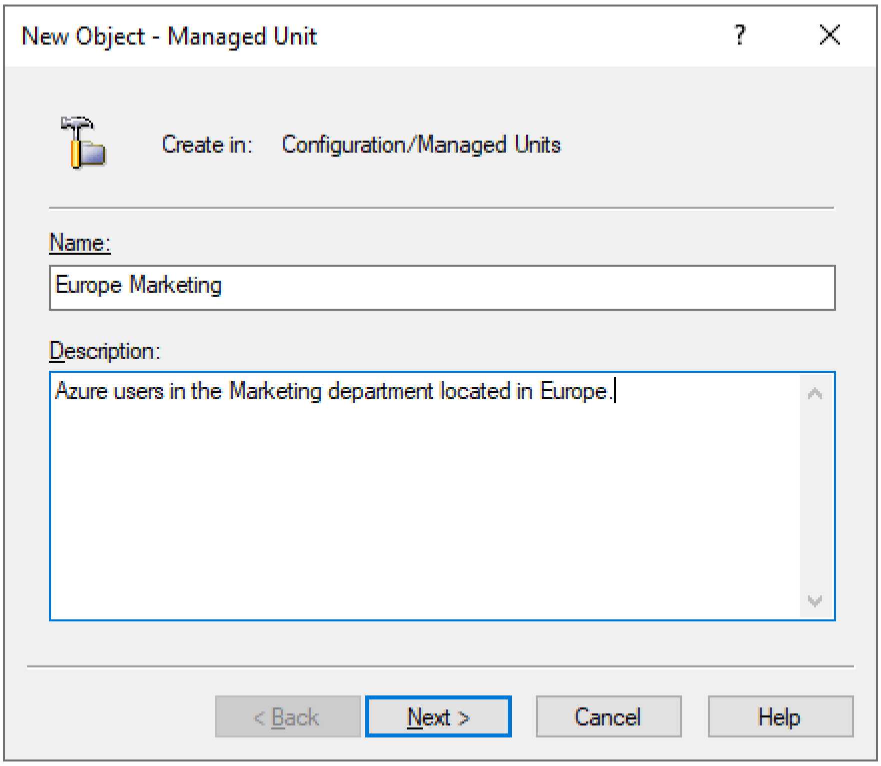
To create a new Managed Unit (MU) in the Active Roles Console

1. In the Active Roles Console, on the **Console tree**, navigate to **Configuration** > **Managed Units**.
2. To open the **New Object - Managed Unit** wizard, right-click the **Managed Units** node, then click **New** > **Managed Units**.

   > TIP: If you need to manage a large number of MUs in your organization, One Identity recommends creating separate MU containers for your specific MUs.
   >
   > To create a new container for the configured MU, right-click on the **Managed Units** node, then click **New** > **Managed Unit Container**.
   >
   > **Figure 1: Active Roles Console – Launching the Managed Unit Container dialog**
   >
   > 
   >
   >
   >
   > Once the new container is created, right-click it in the **Console tree** and select **New** > **Managed Unit** to create a new MU in the container. To move an existing, non built-in MU to the container, right-click the MU, and select **Move**.

3. In the **Name** step, specify a **Name** and optionally, a **Description** for the new MU. This name and description will appear in the Active Roles details pane when selecting the MU.

Figure 2: New Object - Managed Unit wizard – Specifying the Name and Description


New Object - Managed Unit
?
Create in: Configuration/Managed Units

Name: Europe Marketing

Description: Azure users in the Marketing department located in Europe.

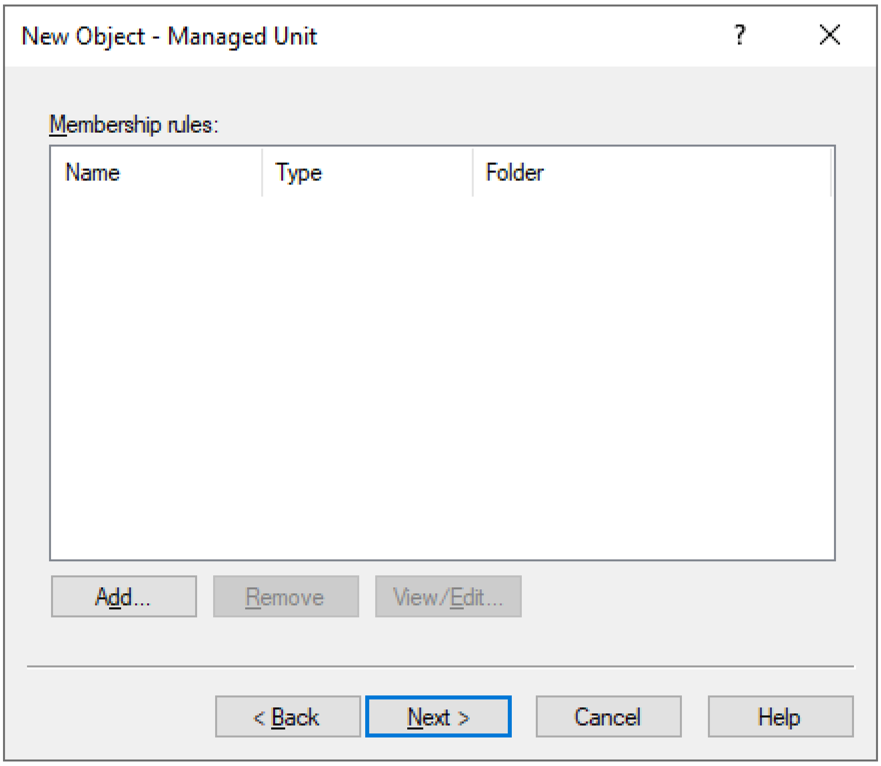
< Back
Next >
Cancel
Help

To continue, click **Next**.

4. To specify a new membership rule for the MU, in the **Membership rule** step, click **Add**.

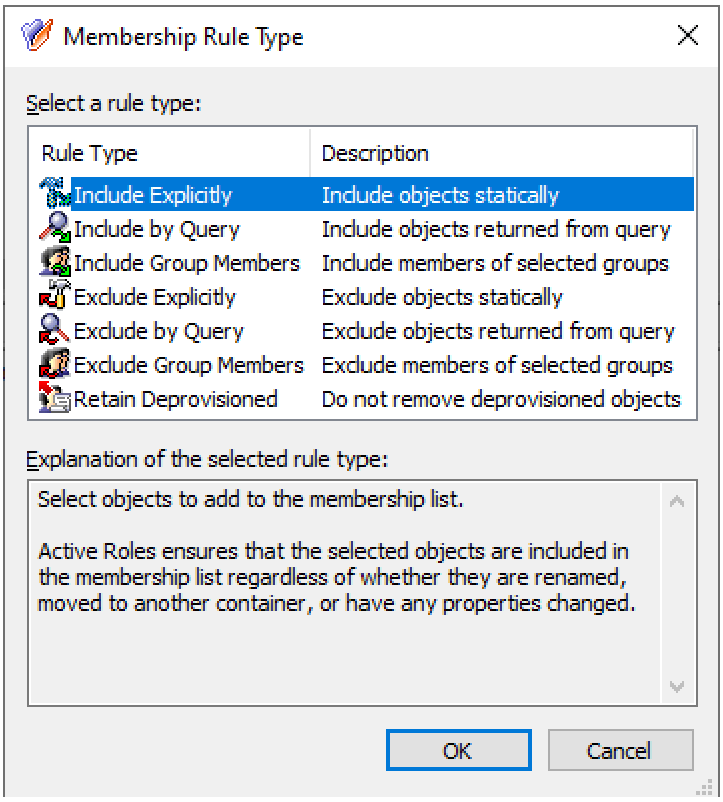
   Membership rules define which directory objects get assigned to the MU. Active Roles populates the MU dynamically based on the configured rules, adding objects that match their criteria and removing those later that no longer do.

Figure 3: New Object - Managed Unit wizard – Membership rule list



5. In the **Membership Rule Type** dialog, select the rule type used to populate the MU. A membership rule can be a search query, a static object inclusion or exclusion rule, or group membership inclusion and exclusion rule.

Figure 4: New Object - Managed Unit wizard – Membership rule type selection



Membership Rule Type

Select a rule type:

Rule Type	Description
Include Explicitly	Include objects statically
Include by Query	Include objects returned from query
Include Group Members	Include members of selected groups
Exclude Explicitly	Exclude objects statically
Exclude by Query	Exclude objects returned from query
Exclude Group Members	Exclude members of selected groups
Retain Deprovisioned	Do not remove deprovisioned objects

Explanation of the selected rule type:

Select objects to add to the membership list.

Active Roles ensures that the selected objects are included in the membership list regardless of whether they are renamed, moved to another container, or have any properties changed.

Active Roles supports the following membership rule types:

Table 1: Managed Unit membership rules

Rule name	Description
Include Explicitly	Includes the Active Directory (AD) or Azure Active Directory (Azure AD) objects you select in the wizard. Once selected, Active Roles will keep the objects included in the MU even if they are updated, renamed, or moved elsewhere within your organization directory.
Include by Query	Lets you define a custom query that the AD or Azure AD objects must match to be included in the MU. The query editor dialog lets you select the object type and location (such as AD domain or Azure tenant), then dynamically populates the dialog with settings according to the object type you selected. The dialog also offers **Advanced** query settings to configure queries by specifying the following elements to check:

Rule name	Description
	• Object types and properties • Logical conditions • Specified values Once you configure a query, you can test it with the **Preview Rule** button. NOTE: Consider the following when configuring a custom query: • The **Include by Query** membership rule does not support Azure contacts and Azure distribution groups. To include Azure contacts or Azure distribution groups in an MU, use the **Include Explicitly** rule type. • The contents of the **Condition** drop-down list are static, and may contain logical conditions that do not work with the selected object attribute (for example, selecting **Greater or equal** for the **edsaAzureManager** Azure AD attribute returns no results). Always make sure to select a logical condition against which Active Roles can enumerate the value of the selected Azure attribute. • When querying Azure object attributes, the **Ends with** condition returns results only if you specify whole words. The only exceptions to this behavior are the **mail**, **otherMails**, **userPrincipalName** and **proxyAddresses** attributes, which you can also query with the **Ends with** condition by specifying them partially. • You can query the **edsaAzureManager** attribute with the **Is not** condition only if the query rule is used in an **AND** relationship with another query rule. Querying the **edsaAzureManager** attribute with the **Is not** condition returns no results if the query rule is used alone or in an **OR** relationship.
Include Group Members	Includes the members of the selected AD or Azure AD groups. Once selected, Active Roles will keep the MU membership dynamically up-to-date: if new members are added to the selected groups, Active Roles will also include them in the MU; and likewise, members removed from the included groups will also be removed from the MU.
Exclude Explicitly	Excludes the AD or Azure AD object you select in the MU.

Rule name	Description
	Once selected, Active Roles will keep the objects excluded from the MU even if they are updated, renamed, or moved elsewhere within your organization directory. NOTE: Consider the following when selecting this membership rule: • The **Exclude Explicitly** rule takes precedence over all other membership rule types. Because of this, Active Roles will exclude the objects specified with this rule, even if another rule specifies that Active Roles must include them in the MU. • This rule excludes only objects that match one of the inclusion rules of the MU.
Exclude by Query	Lets you define a custom query that the AD or Azure AD objects must match to be excluded from the MU. Once configured, Active Roles will automatically exclude objects that meet the query conditions. The query editor works and functions the same way as it does when configuring an **Include by Query** rule, and also shares the same limitations listed there. NOTE: This rule excludes only objects that match one of the inclusion rules of the MU.
Exclude Group Members	Excludes the members of the selected AD or Azure AD groups. Once selected, Active Roles will keep the MU membership dynamically up-to-date: if new members are added to any of the selected groups, Active Roles will exclude them from the MU. Likewise, if a member is removed from all specified groups, Active Roles will add them to the MU, provided that the member meets a configured inclusion rule. NOTE: This rule excludes only objects that match one of the inclusion rules of the MU.
Retain Deprovisioned	Configures the MU to also include and keep deprovisioned objects that meet the membership rules. If this rule is not selected, Active Roles automatically removes deprovisioned objects from the MU.

NOTE: The exclusion rules affect only objects that match one of the inclusion rules configured for the MU.

For example, if a container is explicitly included in an MU, then all objects held in that container are also included in the MU. However, you cannot exclude any of those objects themselves with exclusion rules, as it is their container that meets

> the inclusion rules in this case. To exclude the objects of the container, you must configure an exclusion rule for the container instead.

6. Configure the selected membership rule:

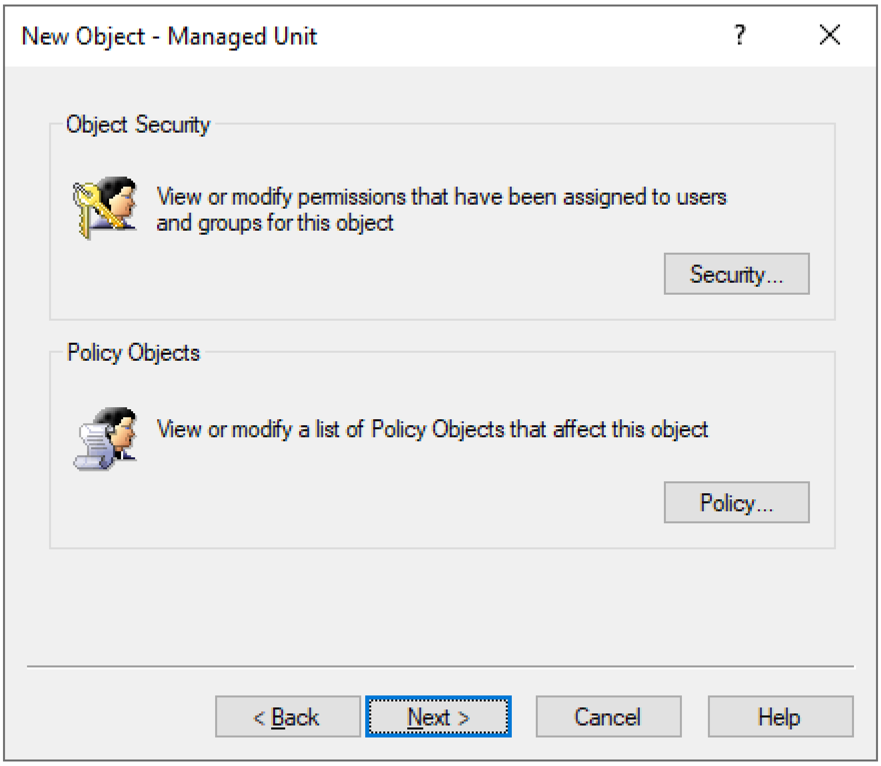
   - If you selected the **Include Explicitly** or **Exclude Explicitly** rule type, the **Select Objects** dialog appears. Select the objects you want to include or exclude from the MU, click **Add**, and then click **OK**.
   - If you selected the **Include Group Members** or **Exclude Group Members** rule type, the **Select Objects** dialog appears, listing the available groups. Select the AD or Azure AD groups you want to include, click **Add**, and then click **OK**. All members of the selected groups will be included or excluded from the MU.
   - If you selected the **Include by Query** or **Exclude by Query** rule type, the **Create Membership Rule** dialog appears. Use the dialog to configure your inclusion or exclusion rule.

7. (Optional) To configure additional rules, click **Add** again.

   > NOTE: If you add several membership rules to an MU, Active Roles runs them in the order you configured them. If some of the configured rules conflict with each other, Active Roles resolves the conflict by prioritizing the configured Exclude rules over the configured Include rules.

8. Once you finished adding all membership rules, click **Next**.
9. (Optional) In the **Object Security / Policy Objects** step, specify the permissions and policy objects related to the configured MU.

Figure 5: New Object - Managed Unit wizard – Access Template and Policy Object links


New Object - Managed Unit
?
Object Security
View or modify permissions that have been assigned to users and groups for this object
Security...
Policy Objects
View or modify a list of Policy Objects that affect this object
Policy...
< Back
Next >
Cancel
Help

- For more information on object security permissions, see Applying Access Templates.
- For more information on Policy Objects, see Applying Policy Objects.

10. To finish configuring the MU, click **Next** and **Finish**.

Modifying Managed Unit properties

You can modify the settings of a Managed Unit with the **Properties** window of the Active Roles Console.

To modify properties of a Managed Unit

1. In the Active Roles Console, on the **Console tree**, navigate to **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 to view or modify properties of the Managed Unit.
4. When finished, click **OK**.

NOTE: Consider the following when modifying Managed Unit properties:

- 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 later 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 Modifying policy settings on a Managed Unit.
- 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 Modifying permission settings on a Managed Unit.

Modifying permission settings on a Managed Unit

You can modify the permission settings of a Managed Unit with the **Delegate Control** window of the Active Roles Console.

To modify permission settings on a Managed Unit

1. In the Active Roles Console, on the **Console tree**, navigate to **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, 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.
   - 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. To do so, select 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.

NOTE: Consider the following when modifying permission settings on a Managed Unit:

- The **Active Roles Security** dialog 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. To do so, select the Managed Unit, then on the **Active Roles Security** tab in the advanced details pane, right-click an Access Template link or a blank area and use the commands on the shortcut menu. For information about the advanced details pane, see *Advanced pane* in the *Active Roles Feature Guide*.

Modifying policy settings on a Managed Unit

You can modify the policy settings of a Managed Unit with the Enforce Policy dialog of the Active Roles Console.

To modify policy settings on a Managed Unit

1. In the Active Roles Console, on the **Console tree**, navigate to **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, 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. To close the **Active Roles Policy** dialog, click **OK**.

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, see *Advanced pane* in the *Active Roles Feature Guide*.

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. You can display and customize the list of members in the Active Roles Console.

To display the members of a Managed Unit

1. In the Active Roles Console, on the **Console tree**, navigate to **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.

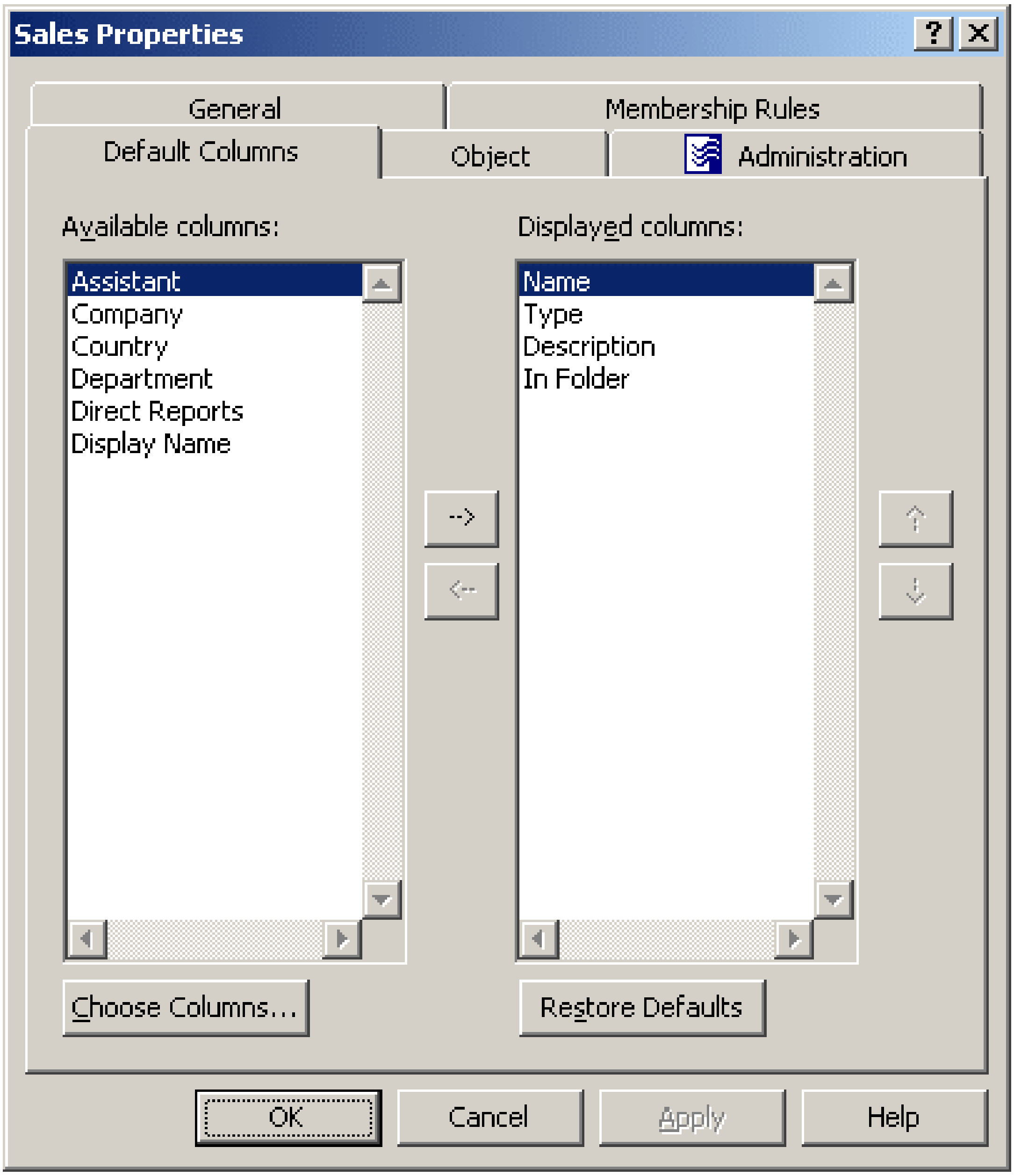
To customize the list of Managed Unit members in the details pane

1. Right-click the Managed Unit, and click **Properties**.
2. In the **Properties** dialog, 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: Consider the following when displaying members of a Managed Unit:

- For each Managed Unit, you can 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, you can select columns and, if necessary, modify column names.
- For your changes to the **Displayed columns** list to take effect, the details pane must be refreshed. To do so, right-click **Managed Units** in the **Console tree** and click **Refresh**.

Figure 6: Managed Unit - Preset columns



Available columns:

- Assistant
- Company
- Country
- Department
- Direct Reports
- Display Name

Displayed columns:

- Name
- Type
- Description
- In Folder

Adding membership rules to a Managed Unit

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

To add a membership rule to a Managed Unit

1. In the Active Roles Console, on the **Console tree**, navigate to **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.
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, the **Create Membership Rule** dialog is displayed. Otherwise (except for the **Retain Deprovisioned** rule type), the **Select Objects** dialog is displayed.
5. Complete the **Create Membership Rule** or **Select Objects** dialog by following the instructions that are given later in this topic.
6. Click **OK** to close the **Properties** dialog.

To complete the Create Membership Rule dialog

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. (Optional) To view a list of objects that match the criteria you have defined, click **Preview Rule**.
5. Click **Add Rule**.

To complete the Select Object dialog

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, 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: Consider the following when adding membership rules to a Managed Unit:
>
> - 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 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, see *Finding objects* in the *Active Roles Console User Guide*.
> - 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 *Building a custom search* and *Using advanced search options* in the *Active Roles Console User Guide*.

Modifying membership rules of a Managed Unit

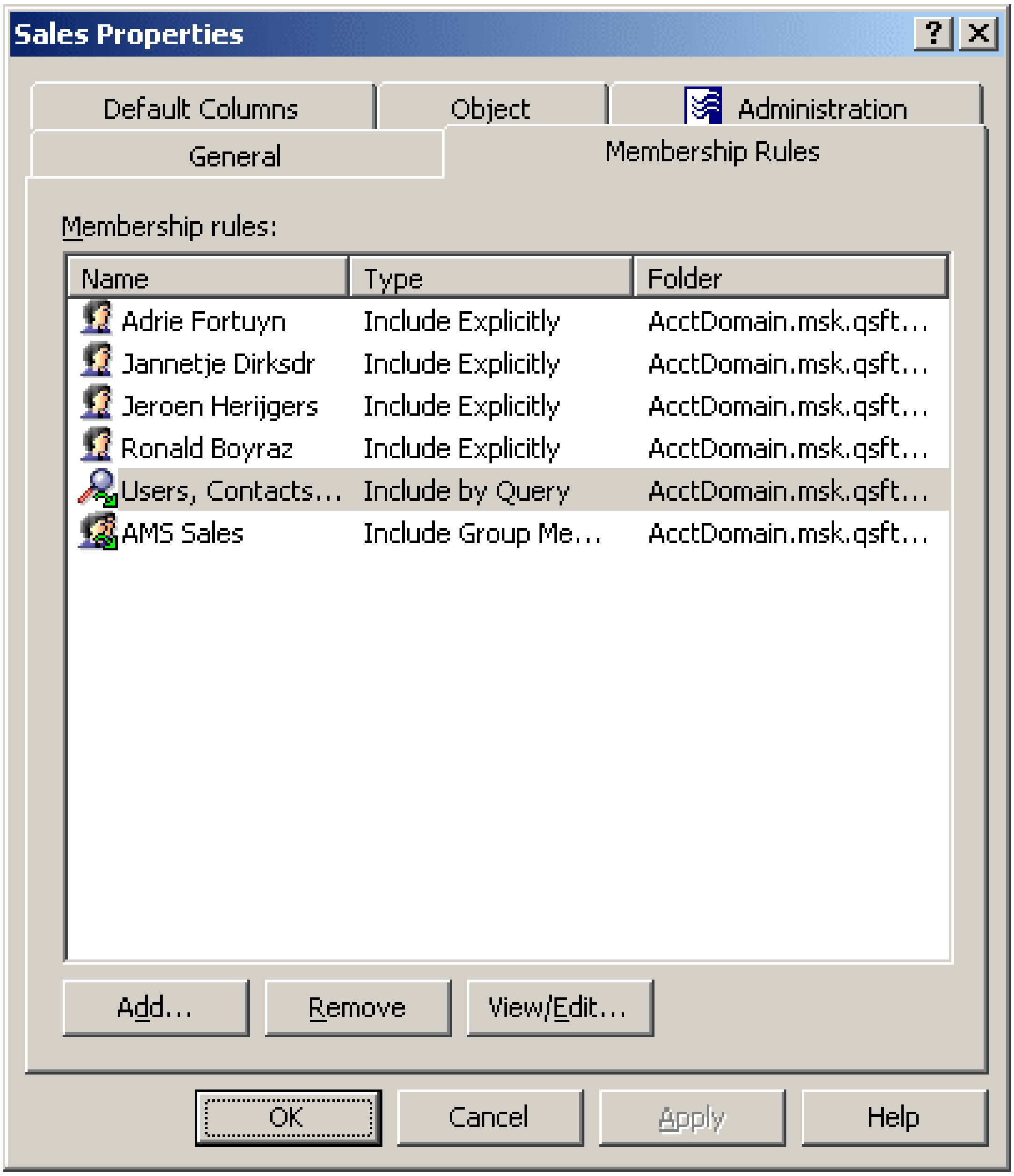
You can modify the existing membership rules of a Managed Unit with the **Properties** window of the Active Roles Console.

To modify membership rules of a Managed Unit

1. In the Active Roles Console, on the **Console tree**, navigate to **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 **View/Edit**.

NOTE: 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.

Figure 7: Managed Unit - Modifying membership rules



Membership rules:

Name	Type	Folder
Adrie Fortuyn	Include Explicitly	AcctDomain.msk.qsft...
Jannetje Dirksdr	Include Explicitly	AcctDomain.msk.qsft...
Jeroen Herijgers	Include Explicitly	AcctDomain.msk.qsft...
Ronald Boyraz	Include Explicitly	AcctDomain.msk.qsft...
Users, Contacts...	Include by Query	AcctDomain.msk.qsft...
AMS Sales	Include Group Me...	AcctDomain.msk.qsft...

Removing membership rules from a Managed Unit

You can remove existing membership rules from a Managed Unit via the **Membership Rules** settings of the Active Roles Console.

To remove a membership rule from a Managed Unit

1. In the Active Roles Console, on the **Console tree**, navigate to **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, then click **Remove**.

Including a member to a Managed Unit

You can add members to a Managed Unit by configuring one or more membership rules in the Active Roles Console.

To include a member to a Managed Unit

1. In the Active Roles Console, on the **Console tree**, navigate to **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.
4. In the **Membership Rule Type** dialog, click **Include Explicitly**, then click **OK**. The **Select Objects** dialog appears.
5. Use the **Select Objects** dialog to locate and select the object (or objects) you want to explicitly include in the Managed Unit.

   For instructions on how to configure a membership rule, see Adding membership rules to a Managed Unit.
6. To close the **Properties** dialog, click **OK**.

Excluding a member from a Managed Unit

You can exclude specific members from a Managed Unit with the Active Roles Console.

To exclude a member to a Managed Unit

1. In the Active Roles Console, on the **Console tree**, navigate to **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.
4. In the **Membership Rule Type** dialog, click **Exclude Explicitly**, and then click **OK**. The **Select Objects** dialog appears.
5. Use the **Select Objects** dialog to locate and select the object (or objects) you want to explicitly exclude in the Managed Unit.

   For instructions on how to configure a membership rule, see Adding membership rules to a Managed Unit.
6. To close the **Properties** dialog, click **OK**.

Adding group members to a Managed Unit

You can add group members to a Managed Unit with the Active Roles Console.

To add group members to a Managed Unit

1. In the Active Roles Console, on the **Console tree**, navigate to **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.
4. In the **Membership Rule Type** dialog, click **Include Group Members**, and then click **OK**. The **Select Objects** dialog appears.
5. Use the **Select Objects** dialog to locate and select the group (or groups) whose members you want to be included in the Managed Unit.

   For instructions on how to configure a membership rule, see Adding membership rules to a Managed Unit.
6. To close the **Properties** dialog, click **OK**.

Removing group members from a Managed Unit

You can remove group members from Managed Units by changing their membership rule settings in the Active Roles Console.

To remove group members from a Managed Unit

1. In the Active Roles Console, on the **Console tree**, navigate to **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.
4. In the **Membership Rule Type** dialog, click **Exclude Group Members**, then click **OK**. The **Select Objects** dialog appears.
5. Use the **Select Objects** dialog to locate and select the group (or groups) whose members you want to be excluded from the Managed Unit.

   For instructions on how to configure a membership rule, see Adding membership rules to a Managed Unit.
6. To close the **Properties** dialog, click **OK**.

Copying a Managed Unit

With the Active Roles Console, you can create copies of Managed Units. This feature helps you reuse existing Managed Units.

To copy a Managed Unit

1. In the Active Roles Console, on the **Console tree**, navigate to **Configuration** > **Managed Units**.
2. Under **Managed Units**, locate the Managed Unit you want to copy.
3. Right-click the Managed Unit, then click **Copy**. The **Copy Object - Managed Unit** wizard starts.
4. On the first page of the wizard, do the following, then click **Next**:
   a. In the **Name** box, enter a name for the Managed Unit.
   b. (Optional) In the **Description** box, enter any 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 Adding membership rules to a Managed Unit.

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 Modifying permission settings on a Managed Unit and Modifying policy settings on a Managed Unit.
8. Click **Next**, 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.

To export Managed Units, select them, right-click the selection, and select **All Tasks** > **Export**. In the **Export Objects** dialog, 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, then click **Import**. In the **Import Directory Objects** dialog, select the file to which the Managed Units were exported, and click **Open**.

> 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.

Renaming a Managed Unit

You can rename a Managed Unit with the **Rename** setting of the Active Roles Console.

To rename a Managed Unit

1. In the Active Roles Console, on the **Console tree**, navigate to **Configuration** > **Managed Units**.
2. Under **Managed Units**, locate the Managed Unit you want to rename, right-click it, and click **Rename**.
3. Enter a new name, then press **Enter**.

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

Deleting a Managed Unit

You can delete existing Managed Units with the Active Roles Console.

To delete a Managed Unit

1. In the Active Roles Console, on the **Console tree**, navigate to **Configuration** > **Managed Units**.
2. Under **Managed Units**, locate the Managed Unit you want to delete, right-click it, and 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.

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.

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 membership rules to a Managed Unit.

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.

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 **Add** 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.

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

Configuring role-based administration

To provide additional flexibility beyond the system-provided Active Directory Users and Computers tool in delegating administrative responsibilities, Active Roles supports:

- Consolidating permissions into customizable administrative roles, known as Access Templates.

  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.

- Claims-based authorization rules (known as "Access Rules") to allow or deny access to Active Directory objects.

  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 (the Active Directory user and computer properties) in the Active Roles authorization model.

> TIP: For more information on these role-based administration features, see *Access Templates* and *Access Rules* in the *Active Roles Feature Guide*.

Access Template management tasks

This section guides you through the Active Roles Console to manage Access Templates.

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**, then selecting one of these containers in the **Console tree**:

- Active Directory
- AD LDS (ADAM)
- Azure
- Builtin
- Computer Resources
- Configuration
- Exchange
- Skype for Business Server
- Starling
- User Interfaces
- User Self-management

For more information on predefined Access Templates and their recommended use, see the *Active Roles Built-in Access Templates Reference Guide*.

Creating an Access Template

You can create a new Access Template for role-based delegation with the Active Roles Console.

> NOTE: Creating and managing Access Templates is done with the **Add Permission Entries Wizard**. For the detailed description of the wizard, see *Add Permission Entries Wizard* in the *Active Roles Feature Guide*.

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.

   > NOTE: Consider the following when creating an Access Template:
   >
   > - You can create a new folder by right-clicking **Access Templates** and selecting **New** > **New Access Template Container**. Similarly, you can create a sub-folder in a folder by right-clicking the folder, and selecting **New** > **Access Template Container**.
   > - One Identity recommends storing custom Access Templates in a separate container.

2. To start the **New Object - Access Template** wizard, right-click the folder, and select **New** > **Access Template**.

3. On the first page of the wizard, do the following, then click **Next**:

    a. In the **Name** box, enter a name for the Access Template.

    b. (Optional) In the **Description** box, type any information about the Access Template.

4. On the second page of the wizard, configure the list of permission entries, then click **Next**.

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. In the Active Roles Console, select the Access Template you want to modify.

2. To start the **Add Permission Entries Wizard**, on the page that displays a list of permission entries included in the Access Template, click **Add**.

3. 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**.

4. Click **Next**.

5. 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.

6. 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.

7. 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.

8. Continue configuring the option you selected in Step 4. then, to add the permission entry to the Access Template, click **Finish**:
   - 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. In the Active Roles Console, select the Access Template you want to modify.
2. 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. Then, to display the **Modify Permission Entry** dialog, click **View/Edit**.
3. Examine the **Apply Onto** tab in the **Modify Permission Entry** dialog. On this tab, you can view or modify the same settings as on the first page of the **Add Permission Entries Wizard**.
4. Examine the **Permissions** tab in the **Modify Permission Entry** dialog. This tab provides the same options as the second page of the **Add Permission Entries Wizard**. 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**.
5. (Optional) 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.
6. If **Object property access** is selected on the **Permissions** tab, use the **Object Properties** tab in the **Modify Permission Entry** dialog to view or modify the

settings that determine which properties are controlled by this permission entry.

7. If **Creation/Deletion of child objects** is selected on the **Permissions** tab, use the **Object Classes** tab in the **Modify Permission Entry** dialog 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. In the Active Roles Console, select the Access Template you want to modify.
2. 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**.
3. To confirm deleting the permission entry, click **Yes**.

Applying Access Templates

You can apply ATs to an AD object with the **Delegation of Control Wizard**. To start the wizard, navigate to either:

- The AT you want to apply on an AD object. When you start the **Delegation of Control Wizard** this way, you can select the securable AD objects for which the access is granted, and the trustees who receive the access to those securable objects.

  For the steps of this procedure, see Applying an Access Template directly.

- The securable AD object (container, Managed Unit or leaf object) whose access and administration permissions you want to configure. When you start the **Delegation of Control Wizard** this way, you can select the trustees who receive the access to the securable object and the ATs defining the permissions of the trustees to the securable object.

  For the steps of this procedure, see Applying Access Templates on a securable object.

- The trustee for which you want to assign permissions. When you start the **Delegation of Control Wizard** this way, you can select the securable AD object to which the trustee will receive access and the ATs defining the permissions of the trustee to the securable object.

  For the steps of this procedure, see Applying Access Templates on a user or group.

NOTE: ATs support propagating their permission settings for the child objects of the securable objects too.

TIP: For more technical details about how Active Roles applies permissions with Access Templates, see *Applying permissions with Access Template* in the *Active Roles Feature Guide*.

Applying an Access Template directly

You can configure permissions for a trustee to a securable Active Directory (AD) object via an Access Template (AT) by selecting the AT directly in the Active Roles Console.

To apply an Access Template on a trustee or trustees

1. In the Active Roles Console, in the Active Directory (AD) tree, navigate to **Configuration** > **Access Templates**.
2. Right-click the AT you want to assign to a trustee (or trustees), then click **Links**.

   > TIP: For more information on the ATs, see the **Description** of the AT or the *Active Roles Built-in Access Templates Reference Guide* document.

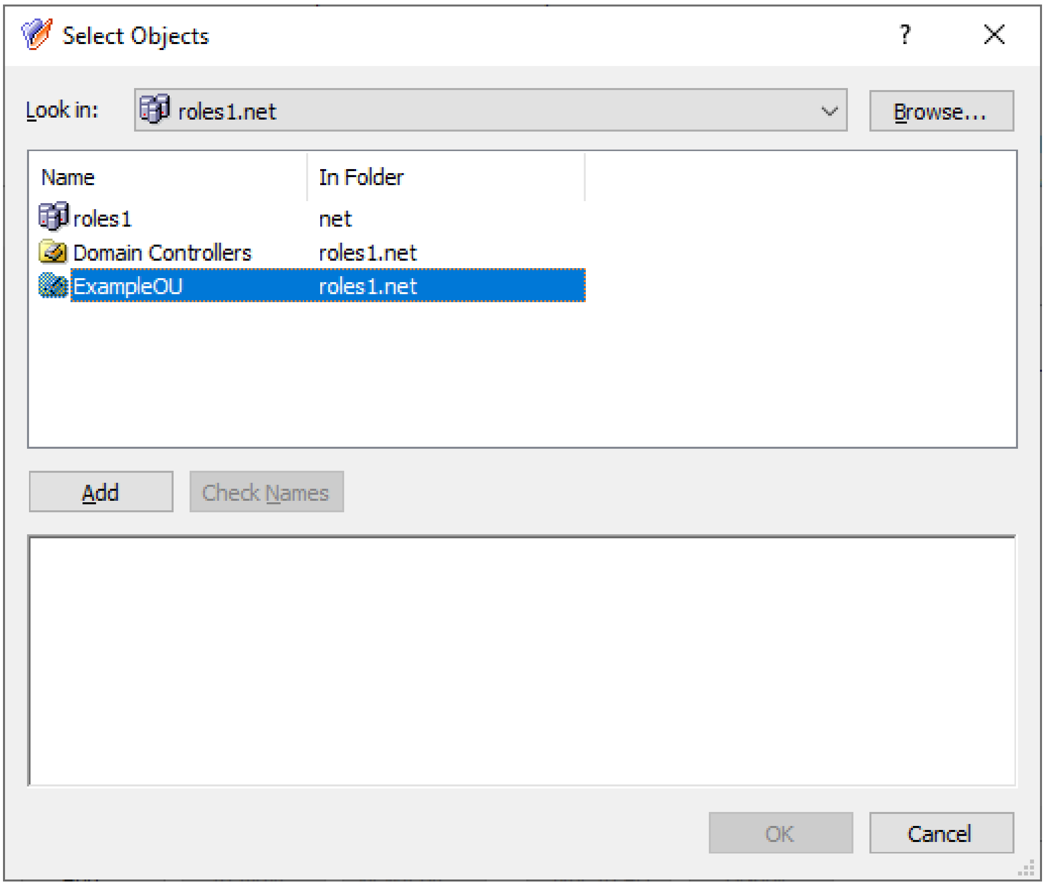
3. In the **Links** dialog, to start the **Delegation of Control Wizard**, click **Add**. Click **Next** on the **Welcome** page, when it appears.
4. In the **Objects** step, specify the securable objects that you want to add to the scope of the AT.

   - To specify a new securable object or objects, click **Add**. Then, in the **Select Objects** window, locate and select the securable objects you want to add to the scope of the AT, and click **Add**.

     Once you finalized the list, to close the **Select Objects** window and apply your selection, click **OK**.

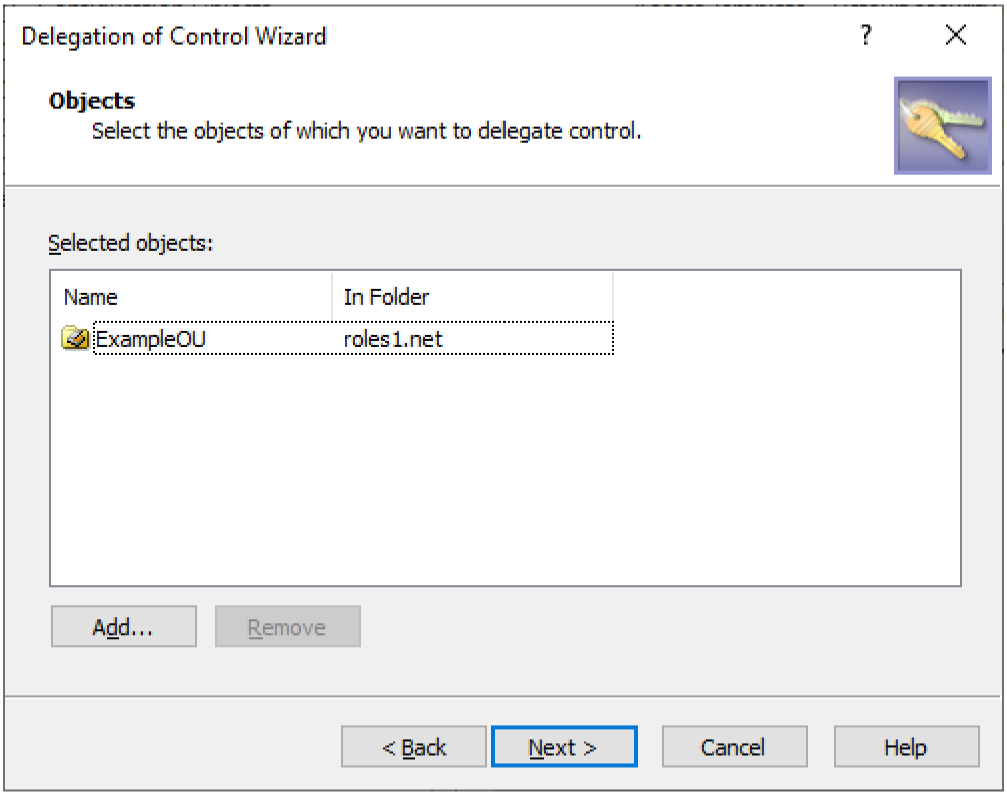
     > TIP: If no securable objects appear in the window, use the **Click here to display objects** link.

Figure 8: Delegation of Control Wizard – Select objects window when specifying securable objects



Name	In Folder
roles1	net
Domain Controllers	roles1.net
ExampleOU	roles1.net

Figure 9: Delegation of Control Wizard – Selecting securable objects



- To remove securable objects added earlier to the scope of the AT, select them in the **Objects** step, and click **Remove**.

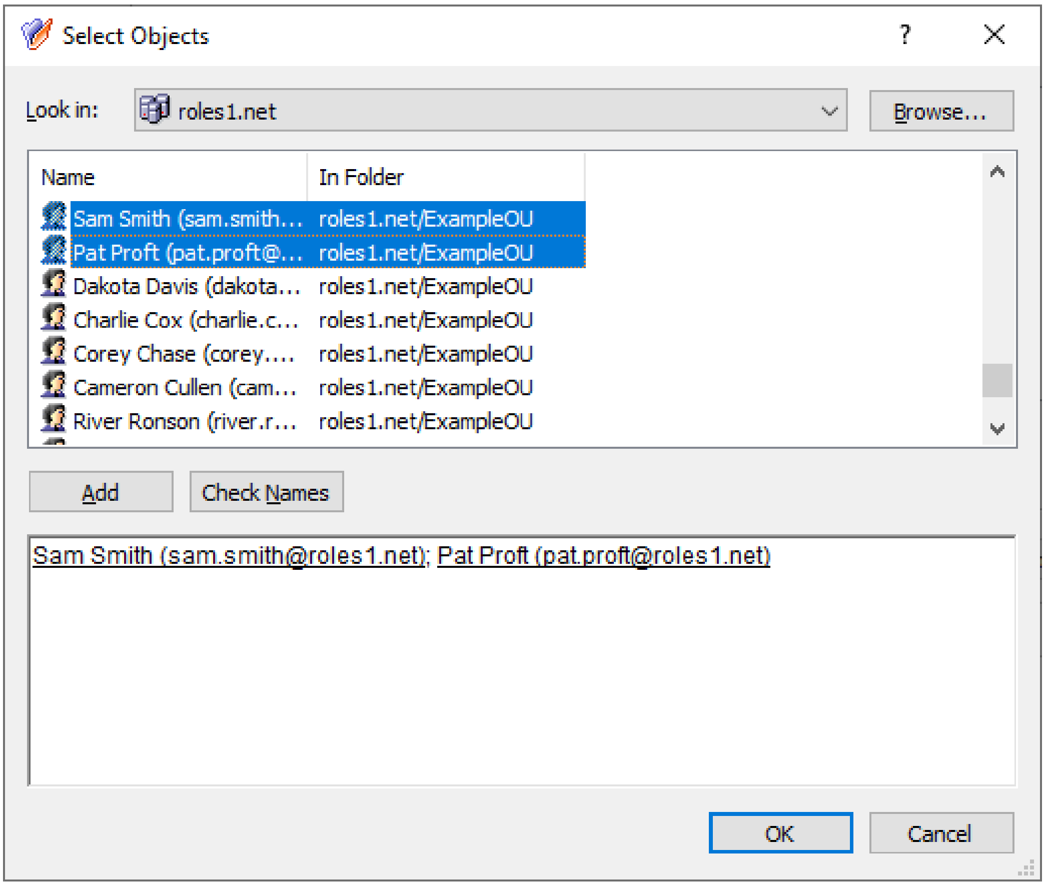
To continue, click **Next**.

5. In the **Users or Groups** step, specify the trustee(s) for which you want to grant the permissions of the AT.

    - To specify a new trustee or new trustees, click **Add**. Then, in the **Select Objects** window, locate and select the users or groups you want to add to the scope of the AT, and click **Add**. Once you finalized the list, to close the **Select Objects** window and apply your selection, click **OK**.

        TIP: If no users or groups appear in the window, use the **Click here to display objects** link.

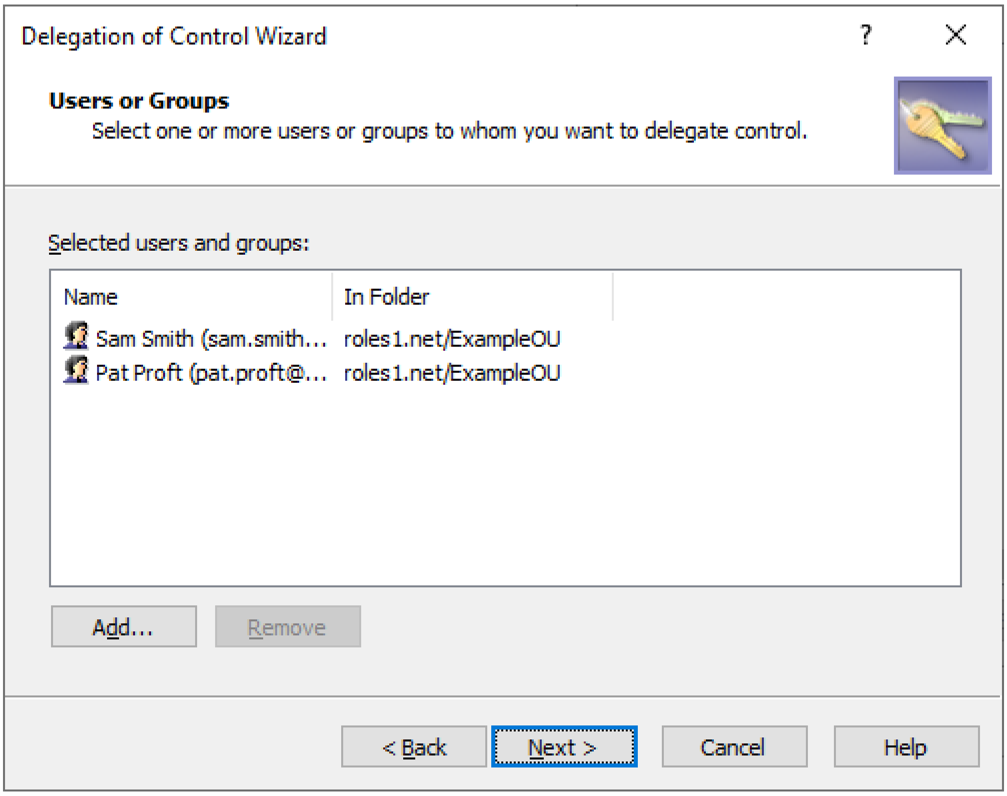
Figure 10: Delegation of Control Wizard – Select Objects window when specifying trustees



Name	In Folder
Sam Smith (sam.smith...	roles1.net/ExampleOU
Pat Proft (pat.proft@...	roles1.net/ExampleOU
Dakota Davis (dakota...	roles1.net/ExampleOU
Charlie Cox (charlie.c...	roles1.net/ExampleOU
Corey Chase (corey....	roles1.net/ExampleOU
Cameron Cullen (cam...	roles1.net/ExampleOU
River Ronson (river.r...	roles1.net/ExampleOU

Sam Smith (sam.smith@roles1.net); Pat Proft (pat.proft@roles1.net)

Figure 11: Delegation of Control Wizard – Selecting trustees



Delegation of Control Wizard

Users or Groups

Select one or more users or groups to whom you want to delegate control.

Selected users and groups:

Name	In Folder
Sam Smith (sam.smith...	roles1.net/ExampleOU
Pat Proft (pat.proft@...	roles1.net/ExampleOU

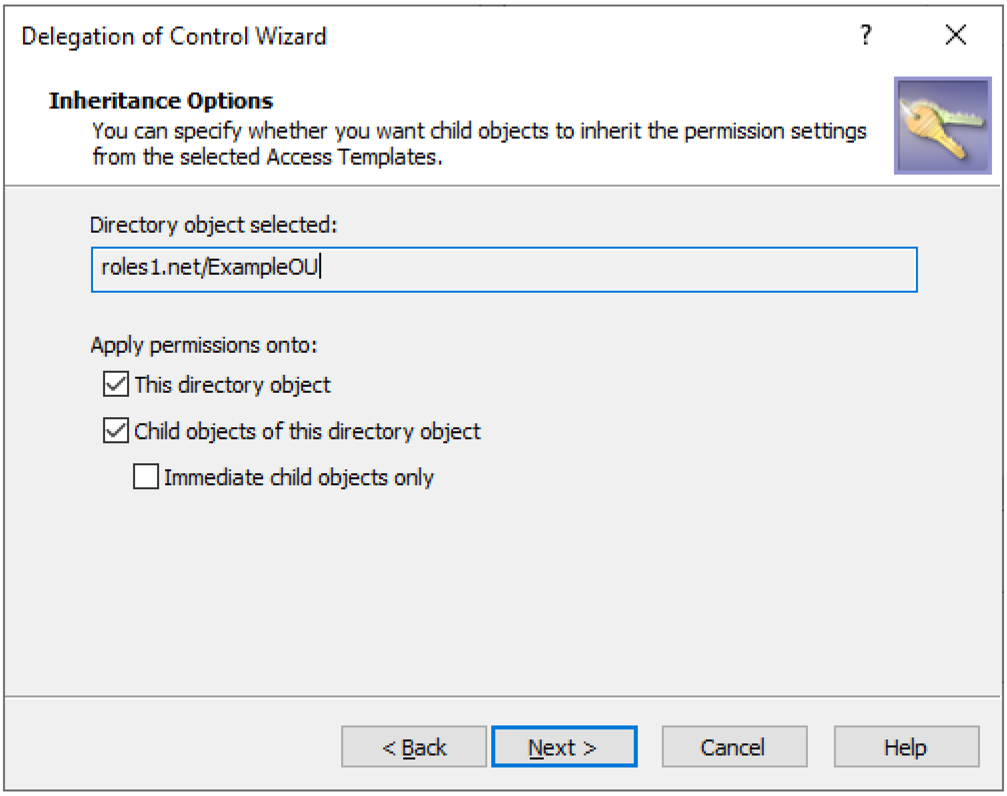
- To remove existing trustees added earlier to the scope of the AT, select them in the **Users or Groups** step, and click **Remove**.

To continue, click **Next**.

6. In the **Inheritance Options** step, specify with the **Apply permissions onto** setting the scope of securable objects to which Active Roles applies the permissions of the AT:

   - **This directory object**: Trustees receive the AT permissions only to the selected securable object.
   - **Child objects of this directory object**: Trustees receive the AT permissions to the children of the securable object. To limit the granted permissions only to the direct children of the object, select **Immediate child objects only** as well.

Figure 12: Delegation of Control Wizard- Inheritance Options



Delegation of Control Wizard

Inheritance Options

You can specify whether you want child objects to inherit the permission settings from the selected Access Templates.

Directory object selected:

roles1.net/ExampleOU

Apply permissions onto:

☑ This directory object

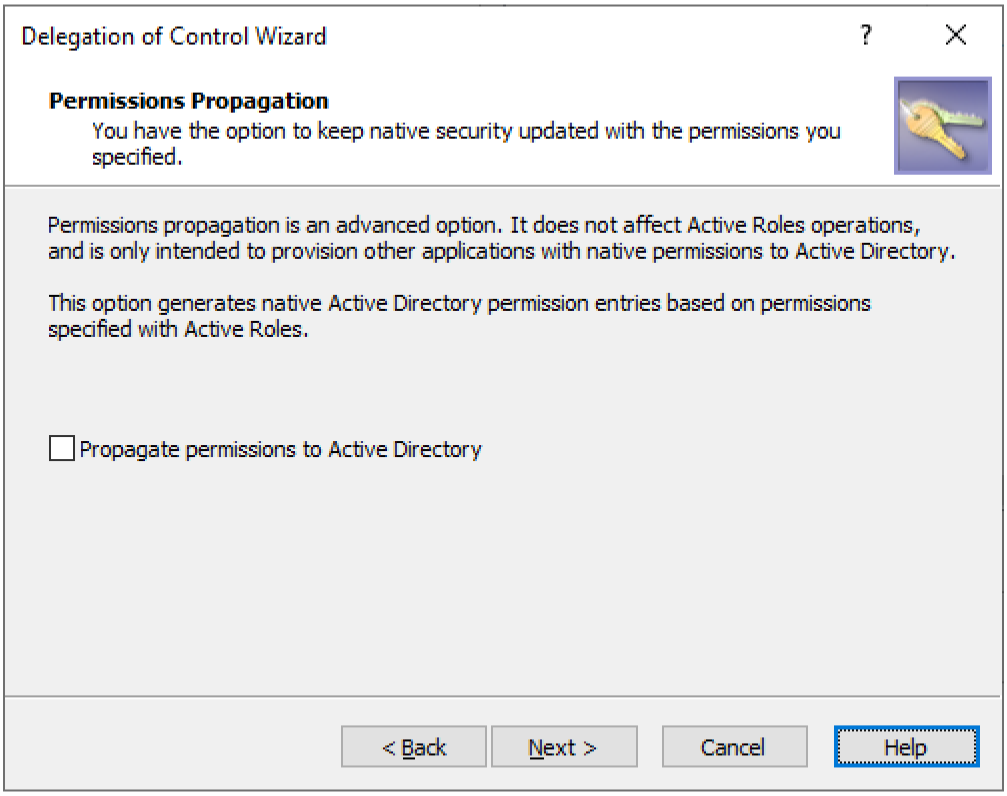
☑ Child objects of this directory object

□ Immediate child objects only

To continue, click **Next**.

7. In the **Permissions Propagation** step, to synchronize the configured permission settings to the native Active Directory (AD) access controls, select **Propagate permissions to Active Directory**.

Figure 13: Delegation of Control Wizard – Permissions propagation



Delegation of Control Wizard

Permissions Propagation

You have the option to keep native security updated with the permissions you specified.

Permissions propagation is an advanced option. It does not affect Active Roles operations, and is only intended to provision other applications with native permissions to Active Directory.

This option generates native Active Directory permission entries based on permissions specified with Active Roles.

□ Propagate permissions to Active Directory

Selecting this setting will modify the authorization information of the AD objects with the permission settings defined in Active Roles, providing more flexibility for users and groups that use native AD management tools besides Active Roles.

> IMPORTANT: Selecting this setting will result in trustees keeping their configured permissions outside of the Active Roles environment, with the potential risk of bypassing policies configured and enforced with Active Roles.
>
> Therefore, select this option only if the selected trustees have the required security clearance and/or meet all security guidelines in effect within your organization.

> TIP: Once **Propagate permissions to Active Directory** is selected and configured, you can change this setting at any time with the **Active Roles Security** > **Sync to AD** setting, or with the **Advanced Details** > **Sync to AD** setting. For more information, see Synchronizing permissions to Active Directory.

To continue, click **Next**.

8. To complete the wizard, click **Finish**.

Applying Access Templates on a securable object

You can configure permissions for a trustee (or trustees) to a securable Active Directory (AD) object via Access Templates (ATs) by selecting the securable object in the Active Roles Console.

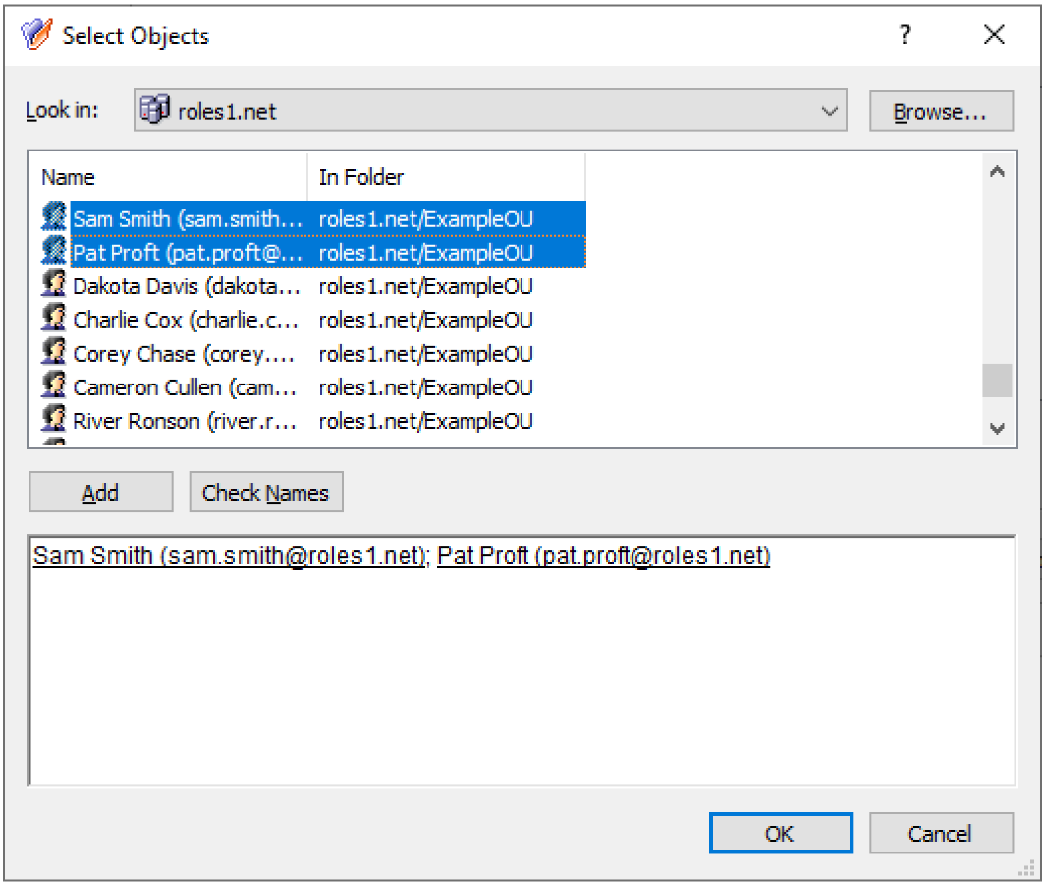
To configure permissions with an Access Template from a securable object

1. In the Active Roles Console, in the Active Directory (AD) tree, navigate to the securable object for which you want to configure an AT.
2. To open the **Delegation of Control Wizard** from the securable object:
   - If the securable object is a container or Managed Unit, right-click the object, then click **Delegate Control** > **Add**.
   - If the securable object is a leaf object, right-click the object and click **Properties**. Then, in the **Properties** window, click **Administration** > **Security** > **Add**.

   When the Welcome screen of the Delegation of Control Wizard appears, click **Next**.
3. In the **Users or Groups** step, specify the trustee(s) for which you want to grant the permissions of the AT.
   - To specify a new trustee or new trustees, click **Add**. Then, in the **Select Objects** window, locate and select the users or groups you want to add to the scope of the AT, and click **Add**. Once you finalized the list, to close the **Select Objects** window and apply your selection, click **OK**.

     > TIP: If no users or groups appear in the window, use the **Click here to display objects** link.

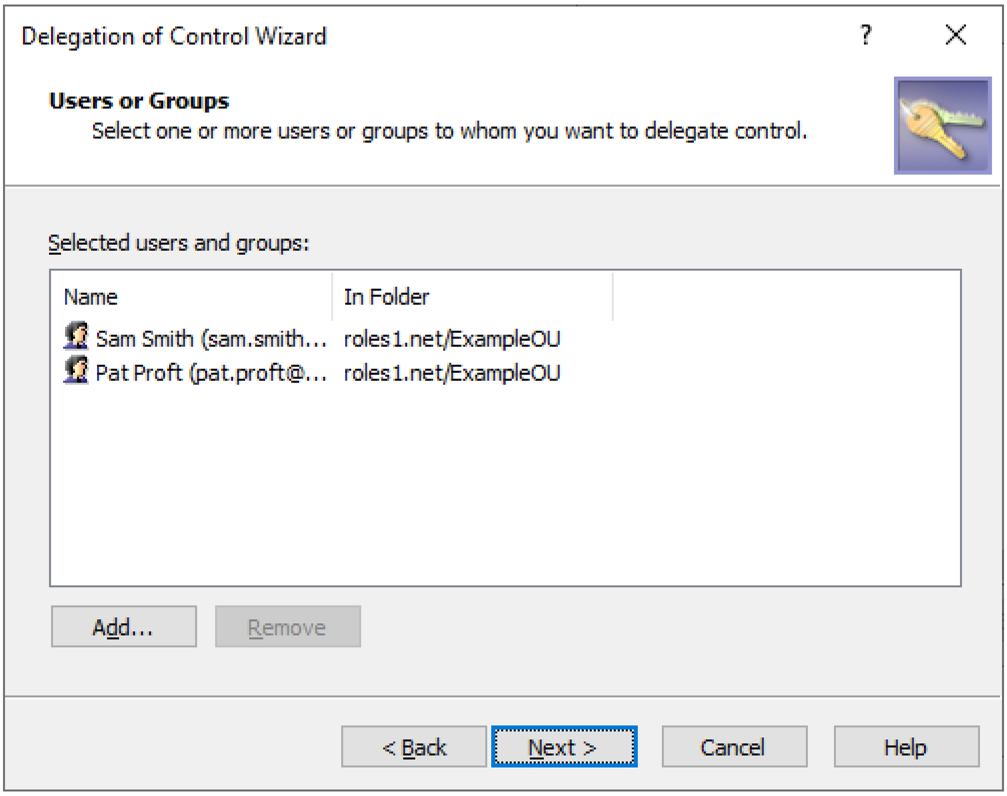
Figure 14: Delegation of Control Wizard – Select Objects window when specifying trustees



Name	In Folder
Sam Smith (sam.smith...	roles1.net/ExampleOU
Pat Proft (pat.proft@...	roles1.net/ExampleOU
Dakota Davis (dakota...	roles1.net/ExampleOU
Charlie Cox (charlie.c...	roles1.net/ExampleOU
Corey Chase (corey....	roles1.net/ExampleOU
Cameron Cullen (cam...	roles1.net/ExampleOU
River Ronson (river.r...	roles1.net/ExampleOU

Sam Smith (sam.smith@roles1.net); Pat Proft (pat.proft@roles1.net)

Figure 15: Delegation of Control Wizard – Selecting trustees


Delegation of Control Wizard

Users or Groups

Select one or more users or groups to whom you want to delegate control.

Selected users and groups:

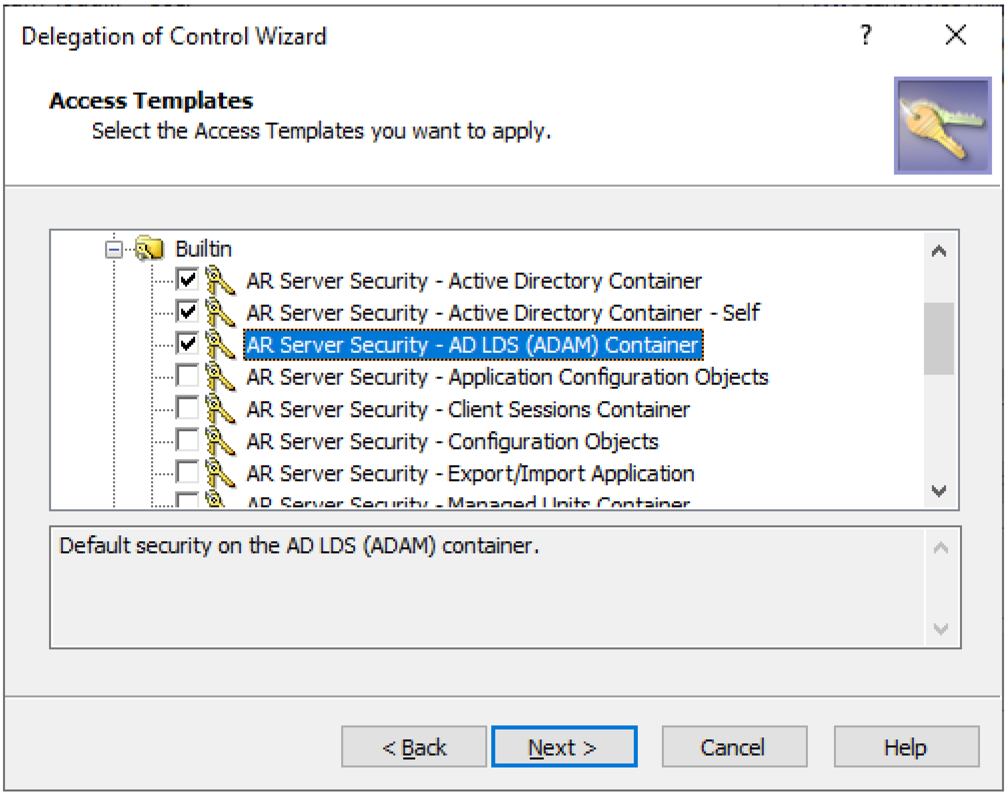
Name	In Folder
Sam Smith (sam.smith...	roles1.net/ExampleOU
Pat Proft (pat.proft@...	roles1.net/ExampleOU

- To remove existing trustees added earlier to the scope of the AT, select them in the **Users or Groups** step, and click **Remove**.

To continue, click **Next**.

4. In the **Access Templates** step, specify the ATs you want to assign to the selected trustees for the configured securable object. Expand the containers of the ATs, then select the AT or ATs you want to apply.

Figure 16: Delegation of Control Wizard – Selecting Access Templates



Delegation of Control Wizard

Access Templates

Select the Access Templates you want to apply.

- Builtin
  - ☑ AR Server Security - Active Directory Container
  - ☑ AR Server Security - Active Directory Container - Self
  - ☑ AR Server Security - AD LDS (ADAM) Container
  - □ AR Server Security - Application Configuration Objects
  - □ AR Server Security - Client Sessions Container
  - □ AR Server Security - Configuration Objects
  - □ AR Server Security - Export/Import Application
  - □ AR Server Security - Managed Units Container

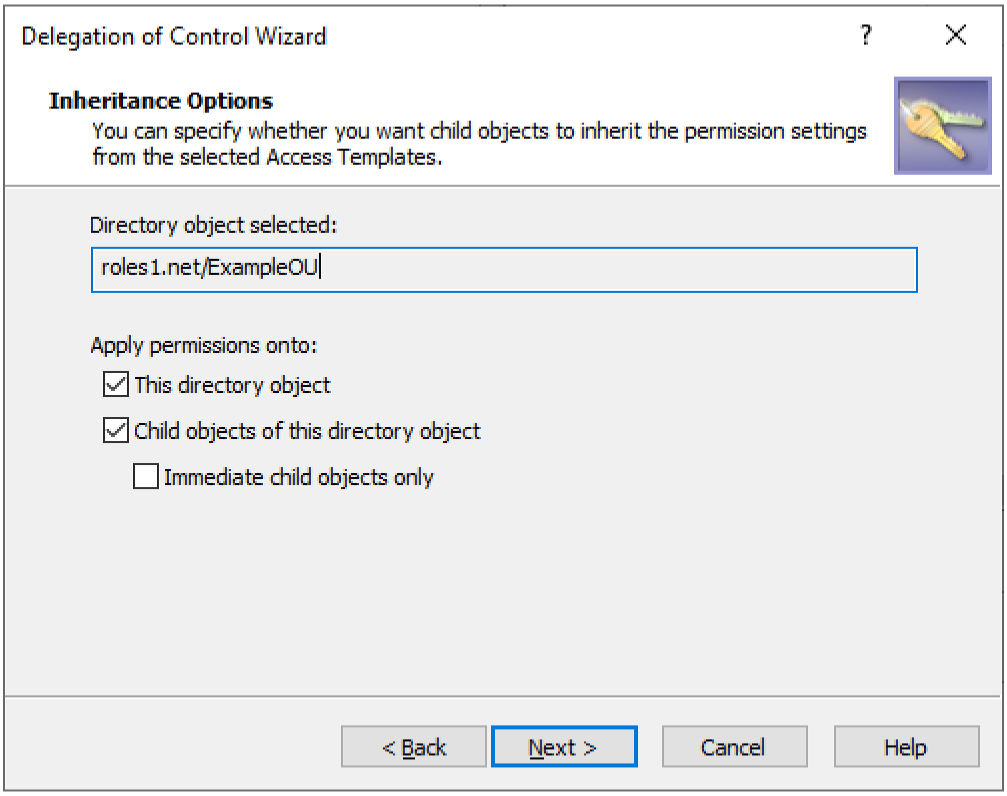
Default security on the AD LDS (ADAM) container.

To continue, click **Next**.

5. In the **Inheritance Options** step, specify with the **Apply permissions onto** setting the scope of securable objects to which Active Roles applies the permissions of the AT:

   - **This directory object**: Trustees receive the AT permissions only to the selected securable object.
   - **Child objects of this directory object**: Trustees receive the AT permissions to the children of the securable object. To limit the granted permissions only to the direct children of the object, select **Immediate child objects only** as well.

Figure 17: Delegation of Control Wizard- Inheritance Options



Delegation of Control Wizard

Inheritance Options

You can specify whether you want child objects to inherit the permission settings from the selected Access Templates.

Directory object selected:

roles1.net/ExampleOU

Apply permissions onto:

☑ This directory object

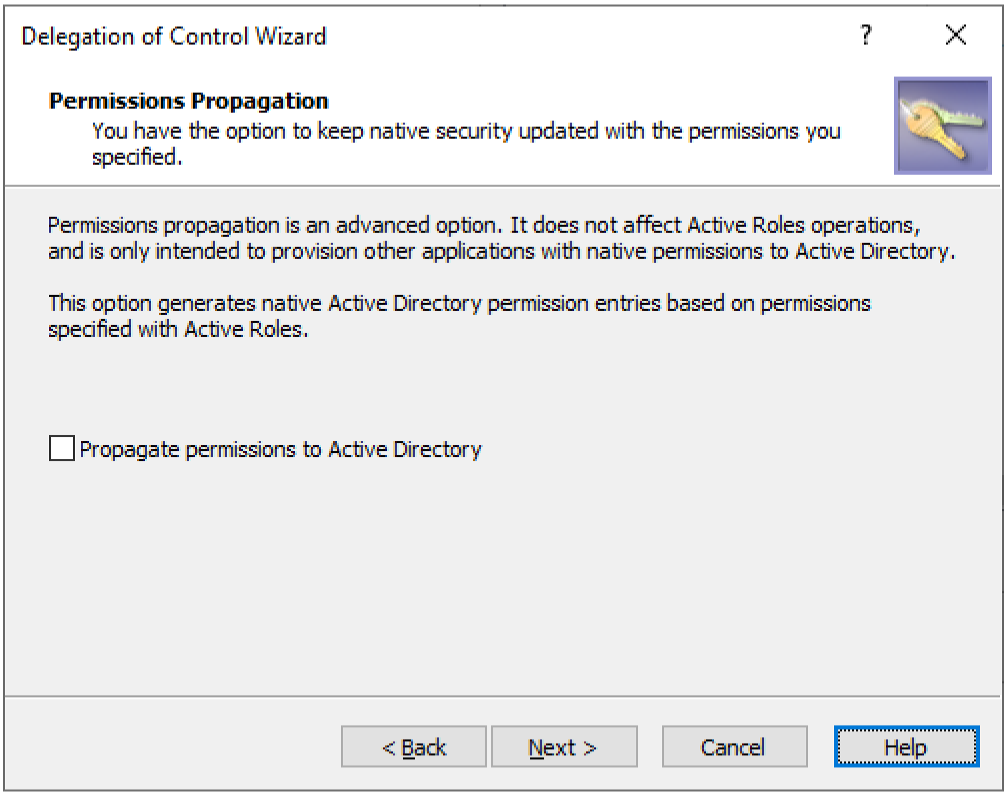
☑ Child objects of this directory object

☐ Immediate child objects only

To continue, click **Next**.

6. In the **Permissions Propagation** step, to synchronize the configured permission settings to the native Active Directory (AD) access controls, select **Propagate permissions to Active Directory**.

Figure 18: Delegation of Control Wizard – Permissions propagation



Delegation of Control Wizard

Permissions Propagation

You have the option to keep native security updated with the permissions you specified.

Permissions propagation is an advanced option. It does not affect Active Roles operations, and is only intended to provision other applications with native permissions to Active Directory.

This option generates native Active Directory permission entries based on permissions specified with Active Roles.

☐ Propagate permissions to Active Directory

Selecting this setting will modify the authorization information of the AD objects with the permission settings defined in Active Roles, providing more flexibility for users and groups that use native AD management tools besides Active Roles.

> IMPORTANT: Selecting this setting will result in trustees keeping their configured permissions outside of the Active Roles environment, with the potential risk of bypassing policies configured and enforced with Active Roles.
>
> Therefore, select this option only if the selected trustees have the required security clearance and/or meet all security guidelines in effect within your organization.

> TIP: Once **Propagate permissions to Active Directory** is selected and configured, you can change this setting at any time with the **Active Roles Security** > **Sync to AD** setting, or with the **Advanced Details** > **Sync to AD** setting. For more information, see Synchronizing permissions to Active Directory.

To continue, click **Next**.

7. To complete the wizard, click **Finish**.

Applying Access Templates on a user or group

You can configure permissions for a trustee (typically a user or group) to a securable Active Directory (AD) object via Access Templates (ATs) by selecting the trustee in the Active Roles Console.

To configure permissions with an Access Template from a trustee

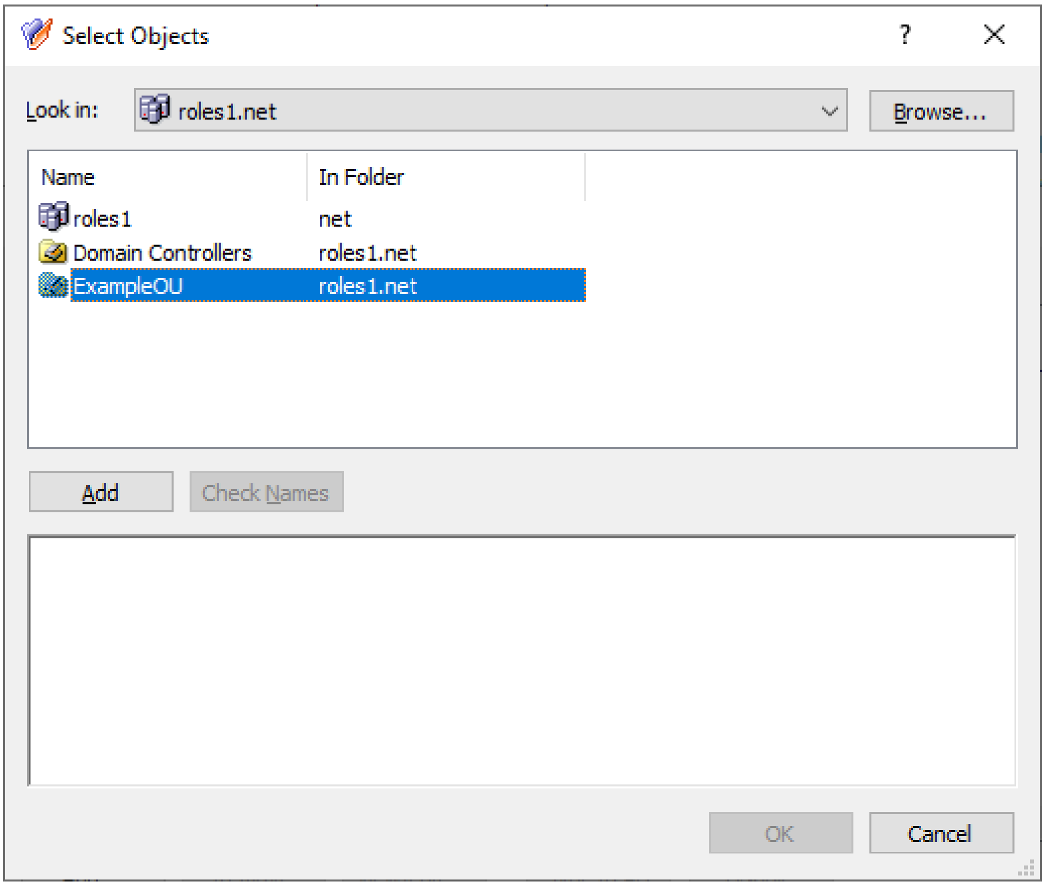
1. In the Active Roles Console, in the Active Directory (AD) tree, navigate to the trustee AD object (such as a user or group) for which you want to configure access with an AT or ATs to a securable object.
2. To open the **Delegation of Control Wizard**, right-click the trustee, then click **Delegated Rights** > **Add**.

   When the Welcome screen appears, click **Next**.
3. In the **Objects** step, specify the securable objects that you want to add to the scope of the AT.
   - To specify a new securable object or objects, click **Add**. Then, in the **Select Objects** window, locate and select the securable objects you want to add to the scope of the AT, and click **Add**.

     Once you finalized the list, to close the **Select Objects** window and apply your selection, click **OK**.

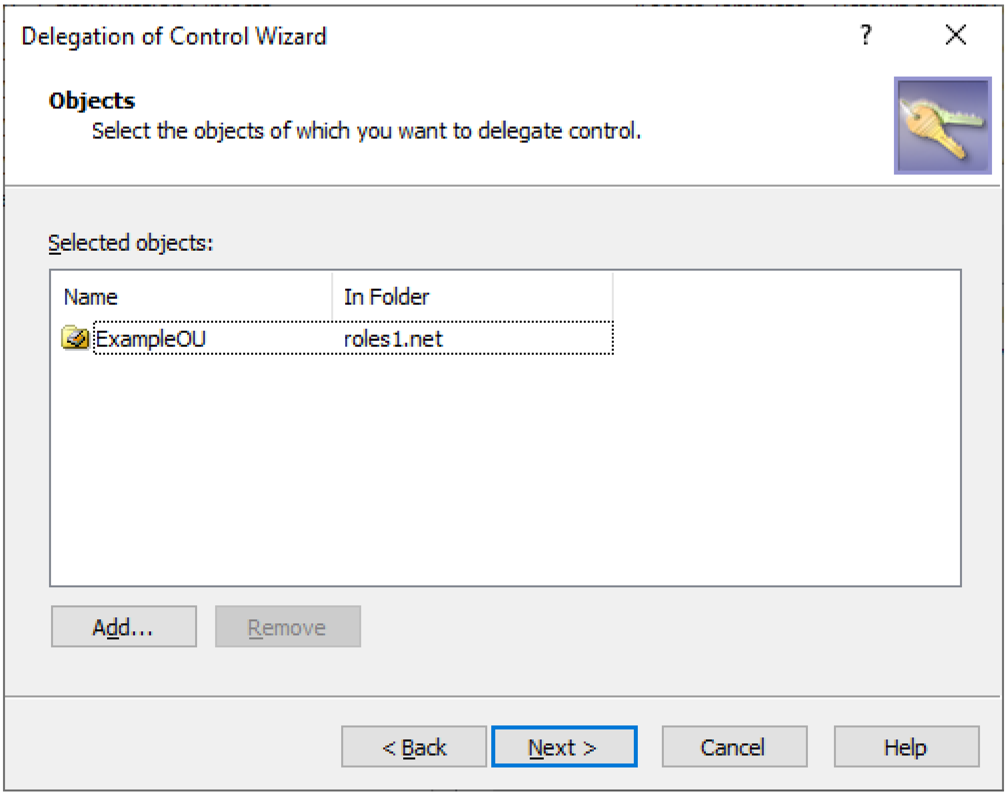
     > TIP: If no securable objects appear in the window, use the **Click here to display objects** link.

Figure 19: Delegation of Control Wizard – Select objects window when specifying securable objects



Name	In Folder
roles1	net
Domain Controllers	roles1.net
ExampleOU	roles1.net

Figure 20: Delegation of Control Wizard – Selecting securable objects



Delegation of Control Wizard

Objects

Select the objects of which you want to delegate control.

Selected objects:

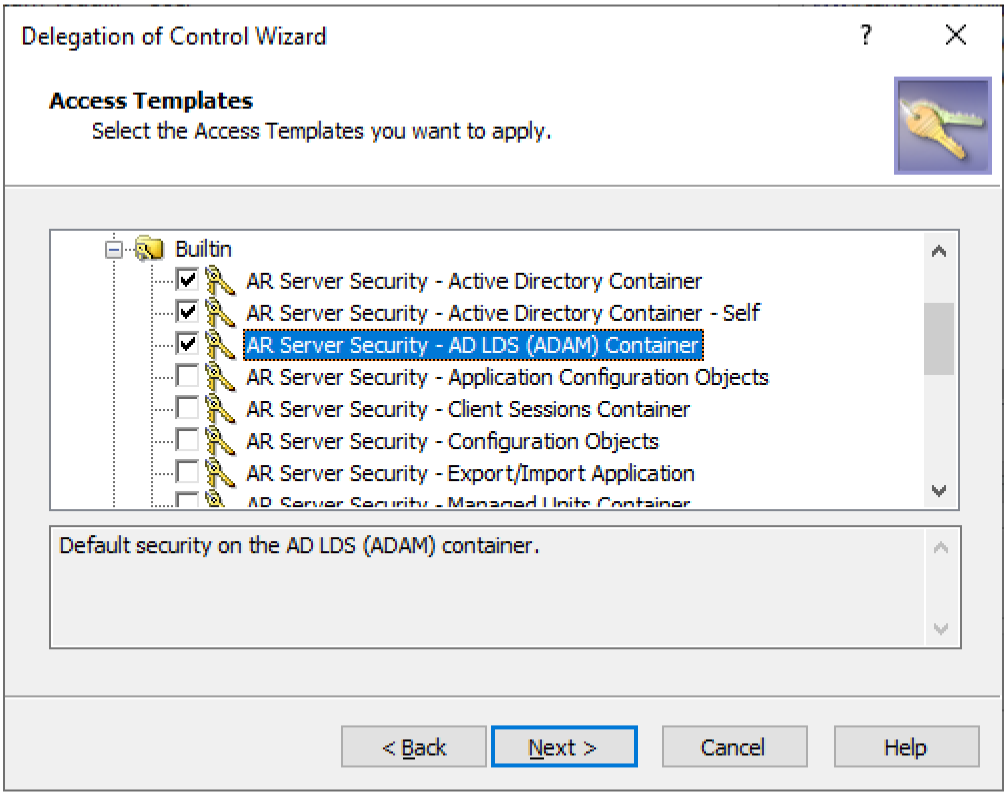
Name	In Folder
ExampleOU	roles1.net

- To remove securable objects added earlier to the scope of the AT, select them in the **Objects** step, and click **Remove**.

To continue, click **Next**.

4. In the **Access Templates** step, specify the ATs you want to assign to the selected trustees for the configured securable object. Expand the containers of the ATs, then select the AT or ATs you want to apply.

Figure 21: Delegation of Control Wizard – Selecting Access Templates



Delegation of Control Wizard

Access Templates

Select the Access Templates you want to apply.

- Builtin
  - ☑ AR Server Security - Active Directory Container
  - ☑ AR Server Security - Active Directory Container - Self
  - ☑ AR Server Security - AD LDS (ADAM) Container
  - □ AR Server Security - Application Configuration Objects
  - □ AR Server Security - Client Sessions Container
  - □ AR Server Security - Configuration Objects
  - □ AR Server Security - Export/Import Application
  - □ AR Server Security - Managed Units Container

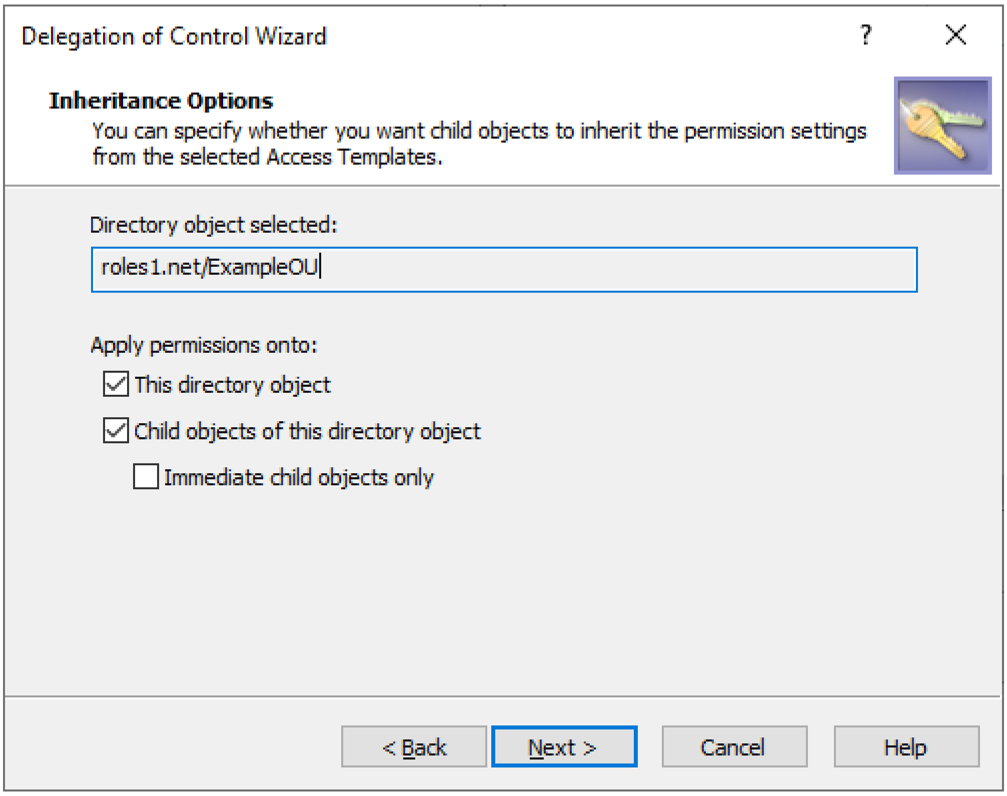
Default security on the AD LDS (ADAM) container.

To continue, click **Next**.

5. In the **Inheritance Options** step, specify with the **Apply permissions onto** setting the scope of securable objects to which Active Roles applies the permissions of the AT:

   - **This directory object**: Trustees receive the AT permissions only to the selected securable object.
   - **Child objects of this directory object**: Trustees receive the AT permissions to the children of the securable object. To limit the granted permissions only to the direct children of the object, select **Immediate child objects only** as well.

Figure 22: Delegation of Control Wizard- Inheritance Options



Delegation of Control Wizard

Inheritance Options

You can specify whether you want child objects to inherit the permission settings from the selected Access Templates.

Directory object selected:

roles1.net/ExampleOU

Apply permissions onto:

☑ This directory object

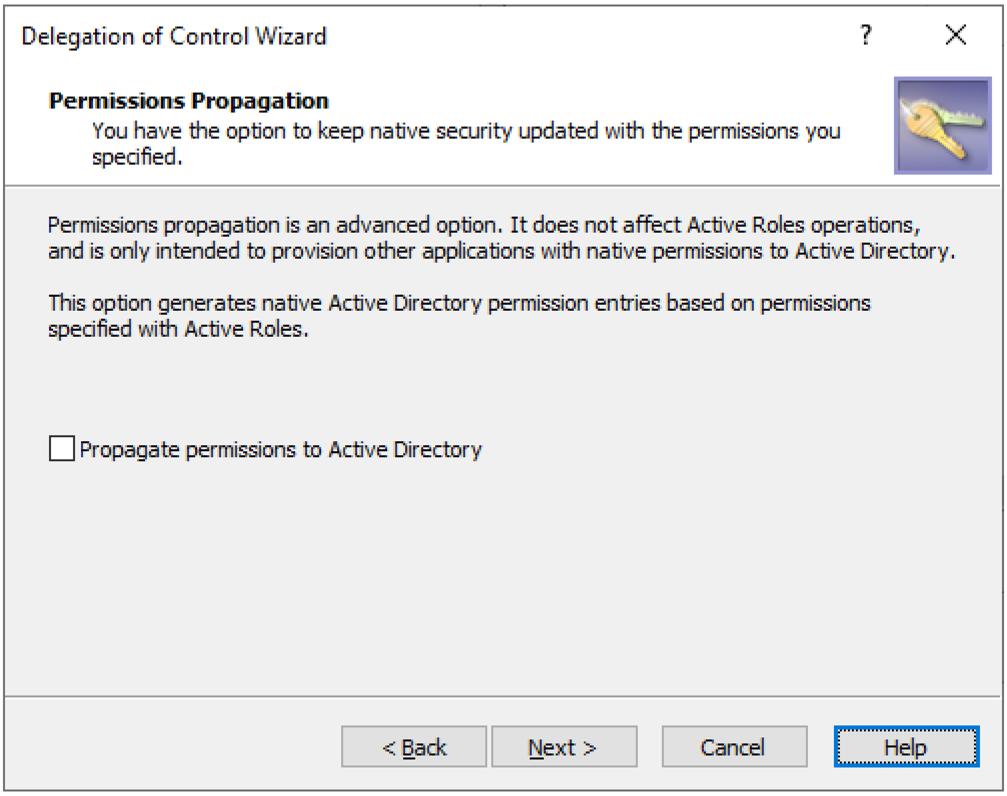
☑ Child objects of this directory object

☐ Immediate child objects only

To continue, click **Next**.

6. In the **Permissions Propagation** step, to synchronize the configured permission settings to the native Active Directory (AD) access controls, select **Propagate permissions to Active Directory**.

Figure 23: Delegation of Control Wizard – Permissions propagation



Delegation of Control Wizard

Permissions Propagation

You have the option to keep native security updated with the permissions you specified.

Permissions propagation is an advanced option. It does not affect Active Roles operations, and is only intended to provision other applications with native permissions to Active Directory.

This option generates native Active Directory permission entries based on permissions specified with Active Roles.

□ Propagate permissions to Active Directory

Selecting this setting will modify the authorization information of the AD objects with the permission settings defined in Active Roles, providing more flexibility for users and groups that use native AD management tools besides Active Roles.

> IMPORTANT: Selecting this setting will result in trustees keeping their configured permissions outside of the Active Roles environment, with the potential risk of bypassing policies configured and enforced with Active Roles.
>
> Therefore, select this option only if the selected trustees have the required security clearance and/or meet all security guidelines in effect within your organization.

> TIP: Once **Propagate permissions to Active Directory** is selected and configured, you can change this setting at any time with the **Active Roles Security** > **Sync to AD** setting, or with the **Advanced Details** > **Sync to AD** setting. For more information, see Synchronizing permissions to Active Directory.

To continue, click **Next**.

7. To complete the wizard, click **Finish**.

Managing Access Template links

When you apply an Access Template (as described in Applying Access Templates), Active Roles creates an Access Template link that stores information about:

- The Access Template used for giving the permissions.
- The directory object on which the Access Template is applied.
- The user or group (Trustee) to whom the permissions are assigned.

If needed, you can modify the link via the Active Roles Console.

> TIP: For more information about Access Template links, see *Access Template link management* in the *Active Roles Feature Guide*.

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, 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. For more information, see Applying Access Templates.
   - 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 for the object with one of the following methods:
   - Right-click the object, and click **Delegate Control**.
   - Right-click the object, and click **Properties**. Then, on the **Administration** tab in the **Properties** dialog, click **Security**.
2. In the **Active Roles Security** dialog, 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 more information, see Applying Access Templates.
   - 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, do the following:
   - 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 more information, see Applying Access Templates.
   - 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.

NOTE: Consider the following when managing Access Template links:

- By default, the **Active Roles Security** dialog 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, 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, 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 the **View/Edit** option. Select the link and click **View/Edit**. Then, click **Properties** next to the **Access Template** box. After that, on the **Administration** tab, click **Links**. Finally, delete the link from the **Links** dialog.
- In the **Active Roles Security** dialog, 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 the **View/Edit** option. 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 for the selected link. This dialog can be considered as a focal point for administration of all elements of the link. Thus, from the **Properties** dialog, 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, 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. For more information, see *Advanced pane* in the *Active Roles Feature Guide*.

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 "permission 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 as follows:

- When applying an Access Template, select the **Propagate permissions to Active Directory** check box in the **Delegation of Control Wizard**. For more information, see Applying Access Templates.
- When managing Access Template links, use the **Sync to AD** button in the dialog that displays a list of links. For more information, see Managing Access Template links.

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 (OU):

To synchronize permission settings on an OU

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

NOTE: Consider the following when configuring permission propagation:

- 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 **Desync to AD**. If you do so, Active Roles deletes all permission entries in Active Directory that were created as a result of synchronization.
- In the **Active Roles Security** dialog, 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 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, 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. For more information, see *Advanced pane* in the *Active Roles Feature Guide*.

Adding, modifying, or removing Access Template permissions

Even after creating a new Access Template, you can:

- Add additional permissions to it.
- Modify any of its permissions.
- Remove any of its permissions.

To change the configured permissions of an existing Access Template, use the Active Roles Console.

Adding permissions to an Access Template

You can add permission entries to an Access Template via the Active Roles Console.

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 Creating an Access Template.

NOTE: Consider the following when working with an Access Template:

- 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 you apply an Access Template in Active Roles to specify directory permissions, any changes to the list of permission entries in the Access Template will result in the directory permissions changing accordingly.
- Active Roles includes a suite of predefined Access Templates. The list of permission entries in a predefined Access Template cannot be modified. If you need to add, modify, or delete permission entries from a predefined Access Template, create a copy of that Access Template, then make changes to the copy. Another option is to create an Access Template and nest the predefined Access Template into the newly created Access Template. For instructions, see Creating an Access Template, Copying an Access Template, and Managing nested Access Templates.

Modifying permissions in an Access Template

You can modify the permissions of an Access Template with the Modify Permission Entry dialog of the Active Roles Console.

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**, then use the tabs in the **Modify Permission Entry** dialog to make changes to the permission entry.

For detailed instructions on how to view or modify a permission entry in an Access Template, see Creating an Access Template.

NOTE: Consider the following when modifying the permissions of an Access Template:

- 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.
- The options on the **Permissions** tab in the **Modify Permission Entry** dialog are read-only. If you need to choose a different option for the permission entry, delete the permission entry, then add a new permission entry with the option you need. For more information, see Adding permissions to an Access Template.
- Once you apply an Access Template in Active Roles to specify directory permissions, any changes to the list of permission entries in the Access Template will result in the directory permissions changing accordingly.
- Active Roles includes a suite of predefined Access Templates. The list of permission entries in a predefined Access Template cannot be modified. If you need to add, modify, or delete permission entries from a predefined Access Template, create a

copy of that Access Template, then make changes to the copy. Another option is to create an Access Template and nest the predefined Access Template into the newly created Access Template. For instructions, see Creating an Access Template, Copying an Access Template, and Managing nested Access Templates.

Removing permissions from an Access Template

You can remove permissions from an Access Template by deleting the related permission entry in the Active Roles Console.

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**, then click **Yes** to confirm the deletion.

NOTE: Consider the following when working with an Access Template:

- 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 you apply an Access Template in Active Roles to specify directory permissions, any changes to the list of permission entries in the Access Template will result in the directory permissions changing accordingly.
- Active Roles includes a suite of predefined Access Templates. The list of permission entries in a predefined Access Template cannot be modified. If you need to add, modify, or delete permission entries from a predefined Access Template, create a copy of that Access Template, then make changes to the copy. Another option is to create an Access Template and nest the predefined Access Template into the newly created Access Template. For instructions, see Creating an Access Template, Copying an Access Template, and Managing nested Access Templates.

Managing nested Access Templates

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

To manage nested ATs, use the **Properties** > **Nesting** settings of the Active Roles Console.

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**, then select the Access Template you want to be included in the Access Template you are configuring.

> NOTE: Consider the following when nesting Access Templates:
>
> - Configuring an Access Template to include another Access Template is called "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 for each of the Access Templates that the selected Access Template is nested into.
> - Nesting allows you to reuse the existing predefined or custom Access Templates. For example, if you need to add permission entries to the predefined **Helpdesk** Access Template, then you can create a new Access Template, nest the **Helpdesk** 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 reuse existing Access Templates. For example, if you need to modify a predefined Access Template, you can create a copy of that Access Template, then modify the copy as needed.

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. To start the **Copy Object - Access Template** wizard, in the details pane, right-click the Access Template, then click **Copy**.
3. On the first page of the wizard, do the following, then click **Next**:
   a. In the **Name** box, enter a name for the new Access Template.
   b. (Optional) In the **Description** box, enter any 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, 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 more information on how to add or modify a permission entry, see Creating an Access Template.
5. To create the copy of the Access Template, click **Finish**.

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 imported the Access Template.

To export Access Templates, select them, right-click the selection, and select **All Tasks** > **Export**. In the **Export Objects** dialog, 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, select the file to which the Access Templates were exported, and click **Open**.

Renaming an Access Template

You can rename an existing Access Template with the Rename setting of the Active Roles Console.

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, then press **Enter**.

> NOTE: Consider the following when renaming an Access Template:
>
> - Renaming an Access Template does not affect its links. This is because Access Templates are referenced by immutable identifier rather than by name.
> - 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 predefined Access Templates. The name of a predefined Access Template cannot be modified. If you need an Access Template with a different name to have the same permission entries as a predefined Access Template, create a copy of the predefined Access Template, and then make changes to the copy. Another option is to create an Access Template and nest the predefined Access Template into the newly- created Access Template. For more information, see Creating an Access Template, Copying an Access Template, and Managing nested Access Templates.

Deleting an Access Template

You can delete existing Access Templates in the Active Roles Console.

Prerequisites

To delete an Access Template, you must remove all existing references to it. To do so:

- Delete the links to the Access Template. For more information, see Managing Access Template links.
- Remove the Access Template from all Access Templates in which the Access Template is nested. For more information, see Managing nested Access Templates.

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, then click **Delete**.

> NOTE: Consider the following when deleting an Access Template:
>
> - 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 by right-clicking the Access Template, and clicking **Links**. If you need to delete the Access Template, first remove all items from the **Links** list. For instructions, see Managing Access Template links.
> - An Access Template cannot be deleted if it is nested into another Access Template. To view the Access Templates into which the selected Access Template is nested, on the **Nesting** tab, click **Nested In**. Then, double-click an item in the **Nested In** list to open a dialog where you can remove the Access Template from nesting. For instructions, see Managing nested Access Templates.
> - Active Roles includes a suite of predefined Access Templates and a number of built-in Access Templates. Neither predefined Access Templates nor built-in Access Templates can be deleted. For more information on the built-in Access Template, see *Active Roles Built-in Access Templates Reference Guide*.

Access Rule management tasks

This section guides you through the Active RolesConsole to manage Windows claims-based Access Rules.

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.
- 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 2016 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.

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`

Enabling claim support

Claims-based authorization requires:

- A domain controller (or controllers) running a version of Windows Server supported by Active Roles, with claim support enabled. For the list of supported operating system, see *System Requirements* in the *Active Roles Release Notes*.
- (Optional) If you need to use device claims, then a domain-joined client computer (or computers) running a supported version of the Windows operating system.

Domain controller requirements for claims-based authorization

The claims-based authorization mechanism has the following requirements on the domain controller (DC) side:

- Extensions to Active Directory, such as claim type objects intended to store the claim configuration data. By adding a Windows Server domain controller (DC), you extend the Active Directory schema to provide the object classes and attributes required to support claims-based authorization.
- Enhancements in the Kerberos Key Distribution Center (KDC) and Security Accounts Manager (SAM) that enable DCs running Windows Server to recognize claim types, retrieve claim information, and transport claims within Kerberos tickets.

  A Windows Server DC 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.
- If the DC does not support claim issuance by default, you must enable it via Group Policy. The Group Policy setting that serves this purpose is located in **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.

> NOTE: Claims-based authorization does not impose domain or forest functional requirements. If your Active Directory domain has a sufficient number of Windows Server DCs to service authentication requests that include claim information, then you can make use of Windows claims.

Client computer

The claims-based authorization mechanism has the following requirements on the client computer side:

- Domain-joined client computers running supported Windows operating systems are required for claims-based authorization when using device claims. For the list of supported operating system, see *System Requirements* in the *Active Roles Release Notes*.

  > NOTE: This requirement does not apply to authorization scenarios that employ user claims only.

- If the client computer does not request user or device claims upon user authentication, you must enable claim support on the client computer via Group Policy. The Group Policy setting that serves this purpose is located in **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.

Managing Windows claims

Claims are statements about an authenticated user or device, issued by an Active Directory domain controller running Windows Server 2016 or later. Claims can contain information about the user or device retrieved from Active Directory.

You can manage claims in Active Roles under the **Active Directory** > **Claim Types** container of the Active Roles **Console tree**.

> NOTE: For more information about Windows claims and claims-based Access Rules, see *Management of Windows claims* in the *Active Roles Feature Guide*.

Managing claim types

Claim types must be created in Active Directory to enable domain controllers (DCs) to issue claims to users or computers. Claims issued by the DC 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 DCs running a Windows Server operating system supported by (see System Requirements in the Release Notes for more information). To identify the forest of a given **Claim Types** container, the container 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, then click **Delete**.
2. Confirm the claim type deletion.

If you encounter a message stating that you do not 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

1. In the Active Roles Console, navigate to the enabled claim type you want to disable.
2. Right-click the claim type object and click **Disable**.

To enable a disabled claim type

1. In the Active Roles Console, navigate to the disabled claim type you want to enable.
2. Right-click the claim type object and click **Enable**.

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.

Access Rules are held in the **Access Rules** container under the **Configuration** node in the **Console tree**.

Creating or modifying 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, then click **Finish**.

To modify an existing Access Rule

1. Right-click the Access Rule you want to modify, 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.

Configuring 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, 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.

Applying an Access Rule to Access Template links

You must 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. 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 the 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 display a list of Access Template links in a number of ways:

   - Right-click a container, 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, 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, then click **Links**. This displays a list of all Access Template links referring to that Access Template.
2. In **Properties**, click **Access Rule**.
3. Click **Change**, then select the Access Rule you want to apply.

From the **Access Rule** tab, you can also perform the following tasks:

- **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.
- **Properties**: Click this button to view or change the Access Rule properties, including the conditional expression of the Access Rule.
- **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 to it, refer to the **Access Rule** field in the list of Access Template links.

Deploying an Access Rule

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 to the **Department** claim of that delegated administrator.

Prerequisites of deploying an Access Rule

To deploy a new Access Rule, you and your organization must meet the following conditions:

- Your organization must have an Active Directory domain, with at least one Domain Controller (DC) running a Windows Server version supported by Active Roles. For the supported operating systems, see *System Requirements* in the *Active Roles Release Notes*.
- The Active Roles Administration Database and Active Roles Console of the latest version of the product must be installed on a member server in your Active Directory domain. The member server must also run a Windows server version supported by Active Roles.
- Your Active Directory domain is registered with Active Roles as a managed domain.

Enabling claim support

To deploy the Access Rule, configure a Group Policy to enable domain controllers (DCs) to issue claims.

To create a Group Policy for claim support

1. On a DC running a supported version of Windows Server, open the Group Policy Management console.

   To open the Console, press **Win+R**. Then, in the **Run** dialog, 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**, 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, click **Enabled** and select **Supported** from the **Options** drop-down list. When finished, click **OK** to close the dialog.
5. Close the **Group Policy Management Editor**.
6. Close **Group Policy Management**.
7. Open the Windows command prompt and enter the following command:

   ```
   gpupdate /force
   ```

Once you are ready, configure the Group Policy to enable the Active Roles Administration Database to retrieve claims for clients by using Kerberos protocol transition.

To configure the Group Policy to retrieve claims

1. On the server running the Active Roles Administration Service, open the **Local Group Policy Editor** console.
2. To open the Console, press **Win+R**. Then, in the **Run** dialog, type `gpmc.msc`, and click **OK**.
3. In the **Console tree**, select **Computer Configuration** > **Policies** > **Administrative Templates** > **System** > **Kerberos**.
4. In the details pane, double-click **Kerberos client support for claims, compound authentication and Kerberos armoring**.
5. In the **Kerberos client support for claims, compound authentication and Kerberos armoring** dialog, click **Enabled**, then click **OK**.
6. Restart the computer to apply the new setting to the Active Roles Administration Service.

   > NOTE: Make sure to restart the computer. Restarting only the Active Roles Administration Service is not sufficient.

Once you are ready, to enable Kerberos authentication, add the Service Principal Names (SPNs) of the Active Roles Administration Service to the service account.

To add SPNs to the service account and enable Kerberos authentication

1. Open the Windows command prompt.
2. Enter the following commands:
   - **setspn -s aradminsvc/<FQDN> <service-account-name>**
   - **setspn -s aradminsvc/<name> <service-account-name>**

In the above commands:

- `<FQDN>` is the fully qualified domain name of the computer running the Active Roles Administration Service (for example, `arsrv.domain.com`).
- `<name>` is the name of the computer (for example, `arsrv`).
- `<service-account-name>` is the name of the service account (that is the domain user account running the Active Roles Administration Service), for example `domain\arsvcacct`.

Creating a claim type

Create a claim type object for your Domain Controller (DC) to issue user claims sourced from the **Department** attribute. Log in as an Active Roles administrator and perform the following steps in the Active Roles Console.

> NOTE: If using a default Active Roles configuration, you must log in with a domain user account that is a member of the Administrators local group of the member server running the Active Roles Administration Service.

To create a claim type

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**, then click **Finish**.

Creating the 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.

To create a new Access Rule

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, then click **Next**.
3. On the **Conditions** page, configure the conditional expression:
   a. Click the **AND group** item, then click **Insert condition**.
   b. Click **Configure condition to evaluate**, then click **User claim**.
   c. On the **Select Claim Type** page that appears, click **Department** in the list of claim types, then click **OK**.
   d. Verify that the comparison operator reads **equals** (this is the default setting).
   e. Click **Define value to compare to**, then click **Target object property**.
   f. On the **Select Target Object Property** page that appears, select the **Department** property, then click **OK**.
4. Click **Finish**.

Applying the Access Rule

To apply the Access Rule you created in the Creating the Access Rule step, you first need to delegate control by using an Access Template, 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:

To apply an Access Rule to a security group

1. In the **Console tree**, under the **Active Directory** node, right-click the name of your domain, then click **Delegate Control**.

2. To start the **Delegation of Control Wizard**, on the **Active Roles Security** page that appears, click **Add**.
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. To close the **Active Roles Security**, click **OK**. This will create the Access Template links.
5. To open the **Active Roles Security** page claim, right-click the name of your Active Directory domain and click **Active Roles Security**.
6. On the **Active Roles Security** page, select the **Users - Modify All Properties** Access Template link, then click **View/Edit**.
7. On the **Access Rule** tab in dialog that appears, click **Change**, and select the **Department Admins** Access Rule. To close the **Select an Access Rule** page, click **OK**, then close the dialog by clicking **OK** again.
8. To close the **Active Roles Security** page, click **OK**.

After you 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 account of the delegated administrator.

5

Rule-based autoprovisioning and deprovisioning

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 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 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 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 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** option.

> 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 2: 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 Configuring a User Logon Name Generation policy.
E-mail 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 uniqueness number to the alias. For more information on how to set up this policy, see 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 Configuring an Exchange Mailbox AutoProvisioning 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.

Policy	Description
	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 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: • 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, seeConfiguring 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

Policy	Description
	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 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 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. 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 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 (Azure AD). For more information on how to set up a Script Execution policy, see Configuring a Script Execution policy.
Microsoft 365 and	Enables configuring multiple assignments to Azure objects. Use this

Policy	Description
Azure Tenant Selection	policy to: - Validate the selected Azure tenants for Azure users, guest users, O365 groups, Azure security 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 in 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 3: 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

Policy	Description
	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. • 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.

Policy	Description
	- 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 email 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 email 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 subject of the email 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 Microsoft 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.

Creating a Policy Object

The Active Roles Console provides separate wizards for creating Policy Objects both for provisioning and deprovisioning.

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 by right-clicking **Administration** and selecting **New** > **Container**. Similarly, you can create a sub-folder in a folder by right-clicking the folder and selecting **New** > **Container**.
2. Right-click the folder, point to **New**, 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:

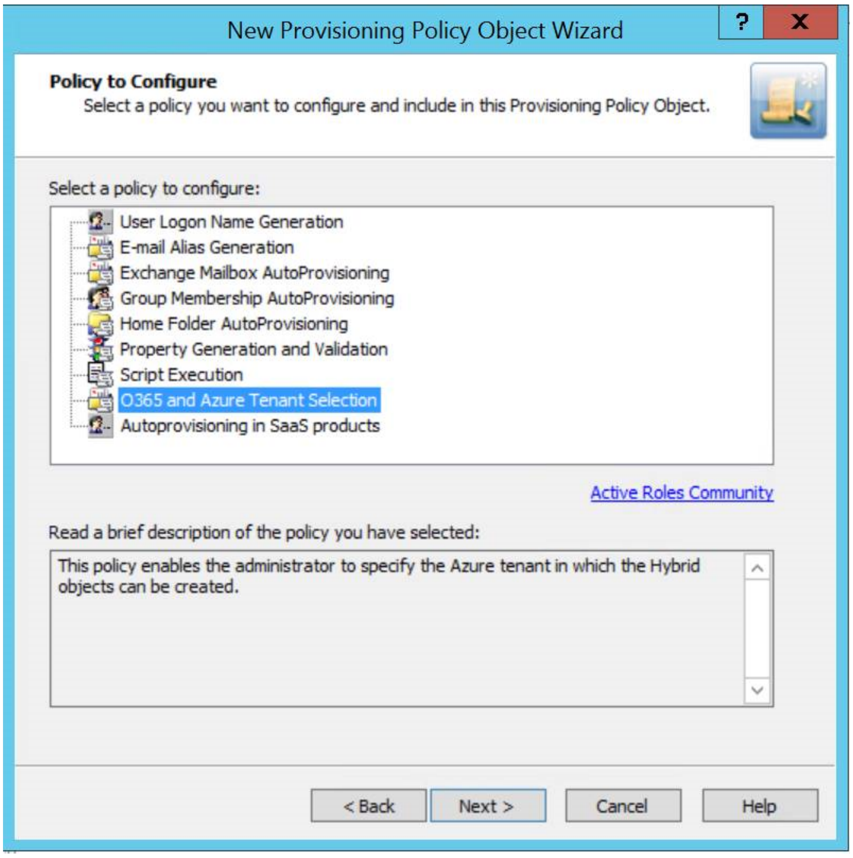
a. In the **Name** box, enter a name for the Policy Object.

b. (Optional) Under **Description**, enter any 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.

Figure 24: Provisioning policies



New Provisioning Policy Object Wizard

Policy to Configure

Select a policy you want to configure and include in this Provisioning Policy Object.

Select a policy to configure:

- User Logon Name Generation
- E-mail Alias Generation
- Exchange Mailbox AutoProvisioning
- Group Membership AutoProvisioning
- Home Folder AutoProvisioning
- Property Generation and Validation
- Script Execution
- O365 and Azure Tenant Selection
- Autoprovisioning in SaaS products

Active Roles Community

Read a brief description of the policy you have selected:

This policy enables the administrator to specify the Azure tenant in which the Hybrid objects can be created.

6. On the **Enforce Policy** page, you can specify the objects to which this Policy Object will be applied. To locate and select the objects you want, Click **Add** and use **Select Objects**.

7. Click **Next**, then click **Finish**.

NOTE: Consider the following when creating Policy Objects:

- For information about available policy types, see Provisioning Policy Objects and Deprovisioning Policy Objects.
- For information on how to configure policies, see Policy configuration tasks.
- To add more policies to the new Policy Object, display the **Properties** dialog, and click **Add** on the **Policies** tab.

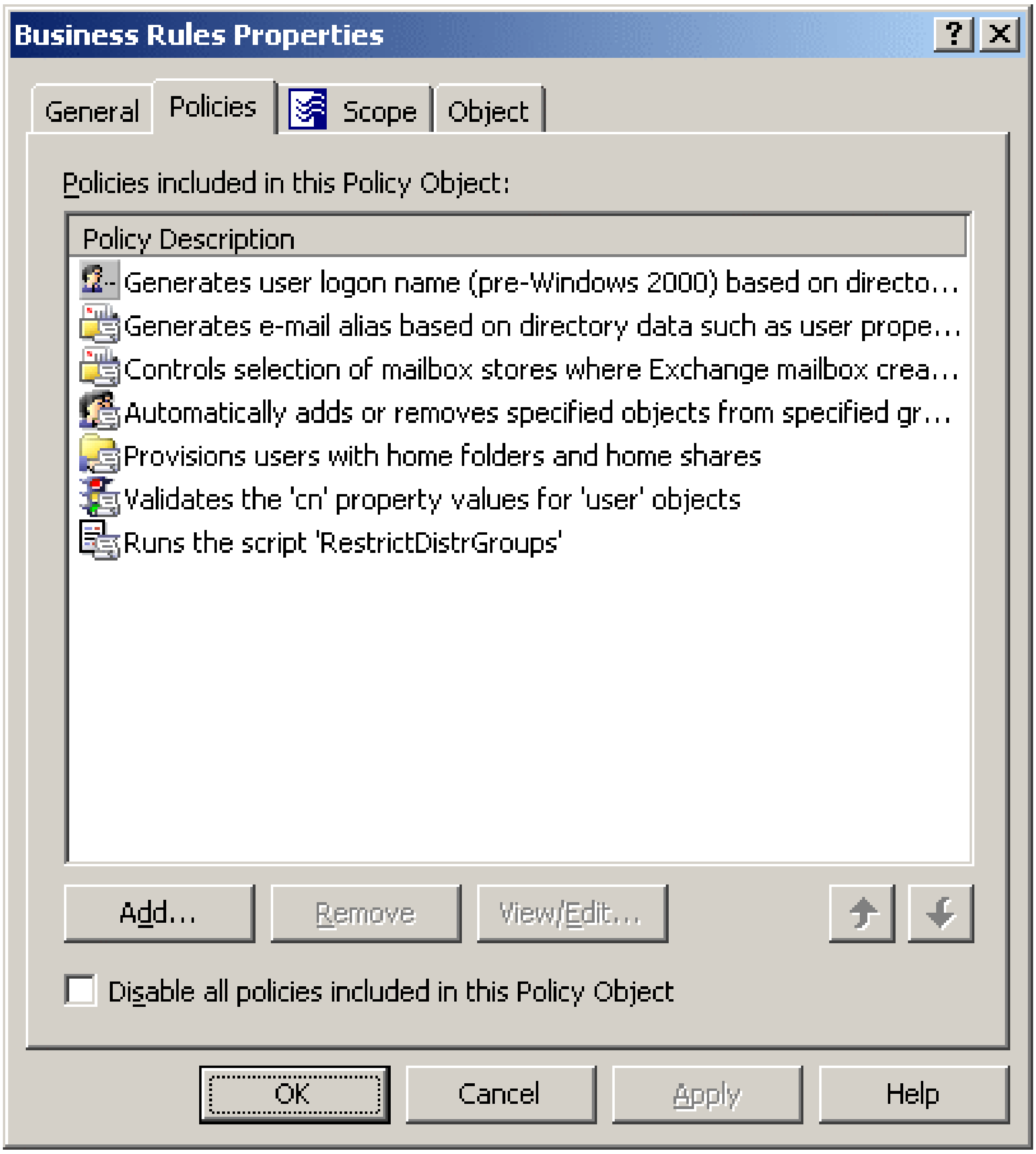
Adding policies to a Policy Object

You can add policies to Policy Objects with the Active Roles Console.

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.

Figure 25: Policy Objects Management



Policies included in this Policy Object:

Policy Description
Generates user logon name (pre-Windows 2000) based on directo...
Generates e-mail alias based on directory data such as user prope...
Controls selection of mailbox stores where Exchange mailbox crea...
Automatically adds or removes specified objects from specified gr...
Provisions users with home folders and home shares
Validates the 'cn' property values for 'user' objects
Runs the script 'RestrictDistrGroups'

□ Disable all policies included in this Policy Object

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: Consider the following when adding policies to Policy Objects:

- 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.

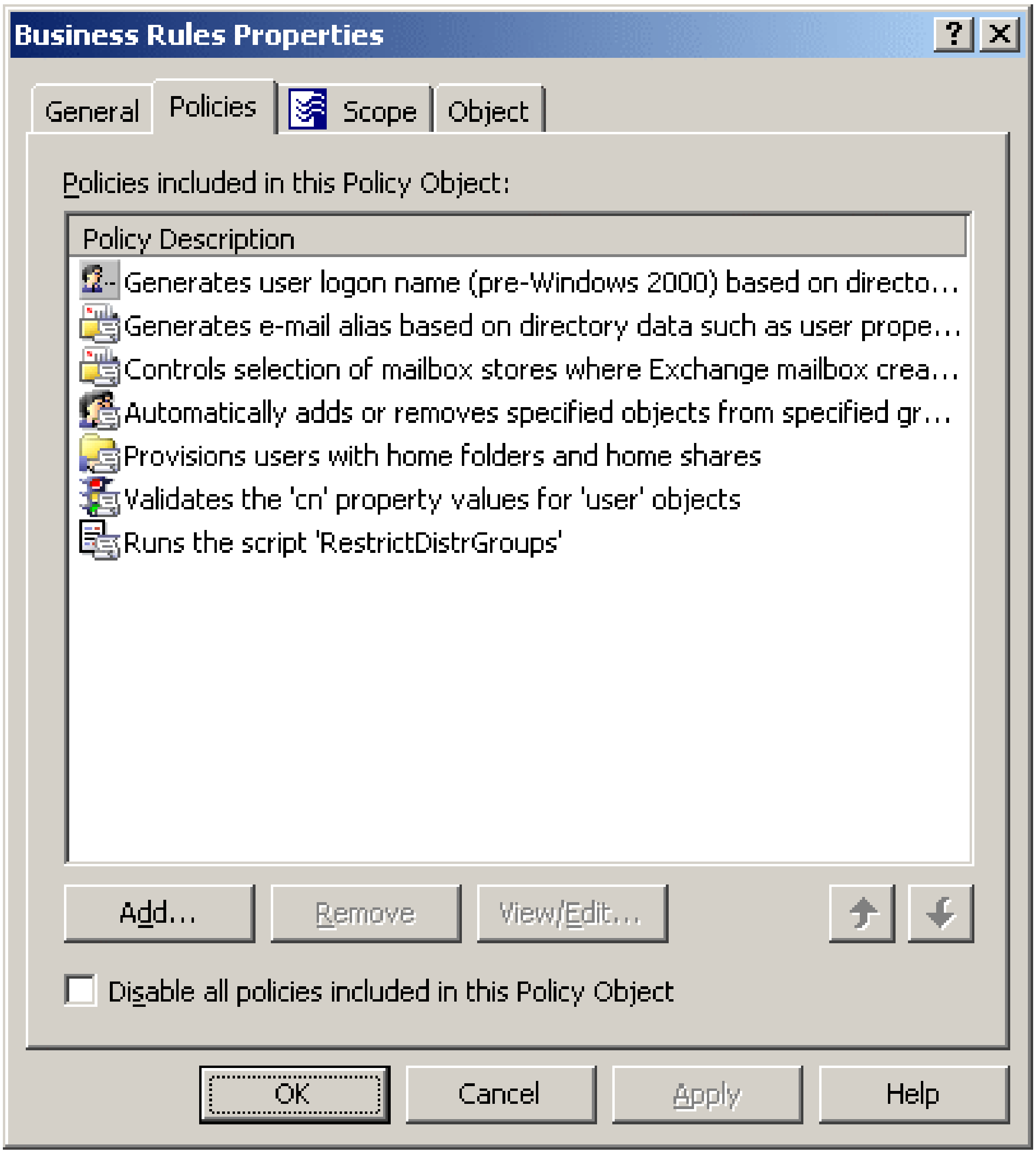
Modifying policies in a Policy Object

You can modifies policies in Policy Objects with the Active Roles Console.

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, then click **Properties**.
3. On the **Policies** tab, select the policy you want to view or modify, and click **View/Edit**.

Figure 26: Policy Objects Management



Business Rules Properties

Policies included in this Policy Object:

Policy Description
Generates user logon name (pre-Windows 2000) based on directo...
Generates e-mail alias based on directory data such as user prope...
Controls selection of mailbox stores where Exchange mailbox crea...
Automatically adds or removes specified objects from specified gr...
Provisions users with home folders and home shares
Validates the 'cn' property values for 'user' objects
Runs the script 'RestrictDistrGroups'

□ Disable all policies included in this Policy Object

4. Use the tabs in the **Policy Properties** dialog to view or modify policy settings.

   The tabs in the **Policy Properties** dialog provide the same options as the wizard for configuring the policy. For information about the options specific to each type of policy, see Policy configuration tasks.

NOTE: Consider the following when modifying policies in a Policy Object:

- 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.

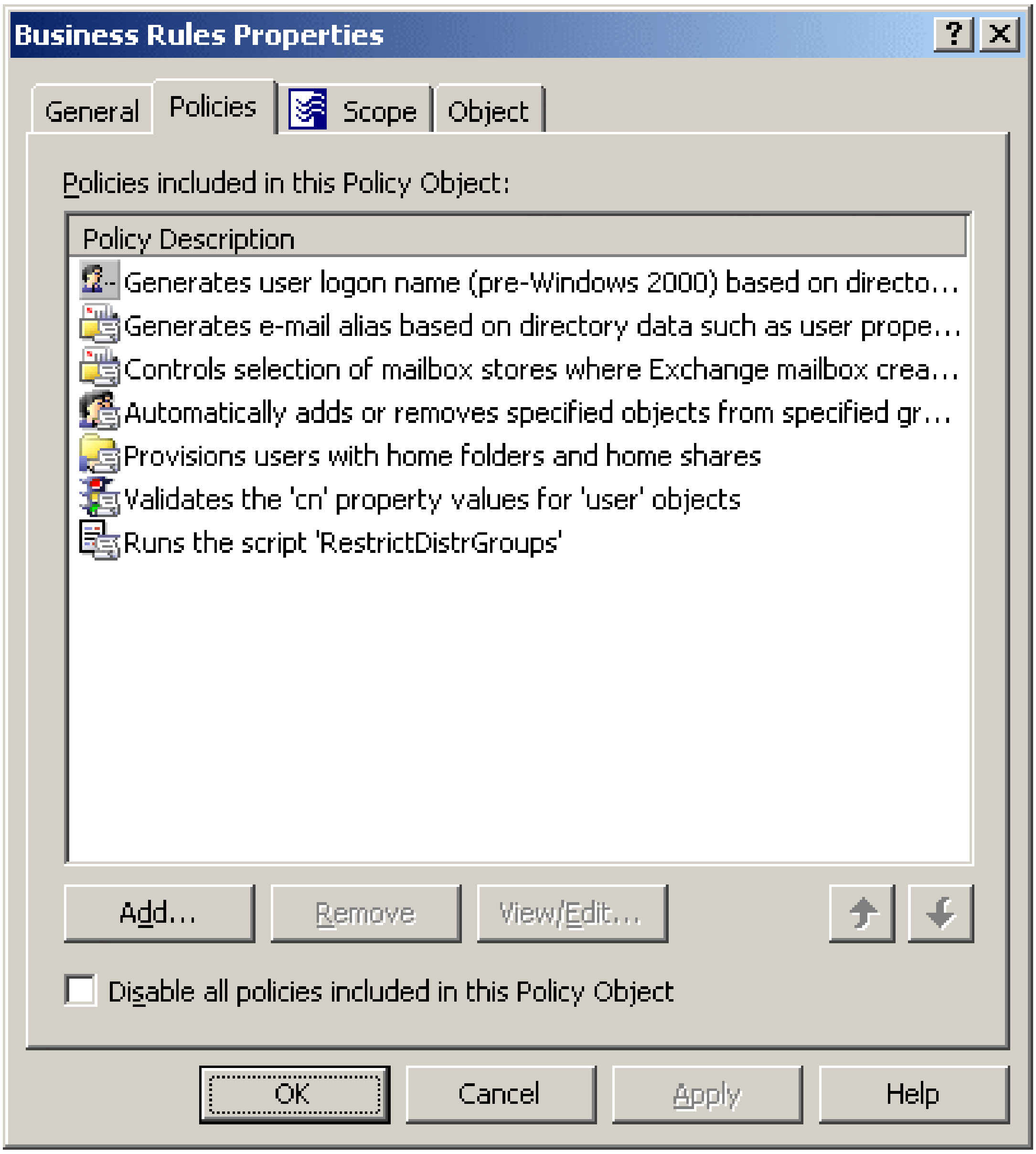
Removing policies from a Policy Object

You can remove policies from Policy Objects with the Active Roles Console.

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, then click **Properties**.
3. On the **Policies** tab, select the policy you want to delete, click **Remove**, then click **Yes** to confirm the deletion.

Figure 27: Policy Objects Management



Policies included in this Policy Object:

Policy Description
Generates user logon name (pre-Windows 2000) based on directo...
Generates e-mail alias based on directory data such as user prope...
Controls selection of mailbox stores where Exchange mailbox crea...
Automatically adds or removes specified objects from specified gr...
Provisions users with home folders and home shares
Validates the 'cn' property values for 'user' objects
Runs the script 'RestrictDistrGroups'

□ Disable all policies included in this Policy Object

NOTE: Consider the following when removing policies for a Policy Object:

- 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.

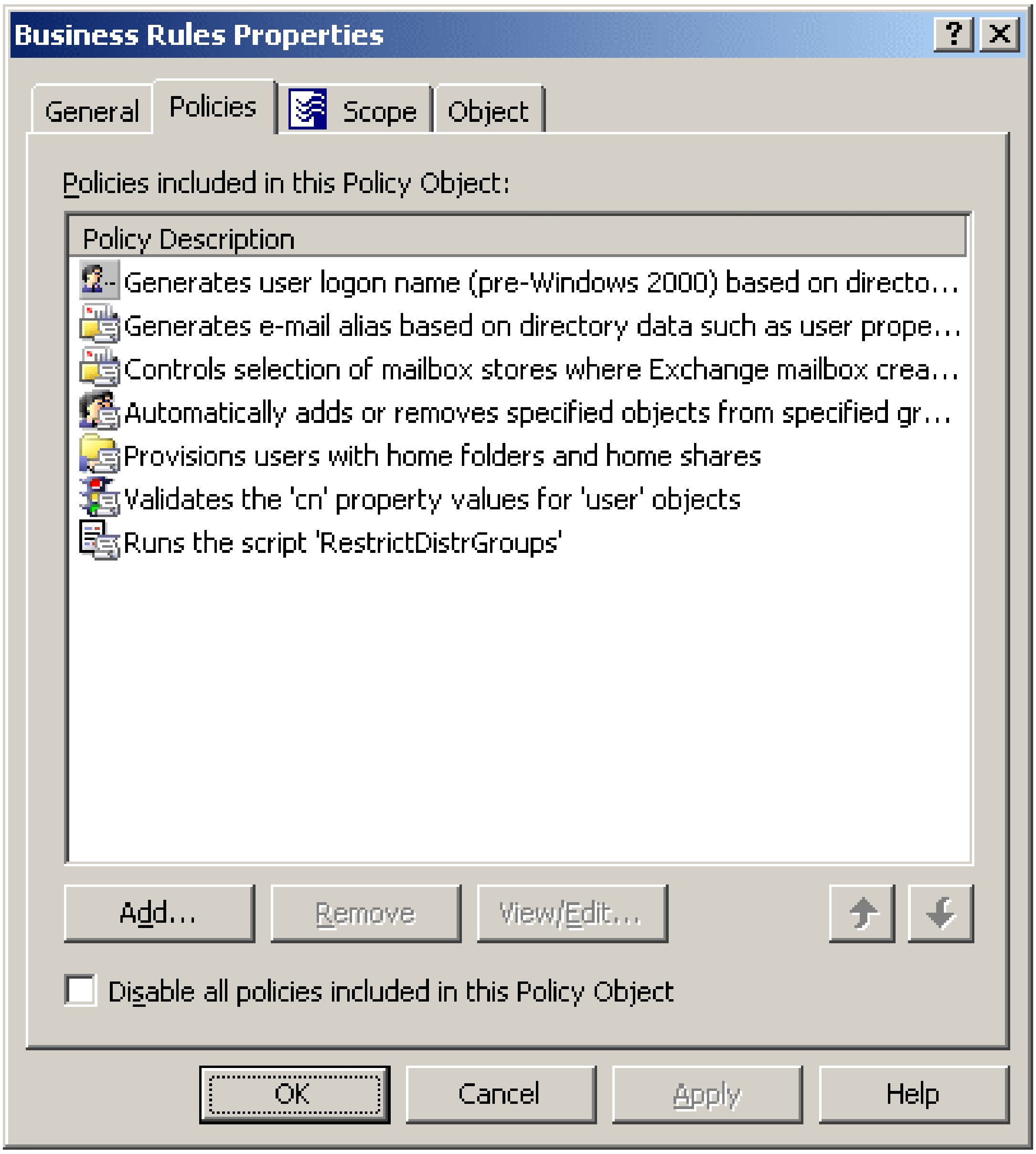
Blocking all policies in a Policy Object

For troubleshooting purposes, you can stop enforcing policies without actually deleting them.

To block all policies 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, then click **Properties**.
3. On the **Policies** tab, select the **Disable all policies included in this Policy Object** check box and click **Apply**.

Figure 28: Policy Objects Management



Policies included in this Policy Object:

Policy Description
Generates user logon name (pre-Windows 2000) based on directo...
Generates e-mail alias based on directory data such as user prope...
Controls selection of mailbox stores where Exchange mailbox crea...
Automatically adds or removes specified objects from specified gr...
Provisions users with home folders and home shares
Validates the 'cn' property values for 'user' objects
Runs the script 'RestrictDistrGroups'

□ Disable all policies included in this Policy Object

4. Click **OK**.

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, that is 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. For more information on how to block policy inheritance, see Managing policy scope.

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, then click **Policy Scope**.
3. In the **Active Roles Policy Scope** dialog, click **Add**.
4. Use the **Select Objects** dialog 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.

To specify policy settings on an object by using a Policy Object

1. Open the **Active Roles Policy** dialog for the object in one of the following ways:
   - Right-click the object, and click **Enforce Policy**.
   - Right-click the object, and click **Properties**. Then, on the **Administration** tab in the **Properties** dialog, click **Policy**.
2. In the **Active Roles Policy** dialog, click **Add**.
3. Use the **Select Policy Objects** dialog 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.

> 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. To do so, 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. For more information, see *Advanced pane* in the *Active Roles Feature Guide*.

To view or modify inheritance options for a Policy Object on a container or Managed Unit

1. Open the **Active Roles Policy Scope** dialog for the Policy Object: Right-click the Policy Object, then click **Policy Scope**.
2. In the **Active Roles Policy Scope** dialog, select the container or Managed Unit to which the Policy Object is applied and on which you want to examine inheritance options, 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.

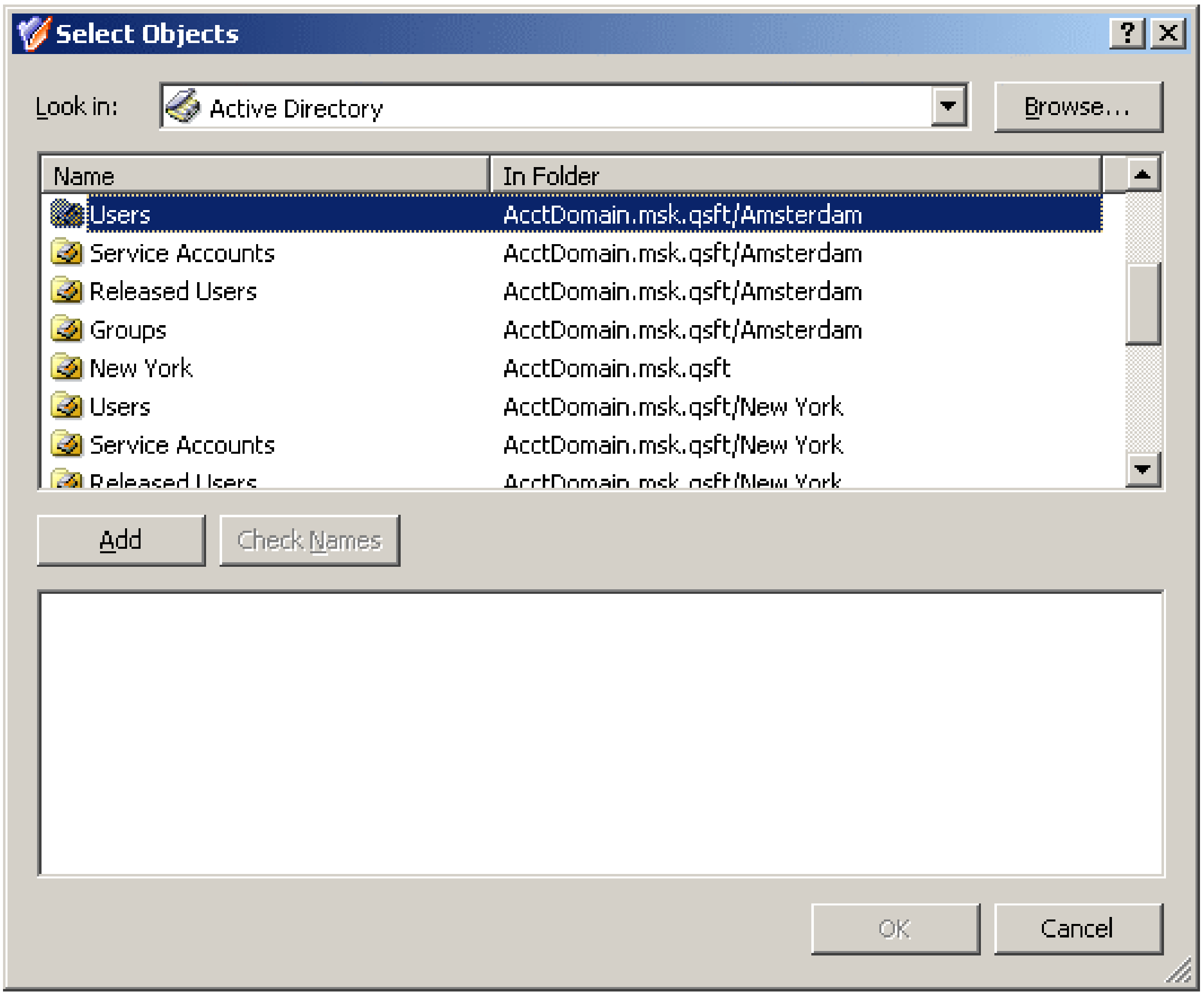
Adding Managed Units or containers to policy scope

You can add administrative views (Managed Units) and directory folders (containers) to the policy scope of a given Policy Object in one of following ways:

- Right-click the Policy Object and click **Policy Scope**. Then, in the **Active Roles Policy Scope** window, click **Add**.
- 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 **Browse** and select **Active Directory** or a container in the hierarchy under **Active Directory**.

Figure 29: Policy Objects



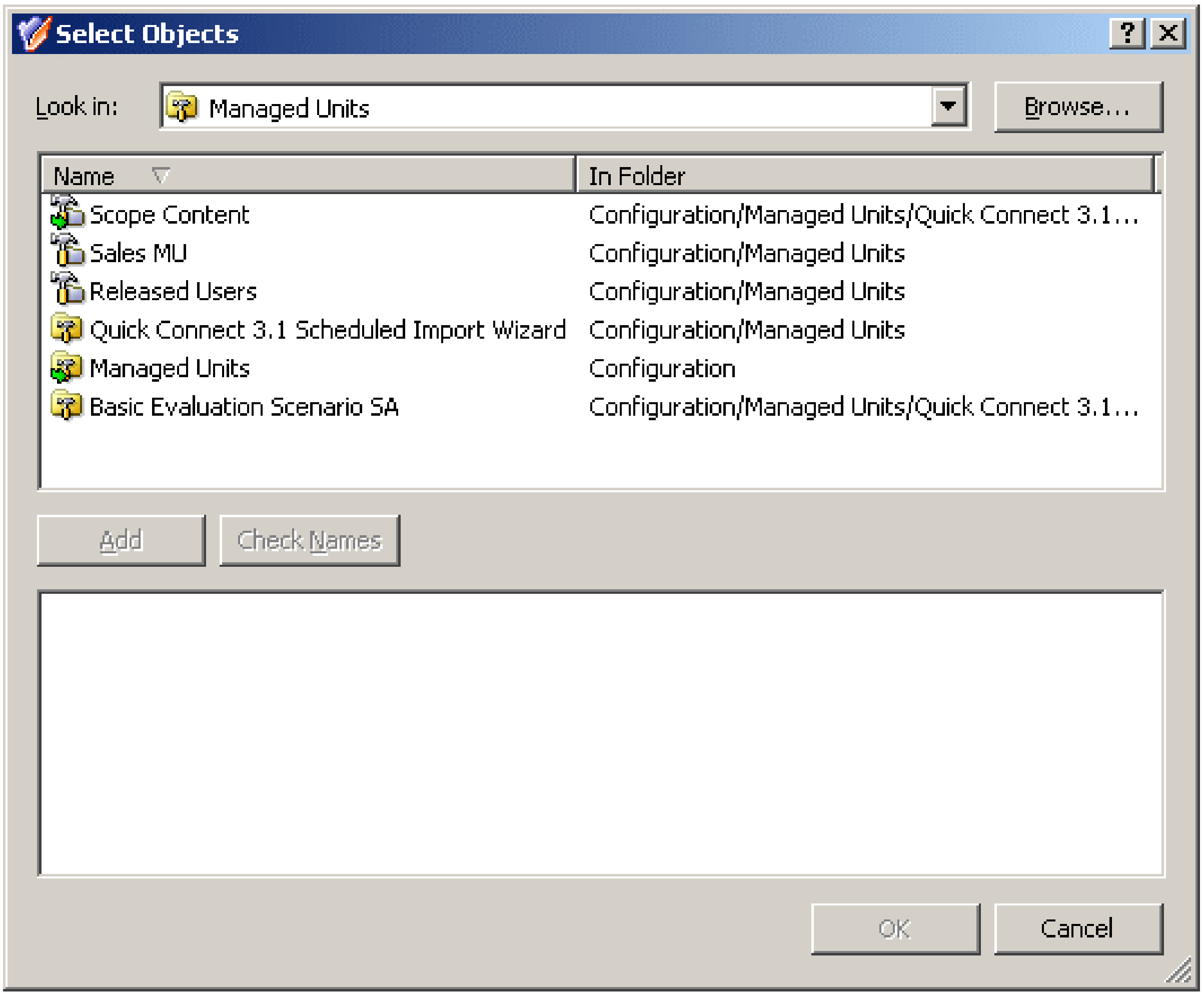
Select Objects

Look in: Active Directory

Name	In Folder
Users	AcctDomain.msk.qsft/Amsterdam
Service Accounts	AcctDomain.msk.qsft/Amsterdam
Released Users	AcctDomain.msk.qsft/Amsterdam
Groups	AcctDomain.msk.qsft/Amsterdam
New York	AcctDomain.msk.qsft
Users	AcctDomain.msk.qsft/New York
Service Accounts	AcctDomain.msk.qsft/New York
Released Users	AcctDomain.msk.qsft/New York

To build a list of Managed Units, click **Browse** and select **Managed Units** or a container in the hierarchy under **Managed Units**.

Figure 30: Managed Units



Look in: Managed Units

Name	In Folder
Scope Content	Configuration/Managed Units/Quick Connect 3.1...
Sales MU	Configuration/Managed Units
Released Users	Configuration/Managed Units
Quick Connect 3.1 Scheduled Import Wizard	Configuration/Managed Units
Managed Units	Configuration
Basic Evaluation Scenario SA	Configuration/Managed Units/Quick Connect 3.1...

In the **Select Objects** window, select containers or Managed Units from the list and click **Add** 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:

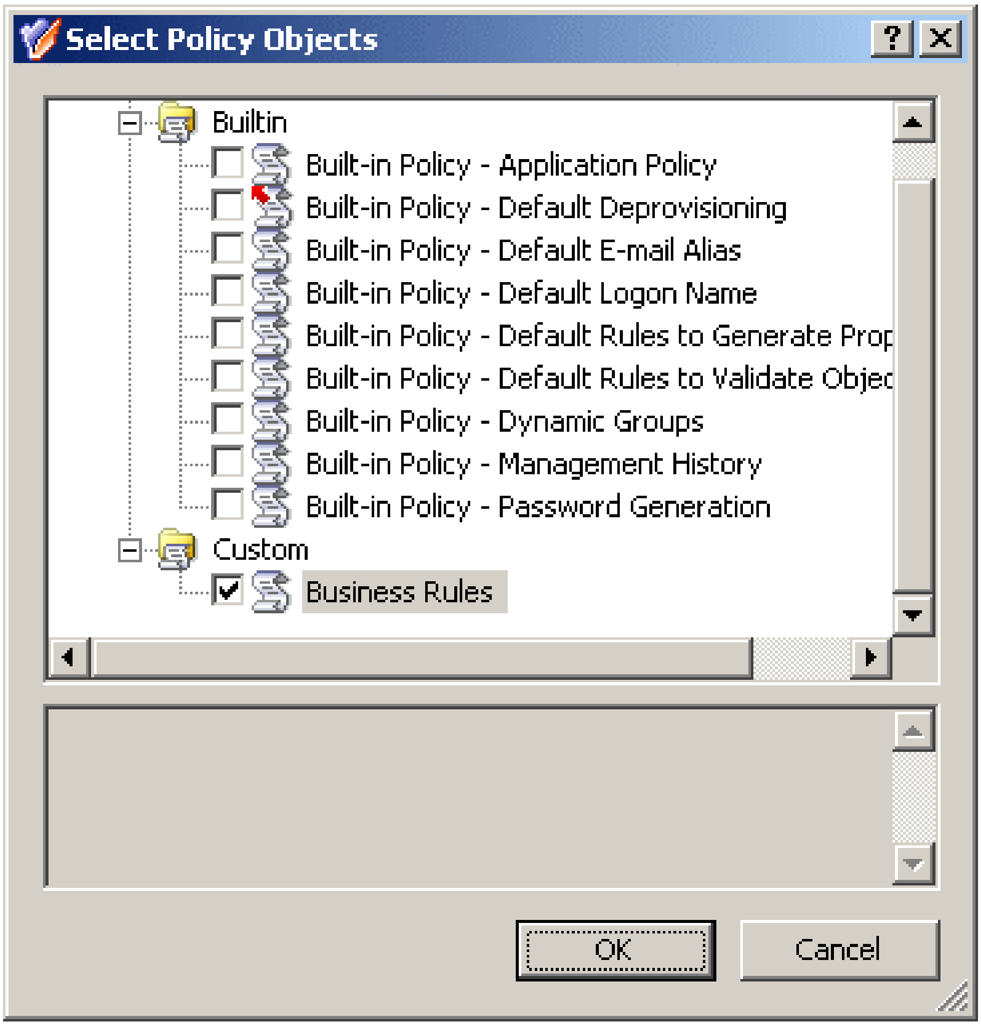
- Right-click a Managed Unit or container and click **Enforce Policy**. Then, in the **Active Roles Policy** window, click **Add**.
- Right-click a leaf object (user, group, or the like), click **Properties**, go to the **Administration** tab, and click **Policy**. Then, in the **Active Roles Policy** window, click **Add**.

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 the Policy Objects, then click **OK**.

Figure 31: Policy Objects



Select Policy Objects

Builtin
- □ Built-in Policy - Application Policy
- □ Built-in Policy - Default Deprovisioning
- □ Built-in Policy - Default E-mail Alias
- □ Built-in Policy - Default Logon Name
- □ Built-in Policy - Default Rules to Generate Prop
- □ Built-in Policy - Default Rules to Validate Objec
- □ Built-in Policy - Dynamic Groups
- □ Built-in Policy - Management History
- □ Built-in Policy - Password Generation

Custom
- ☑ Business Rules

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:

- The Policy Object that defines the policies.
- The Directory object that is the target of the link.
- An **Include** or **Exclude** flag 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 **Advanced**.

  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 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 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, that is 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 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. However, it 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 for the Policy Object you select from the list. You can use the **Properties** dialog 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 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.

Managing Policy Object links

When you apply a Policy Object (see Applying Policy Objects), 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, do the following:
   - To create a new link, click **Add**, and then use the **Select Objects** dialog 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 Applying Policy Objects.
- 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 for the object in one of the following ways:
    - Right-click the object, and click **Enforce Policy**.
    - Right-click the object, and click **Properties**. Then, on the **Administration** tab in the **Properties** dialog, click **Policy**.

    The **Active Roles Policy** dialog 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, 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 more information, see Modifying policies in a Policy Object.
    - 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, click **Advanced**.
2. In the **Active Roles Policy - Advanced View** dialog, 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: Consider the following when managing Policy Object links:
>
> - By default, the **Active Roles Policy - Advanced View** dialog 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 displays the **Properties** dialog for the selected link. From the **Properties** dialog, 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. For more information, see Applying Policy Objects.
> - 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, 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* in the *Active Roles Feature Guide*).

Excluding an object from a 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. To remove the effects of the Policy Object from specific objects, perform the following steps.

To exclude an object from the policy scope of a Policy Object

1. Open the **Active Roles Policy** dialog for the object in one of the following ways:
   - Right-click the object, and click **Enforce Policy**.
   - Right-click the object, and click **Properties**. Then, on the **Administration** tab in the **Properties** dialog, click **Policy**.
2. In the **Active Roles Policy** dialog, select the **Blocked** check box next to the name of the Policy Object.
3. Click **OK** to close the **Active Roles Policy** dialog.

> NOTE: Consider the following when excluding an object from the policy scope of a Policy Object:
>
> - 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 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 more information on how to manage Policy Object links, see Managing Policy Object links.

Copying a Policy Object

With the Active Roles Console, you can create copies of Policy Objects. This feature helps you reuse existing Policy Objects.

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. To start the **Copy Object - Policy Object Wizard**, in the details pane, right-click the Policy Object, then click **Copy**.
3. On the first page of the wizard, do the following:
   a. In the **Name** box, enter a name for the new Policy Object.
   b. (Optional) In the **Description** box, enter any information about the new Policy Object.

   Click **Next**.
4. To create the copy of the Policy Object, click **Finish**.

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 for the newly created Policy Object. To have the console display the **Properties** dialog, select **Display the object properties when this wizard closes** on the completion page of the **Copy Object - Policy Object Wizard**. For more information on how to add, modify, and remove policies from a Policy Object, see Adding policies to a Policy Object, Modifying policies in a Policy Object, and Removing policies from a Policy Object.

Renaming a Policy Object

You can rename Policy Objects with the Active Roles Console.

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. Enter a new name, 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, 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, select the file to which the Policy Objects were exported, and click **Open**.

Deleting a Policy Object

You can delete Policy Objects with the Active Roles Console.

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, 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.

Policy configuration tasks

This section discusses how to configure policies of the following types, grouped by Policy Object category.

Table 4: Policy Configuration Tasks

Policy Object category	Policy type
Provisioning Policy Object	Property Generation and Validation
	User Logon Name Generation
	Group Membership AutoProvisioning
	E-mail Alias Generation
	Exchange Mailbox AutoProvisioning
	Home Folder AutoProvisioning
	Script Execution
	Microsoft 365 and Azure Tenant Selection
	AutoProvisioning in 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 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.

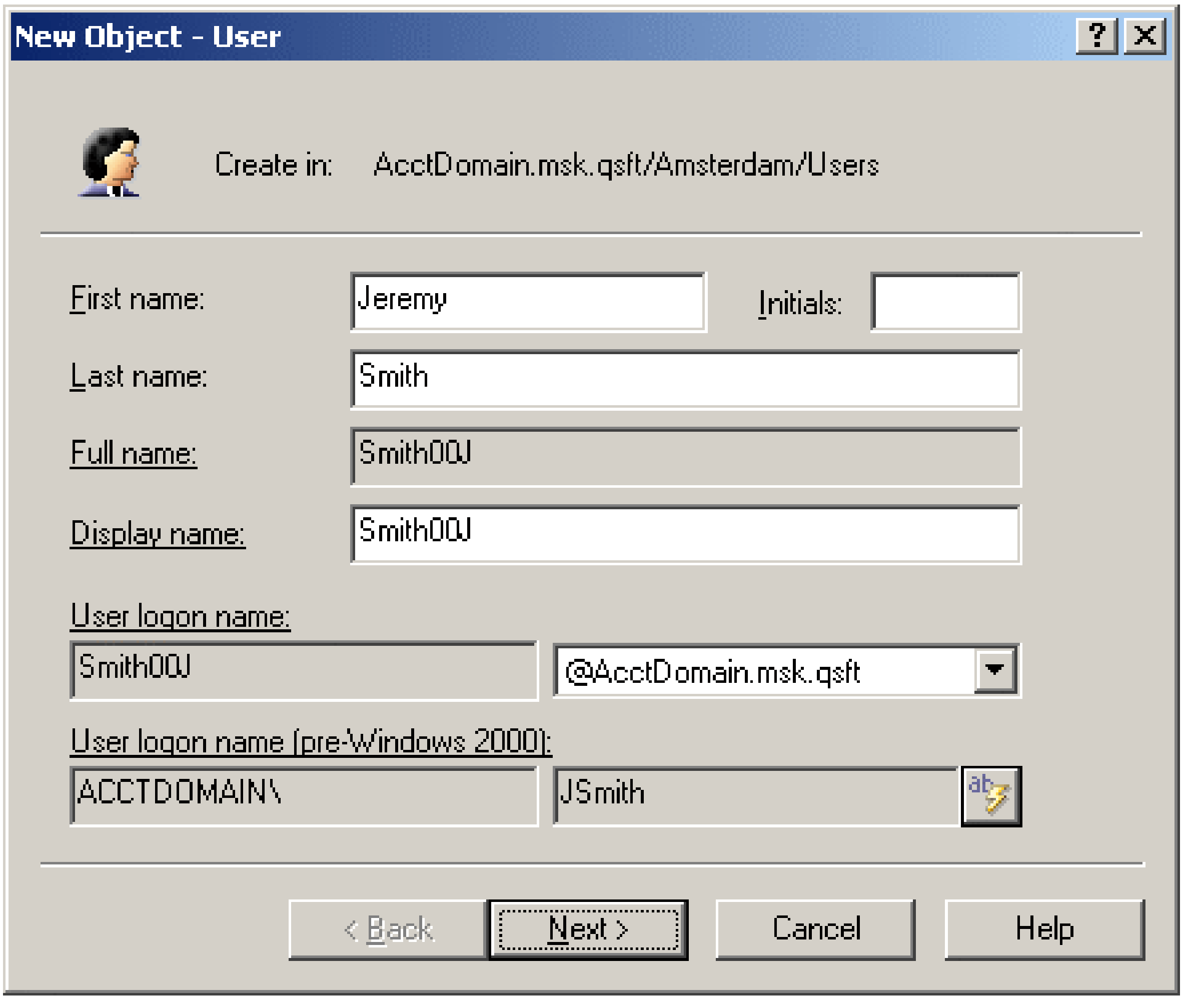
> NOTE: Policy Object settings specific to Azure cloud-only objects (such as cloud-only Azure users, guest users, or contacts) are available only if your Active Roles deployment is licensed for managing cloud-only Azure objects. Contact One Identity support for more information.
>
> Also, Policy Objects 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 the Property Generation and Validation policy works

When creating or modifying an object, Active Roles checks whether the property values satisfy the criteria defined in the policy. If they do not, Active Roles prevents you from creating or modifying the object.

In the **Object Creation Wizard** and **Properties** dialogs, 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 32: Object creation



New Object - User

Create in: AcctDomain.msk.qsft/Amsterdam/Users

First name: Jeremy

Initials:

Last name: Smith

Full name: Smith00J

Display name: Smith00J

User logon name:

Smith00J @AcctDomain.msk.qsft

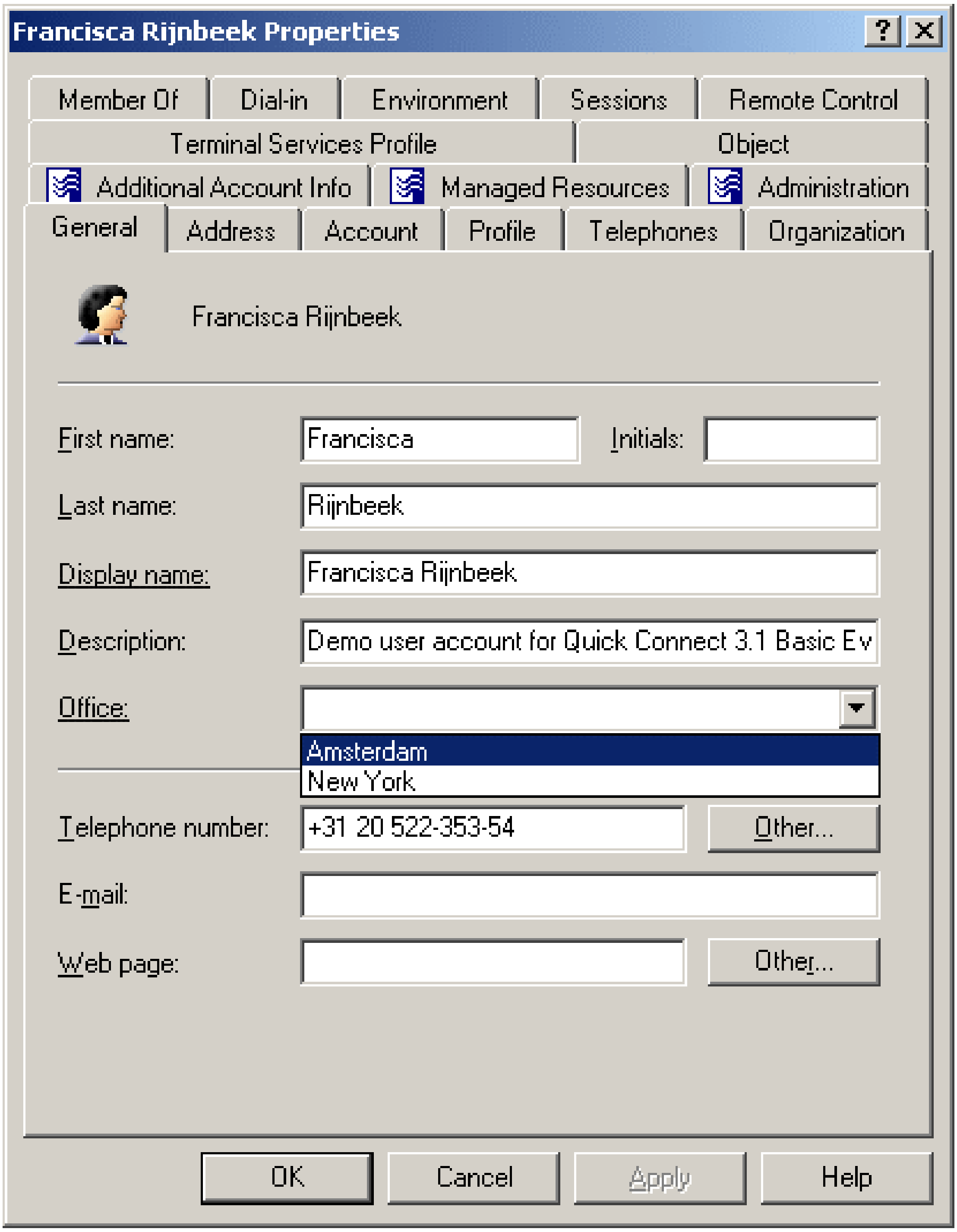
User logon name (pre-Windows 2000):

ACCTDOMAIN\ JSmith

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, where the **Office** box provides a list of acceptable values that are prescribed by policy.

Figure 33: Acceptable values for a policy



Francisca Rijnbeek

First name: Francisca

Initials:

Last name: Rijnbeek

Display name: Francisca Rijnbeek

Description: Demo user account for Quick Connect 3.1 Basic Ev

Office:

- Amsterdam
- New York

Telephone number: +31 20 522-353-54

E-mail:

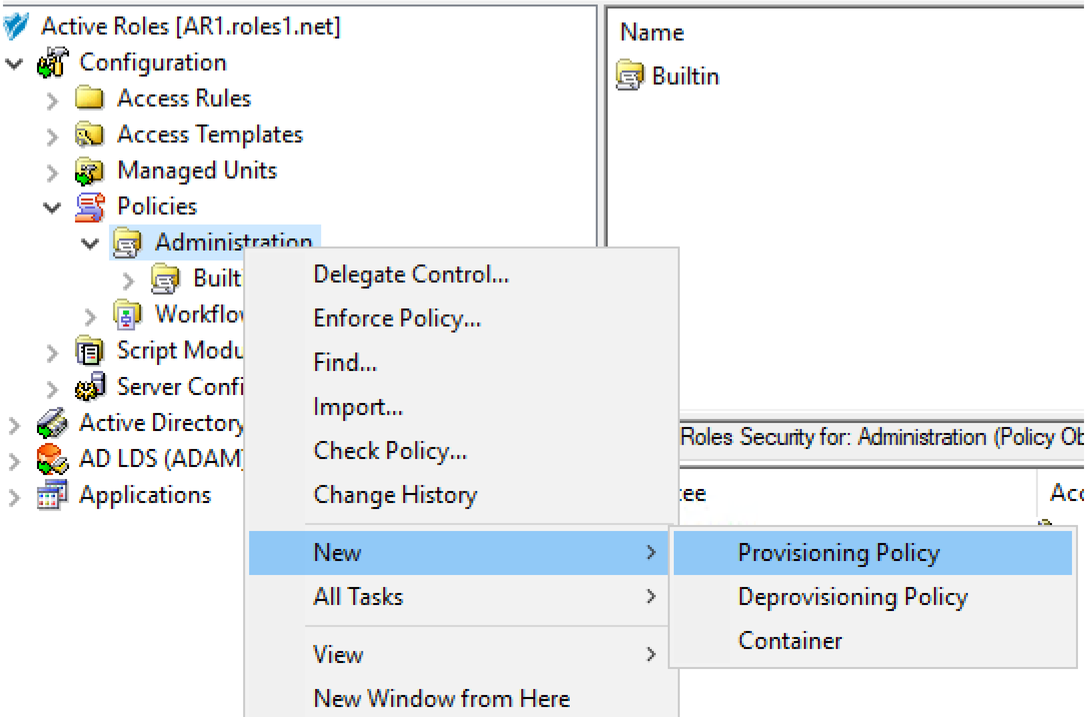
Web page:

Configuring a Property Generation and Validation policy

To configure a Property Generation and Validation policy via the Active Roles Console (also known as the MMC interface), 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 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. To complete creating the Policy Object, click **Next**, then **Finish**.

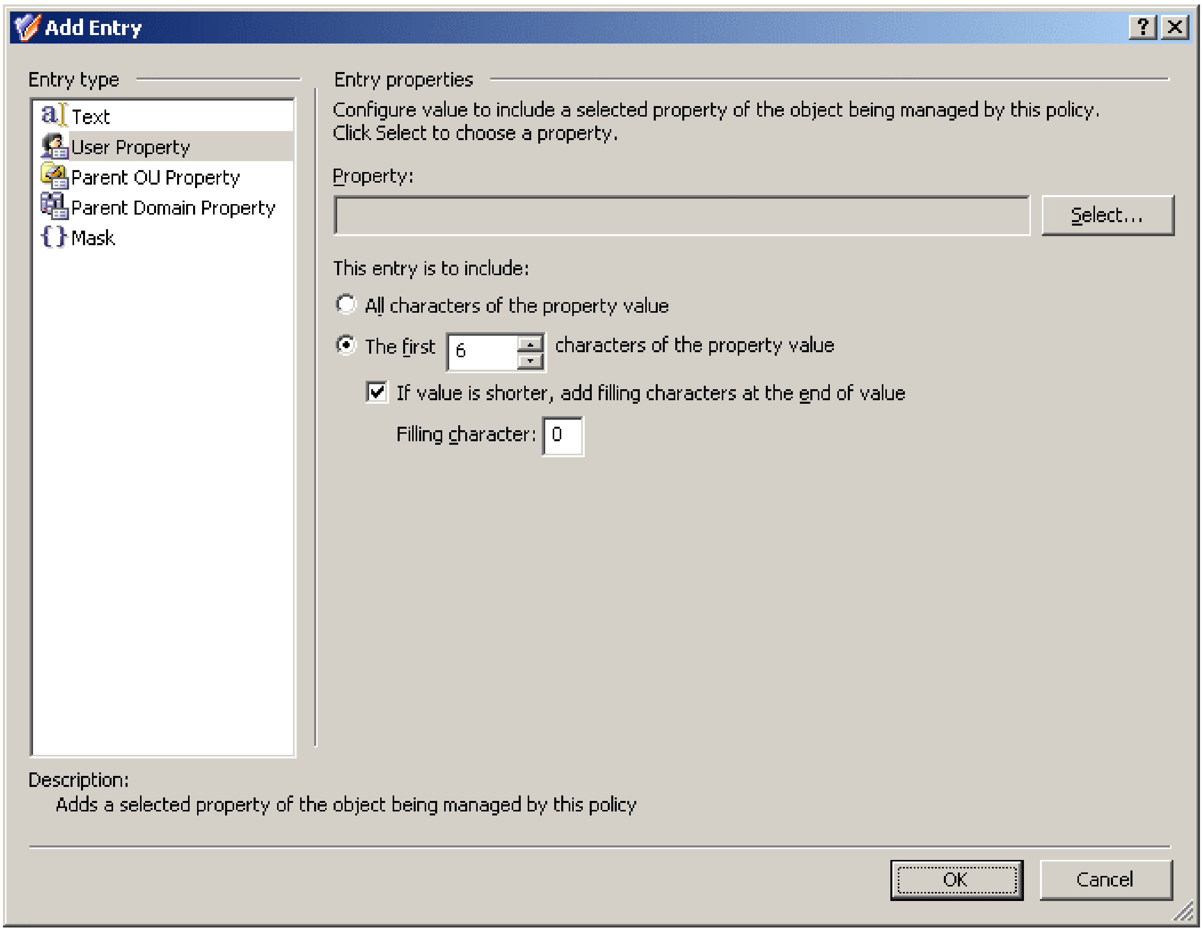
Entry type: Text

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 as follows.



Add Entry

Entry type

- Text
- User Property
- Parent OU Property
- Parent Domain Property
- Mask

Entry properties

Configure value to include a selected property of the object being managed by this policy. Click Select to choose a property.

Property:

This entry is to include:

○ All characters of the property value

● The first 6 characters of the property value

☑ If value is shorter, add filling characters at the end of value

Filling character: 0

Description:
Adds a selected property of the object being managed by this policy

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.

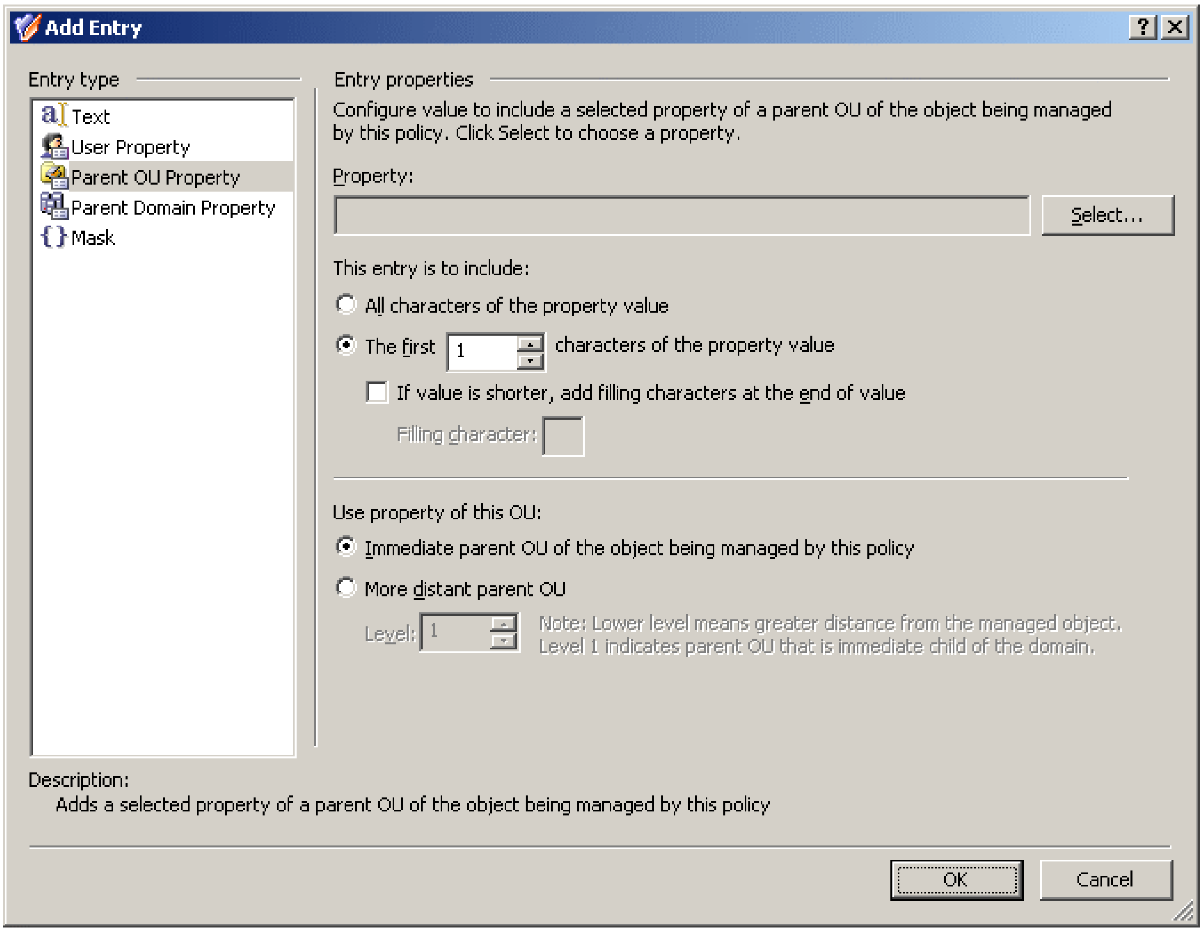
In the latter case, you can select the **If value is shorter, add filling characters at the end of value** check box, and type a character in the **Filling character** box. This character will fill the missing characters in the value of the object property if the value is shorter than specified in the 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.

Entry type: Parent OU Property

When you select **Parent OU Property** under **Entry type** in the **Add Entry** window, the **Entry properties** area looks as follows.

Figure 34: Add Entry: Parent OU Property



Add Entry

Entry type

- Text
- User Property
- Parent OU Property
- Parent Domain Property
- Mask

Entry properties

Configure value to include a selected property of a parent OU of the object being managed by this policy. Click Select to choose a property.

Property:

This entry is to include:

○ All characters of the property value

● The first 1 characters of the property value

□ If value is shorter, add filling characters at the end of value

Filling character:

Use property of this OU:

● Immediate parent OU of the object being managed by this policy

○ More distant parent OU

Level: 1 Note: Lower level means greater distance from the managed object. Level 1 indicates parent OU that is immediate child of the domain.

Description:
Adds a selected property of a parent OU of the object being managed by this policy

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

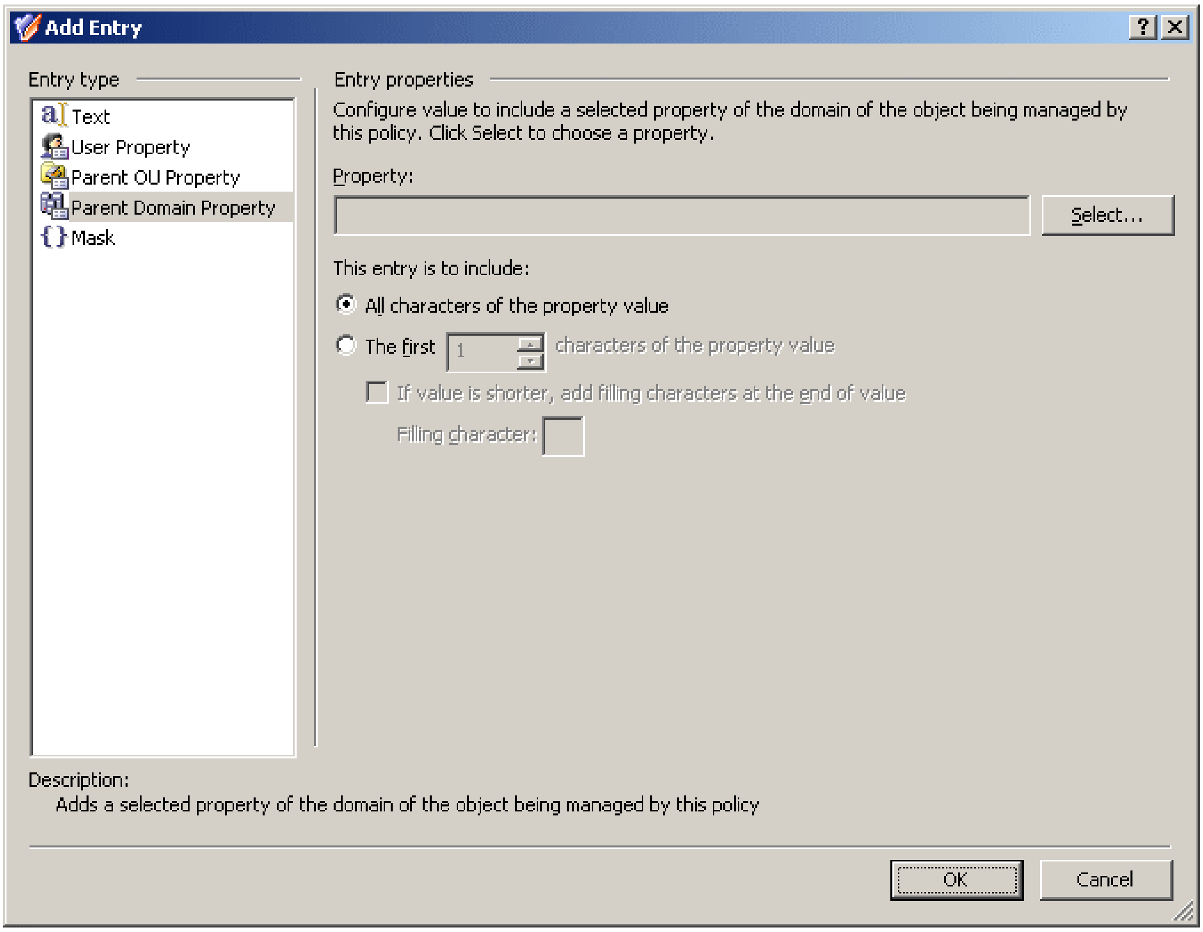
More distant parent OU and then, in the **Level** box, specify the level of the OU. Lower level means greater distance from the managed object in the hierarchy of containers above that object. OU level 1 is an immediate child OU of the domain.

When you are done configuring an entry, click **OK** to close the **Add Entry** window. The entry is added to the **Configure Value** dialog.

Entry type: Parent Domain Property

When you select **Parent Domain Property** under **Entry type** in the **Add Entry** window, the **Entry properties** area looks as follows.

Figure 35: Add Entry: Parent Domain Property



Configure value to include a selected property of the domain of the object being managed by this policy. Click Select to choose a property.

Property:

Select...

This entry is to include:

● All characters of the property value

○ The first 1 characters of the property value

□ If value is shorter, add filling characters at the end of value

Filling character:

Description:
Adds a selected property of the domain of the object being managed by this policy

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.

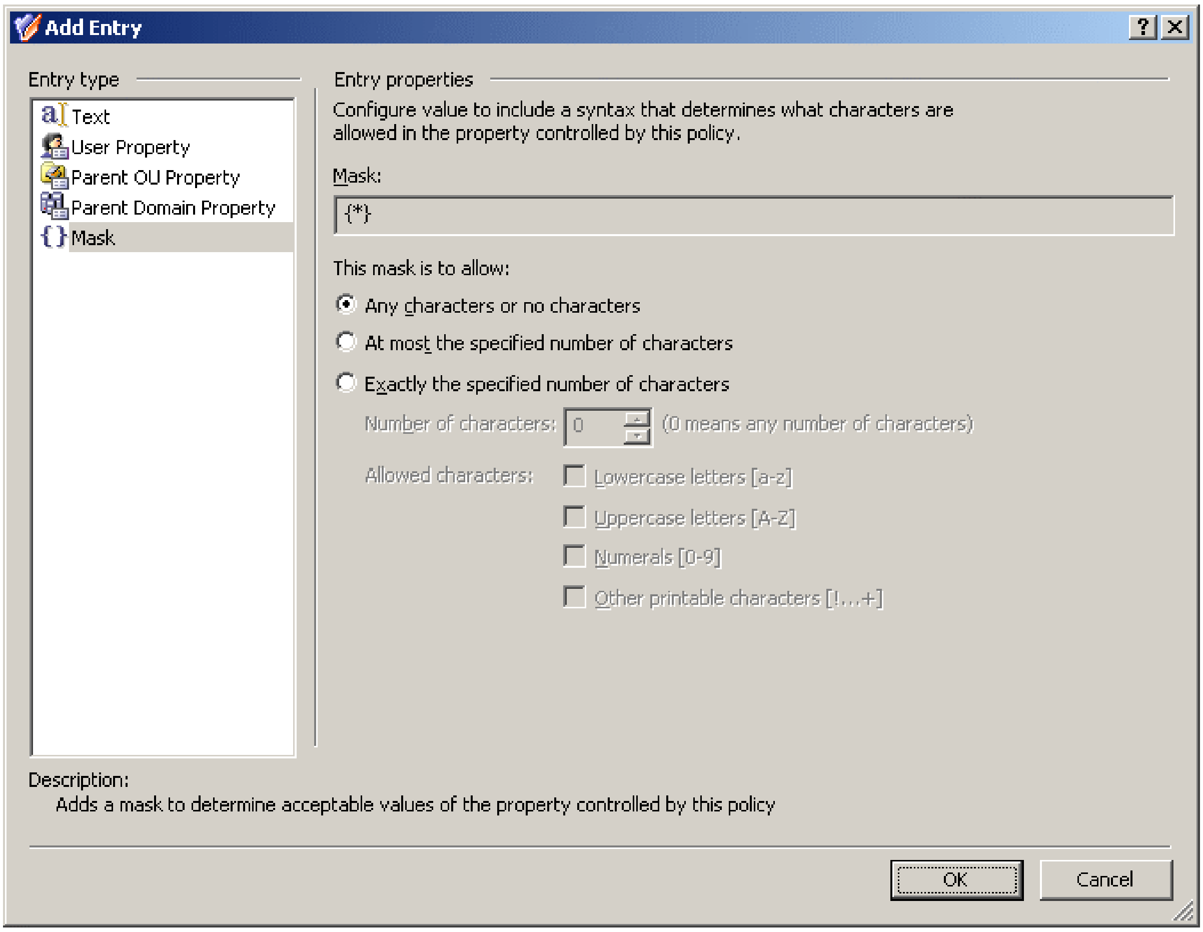
In the latter case, you can select the **If value is shorter, add filling characters at the end of value** check box, and type a character in the **Filling character** box. This character will fill the missing characters in the value of the domain property if the value is shorter than specified in the 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.

Entry type: Mask

When you select **Mask** under **Entry type** in the **Add Entry** window, the **Entry properties** area looks as follows.

Figure 36: Add Entry: Mask



Add Entry

Entry type

- Text
- User Property
- Parent OU Property
- Parent Domain Property
- Mask

Entry properties

Configure value to include a syntax that determines what characters are allowed in the property controlled by this policy.

Mask: {*}

This mask is to allow:

● Any characters or no characters

○ At most the specified number of characters

○ Exactly the specified number of characters

Number of characters: 0 (0 means any number of characters)

Allowed characters: □ Lowercase letters [a-z] □ Uppercase letters [A-Z] □ Numerals [0-9] □ Other printable characters [!...+]

Description:
Adds a mask to determine acceptable values of the property controlled by this policy

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.

Configuring entries

Use the following step-by-step instructions to configure an entry in the **Add Entry** dialog. 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 enter 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**: Allows the entry to include any series of characters.
- **At most the specified number of characters**: Specify the maximum number of allowed characters the entry may include.
- **Exactly the specified number of characters**: Specify an exact number of allowed characters that the entry must include.

3. If you selected **At most the specified number of characters** or **Exactly the specified number of characters**, then in **Number of characters**, specify the number of characters allowed in this entry.

   If you selected **At most the specified number of characters**, the entry may include any number of characters not exceeding the number specified.

   If you selected **Exactly the specified number of characters**, 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 **Initiator ID built using a custom rule**, click **Configure**, and use the **Configure Value** dialog 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).

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: Consider the following when configuring entries:
>
> - You may need to configure an entry when configuring a policy such as Property Generation and Validation, User Logon Name Generation, Group Membership AutoProvisioning, E-mail Alias Generation, User Account Deprovisioning, or Group Object Deprovisioning.
> - The contents of the **Entry Type** list in the **Add Entry** dialog depend on 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.

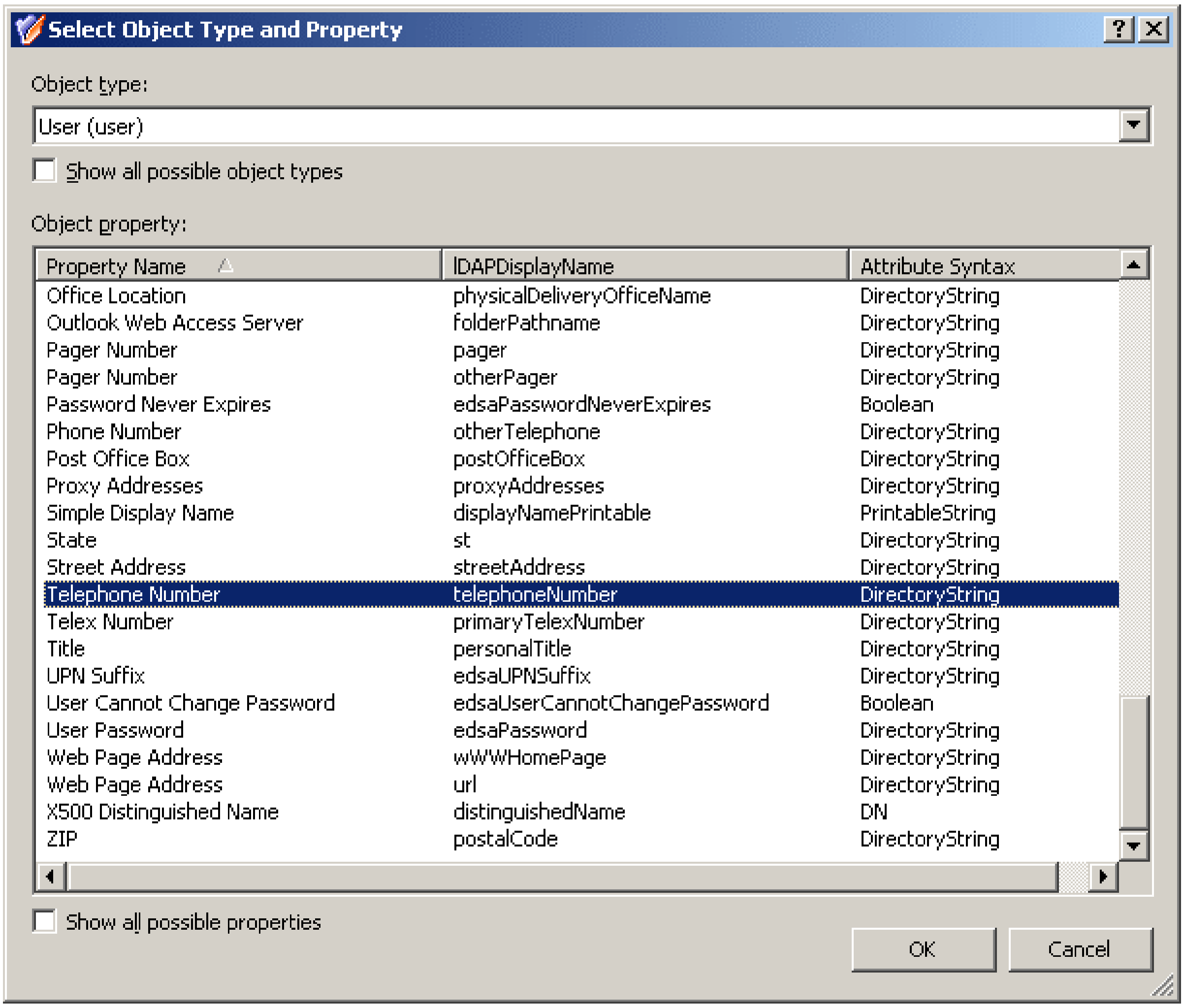
Creating and configuring the Policy Object

You can create and configure the Policy Object you need by using the **New Provisioning Policy Object Wizard**. For more information about the wizard, see Creating a Policy Object.

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, select **User** from the **Object type** list, and click **Telephone number** in the **Object property** list, as shown in the following figure.

Figure 37: Select Object type and property



Object type:

User (user)

☐ Show all possible object types

Object property:

Property Name	lDAPDisplayName	Attribute Syntax
Office Location	physicalDeliveryOfficeName	DirectoryString
Outlook Web Access Server	folderPathname	DirectoryString
Pager Number	pager	DirectoryString
Pager Number	otherPager	DirectoryString
Password Never Expires	edsaPasswordNeverExpires	Boolean
Phone Number	otherTelephone	DirectoryString
Post Office Box	postOfficeBox	DirectoryString
Proxy Addresses	proxyAddresses	DirectoryString
Simple Display Name	displayNamePrintable	PrintableString
State	st	DirectoryString
Street Address	streetAddress	DirectoryString
Telephone Number	telephoneNumber	DirectoryString
Telex Number	primaryTelexNumber	DirectoryString
Title	personalTitle	DirectoryString
UPN Suffix	edsaUPNSuffix	DirectoryString
User Cannot Change Password	edsaUserCannotChangePassword	Boolean
User Password	edsaPassword	DirectoryString
Web Page Address	wWWHomePage	DirectoryString
Web Page Address	url	DirectoryString
X500 Distinguished Name	distinguishedName	DN
ZIP	postalCode	DirectoryString

☐ Show all possible properties

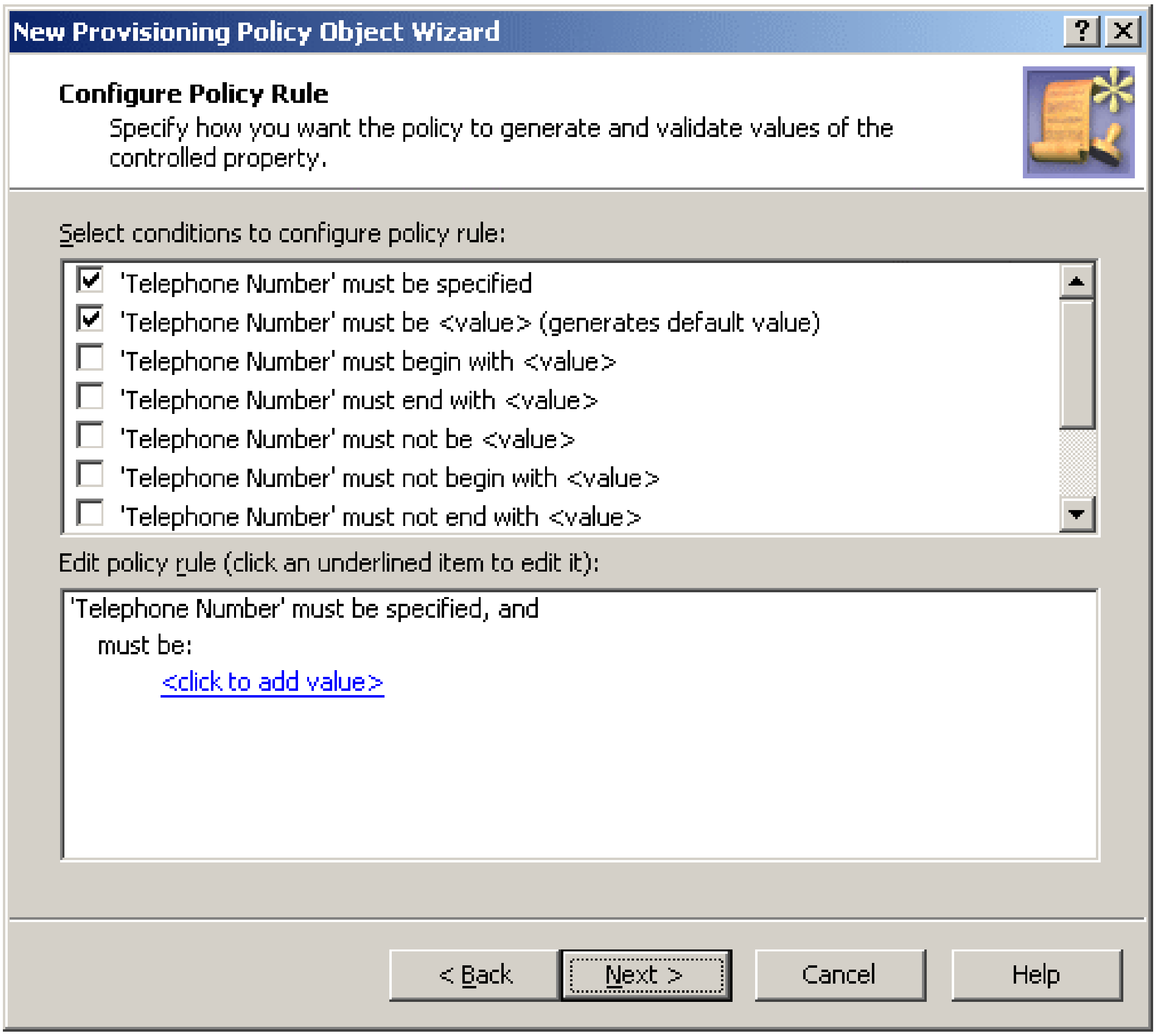
Click **OK**, then click **Next**.

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 38: Configure policy rules



New Provisioning Policy Object Wizard

Configure Policy Rule

Specify how you want the policy to generate and validate values of the controlled property.

Select conditions to configure policy rule:

- ☑ 'Telephone Number' must be specified
- ☑ 'Telephone Number' must be <value> (generates default value)
- □ 'Telephone Number' must begin with <value>
- □ 'Telephone Number' must end with <value>
- □ 'Telephone Number' must not be <value>
- □ 'Telephone Number' must not begin with <value>
- □ 'Telephone Number' must not end with <value>

Edit policy rule (click an underlined item to edit it):

'Telephone Number' must be specified, and
must be:
<click to add value>

The next phase is to configure the value.

Click **<click to add value>**. In the **Add Value** dialog, click **Configure**. In the **Configure Value** dialog, click **Add**. In the **Add Entry** window, under **Entry type**, click **Mask**.

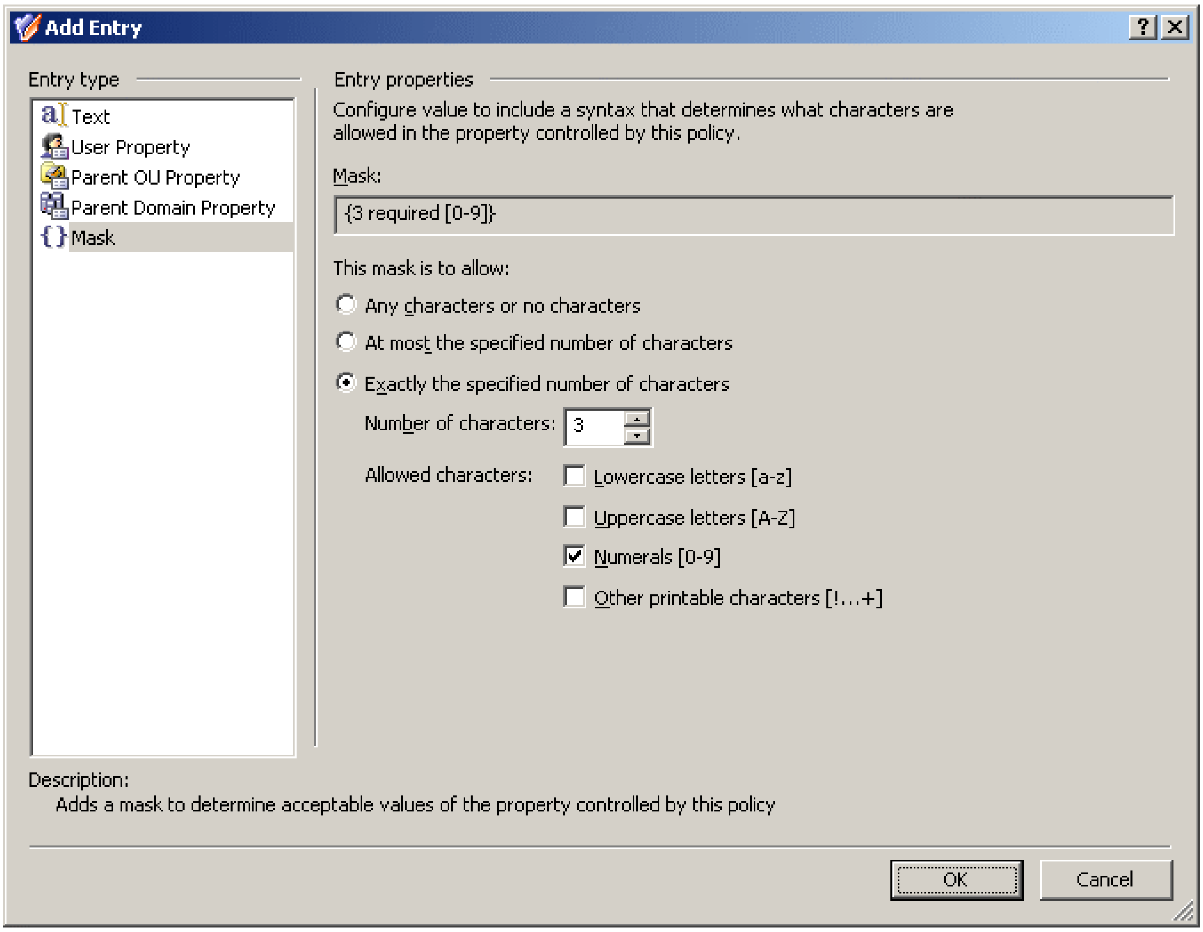
Now you can use the **Entry properties** area in the **Add Entry** window to configure a mask.

The format consists of four groups of numerals divided by certain characters: spaces, 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 39: Add entry



Add Entry

Entry type
- Text
- User Property
- Parent OU Property
- Parent Domain Property
- Mask

Entry properties

Configure value to include a syntax that determines what characters are allowed in the property controlled by this policy.

Mask: {3 required [0-9]}

This mask is to allow:

○ Any characters or no characters

○ At most the specified number of characters

● Exactly the specified number of characters

Number of characters: 3

Allowed characters: □ Lowercase letters [a-z]

□ Uppercase letters [A-Z]

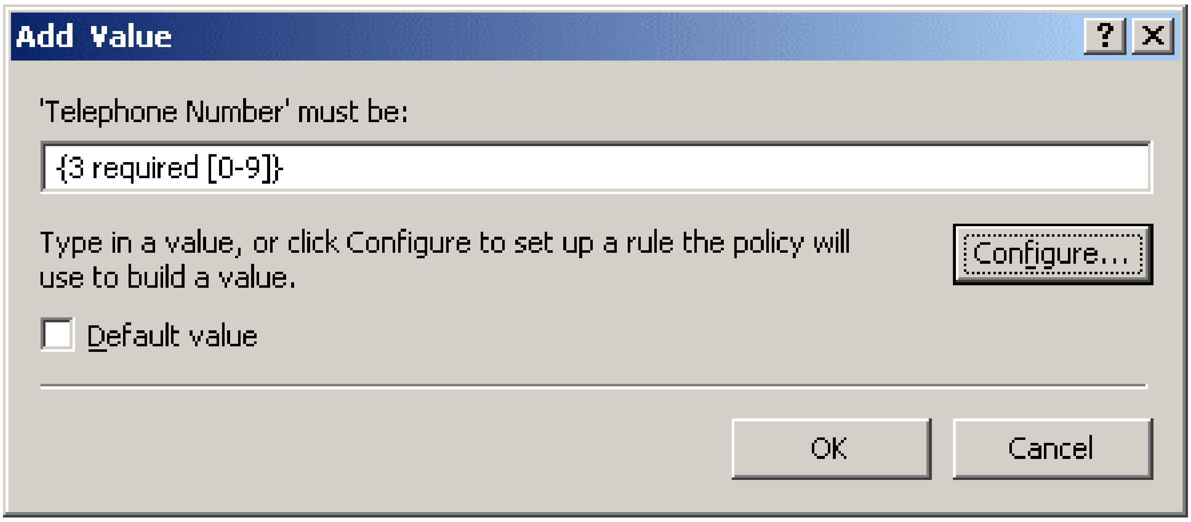
☑ Numerals [0-9]

□ Other printable characters [!...+]

Description:
Adds a mask to determine acceptable values of the property controlled by this policy

Click **OK** to close the **Add Entry** window. Then, click **OK** to close the **Configure Value** dialog. As a result, the **Add Value** dialog looks as shown in the following figure.

Figure 40: Add value dialog



Add Value

'Telephone Number' must be:

{3 required [0-9]}

Type in a value, or click Configure to set up a rule the policy will use to build a value.

□ Default value

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. 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. Click **Next** and follow the instructions in the wizard to create the Policy Object.

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.

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 5: 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.

Configuring the Policy Object

You can configure the Policy Object you need by modifying the Policy Object that implements Scenario 1: Using mask to control phone number format.

Display the **Properties** dialog 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.

The **Policy Rule** tab in the **Property Generation and Validation Policy Properties** dialog 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 Using regular expressions.

Table 6: 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, enter **`^\+([0-9]+ )+[0-9]+$`**, and click **OK**.
3. On the **Policy Rule** tab, in the lower box, click **<click to add value>**.
4. In the **Add Value** dialog, enter **`^\+([0-9]+ )+x[0-9]+$`**, and click **OK**.
5. Click **OK** to close the **Property Generation and Validation Policy Properties** dialog.

Applying the Policy Object

You can apply the Policy Object without closing its **Properties** dialog. 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 to.

   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 for the Policy Object.

For more information on how to apply a Policy Object, see Applying Policy Objects and Managing policy scope.

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 the User Logon Name Generation 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, 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.

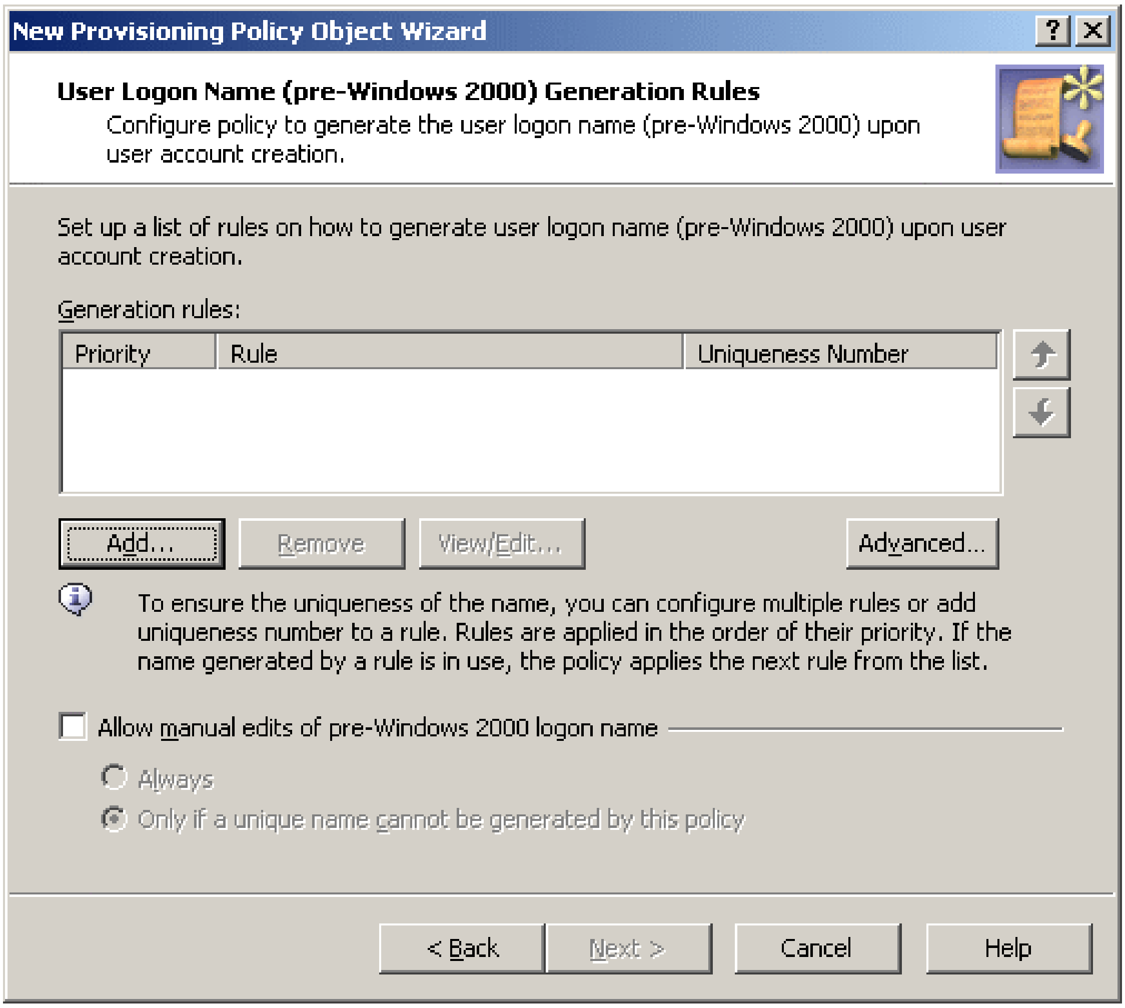
Configuring a User Logon Name Generation policy

You can configure a new User Logon Name Generation policy with the New Provisioning Policy Object Wizard of the Active Roles Console.

To configure a User Logon Name Generation policy

1. On the **Policy to Configure** page, select **User Logon Name Generation**, then click **Next**.

   Figure 41: New Provisioning Policy Object Wizard



New Provisioning Policy Object Wizard

User Logon Name (pre-Windows 2000) Generation Rules

Configure policy to generate the user logon name (pre-Windows 2000) upon user account creation.

Set up a list of rules on how to generate user logon name (pre-Windows 2000) upon user account creation.

Generation rules:

Priority	Rule	Uniqueness Number

To ensure the uniqueness of the name, you can configure multiple rules or add uniqueness number to a rule. Rules are applied in the order of their priority. If the name generated by a rule is in use, the policy applies the next rule from the list.

☐ Allow manual edits of pre-Windows 2000 logon name

○ Always

● Only if a unique name cannot be generated by this policy

2. 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, discussed in Configuring a Property Generation and Validation policy. Use that dialog 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.
- **Remove**: Deletes the rules you select from the list.
- **View/Edit**: Opens the **Configure Value** dialog for the rule you select from the list. Modify the selected rule by managing the list of entries in that dialog.
- **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. Complete the **Advanced** dialog by using the procedure outlined later in this topic.
    - If you want the logon name to be allowed for manual edit, select the **Allow manual edits of pre-Windows 2000 logon name** check box. 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.

3. Click **Next**.
4. On the **Enforce Policy** page, specify the objects to which this Policy Object will be applied. To do so, click **Add**, and use the **Select Objects** dialog to locate and select the objects you want.
5. Click **Next**, then click **Finish**.

To complete the Configure Value dialog

1. Click **Add**.
2. Configure an entry to include in the value. For more information, see Configuring entries.

3. In the **Configure Value** dialog, add more entries, delete or edit existing ones, then click **OK**.

To complete the Advanced dialog

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. (Optional) 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.

Configuring a logon name generation rule

To configure a generation rule, click **Add** beneath the **Generation rules** list. This displays the **Configure Value** dialog, 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. 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, then configure the entry. The following table summarizes the available types of entries.

Table 7: 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.

Type of entry	Description
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 Configuring a Property Generation and Validation policy to configure an entry of any of these types:

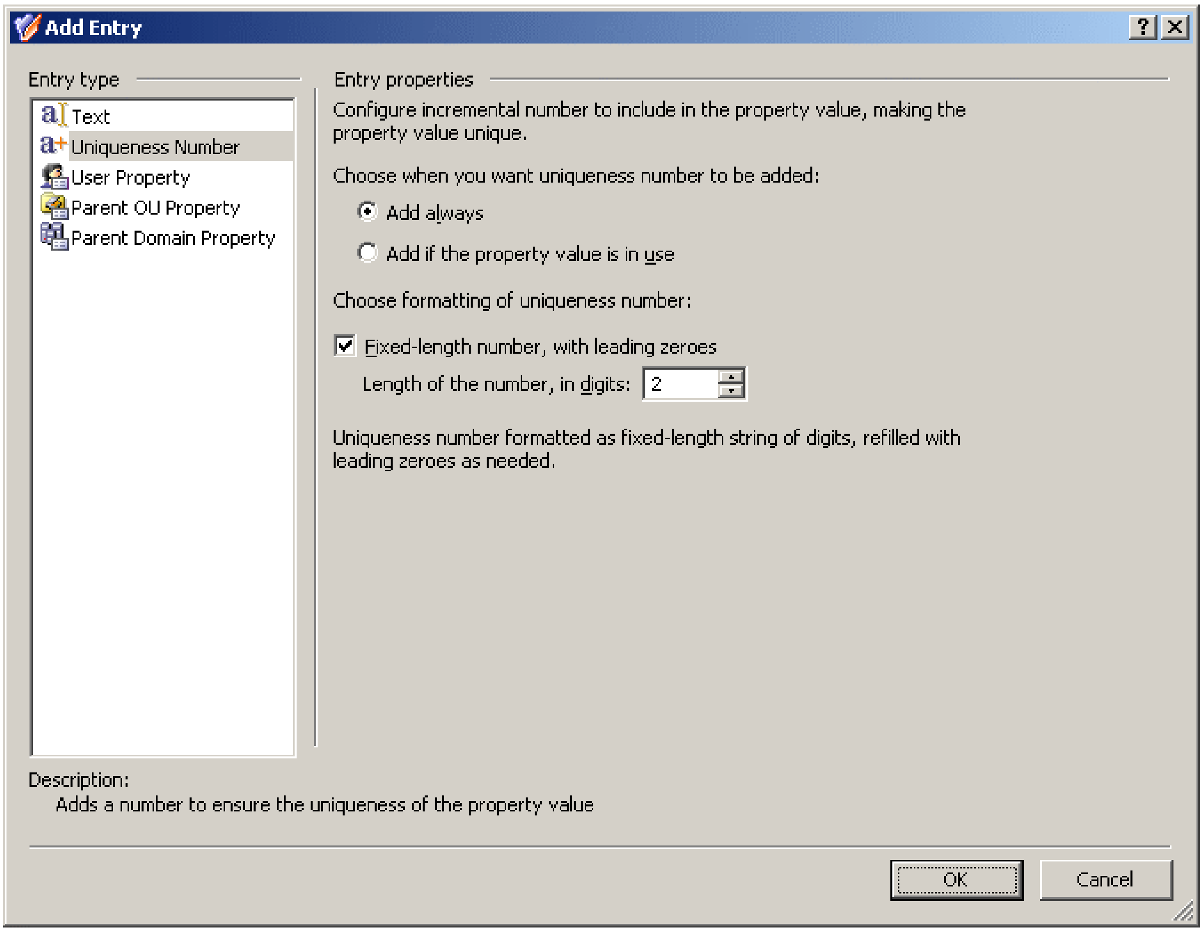
- **Text**: See Entry type: Text.
- **User Property**: See Entry type: <Object> Property.
- **Parent OU Property**: See Entry type: Parent OU Property.
- **Parent Domain Property**: Refer to the Entry type: Parent Domain Property.

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 as follows.

Figure 42: Entry Type: Uniqueness Number



Configure incremental number to include in the property value, making the property value unique.

Choose when you want uniqueness number to be added:

● Add always

○ Add if the property value is in use

Choose formatting of uniqueness number:

☑ Fixed-length number, with leading zeroes

Length of the number, in digits: 2

Uniqueness number formatted as fixed-length string of digits, refilled with leading zeroes as needed.

Description:
Adds a number to ensure the uniqueness of the property value

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, and so on.

When you are done configuring an entry, click **OK** to close the **Add Entry** window. The entry is added to the **Configure Value** dialog.

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 **Generate** causes the policy to add a uniqueness number to the name.

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

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.

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 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, 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**.

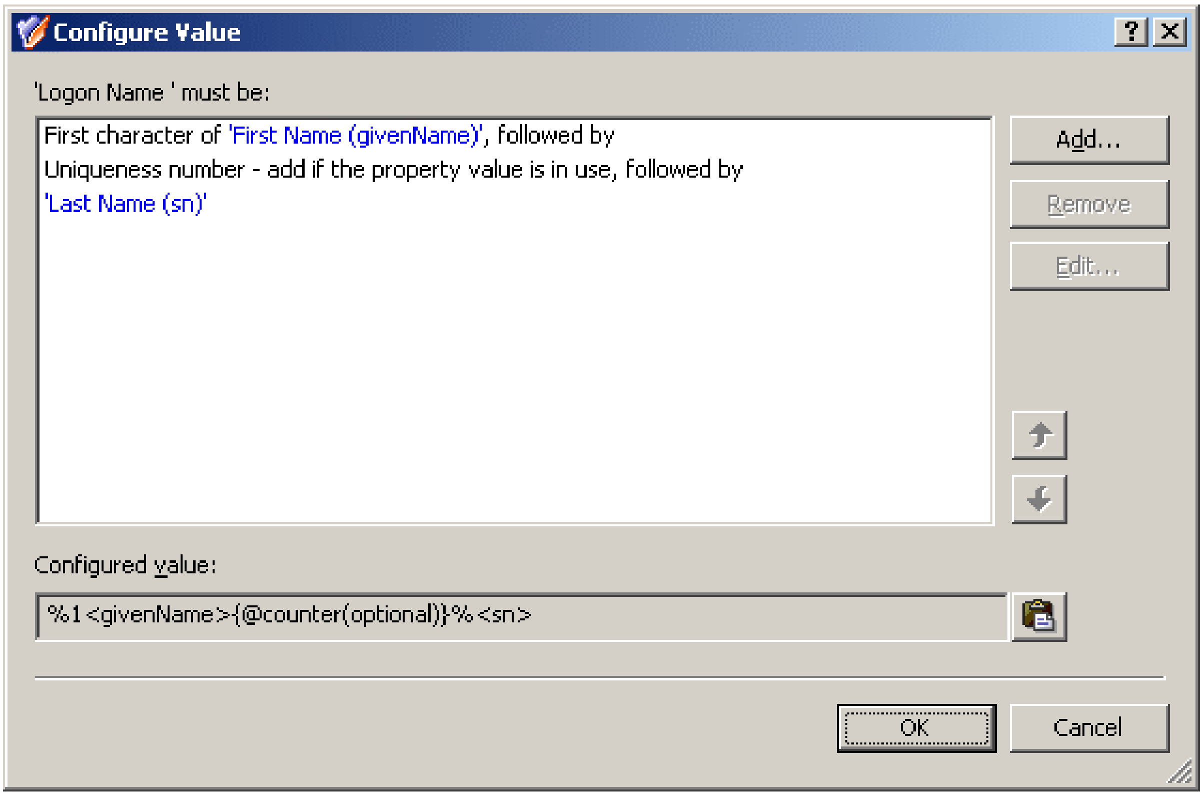
   b. Under **Entry properties**, click **Select**.

   c. In the **Select Object Property** window, click **Last Name** in the **Object property** list, then click **OK**.

   d. Click **OK**.

After you complete these steps, the list of entries in the **Configure Value** dialog should look like the following figure.

Figure 43: Configure Value



'Logon Name ' must be:

First character of 'First Name (givenName)', followed by
Uniqueness number - add if the property value is in use, followed by
'Last Name (sn)'

Configured value:

%1<givenName>{@counter(optional)}%<sn>

Click **OK** to close the **Configure Value** dialog.

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 **Advanced**. In the **Advanced** dialog, in the **Maximum length, in characters** box, type **8**, then click **OK**.

Click **Next** and follow the instructions in the wizard to create the Policy Object.

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.

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 **Generate** causes the policy to apply a subsequent rule.

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

Configuring the Policy Object

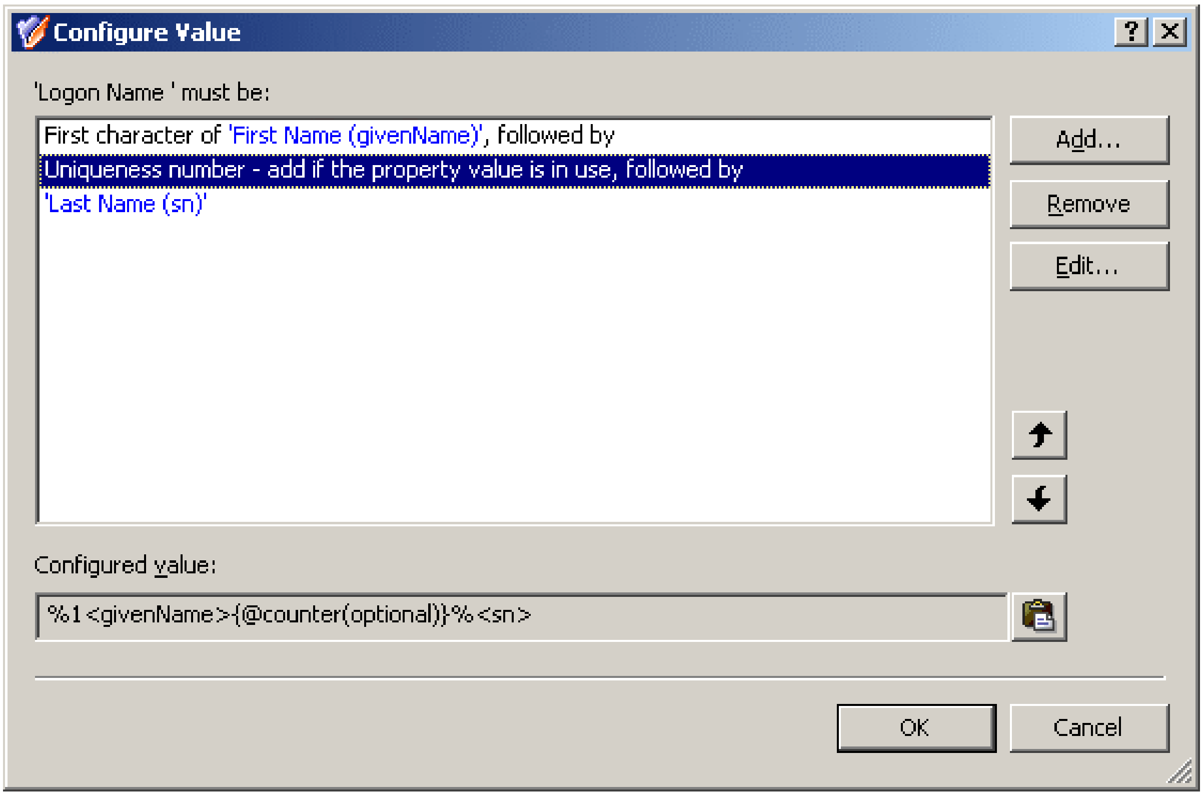
You can configure the Policy Object you need by modifying the Policy Object that implements the previous scenario; see Scenario 1: Using uniqueness number.

Display the **Properties** dialog for that Policy Object and go to **Policies**. Then, select the policy from the list, and click **View/Edit** to display the **User Logon Name Generation Policy Properties** dialog.

The **Generation Rules** tab in the **User Logon Name Generation Policy Properties** dialog 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. Then, select the uniqueness number entry as shown in the following figure, and click **Remove**.

Figure 44: Configure Value



First character of 'First Name (givenName)', followed by
Uniqueness number - add if the property value is in use, followed by
'Last Name (sn)'

%1<givenName>{@counter(optional)}%<sn>

Click **OK** to close the **Configure Value** dialog.

Next, configure the additional policy rules as follows.

1. On the **Generation Rules** tab, click **Add** to display the **Configure Value** dialog.
2. In the **Configure Value** dialog, 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, 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**.

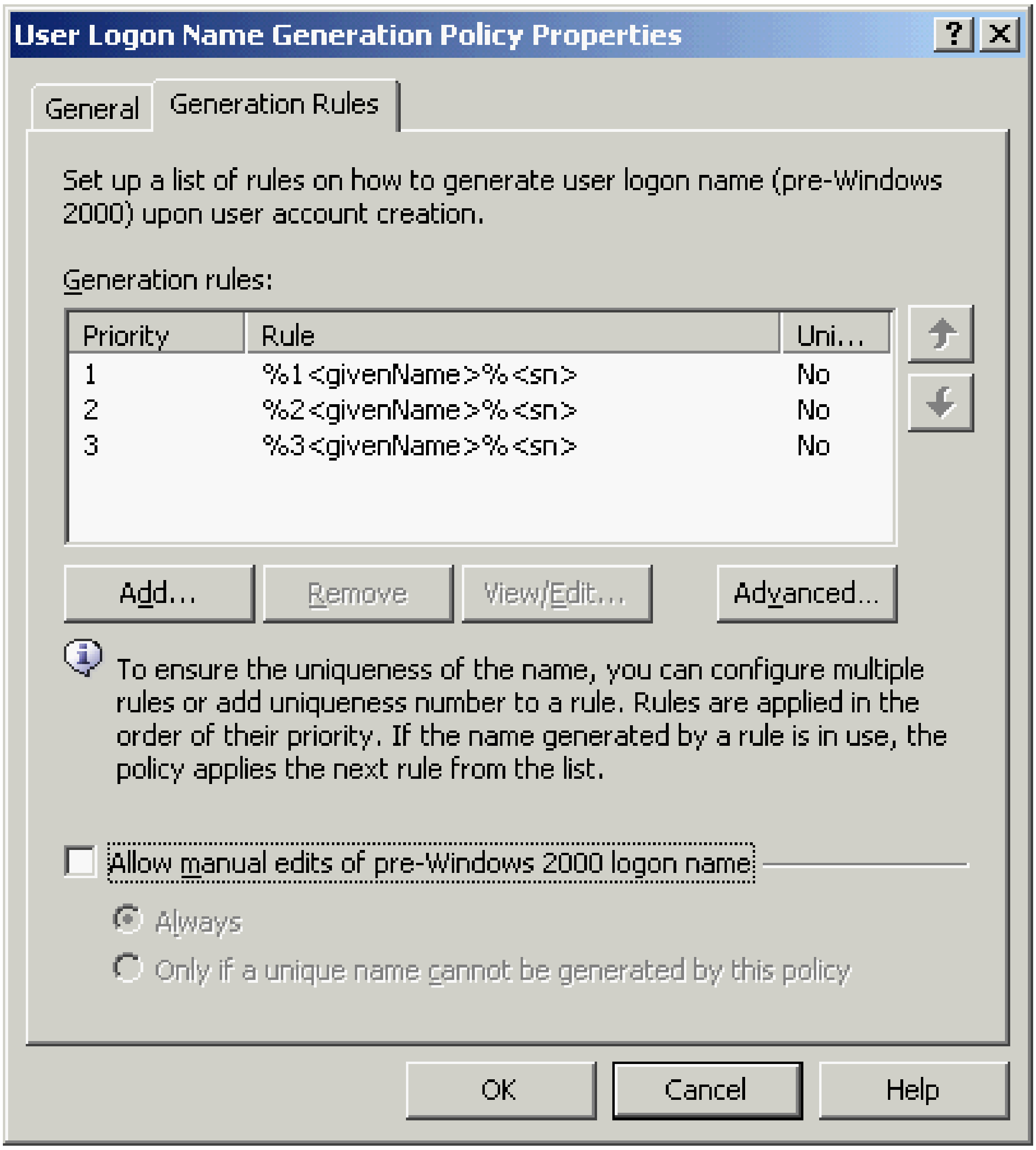
d. Click **OK** to close the **Add Entry** window.

6. Click **OK** to close the **Configure Value** dialog.
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 45: Generation rules



User Logon Name Generation Policy Properties

Set up a list of rules on how to generate user logon name (pre-Windows 2000) upon user account creation.

Generation rules:

Priority	Rule	Uni...
1	%1<givenName>%<sn>	No
2	%2<givenName>%<sn>	No
3	%3<givenName>%<sn>	No

To ensure the uniqueness of the name, you can configure multiple rules or add uniqueness number to a rule. Rules are applied in the order of their priority. If the name generated by a rule is in use, the policy applies the next rule from the list.

□ Allow manual edits of pre-Windows 2000 logon name

● Always

○ Only if a unique name cannot be generated by this policy

Click **OK** to close the **User Logon Name Generation Policy Properties** dialog.

Applying the Policy Object

You can apply the Policy Object without closing its **Properties** dialog. 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 to.

   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 for the Policy Object.

For more information on how to apply a Policy Object, see Applying Policy Objects and Managing policy scope.

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 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 the Group Membership AutoProvisioning 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.

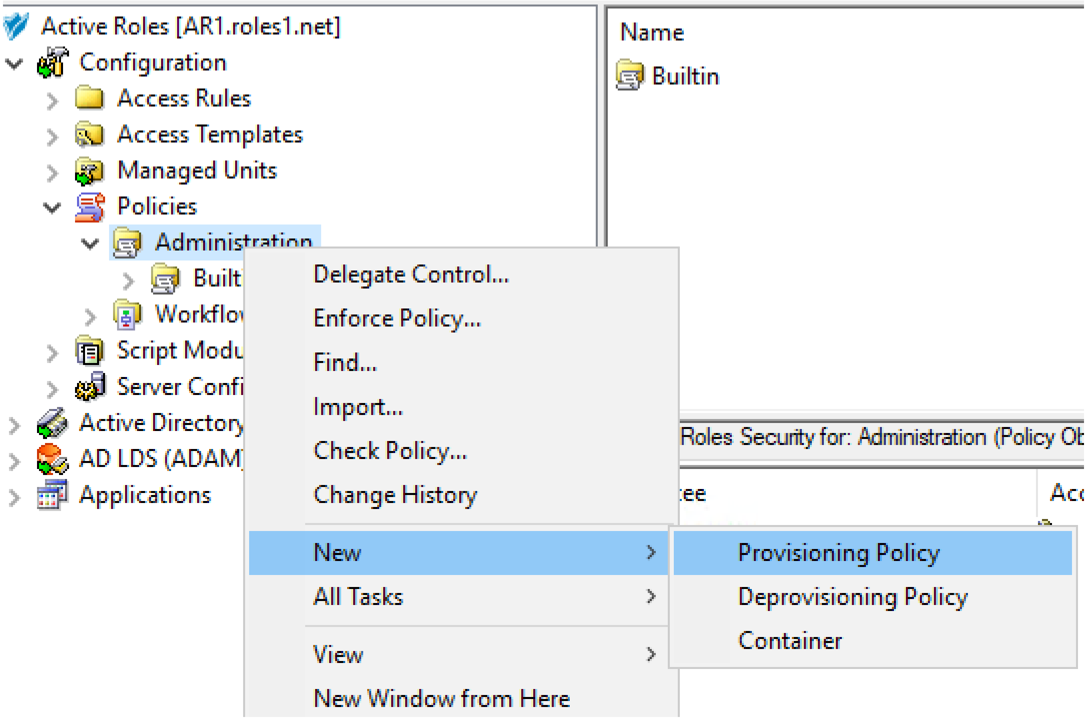
Whereas the Dynamic Groups feature delivers a rules-based mechanism for managing a group membership list as a whole, a Group Membership AutoProvisioning policy allows the administrator to define membership rules on a per-user basis. This policy automates the process of adding particular users to particular groups without affecting the other members of those groups.

Configuring a Group Membership AutoProvisioning policy

To configure a Group Membership AutoProvisioning policy via the Active Roles Console (also known as the MMC interface), 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 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 has the **Department** property set to **Sales**, Active Roles automatically adds that account in the **Sales** group.

Configuring the Group Membership AutoProvisioning 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.

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**).

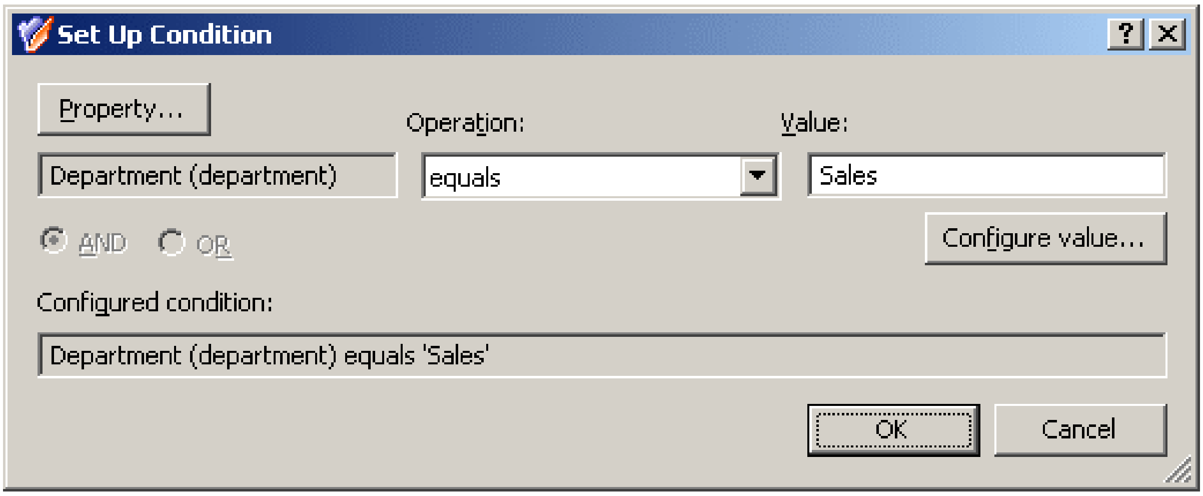
On the **Policy Conditions** page, click **Add** to display the **Set Up Condition** dialog.

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 must look as follows.

Figure 46: Set Up Condition



Set Up Condition

Property...

Operation: equals

Value: Sales

Department (department)

● AND ○ OR

Configure value...

Configured condition: Department (department) equals 'Sales'

Click **OK** to close the **Set Up Condition** dialog.

On the **Policy Conditions** page, click **Next**.

On the **Policy Action** page, click **Add object to groups if object satisfies policy conditions**, then click **Next**.

On the **Group Selection** page, click **Add** and use the **Select Objects** dialog 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.

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.

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 the Exchange Mailbox AutoProvisioning 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.

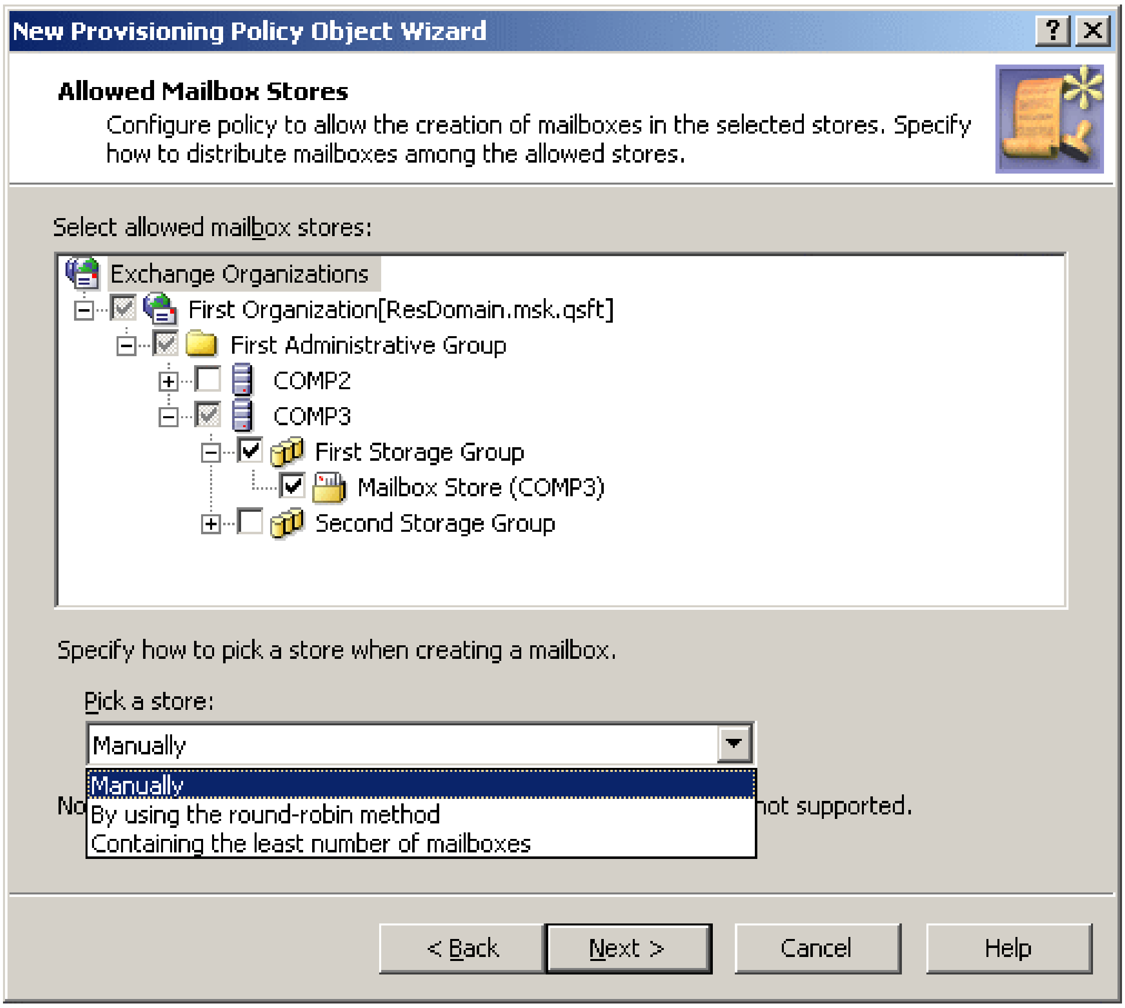
Configuring an Exchange Mailbox AutoProvisioning policy

You can configure a new Exchange Mailbox AutoProvisioning policy with the Active Roles Console.

To configure an Exchange Mailbox AutoProvisioning Policy

1. On the **Policy to Configure** page, select **Exchange Mailbox AutoProvisioning**, then click **Next**.

Figure 47: Allowed mailbox stores



New Provisioning Policy Object Wizard

Allowed Mailbox Stores

Configure policy to allow the creation of mailboxes in the selected stores. Specify how to distribute mailboxes among the allowed stores.

Select allowed mailbox stores:

- Exchange Organizations
  - ☑ First Organization[ResDomain.msk.qsft]
    - ☑ First Administrative Group
      - □ COMP2
      - ☑ COMP3
        - ☑ First Storage Group
          - ☑ Mailbox Store (COMP3)
        - □ Second Storage Group

Specify how to pick a store when creating a mailbox.

Pick a store: Manually

- Manually
- By using the round-robin method
- Containing the least number of mailboxes

No … not supported.

2. Under **Select allowed mailbox stores**, select servers and stores to be allowed for mailbox creation, then click **Next**.

   In case of multiple stores, from the **Pick a store** list, select one of following options:

   - **Manually**
   - **By using the round-robin method**
   - **Containing the least number of mailboxes**

   For information about the methods of picking a store in case of multiple stores, see How the Exchange Mailbox AutoProvisioning policy works.

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 to locate and select the objects you want.

4. Click **Next**, 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.

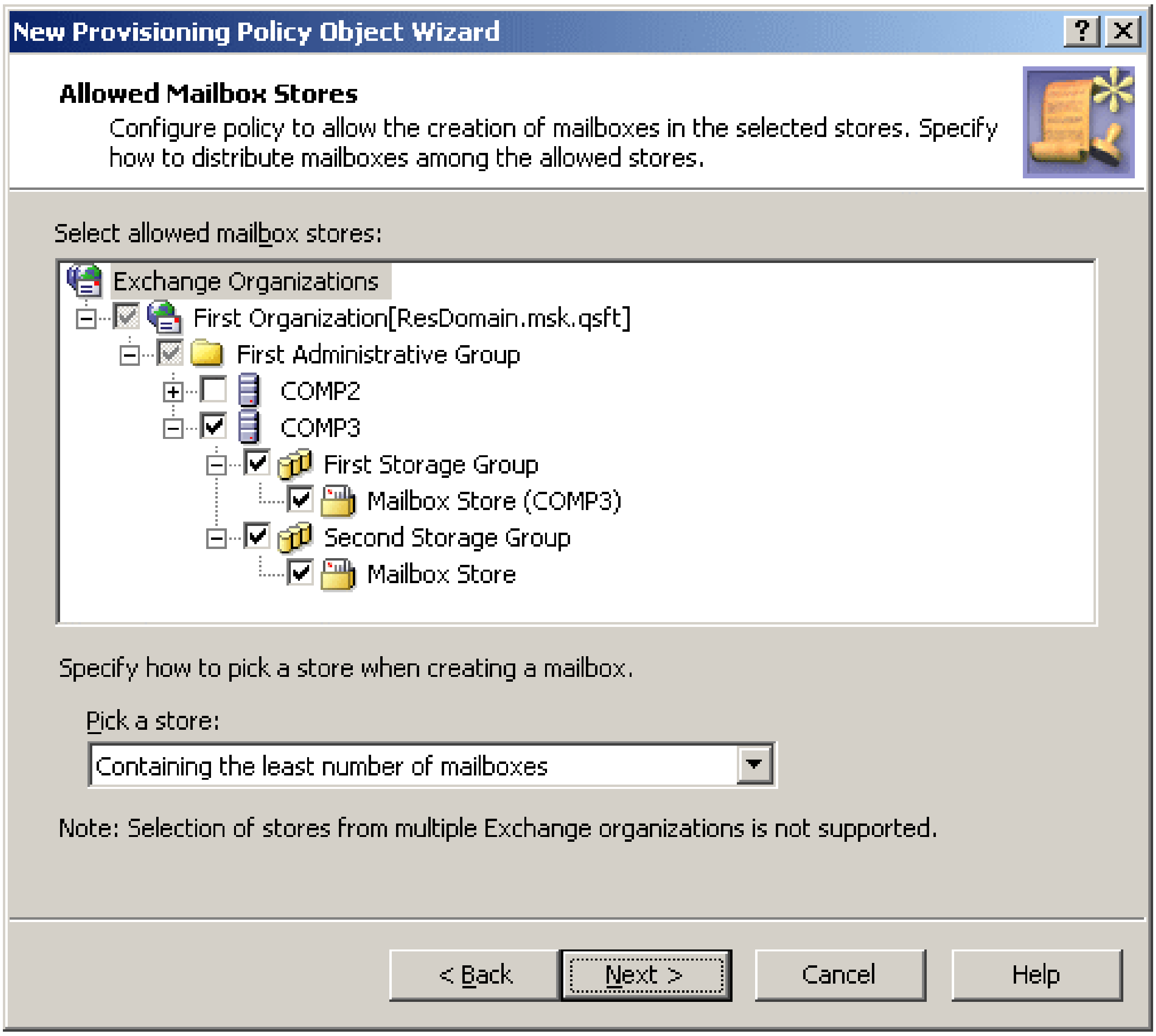
Creating and configuring the Exchange Mailbox AutoProvisioning Policy Object

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.

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**.

Figure 48: Allowed mailbox stores



New Provisioning Policy Object Wizard

Allowed Mailbox Stores

Configure policy to allow the creation of mailboxes in the selected stores. Specify how to distribute mailboxes among the allowed stores.

Select allowed mailbox stores:

- Exchange Organizations
  - ☑ First Organization[ResDomain.msk.qsft]
    - ☑ First Administrative Group
      - □ COMP2
      - ☑ COMP3
        - ☑ First Storage Group
          - ☑ Mailbox Store (COMP3)
        - ☑ Second Storage Group
          - ☑ Mailbox Store

Specify how to pick a store when creating a mailbox.

Pick a store: Containing the least number of mailboxes

Note: Selection of stores from multiple Exchange organizations is not supported.

Click **Next**, and follow the instructions in the wizard to create the Policy Object.

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.

Default creation options for Exchange mailboxes

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 mailboxes

1. Create a Policy Object containing an Exchange Mailbox AutoProvisioning policy.
2. Open the **Properties** dialog for the Policy Object you created.
3. On the **Policies** tab in the **Properties** dialog, double-click the **Exchange Mailbox AutoProvisioning** policy entry.
4. On the **Mailbox Creation** tab in the **Exchange Mailbox AutoProvisioning Policy Properties** dialog, 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, clear this policy option.
   - **Enforce creation of the mailbox**: Causes the **Create anExchange mailbox** option to be selected and unavailable so that the administrator who creates a user account cannot clear that option.
5. Click **OK** to close the dialogs 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 in 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 the AutoProvisioning in SaaS products 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.

Creating a provisioning policy for Starling Connect

You can create a new provisioning policy for Starling Connect in the Active Roles Console by configuring a new Policy Object based on the **Autoprovisioning in SaaS products** policy.

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**, then click **Provisioning Policy**.
3. On the **Welcome** page of the wizard, click **Next**.
4. On the **Name and Description** page, do the following, then click **Next**:
   a. In the **Name** box, enter a name for the Policy Object.
   b. (Optional) Under **Description**, enter any 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: Consider the following when configuring a Policy Object for Starling Connect:

- You must apply the Starling Connect policy 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 if 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. If this happens, you 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 the OneDrive Provisioning policy works

Active Roles relies on this policy during user creation to provision Azure AD users for OneDrive access.

Creating a 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 more information on how to create a Policy Object, see Creating a Policy Object.
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: Consider the following when creating and applying a new policy:

- If the policy conditions are not met, for example by specifying an incorrect SharePoint Admin URL or a storage size that is not within the acceptable range, an error will appear.
- The policy accepts a minimum storage size of 1 GB and a maximum of 10 TB.

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 the Home Folder AutoProvisioning policy works

When running 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.

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 preconfigured 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.

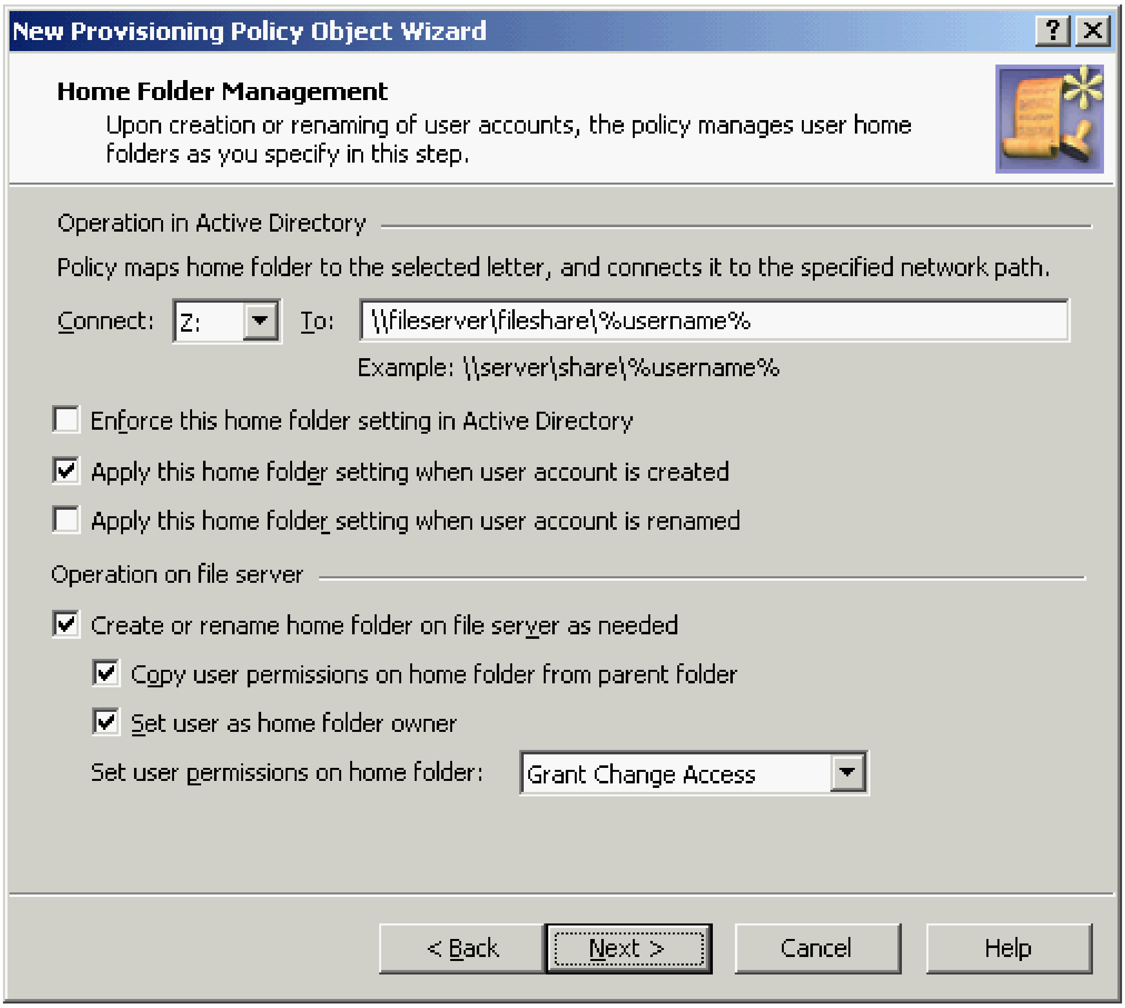
Configuring a Home Folder AutoProvisioning policy

You can create a new Home Folder AutoProvisioning policy with the Active Roles Console.

To configure a Home Folder AutoProvisioning policy

1. On the **Policy to Configure** page, select **Home Folder AutoProvisioning**, then click **Next**.

   Figure 49: Home folder management

   New Provisioning Policy Object Wizard

   Home Folder Management

   Upon creation or renaming of user accounts, the policy manages user home folders as you specify in this step.

   Operation in Active Directory

   Policy maps home folder to the selected letter, and connects it to the specified network path.

   Connect: Z: To: \\fileserver\fileshare\%username%

   Example: \\server\share\%username%

   □ Enforce this home folder setting in Active Directory

   ☑ Apply this home folder setting when user account is created

   □ Apply this home folder setting when user account is renamed

   Operation on file server

   ☑ Create or rename home folder on file server as needed

   ☑ Copy user permissions on home folder from parent folder

   ☑ Set user as home folder owner

   Set user permissions on home folder: Grant Change Access

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, if you want to create home folders on the share `Home` of the server `Ant`, use the following path:

     ```
     \\Ant\Home\%username%
     ```

> NOTE: Paths containing just the server and username, like `\\SERVER\%username%` are 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.

- (Optional) In **Share name prefix** and **Share name suffix**, type a prefix and suffix for the name of the home share.
- (Optional) 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 to locate and select the objects you want.

5. Click **Next**, then click **Finish**.

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.
- If you want the policy to create home shares (see Set user permissions on home folder), 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.

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.

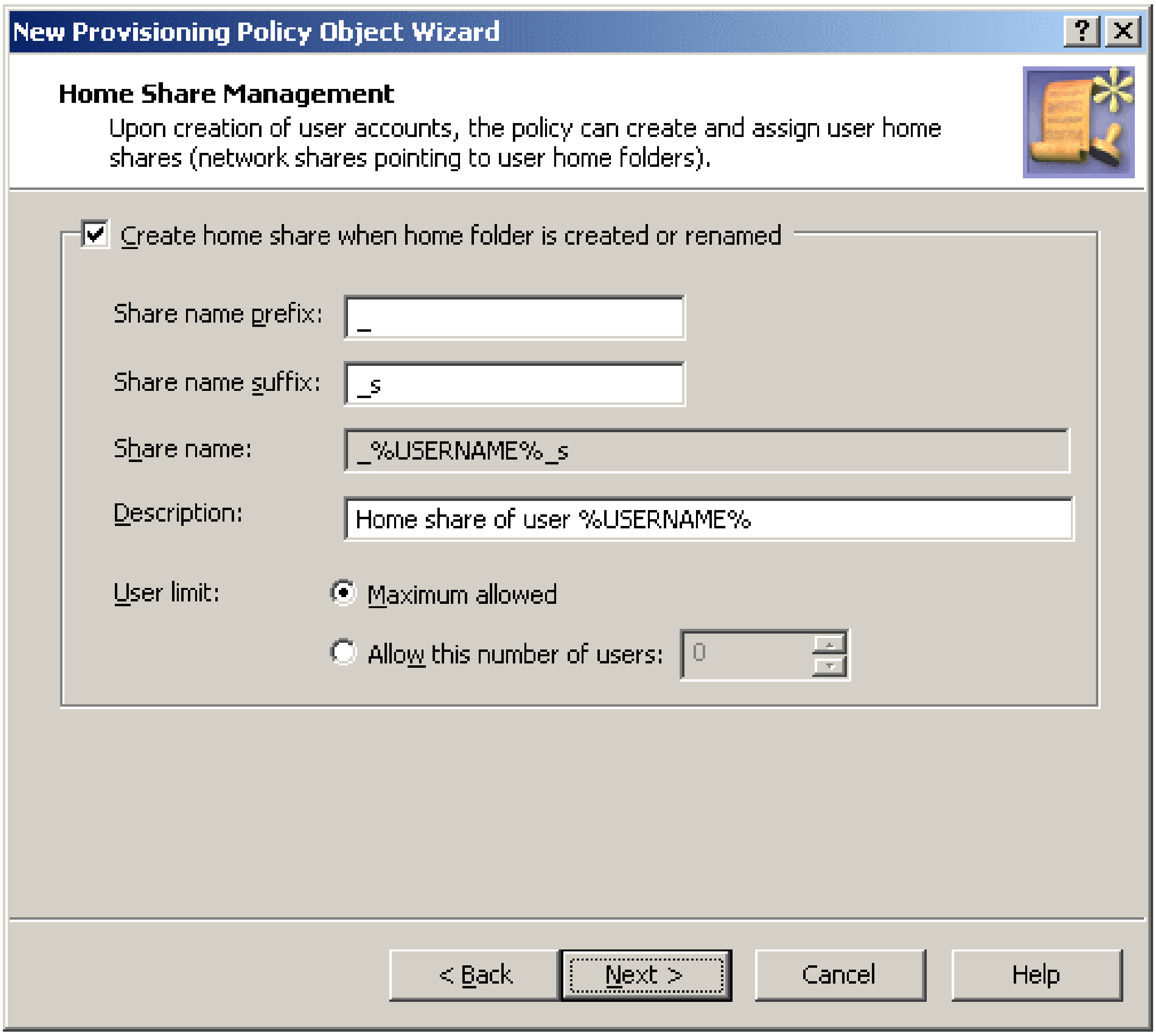
Set user permissions on home folder

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 50: Home share management



New Provisioning Policy Object Wizard

Home Share Management

Upon creation of user accounts, the policy can create and assign user home shares (network shares pointing to user home folders).

☑ Create home share when home folder is created or renamed

Share name prefix: _

Share name suffix: _s

Share name: _%USERNAME%_s

Description: Home share of user %USERNAME%

User limit: ● Maximum allowed ○ Allow this number of users: 0

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.

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 dialogs 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 cleared, 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.

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.

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.

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, enter the path in the following format:

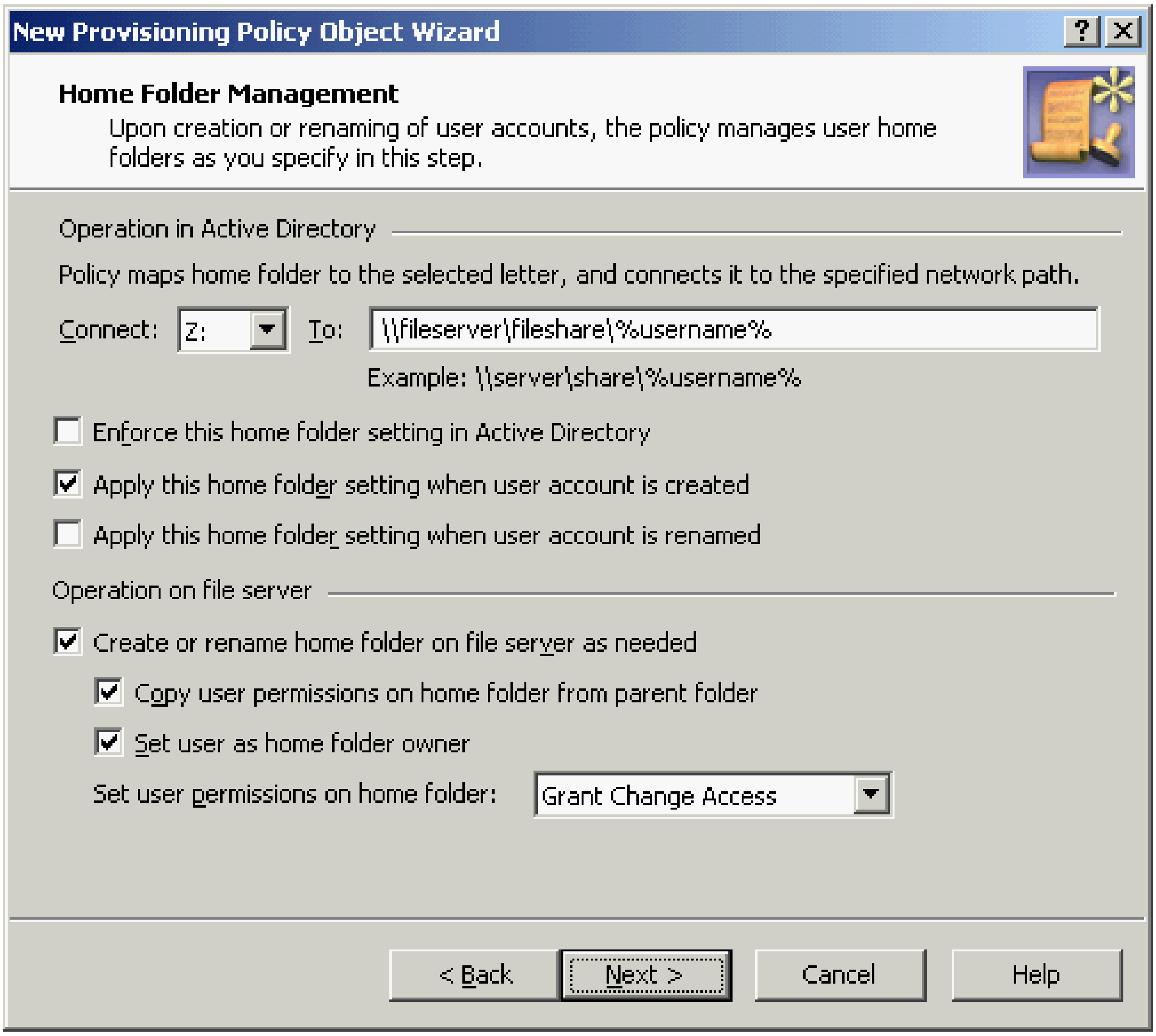
  `\\server\share\%username%`

  In the above format, `\\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`**, 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 51: Policy Object: Home folder management



New Provisioning Policy Object Wizard

Home Folder Management

Upon creation or renaming of user accounts, the policy manages user home folders as you specify in this step.

Operation in Active Directory

Policy maps home folder to the selected letter, and connects it to the specified network path.

Connect: Z: To: \\fileserver\fileshare\%username%

Example: \\server\share\%username%

☐ Enforce this home folder setting in Active Directory

☑ Apply this home folder setting when user account is created

☐ Apply this home folder setting when user account is renamed

Operation on file server

☑ Create or rename home folder on file server as needed

☑ Copy user permissions on home folder from parent folder

☑ Set user as home folder owner

Set user permissions on home folder: Grant Change Access

Click **Next** and follow the steps in the wizard to create the Policy Object.

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.

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 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 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 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 (Azure 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 the Script Execution 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.

Configuring a 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.

To configure a Script Execution policy

1. On the **Policy to Configure** page, select **Script Execution**, 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 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. (Optional) 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.

   b. 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 will be applied. To do so, click **Add**, and use the **Select Objects** dialog to locate and select the objects you want.
6. Click **Next**, 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.

   To create a new folder, right-click **Script Modules** and select **New** > **Scripts Container**. Similarly, you can create a sub-folder in a folder by right-clicking the folder and selecting **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, 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, 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.

   To create a new folder, right-click **Script Modules** and select **New** > **Scripts Container**. Similarly, you can create a sub-folder in a folder by right-clicking the folder and selecting **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, then click **Save**.

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 where you can select and open a script file.

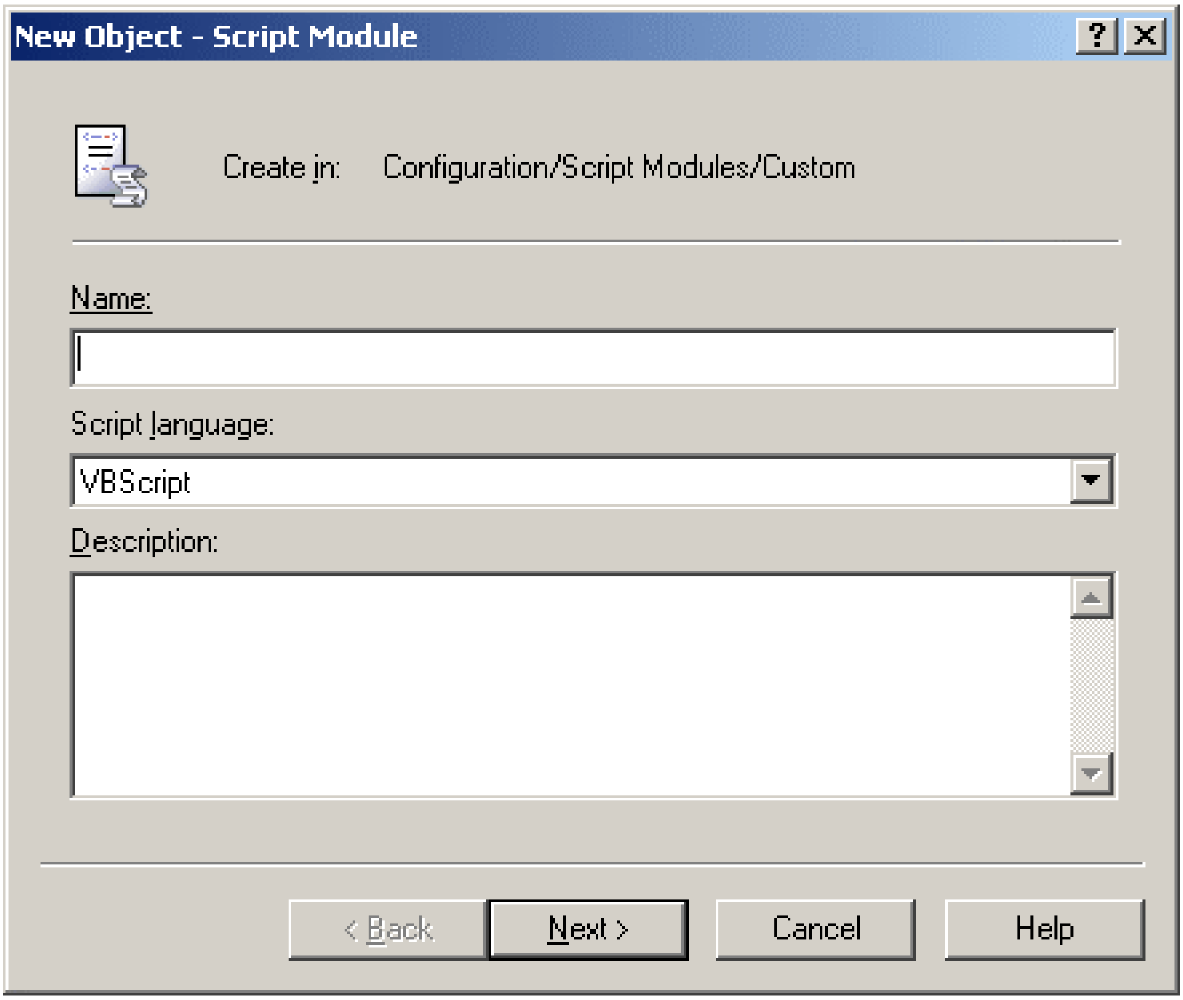
Creating a script

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 52: Script module: Creating a script



New Object - Script Module

Create in: Configuration/Script Modules/Custom

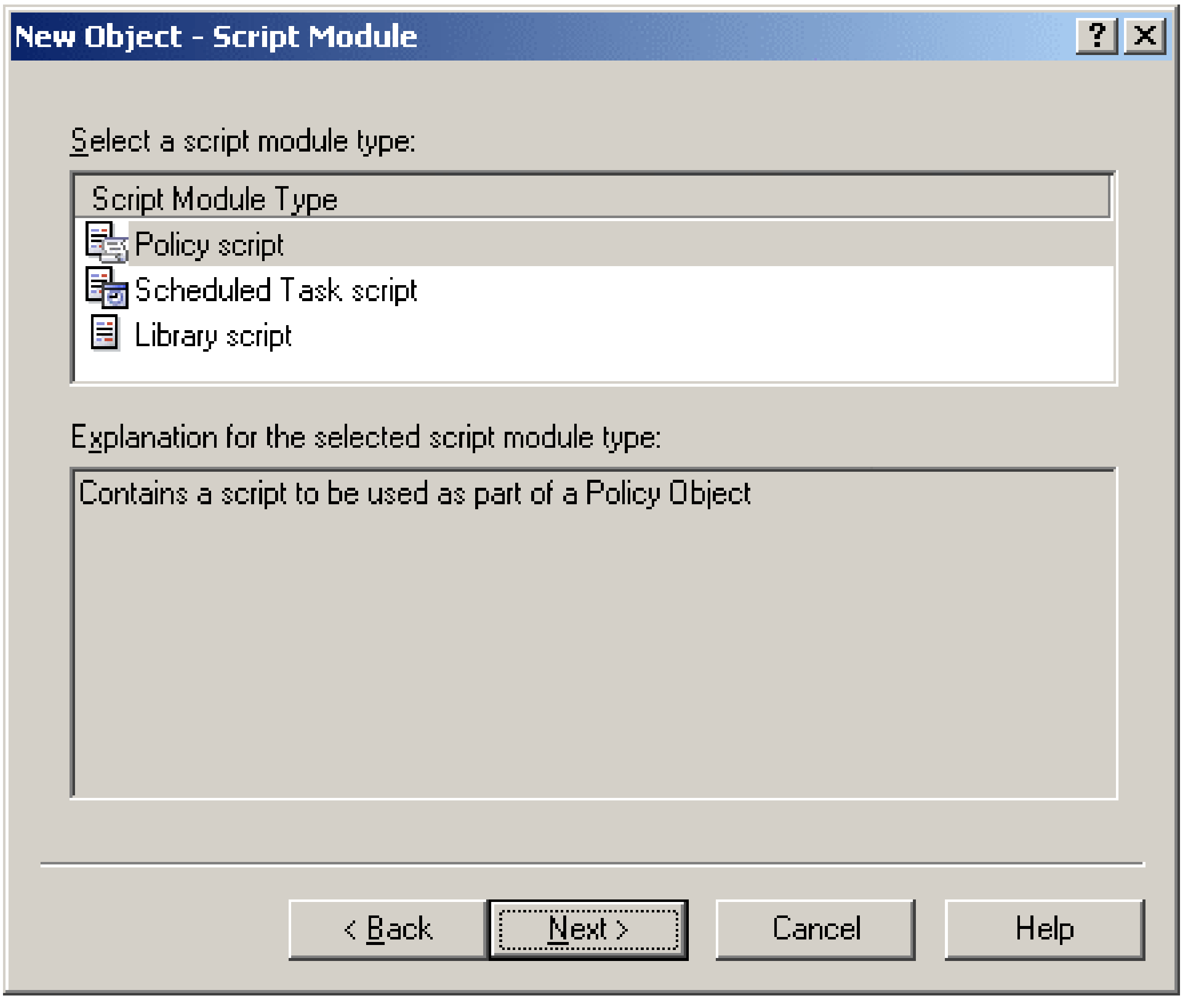
Name:

Script language: VBScript

Description:

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 53: Script Module: Policy script



New Object - Script Module
Select a script module type:
Script Module Type
Policy script
Scheduled Task script
Library script
Explanation for the selected script module type:
< Back
Next >
Cancel
Help

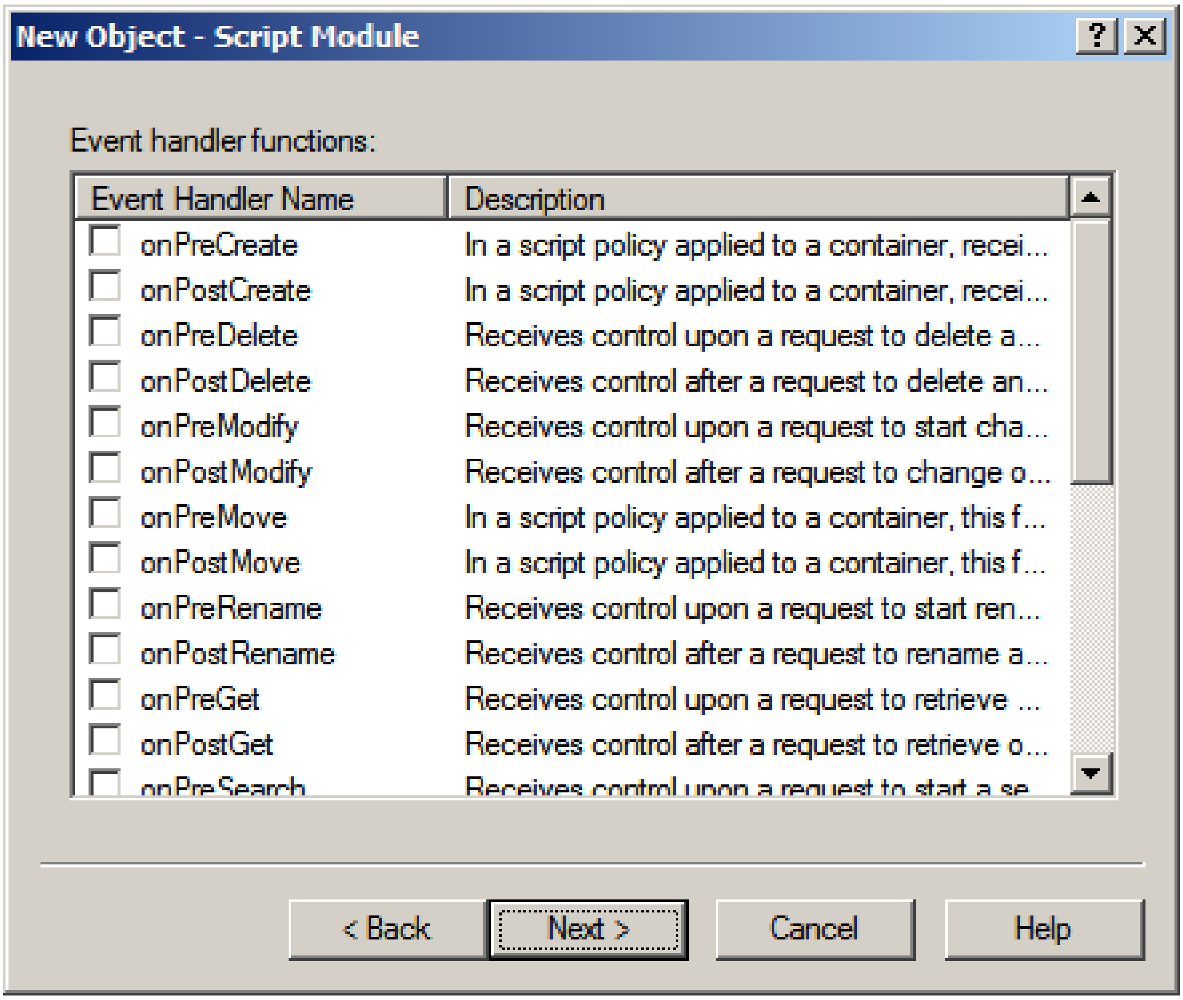
Contains a script to be used as part of a Policy Object

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 54: Script Module: Event handler functions



Event handler functions:

Event Handler Name	Description
□ onPreCreate	In a script policy applied to a container, recei...
□ onPostCreate	In a script policy applied to a container, recei...
□ onPreDelete	Receives control upon a request to delete a...
□ onPostDelete	Receives control after a request to delete an...
□ onPreModify	Receives control upon a request to start cha...
□ onPostModify	Receives control after a request to change o...
□ onPreMove	In a script policy applied to a container, this f...
□ onPostMove	In a script policy applied to a container, this f...
□ onPreRename	Receives control upon a request to start ren...
□ onPostRename	Receives control after a request to rename a...
□ onPreGet	Receives control upon a request to retrieve ...
□ onPostGet	Receives control after a request to retrieve o...
□ onPreSearch	Receives control upon a request to start a se

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 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.

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 orWeb 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.

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.

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.

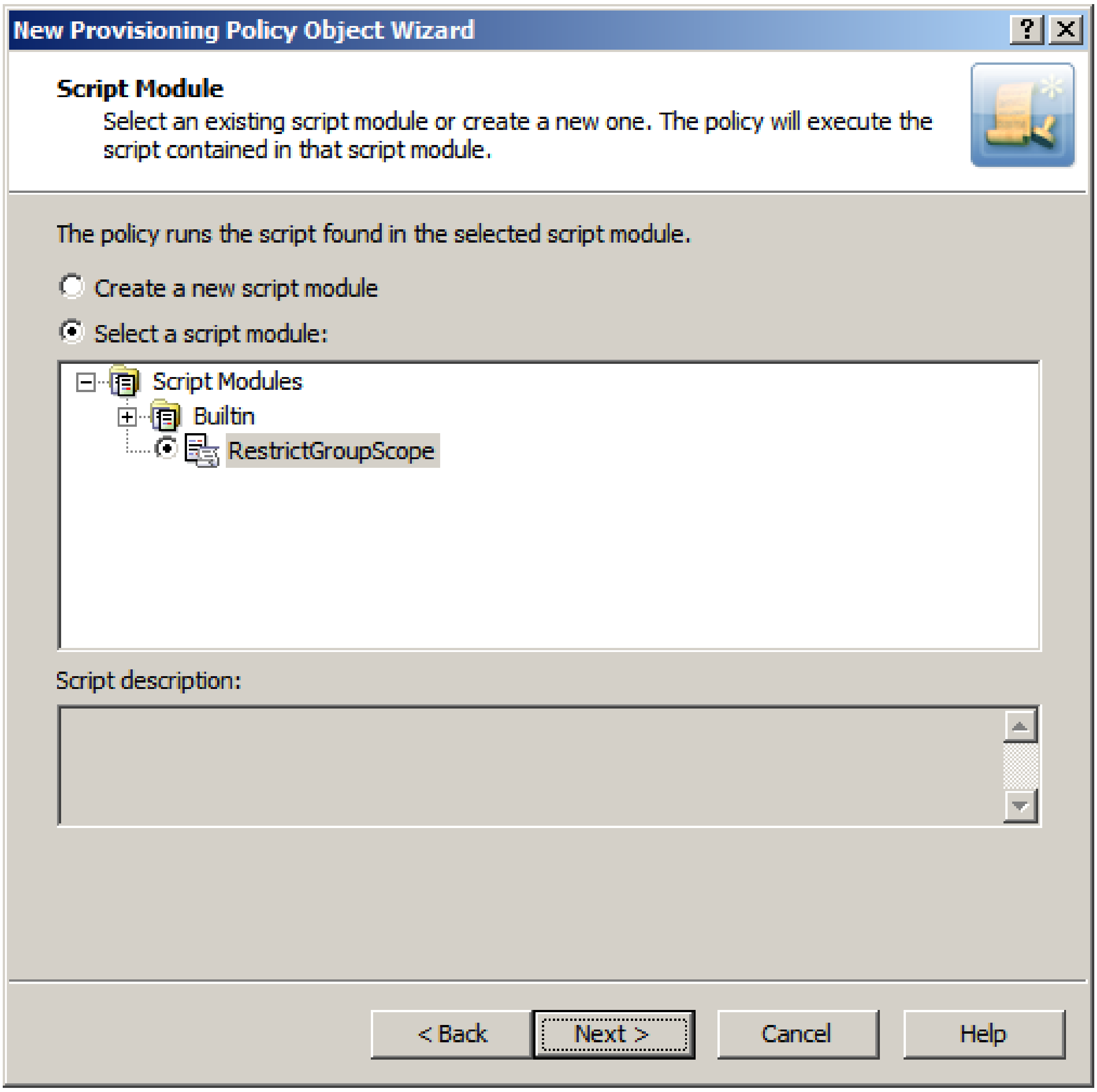
Creating and configuring the Policy Object

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.

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 55: Script Module: Creating/configuring Policy Object



New Provisioning Policy Object Wizard

Script Module

Select an existing script module or create a new one. The policy will execute the script contained in that script module.

The policy runs the script found in the selected script module.

○ Create a new script module

● Select a script module:

- Script Modules
  - Builtin
  - ● RestrictGroupScope

Script description:

Click **Next** and follow the instructions in the wizard to create the Policy Object.

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.

Microsoft 365 and Azure Tenant Selection

Microsoft 365 and Azure Tenant Selection policies help you manage Azure tenant selection, Microsoft 365 (M365) license and role selection, and OneDrive provisioning for hybrid Azure users in the Azure tenant.

How the Microsoft 365 and Azure Tenant Selection 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 forAzure 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, Azure security 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.

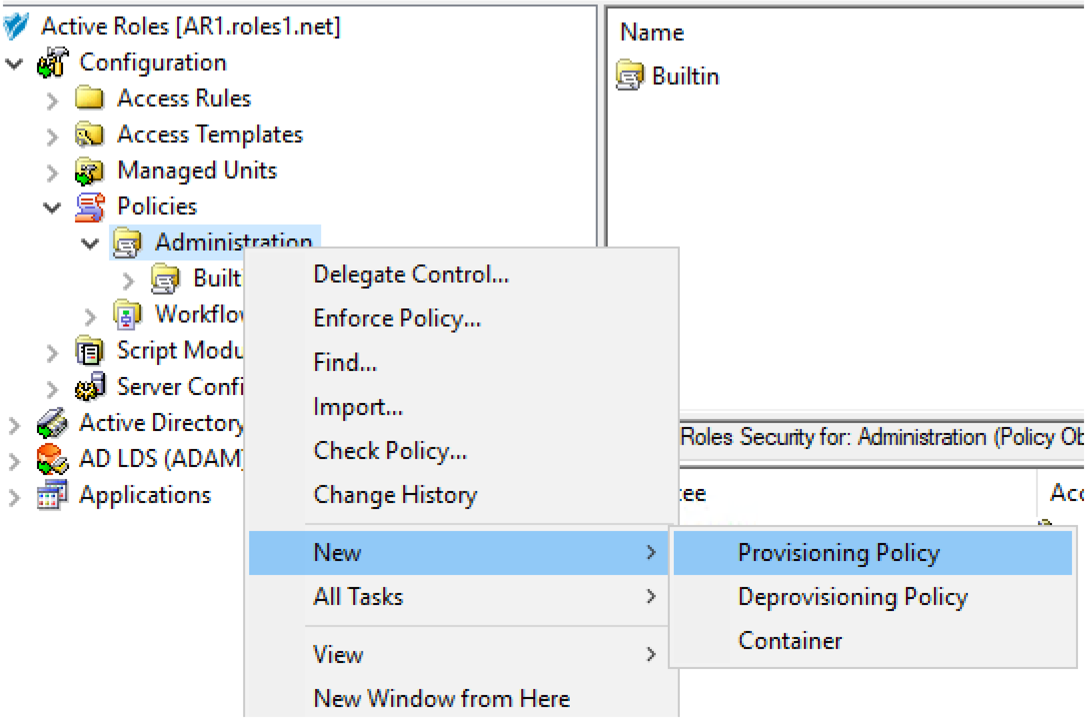
Prerequisites

Consider the following before configuring an **O365 and Azure Tenant Selection** policy:

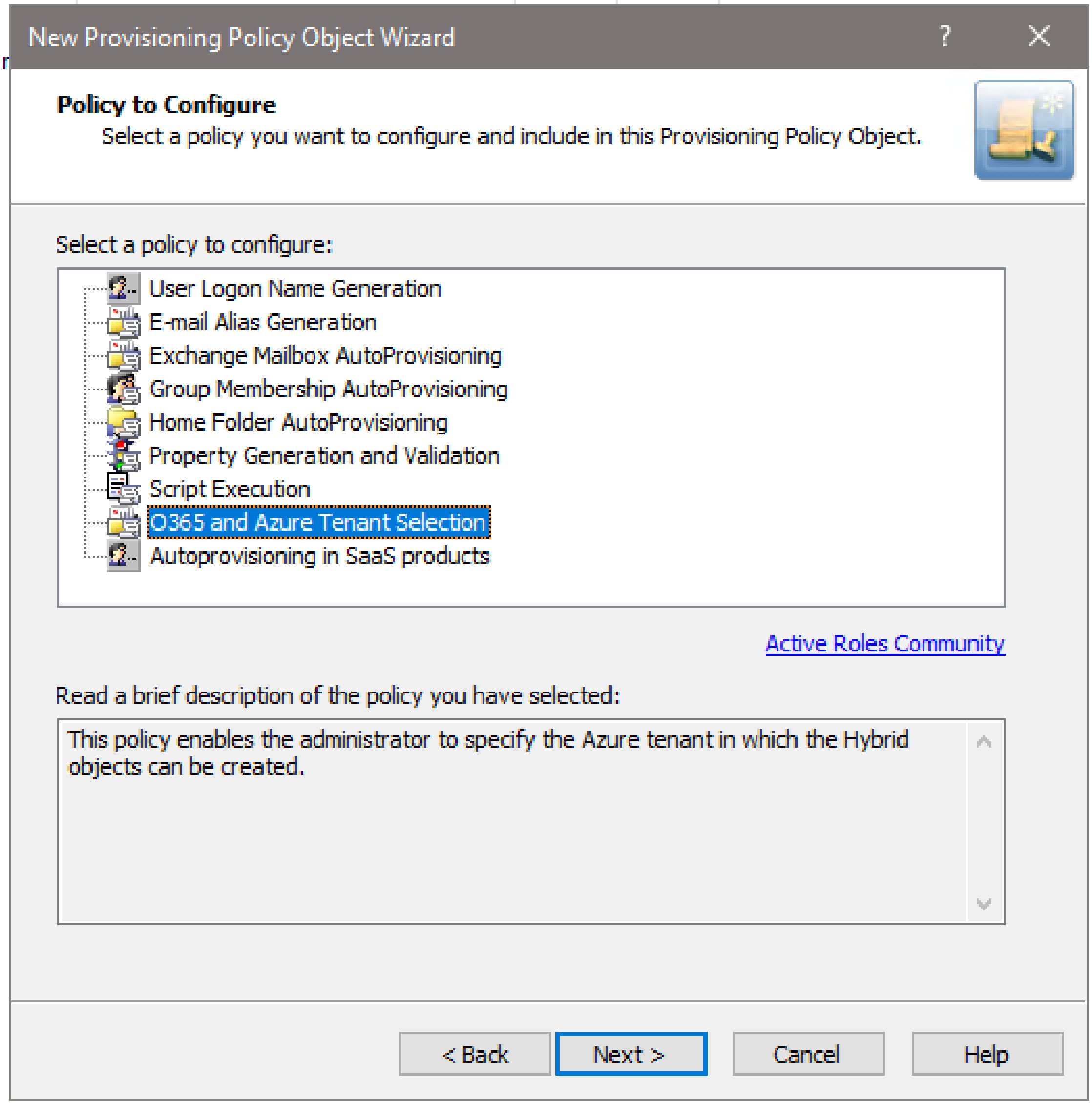
- The OneDrive settings of this policy are applicable to hybrid Azure users only, and will work only if you have already enabled OneDrive for your Azure tenant in the **Azure AD Configuration** > **Modify** (**Tenant details**) window of the Active Roles Configuration Center. For more information on 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**.



New Provisioning Policy Object Wizard

Policy to Configure

Select a policy you want to configure and include in this Provisioning Policy Object.

Select a policy to configure:

- User Logon Name Generation
- E-mail Alias Generation
- Exchange Mailbox AutoProvisioning
- Group Membership AutoProvisioning
- Home Folder AutoProvisioning
- Property Generation and Validation
- Script Execution
- O365 and Azure Tenant Selection
- Autoprovisioning in SaaS products

Active Roles Community

Read a brief description of the policy you have selected:

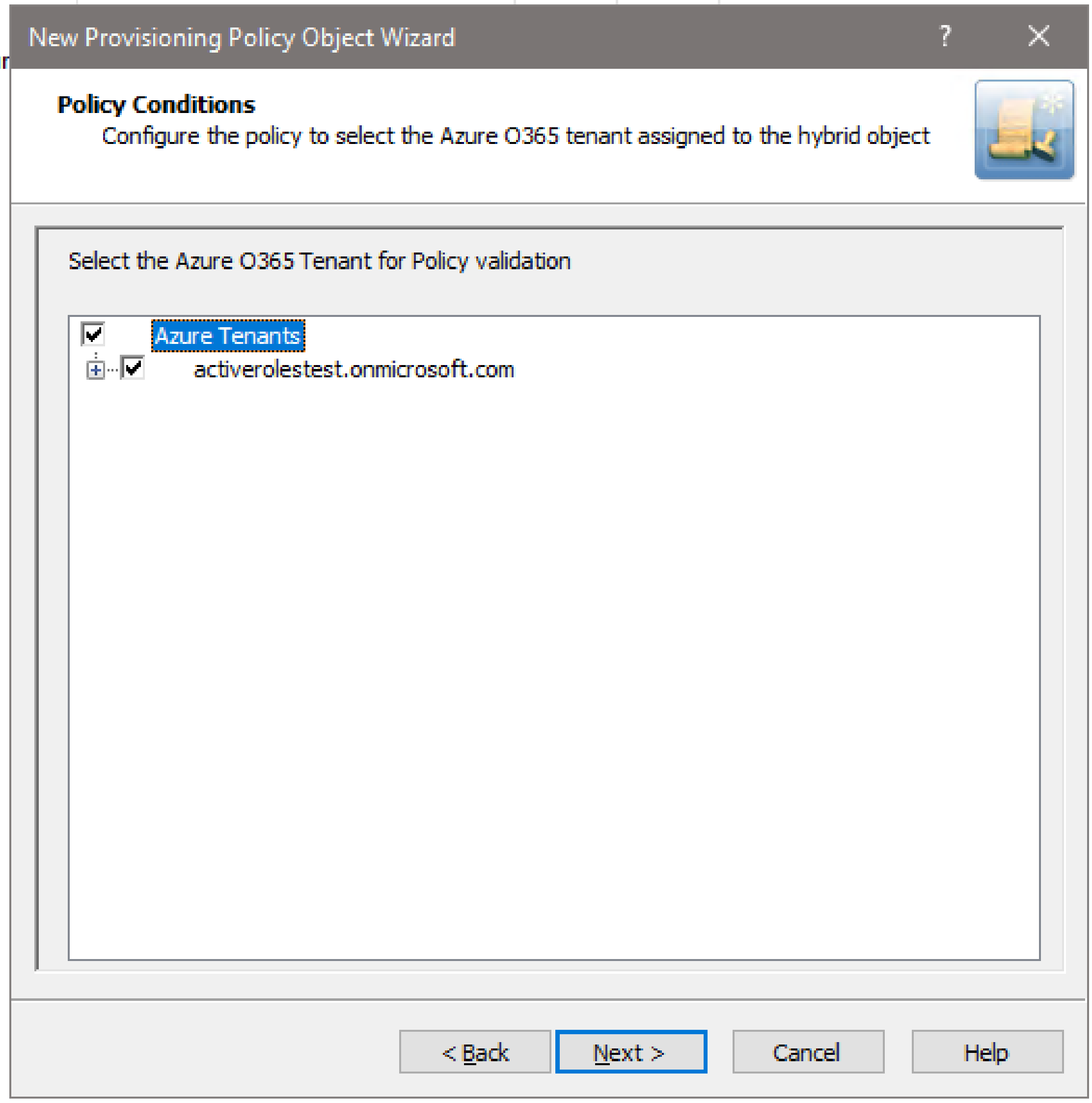
This policy enables the administrator to specify the Azure tenant in which the Hybrid objects can be created.

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**.



New Provisioning Policy Object Wizard

Policy Conditions

Configure the policy to select the Azure O365 tenant assigned to the hybrid object

Select the Azure O365 Tenant for Policy validation

☑ Azure Tenants

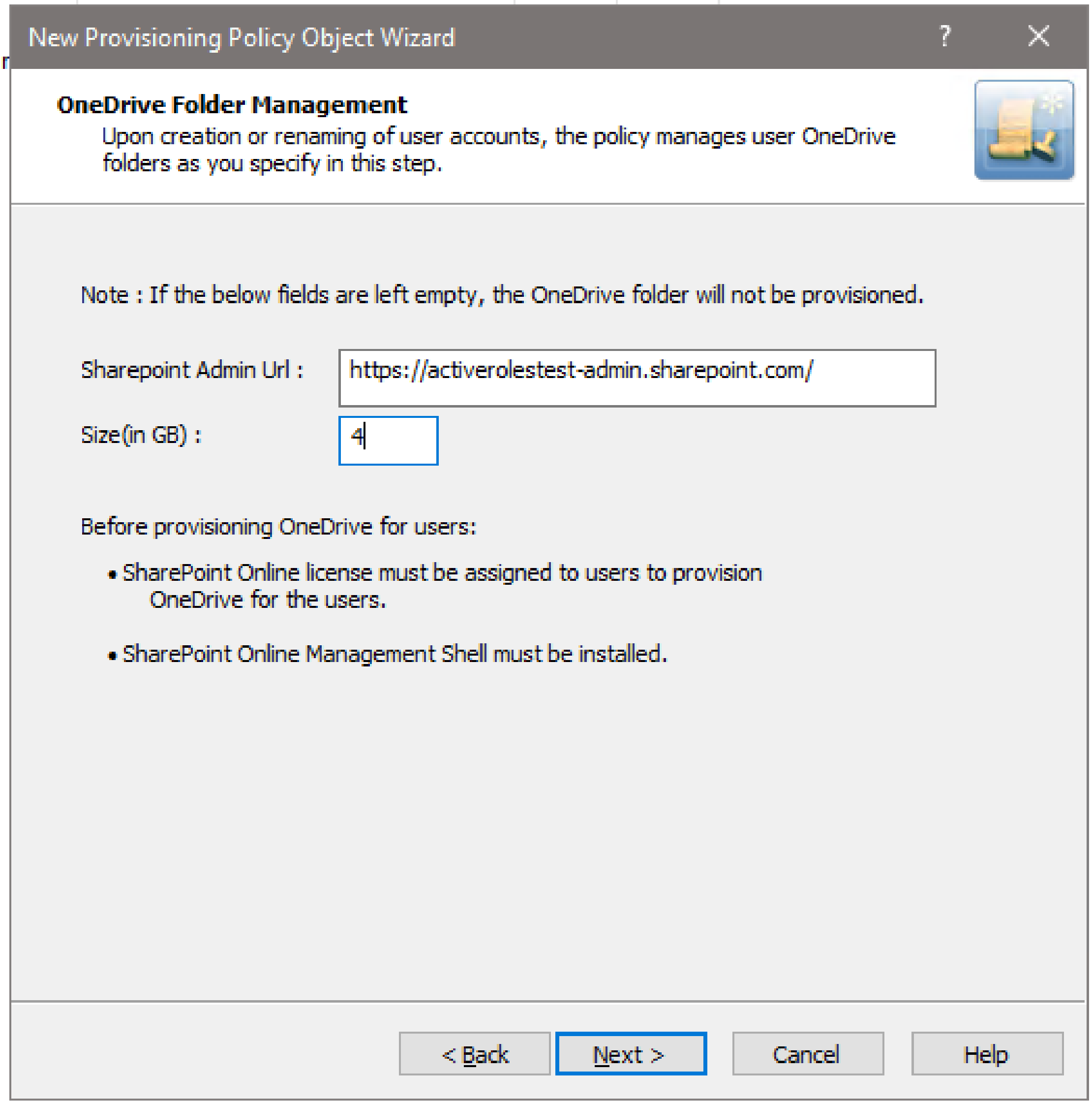
☑ activerolestest.onmicrosoft.com

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* in the *Active Roles Web Interface User Guide*.

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:



OneDrive Folder Management

Upon creation or renaming of user accounts, the policy manages user OneDrive folders as you specify in this step.

Note : If the below fields are left empty, the OneDrive folder will not be provisioned.

Sharepoint Admin Url : https://activerolestest-admin.sharepoint.com/

Size(in GB) : 4

Before provisioning OneDrive for users:

- SharePoint Online license must be assigned to users to provision OneDrive for the users.
- SharePoint Online Management Shell must be installed.

- **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

To manage Office 365 user licenses

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. To check if there are any policy violations, right-click and click **Check Policy**.

   For a container object, this displays the **Check Policy** dialog.
3. Review the options in the **Check Policy** dialog 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.

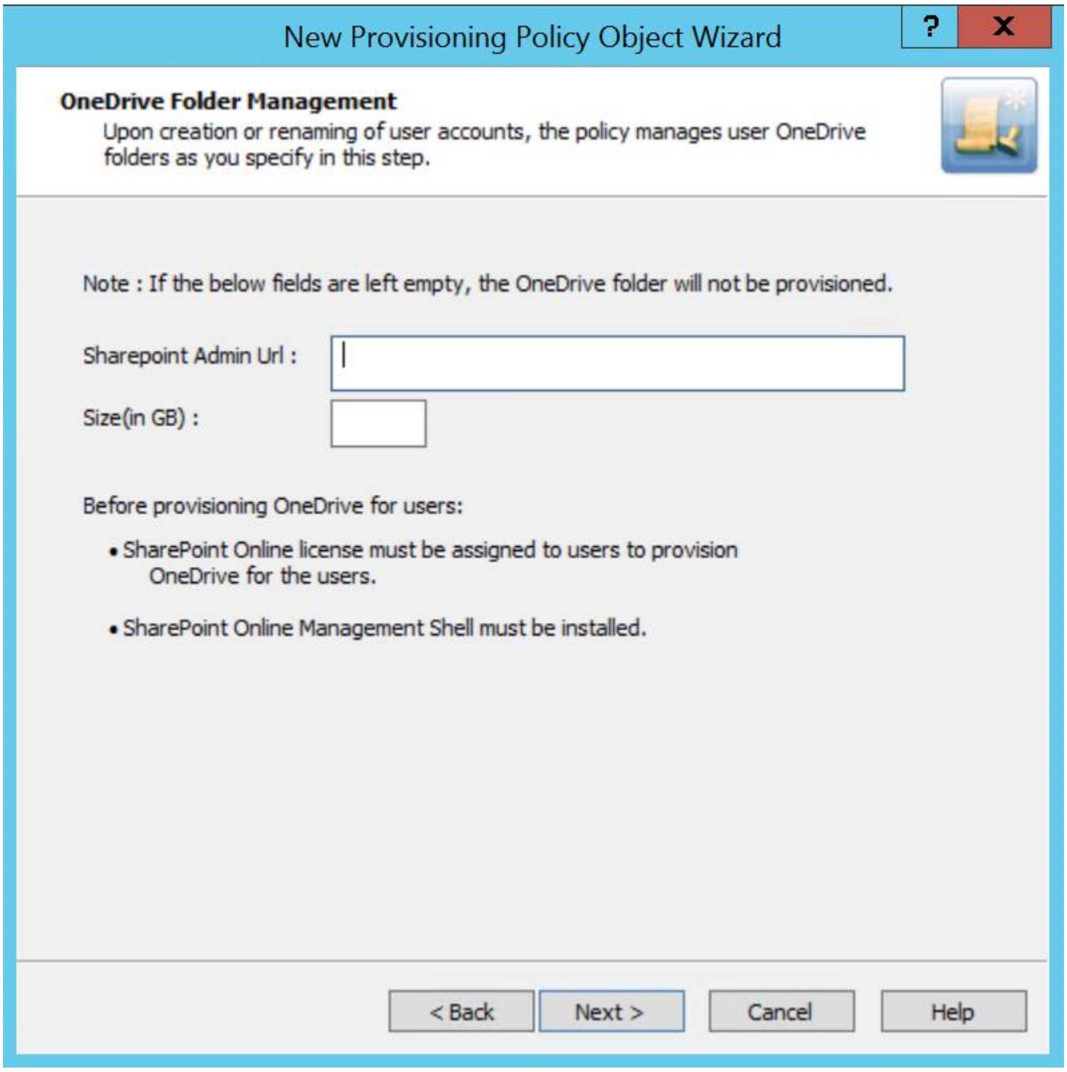
Office 365 user roles management through provisioning policy

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 56: OneDrive folder management wizard



OneDrive Folder Management

Upon creation or renaming of user accounts, the policy manages user OneDrive folders as you specify in this step.

Note : If the below fields are left empty, the OneDrive folder will not be provisioned.

Sharepoint Admin Url :

Size(in GB) :

Before provisioning OneDrive for users:

- SharePoint Online license must be assigned to users to provision OneDrive for the users.
- SharePoint Online Management Shell must be installed.

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: Consider the following when provisioning OneDrive:
   >
   > - 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 the **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.

E-mail Alias Generation

Policies in this category are intended to automate the assignment of the email alias when designating a user as mailbox-enabled on Microsoft Exchange Server. By default, Microsoft Exchange Server provides for the following recipient email address format: `<email-alias>@<domain-name>`

You can use predefined rules to generate email aliases, or configure custom rules. For example, you can configure a policy to compose the email 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 the E-Mail Alias Generation policy works

When making a user mailbox-enabled, Active Roles relies on this policy to assign a certain email 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 **Generate** 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, 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:

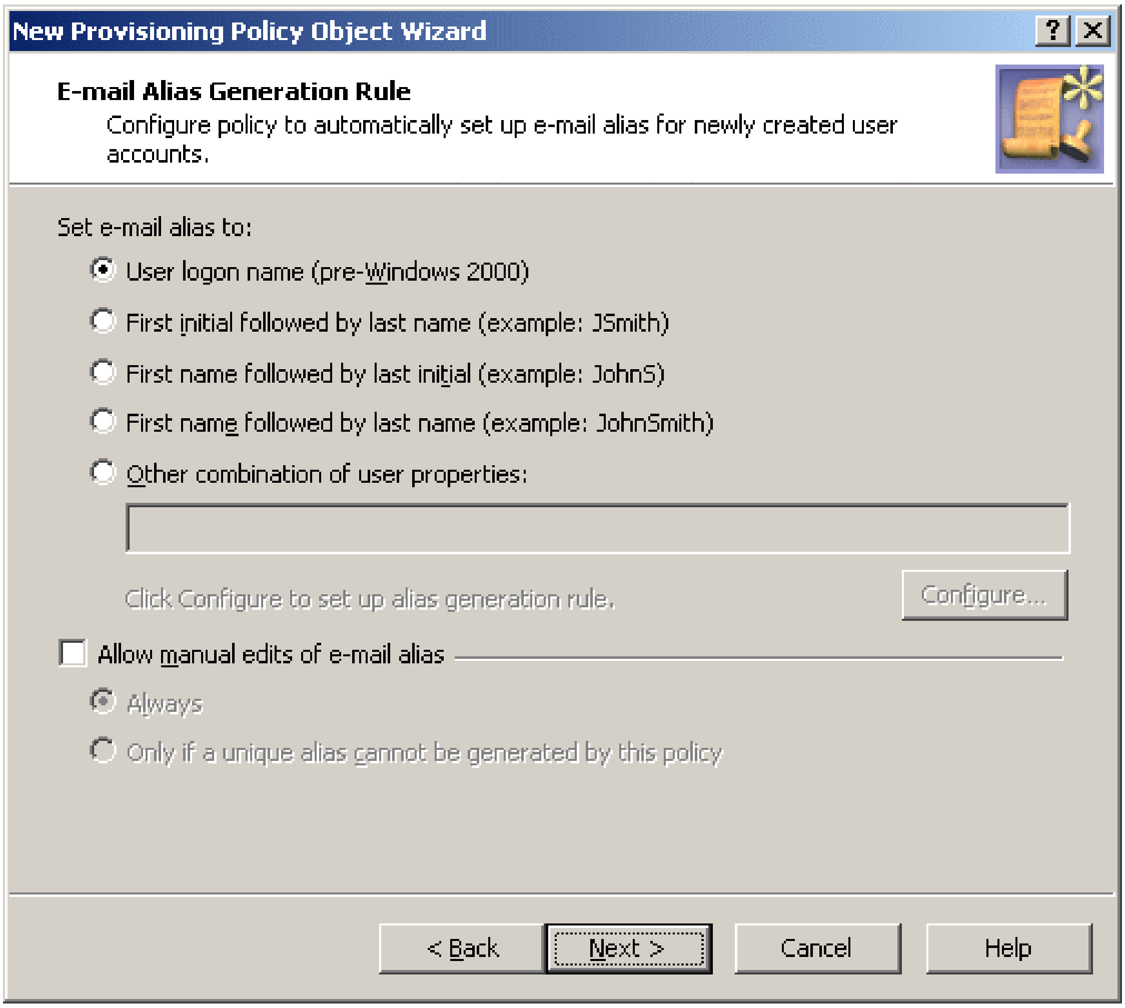
  @ * + | = \ ; : ? [ ] , < > /
- The policy denies processing of operation requests that assign the empty value to the e-mail alias.
- When checking user accounts for Active Roles policy compliance, Active Roles detects, and reports on, the aliases that are not set up as prescribed by the alias generation policy.

Configuring an E-mail Alias Generation policy

You can configure a new E-mail Alias Generation policy with the Active Roles Console.

To configure an E-mail Alias Generation policy

1. On the **Policy to Configure** page, select **E-mail Alias Generation**, and click **Next**.



New Provisioning Policy Object Wizard

E-mail Alias Generation Rule

Configure policy to automatically set up e-mail alias for newly created user accounts.

Set e-mail alias to:

- ● User logon name (pre-Windows 2000)
- ○ First initial followed by last name (example: JSmith)
- ○ First name followed by last initial (example: JohnS)
- ○ First name followed by last name (example: JohnSmith)
- ○ Other combination of user properties:

Click Configure to set up alias generation rule.

□ Allow manual edits of e-mail alias

- ● Always
- ○ Only if a unique alias cannot be generated by this policy

2. On the **E-mail Alias Generation Rule** page, do the following:
   - Select one of the preconfigured 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 as described later in the procedure.
   - If you want the email 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 email 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 to locate and select the objects you want.
4. Click **Next**, then click **Finish**.

To complete the Configure Value dialog

1. Click **Add**.
2. Configure an entry to include in the value. For more information, see Configuring entries.
3. In the **Configure Value** dialog, add more entries, delete or edit existing ones, and click **OK**.

Configuring a custom generation rule

To configure a custom rule, click **Other combination of user properties**, then click **Configure**. This displays the **Configure Value** dialog, as described in Configuring a Property Generation and Validation policy. You can use that dialog 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. 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 8: 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.

Type of entry	Description
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 more information on how to configure each entry type, see the following resources:

- **Text**: See Entry type: Text.
- **Uniqueness Number**: See Entry type: Uniqueness Number.
- **User Property**: See Entry type: <Object> Property.
- **Parent OU Property**: See Entry type: Parent OU Property.
- **Parent Domain Property**: See Entry type: Parent Domain Property.

When you are done configuring a value, click **OK** to close the **Configure Value** dialog. This will add the value to the policy rule. If necessary, you can modify the value by clicking **Configure**, then managing the list of entries in the **Configure Value** dialog.

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.

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 email 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 **Generate** causes the policy to add a uniqueness number to the alias.

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

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.

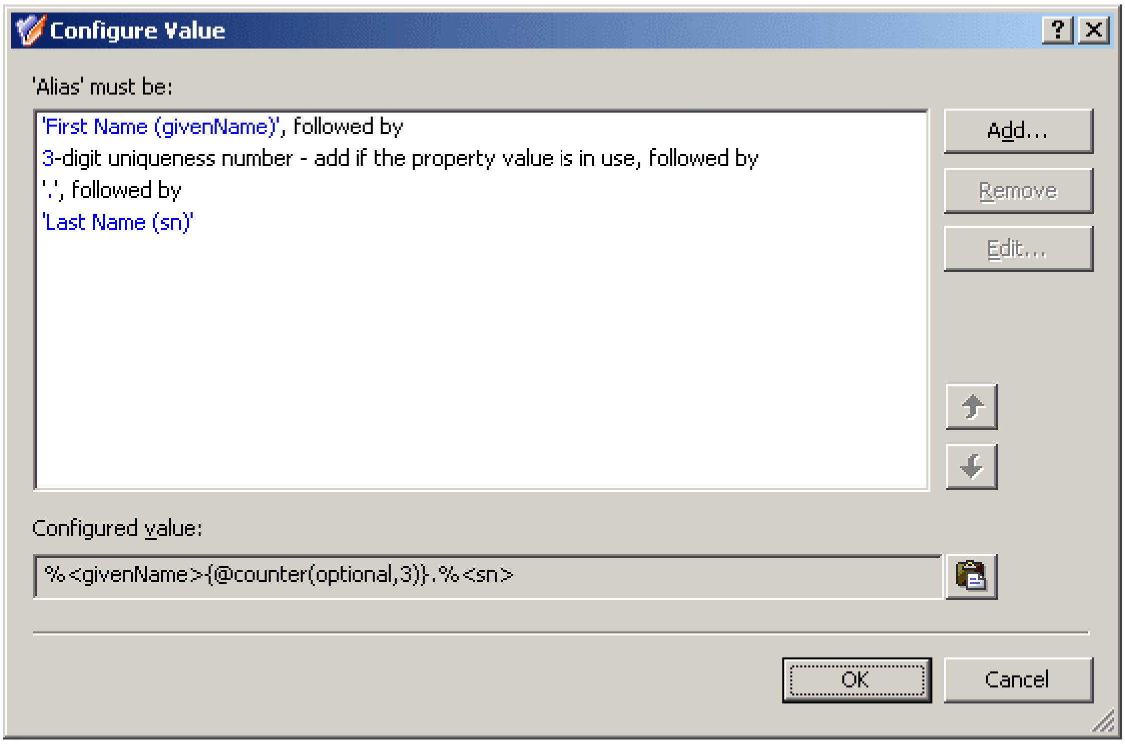
To configure the policy, click **E-mail Alias Generation** on the **Select Policy Type** page of the wizard. Then, click **Next**.

On the **E-mail Alias Generation Rule** page, click **Other combination of user properties**, and then click **Configure**.

Complete the **Configure Value** dialog 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 property** 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 must look like the following figure.



'Alias' must be:

'First Name (givenName)', followed by
3-digit uniqueness number - add if the property value is in use, followed by
'.', followed by
'Last Name (sn)'

Configured value:

%<givenName>{@counter(optional,3)}.%<sn>

9. Click **OK** to close the **Configure Value** dialog. Then, click **Next** and follow the instructions in the wizard to create the Policy Object.

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.

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 the User Account Deprovisioning 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.

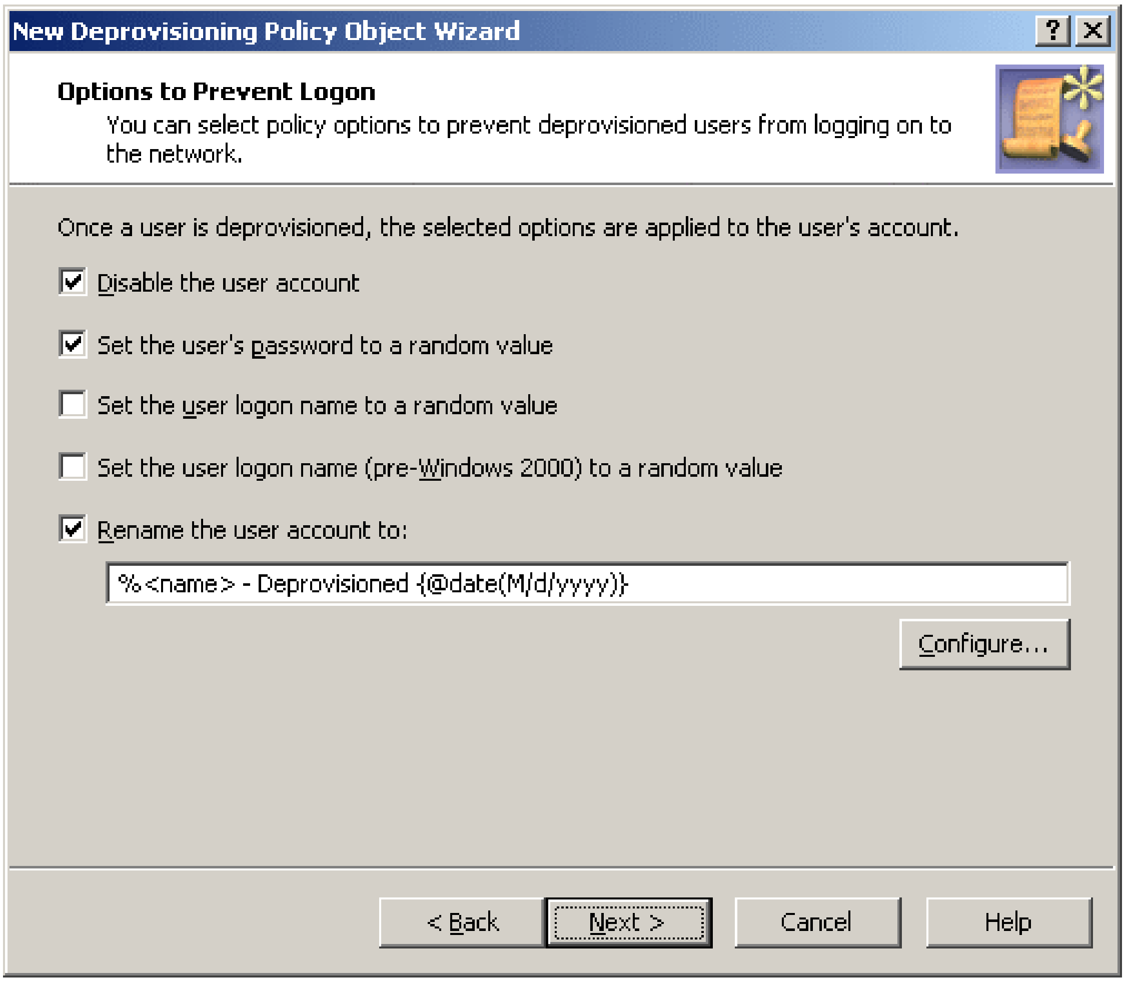
Thus, when deprovisioning a user, Active Roles modifies the user's account in Active Directory as determined by the User Account Deprovisioning policy that is in effect.

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**.

Figure 57: User Account Deprovisioning



New Deprovisioning Policy Object Wizard

Options to Prevent Logon

You can select policy options to prevent deprovisioned users from logging on to the network.

Once a user is deprovisioned, the selected options are applied to the user's account.

☑ 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:

%<name> - Deprovisioned {@date(M/d/yyyy)}

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 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 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 to locate and select the objects you want.
8. Click **Next**, and then click **Finish**.

To complete the Configure Value dialog

1. Click **Add**.
2. Configure an entry to include in the value. For more information, see Configuring entries.
3. In the **Configure Value** dialog, add more entries, delete or edit existing ones, and then click **OK**.

To complete Select Object Property dialog

1. From the **Object property** list, select an object property, and then click **OK**. The **Add Value** dialog appears.

   If you select multiple properties, the **Add Value** dialog 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, 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 by using the instructions given earlier in this topic.

Configuring a property update rule

To configure a property update rule for the user name, click **Configure**. This shows the **Configure Value** dialog, as described in Configuring a Property Generation and Validation policy. You can use that dialog 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. 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 9: 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. For more information on how to configure each entry type, see the following resources:

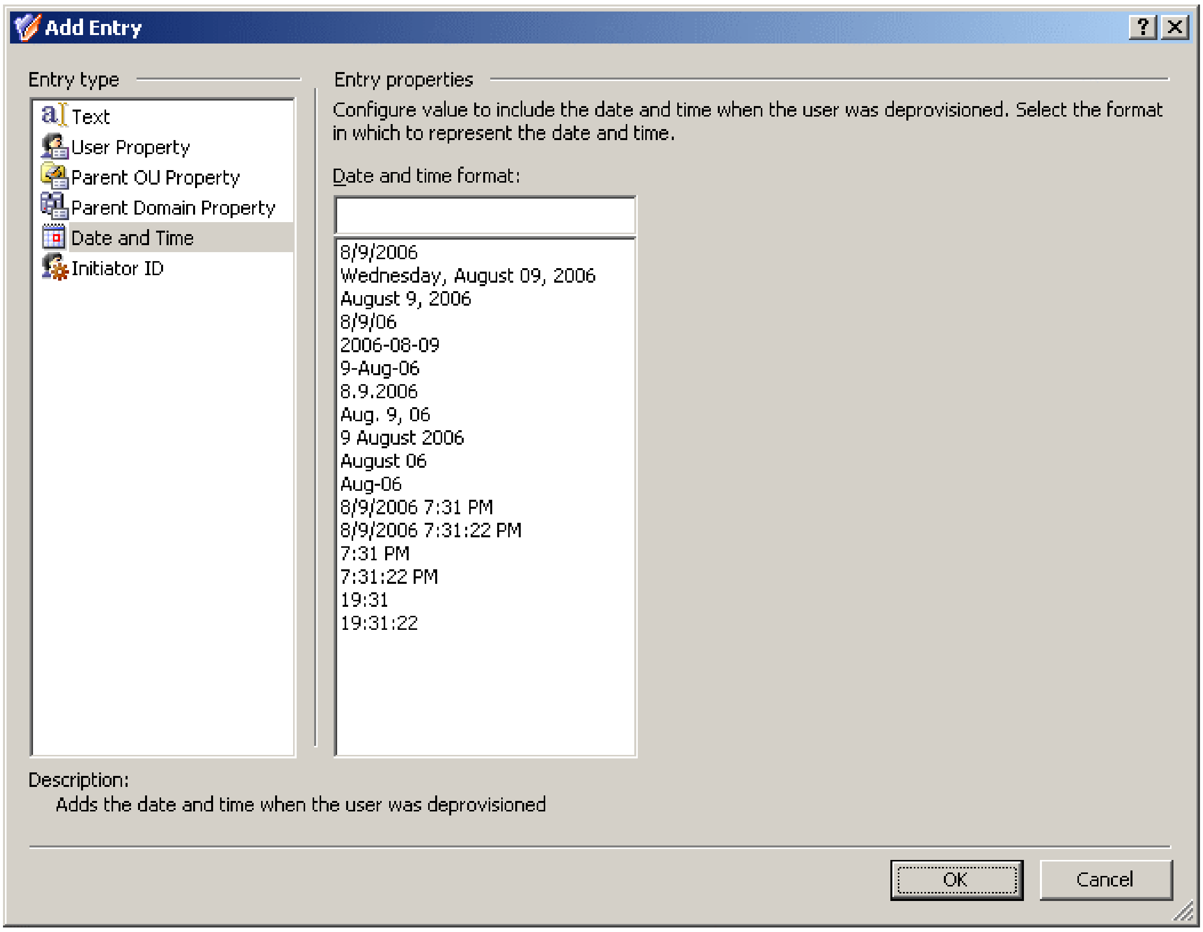
- **Text**: See Entry type: Text.
- **User Property**: See Entry type: <Object> Property.
- **Parent OU Property**: See Entry type: Parent OU Property.
- **Parent Domain Property**: See Entry type: Parent Domain Property.

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 as follows.

Figure 58: Entry type: Date and Time



Configure value to include the date and time when the user was deprovisioned. Select the format in which to represent the date and time.

Date and time format:

8/9/2006
Wednesday, August 09, 2006
August 9, 2006
8/9/06
2006-08-09
9-Aug-06
8.9.2006
Aug. 9, 06
9 August 2006
August 06
Aug-06
8/9/2006 7:31 PM
8/9/2006 7:31:22 PM
7:31 PM
7:31:22 PM
19:31
19:31:22

Description:
Adds the date and time when the user was deprovisioned

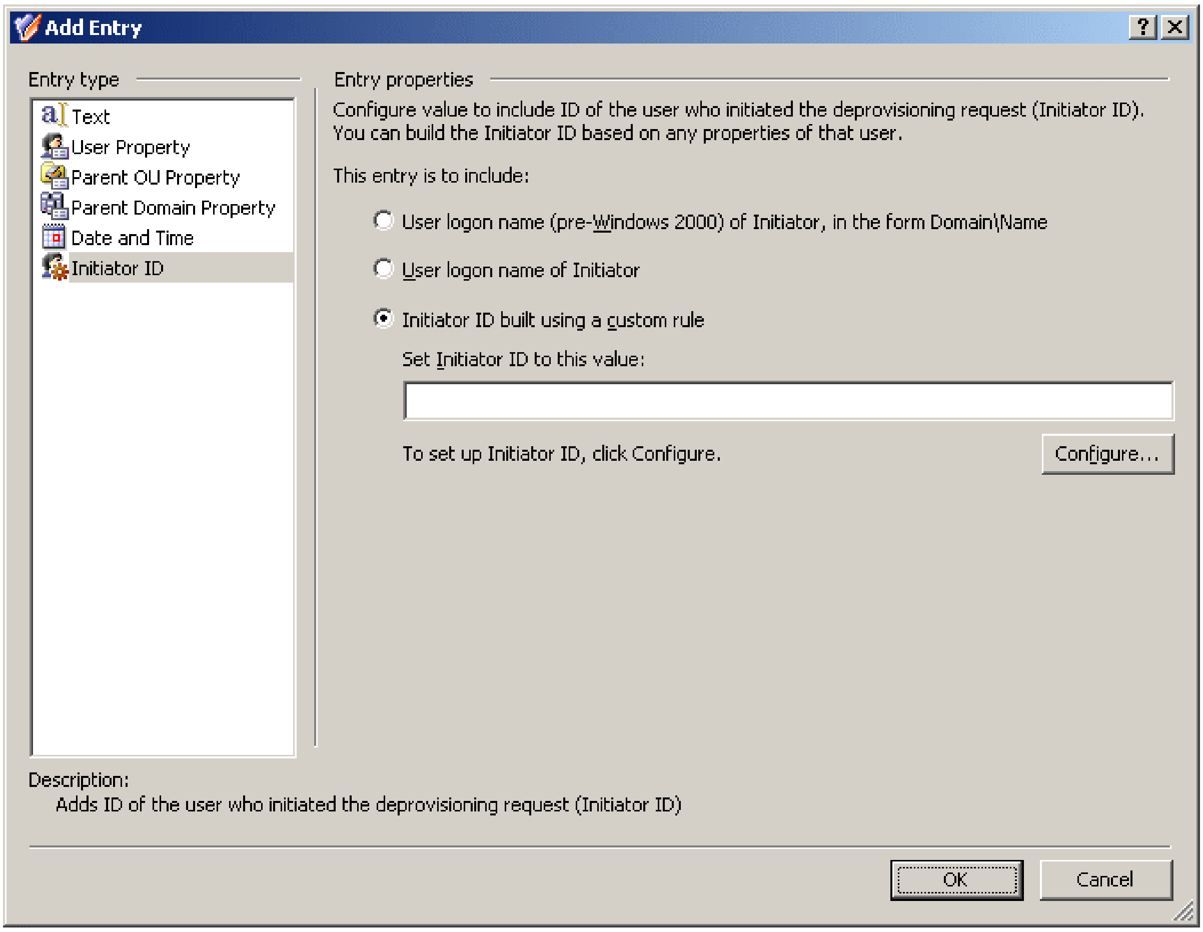
Using this entry type, you can add an entry that represents the date and time when the user account was deprovisioned.

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 as follows.

Figure 59: Entry type: Initiator ID



Configure value to include ID of the user who initiated the deprovisioning request (Initiator ID). You can build the Initiator ID based on any properties of that user.

This entry is to include:

○ User logon name (pre-Windows 2000) of Initiator, in the form Domain\Name

○ User logon name of Initiator

● Initiator ID built using a custom rule

Set Initiator ID to this value:

To set up Initiator ID, click Configure.

Description:
Adds ID of the user who initiated the deprovisioning request (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**, then click **Configure**. This displays the **Configure Value** dialog, described in Configuring a Property Generation and Validation policy. You can use that dialog 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. 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 10: 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 more information on how to configure each entry type, see the following resources:

- **Text**: See Entry type: Text.
- **Initiator Property**: See Entry type: <Object> Property.
- **Parent OU Property**: See Entry type: Parent OU Property.
- **Parent Domain Property**: See Entry type: Parent Domain Property.

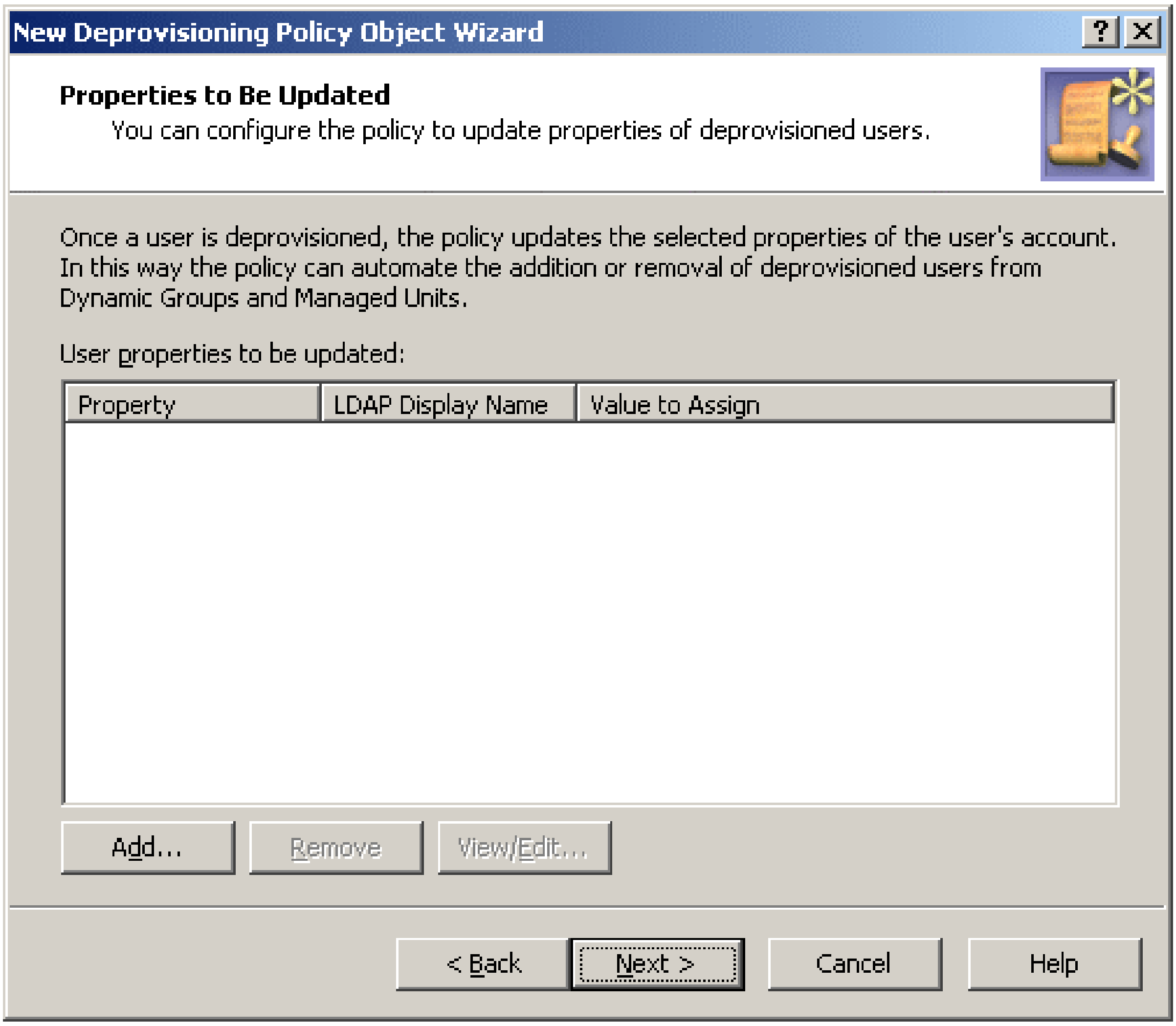
When you are done configuring a value for the **'Initiator ID' must be** condition, click **OK** to close the **Configure Value** dialog. This will add the value to the Initiator ID entry properties. If necessary, you can modify the value by clicking the **Configure** button in the **Add Entry** window and then managing the list of entries in the **Configure Value** dialog.

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 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. This will add the rule to the **Options to Prevent Logon** page of the wizard. If necessary, you can modify the rule by clicking **Configure** on that page and then managing the list of entries in the **Configure Value** dialog.

Once you have completed the **Options to Prevent Logon** page, click **Next** to display the **Properties to Be Updated** page.

Figure 60: Properties to Be Updated



New Deprovisioning Policy Object Wizard

Properties to Be Updated

You can configure the policy to update properties of deprovisioned users.

Once a user is deprovisioned, the policy updates the selected properties of the user's account. In this way the policy can automate the addition or removal of deprovisioned users from Dynamic Groups and Managed Units.

User properties to be updated:

Property	LDAP Display Name	Value to Assign

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 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**.

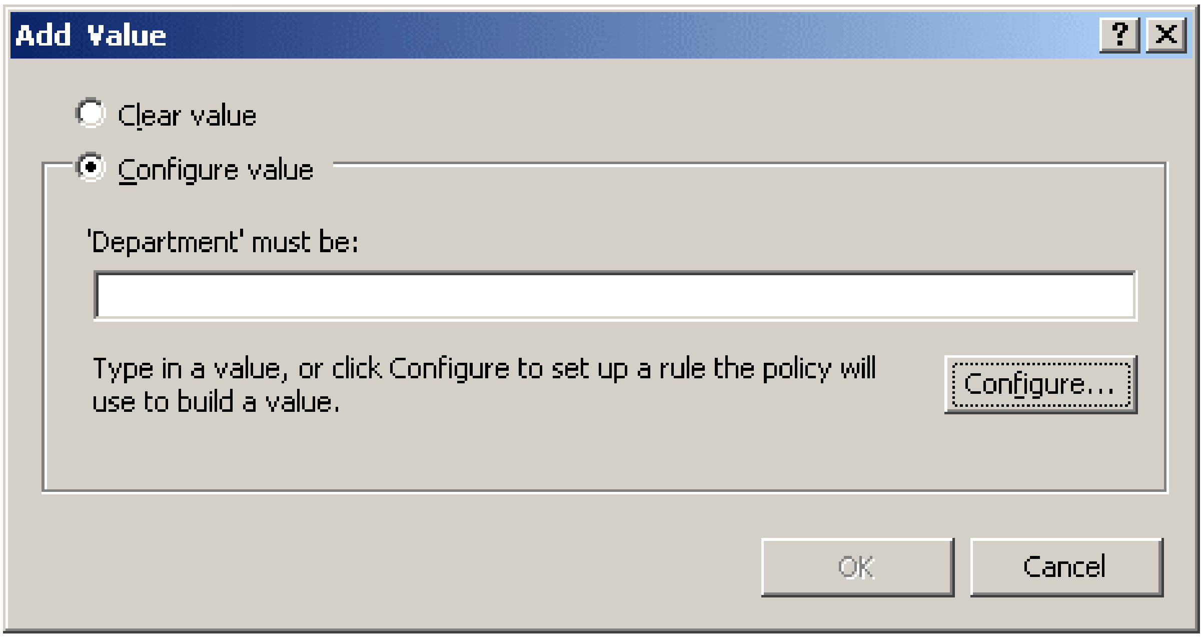
- **View/Edit**: Allows you to modify the update rule for the property you select from the list.

Clicking **Add** displays the **Select Object Property** dialog 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, you are presented with the **Add Value** dialog so you can proceed to configuring a property update rule.

Figure 61: Add Value



Add Value

○ Clear value

● Configure value

'Department' must be:

Type in a value, or click Configure to set up a rule the policy will use to build a 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 **Configure** and follow the instructions provided in Configuring a property update rule.

When you are done configuring a value, click **OK** to close the **Add Value** dialog. 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 **View/Edit** beneath the list of properties. This displays a dialog, similar to the **Add Value** dialog, 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.

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.

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.

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 **Configure**, and use the following instructions to complete the **Configure Value** dialog.

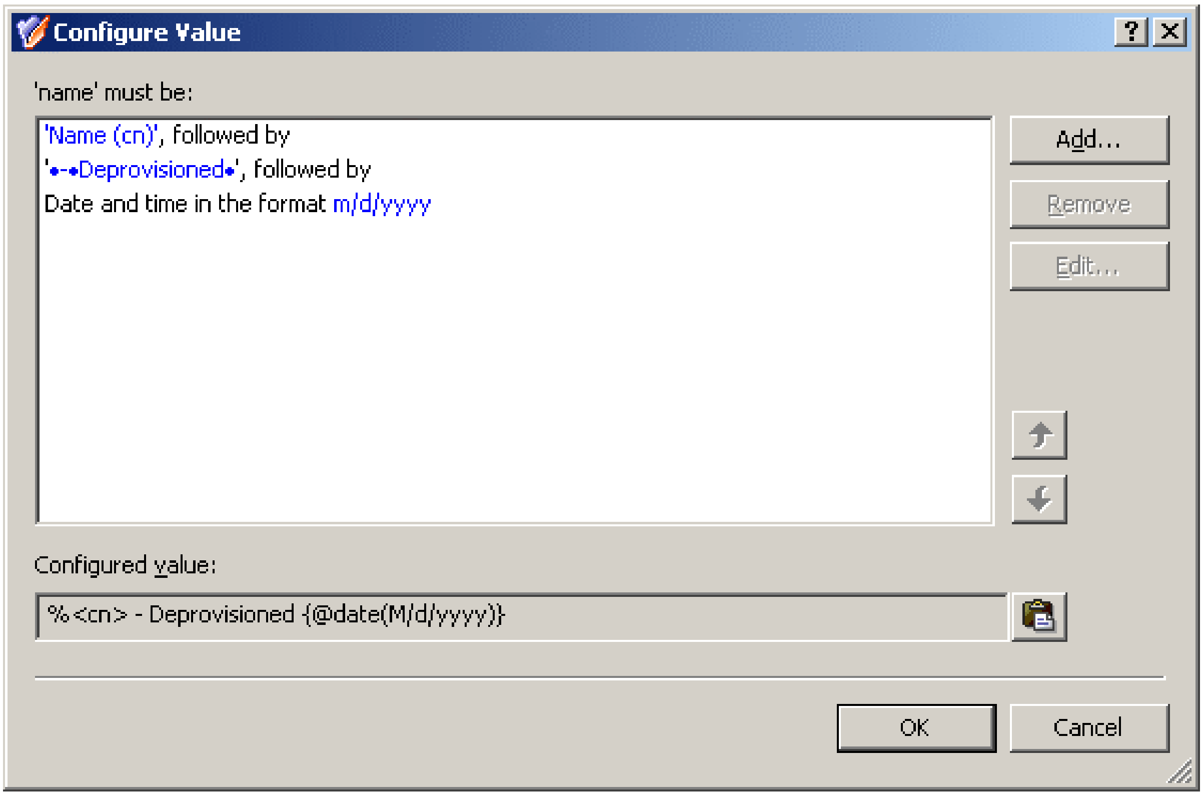
1. Click **Add**.
2. In the **Add Entry** window, click **User Property** under **Entry type**, and configure the entry as follows:
   a. Click **Select** and choose the **Name** property.
   b. Click **All characters of the property value**.

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 should look like the following figure.

Figure 62: Configure Value



Configure Value

'name' must be:

'Name (cn)', followed by
'•-•Deprovisioned•', followed by
Date and time in the format m/d/yyyy

Configured value:

%<cn> - Deprovisioned {@date(M/d/yyyy)}

Click **OK** to close the **Configure Value** dialog. Then, click **Next** and follow the instructions in the wizard to create the Policy Object.

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.

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.

Creating and configuring the Managed Unit

You can create and configure the Managed Unit by using the Active Roles Console.

To create and configure the Managed Unit for deprovisioned users

1. In the Console tree, under **Configuration**, right-click **Managed Units**, and select **New** > **Managed Unit**.
2. In **Name**, enter a name for the Managed Unit. For example, enter `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**.

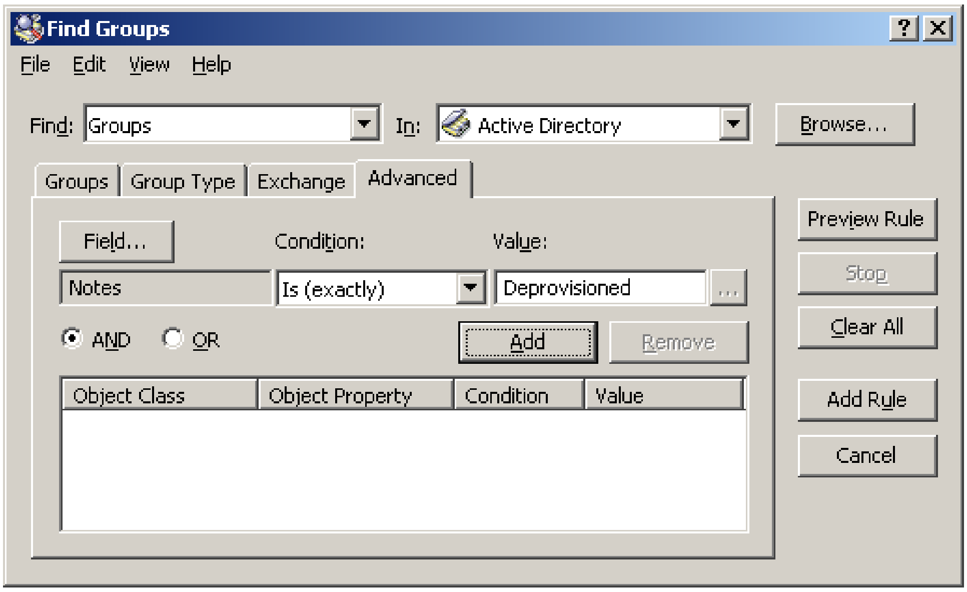
b. In the **Membership Rule Type** dialog, click **Include by Query**, and then click **OK**.

c. Use the **Create Membership Rule** window to set up the rule:

  - In **Find**, click **Users**.
  - Click **Browse** and select **Active Directory**.
  - Click **Advanced**.
  - Click **Field**, then click **Notes**.
  - In **Condition**, click **Is (exactly)**.
  - In **Value**, enter **`Deprovisioned`**.

  At this point, the window must look like the following figure.

Figure 63: Find Groups



Find: Groups

In: Active Directory

Field: Notes

Condition: Is (exactly)

Value: Deprovisioned

● AND ○ OR

  - Click **Add**.
  - Click **Add Rule**.

5. On the wizard page, click **Add**.
6. In the **Membership Rule Type** dialog, click **Retain Deprovisioned**, then click **OK**.
7. Click **Next**, click **Next**, and then click **Finish**.

Configuring the Policy Object

You can configure the Policy Object you need by modifying the Policy Object that implements the scenario described in Scenario 1: Disabling and renaming the user account upon deprovisioning.

Display the **Properties** dialog 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. Click **Change Properties**.

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.
2. Select the check box next to the **Notes** property, and then click **OK**.
3. In the **Add Value** dialog, type **Deprovisioned** in the **'Notes' must be** box, and then click **OK**.

Click **OK** to close the **Group Object Deprovisioning Policy Properties** dialog.

Applying the Policy Object

You can apply the Policy Object without closing its **Properties** dialog. 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 to.

   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 for the Policy Object.

For more information on how to apply a Policy Object, see Applying Policy Objects and Managing policy scope.

Office 365 Licenses Retention

This policy is intended to automate retention of all or selected Microsoft 365 licenses assigned to an Azure AD user after the Azure AD user is deprovisioned successfully.

How the Microsoft 365 Licenses Retention 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 Microsoft 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 Microsoft 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 Creating a Policy Object. The **Office 365 Licenses Retention** policy from the **User Deprovisioning Policies** must be selected to enable retention of the required Microsoft 365 licenses upon Azure AD user deprovisioning.

> NOTE: The **Office 365 Licenses Retention** policy is enabled only if Azure AD is configured.

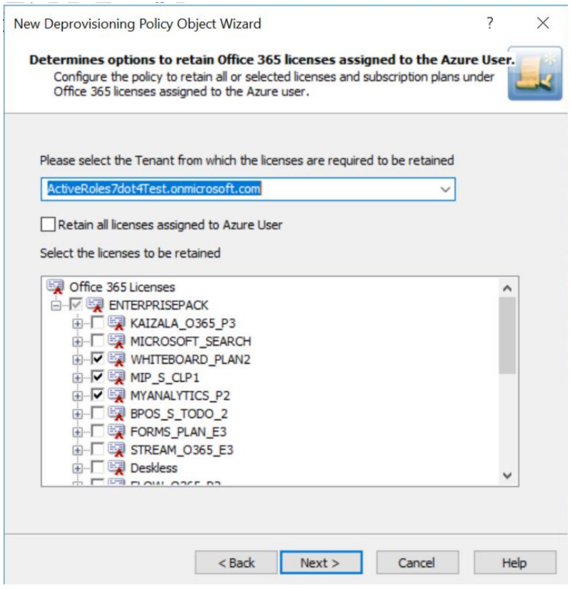
Configuring a Microsoft 365 license retention policy

You can configure a new Microsoft 365 license retention policy with the **Office 365 License Retention** policy type in the Active Roles Console.

To configure an Microsoft 365 license retention policy

1. On the **Policy to Configure** page, select **Office 365 License Retention**, then click **Next**.

Figure 64: Office 365 Licenses Retention page



New Deprovisioning Policy Object Wizard

Determines options to retain Office 365 licenses assigned to the Azure User.
Configure the policy to retain all or selected licenses and subscription plans under Office 365 licenses assigned to the Azure user.

Please select the Tenant from which the licenses are required to be retained

ActiveRoles7dot4Test.onmicrosoft.com

□ Retain all licenses assigned to Azure User

Select the licenses to be retained

- Office 365 Licenses
  - ☑ ENTERPRISEPACK
    - □ KAIZALA_O365_P3
    - □ MICROSOFT_SEARCH
    - ☑ WHITEBOARD_PLAN2
    - ☑ MIP_S_CLP1
    - ☑ MYANALYTICS_P2
    - □ BPOS_S_TODO_2
    - □ FORMS_PLAN_E3
    - □ STREAM_O365_E3
    - □ Deskless

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 Microsoft 365 licenses after successful deprovisioning.

   - Select the check boxes corresponding to the specific Microsoft 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 to locate and select the objects on which you want to enforce the policy.

5. Click **Next**, then click **Finish**.

NOTE: Consider the following when configuring an Microsoft 365 licenses retention policy:

- 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 Deprovisioning Policy Objects and Creating a Policy Object.

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 11: 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.	Not applicable
Azure User Office 365 licenses are retained.	Not applicable

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 the Group Membership Removal 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 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 email 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 12: 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 an Active Roles Dynamic Group (for more information, see Dynamic groups). The Dynamic Group policy detects the removal, and might add the deprovisioned user back to the Dynamic Group. To avoid this, Active Roles does not allow Dynamic Groups to hold deprovisioned users. Once a user is deprovisioned, the user account is removed from all Dynamic Groups.

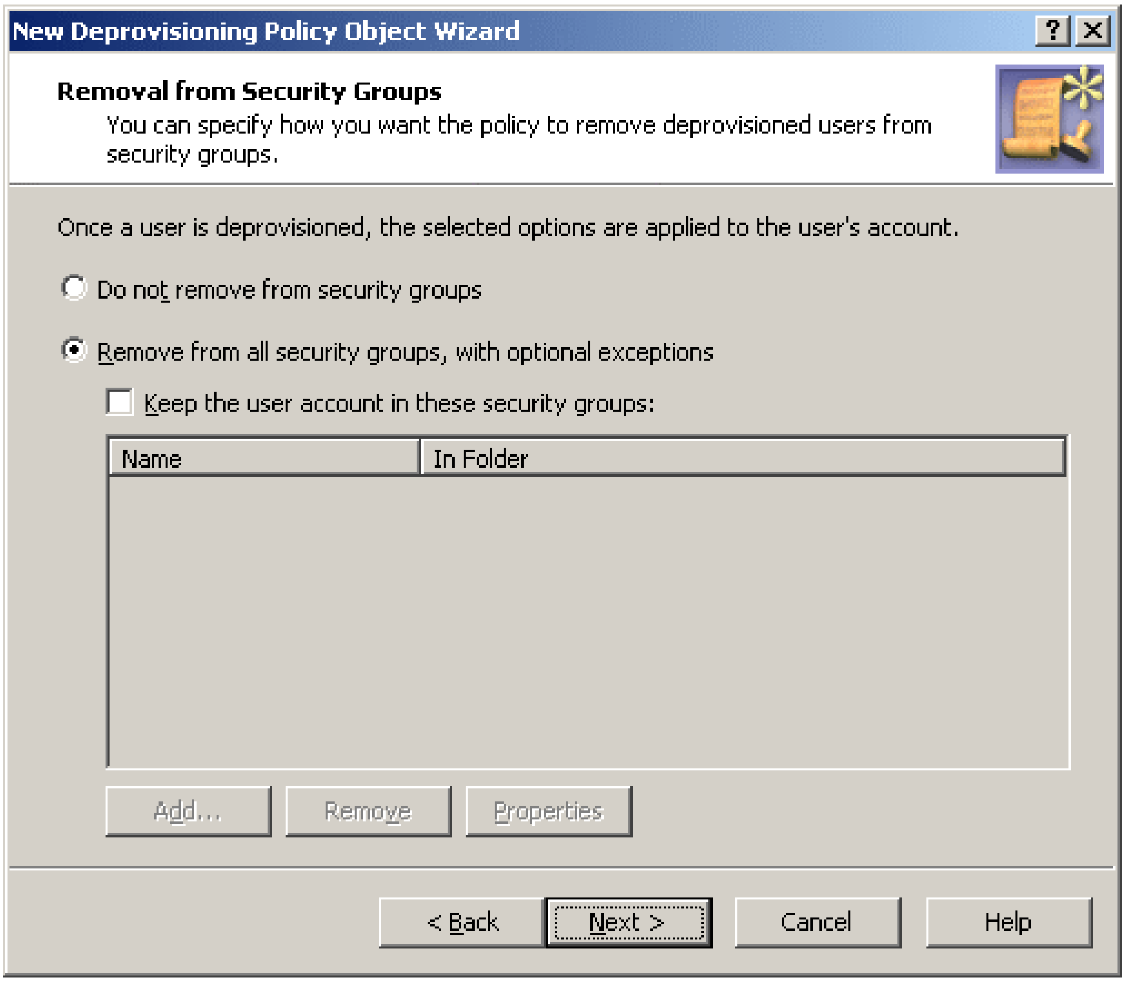
Configuring a Group Membership Removal policy

You can configure a new Group Membership Removal policy with the Active Roles Console.

To configure a Group Membership Removal policy

1. On the **Policy to Configure** page, select **Group Membership Removal**, then click **Next**.

   Figure 65: Removal from Security Groups

   New Deprovisioning Policy Object Wizard

   Removal from Security Groups

   You can specify how you want the policy to remove deprovisioned users from security groups.

   Once a user is deprovisioned, the selected options are applied to the user's account.

   ○ Do not remove from security groups

   ● Remove from all security groups, with optional exceptions

   □ Keep the user account in these security groups:

Name	In Folder

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.

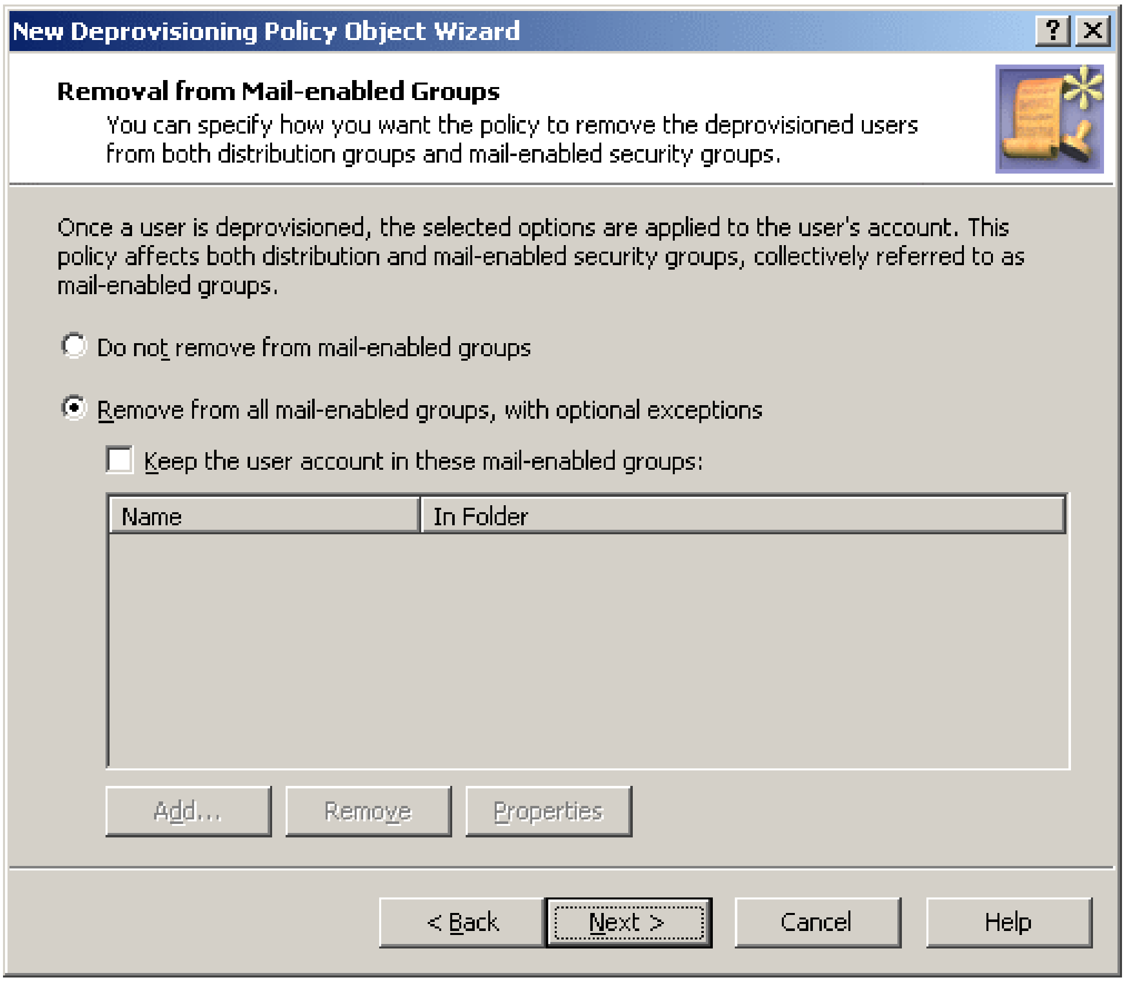
3. If you selected **Remove from all security groups, with optional exceptions**, specify whether you want the policy not to remove the user account from certain security groups. Do one of the following:
   - Select the **Keep the user account in these security groups** check box and set up the list 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.

Figure 66: Removal from Mail-enabled Groups



New Deprovisioning Policy Object Wizard

Removal from Mail-enabled Groups

You can specify how you want the policy to remove the deprovisioned users from both distribution groups and mail-enabled security groups.

Once a user is deprovisioned, the selected options are applied to the user's account. This policy affects both distribution and mail-enabled security groups, collectively referred to as mail-enabled groups.

○ Do not remove from mail-enabled groups

● Remove from all mail-enabled groups, with optional exceptions

□ Keep the user account in these mail-enabled groups:

Name	In Folder

6. If you selected **Remove from all mail-enabled groups, with optional exceptions**, 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 to locate and select the objects you want.
9. Click **Next**, 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, Active Roles removes the user account from all groups.

Creating and configuring the Group Membership Removal 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.

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.

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.

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 email addressed to deprovisioned users.
- Force the mailbox of the deprovisioned user to send automatic replies.

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 email sent to the mailbox of the deprovisioned user, and to authorize designated persons to monitor such email.

How the Exchange Mailbox Deprovisioning 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 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 mailbox.

Table 13: Policy effect on a user’s mailbox

Option	Policy effect
Hide the mailbox from the Global Address List (GAL), to prevent access to the mailbox	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 emails 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 mailbox.
Disallow forwarding messages to alternate recipients	Email addressed to the deprovisioned user is not forwarded to an alternate recipient.
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	Email 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 email 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.
Auto-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

Option	Policy effect
	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.

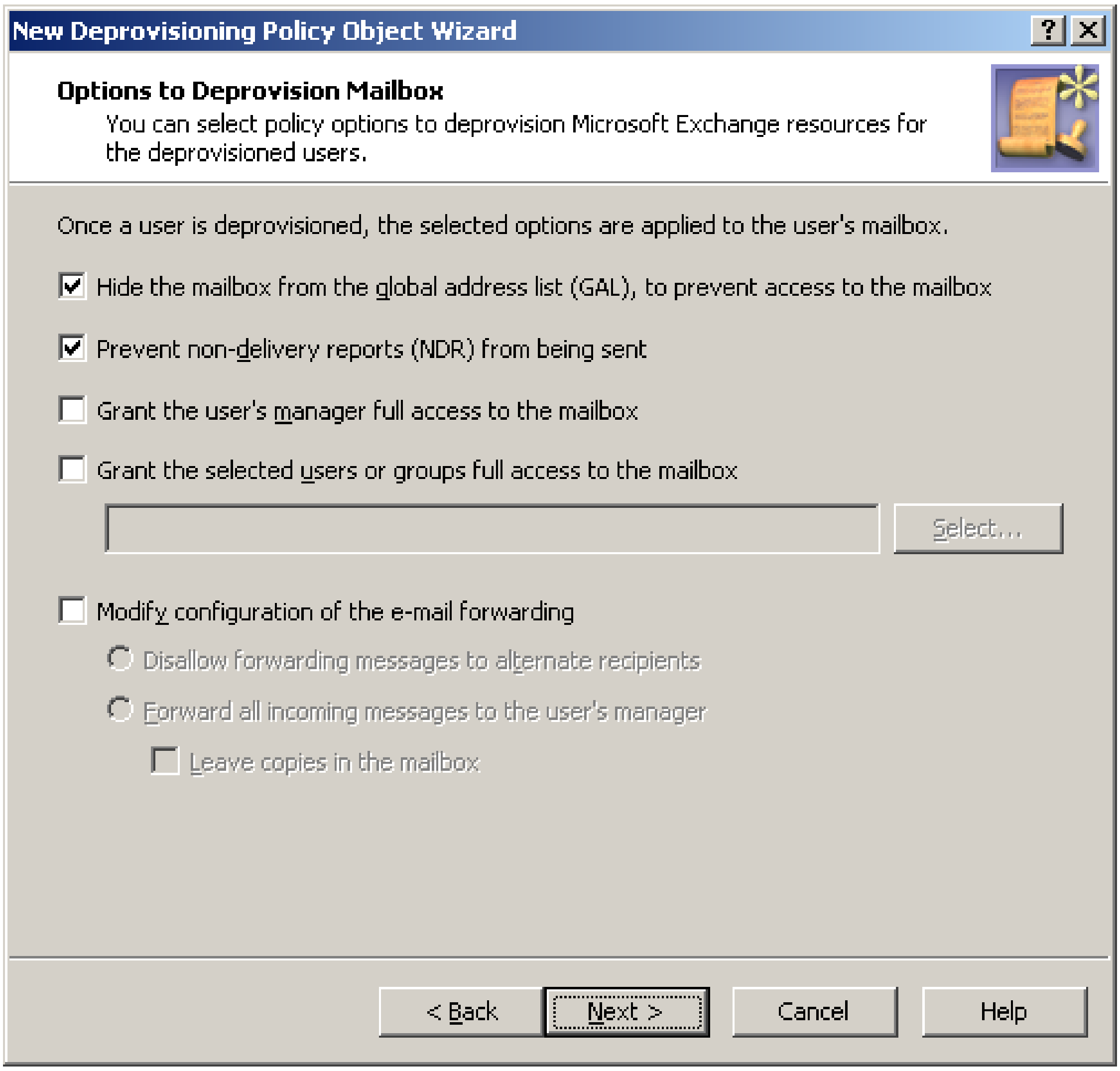
Configuring an Exchange Mailbox Deprovisioning policy

You can configure a new Exchange Mailbox Deprovisioning policy with the Active Roles Console.

To configure an Exchange Mailbox Deprovisioning policy

1. On the **Policy to Configure** page, select **Exchange Mailbox Deprovisioning**, then click **Next**.

Figure 67: Options to Deprovision Mailbox



Options to Deprovision Mailbox

You can select policy options to deprovision Microsoft Exchange resources for the deprovisioned users.

Once a user is deprovisioned, the selected options are applied to the user's mailbox.

☑ 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

☐ Modify configuration of the e-mail forwarding

○ Disallow forwarding messages to alternate recipients

○ Forward all incoming messages to the user's manager

☐ Leave copies in the mailbox

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), 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**
   - **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 email 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 email 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, 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 to locate and select the objects you want.
10. Click **Next**, then click **Finish**.

Scenario: Hide mailbox and forward email 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 email forwarding so that email messages addressed to the deprovisioned user are sent to the user’s manager, without delivering them to the user 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 selected container, Active Roles hides the deprovisioned user from the Exchange address lists and configures the forwarding address for that user as prescribed by this policy.

Creating and configuring the Exchange Mailbox Deprovisioning 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.

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 email 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.

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.

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 the Home Folder Deprovisioning 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 14: 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	Designates the specified user or group as the owner of the deprovisioned

Option	Policy effect
selected user or group the owner of the home folder	user’s home folder. The owner is authorized to control how permissions are set on the folder, and can grant permissions to others.
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.

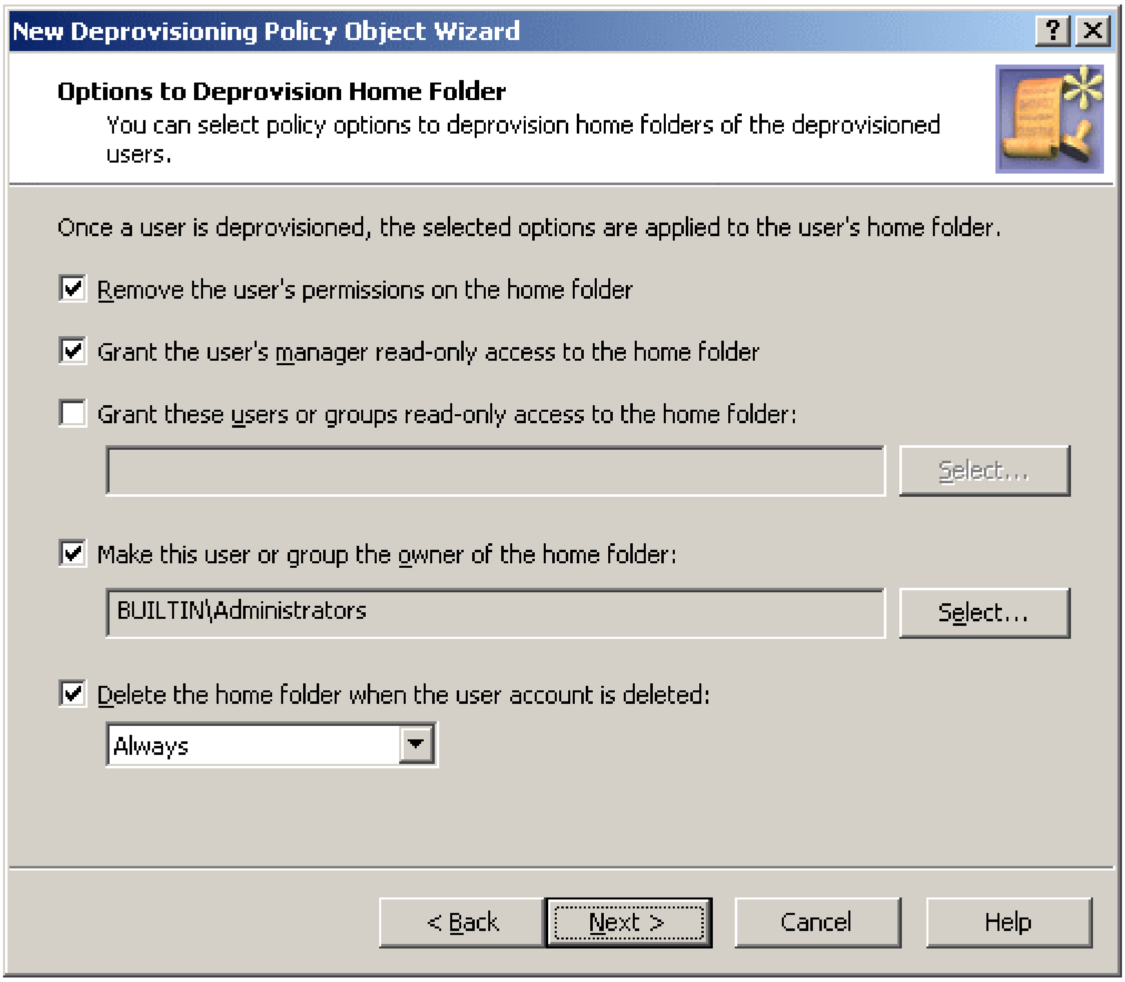
Configuring a Home Folder Deprovisioning policy

You can configure a Home Folder Deprovisioning policy with the Active Roles Console.

To configure a Home Folder Deprovisioning policy

1. On the **Policy to Configure** page, select **Home Folder Deprovisioning**, and then click **Next**.

Figure 68: Options to Deprovision Home Folder



New Deprovisioning Policy Object Wizard

Options to Deprovision Home Folder

You can select policy options to deprovision home folders of the deprovisioned users.

Once a user is deprovisioned, the selected options are applied to the user's home folder.

☑ 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: BUILTIN\Administrators

☑ Delete the home folder when the user account is deleted: Always

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 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 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 must be applied. To do so, click **Add**, and use the **Select Objects** dialog to locate and select the objects you want.
8. Click **Next**, 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.

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.

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.

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.

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 the User Account Relocation 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 Roles to move deprovisioned user accounts 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.

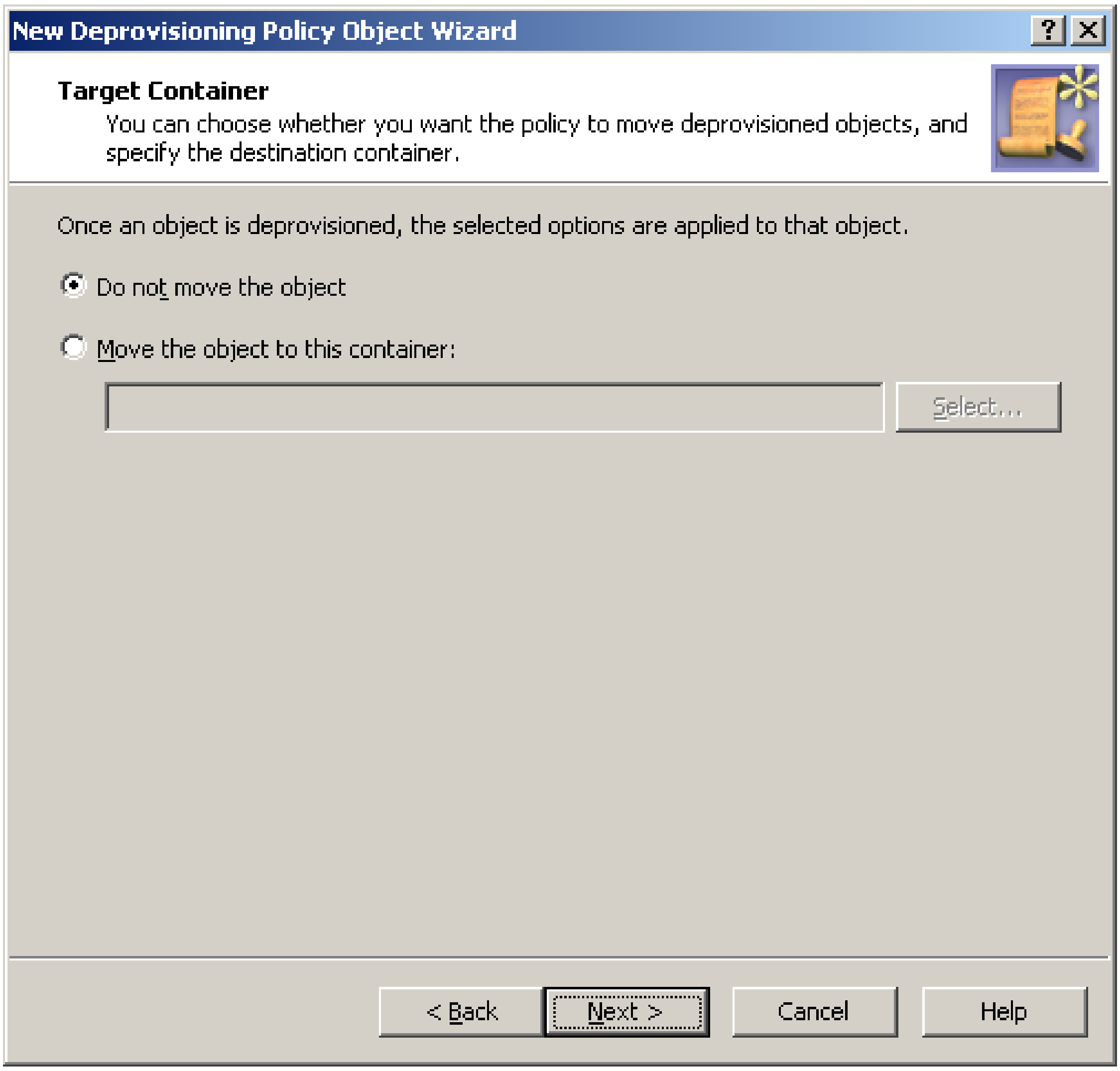
Configuring a User Account Relocation policy

You can configure a new User Account Relocation policy for deprovisioned user accounts with the New Deprovisioning Policy Object Wizard of the Active Roles Console.

To configure a User Account Relocation policy

1. On the **Policy to Configure** page, select **User Account Relocation**, then click **Next**.

   Figure 69: Target container

   New Deprovisioning Policy Object Wizard

   Target Container

   You can choose whether you want the policy to move deprovisioned objects, and specify the destination container.

   Once an object is deprovisioned, the selected options are applied to that object.

   ● Do not move the object

   ○ Move the object to this container:

2. On the **Target Container** page, do one of the following, 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 will be applied. To do so, click **Add**, and use the **Select Objects** dialog 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.

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.

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.

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.

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 the User Account Permanent Deletion 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.

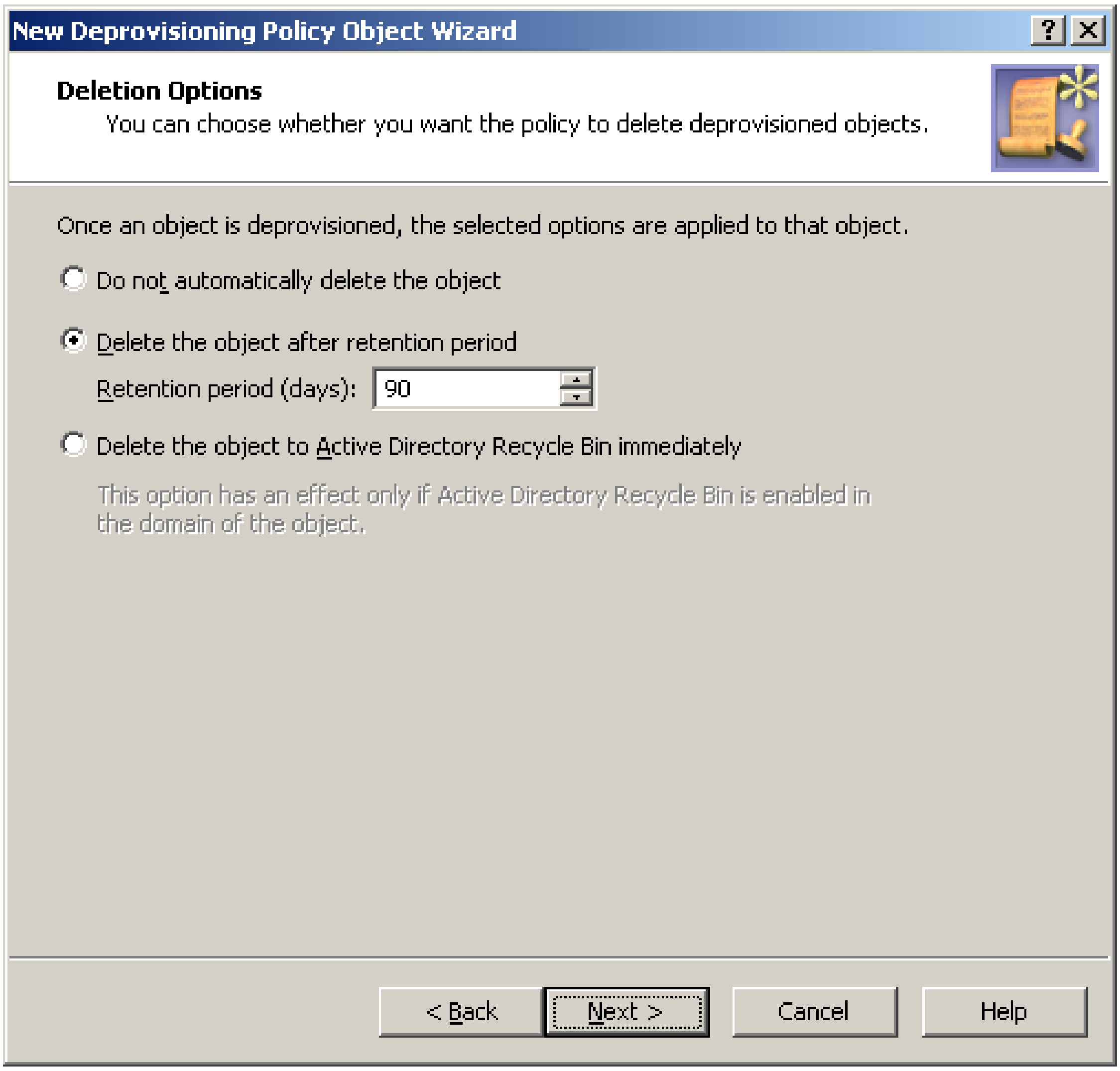
Configuring a User Account Permanent Deletion policy

You can configure a new User Account Permanent Deletion policy with the New Deprovisioning Policy Object Wizard of the Active Roles Console.

To configure a User Account Permanent Deletion policy

1. On the **Policy to Configure** page, select **User Account Permanent Deletion**, then click **Next**.

   Figure 70: Deletion options

   New Deprovisioning Policy Object Wizard

   Deletion Options

   You can choose whether you want the policy to delete deprovisioned objects.

   Once an object is deprovisioned, the selected options are applied to that object.

   ○ Do not automatically delete the object

   ● Delete the object after retention period

   Retention period (days): 90

   ○ Delete the object to Active Directory Recycle Bin immediately

   This option has an effect only if Active Directory Recycle Bin is enabled in the domain of the object.

2. On the **Deletion Options** page, do one the following, 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.

  > NOTE: If you select this option, apply the policy to domains that have Active Directory Recycle Bin enabled, or the policy will have no effect.
  >
  > With this option, once a user account is deprovisioned, the policy causes Active Roles to delete the user account immediately. In a domain where Active Directory Recycle Bin is enabled, this deletion means that the account is marked as deleted and moved to a specified container from which it can be restored later if necessary without any data loss.

Click **Next**.

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 to locate and select the objects you want.
4. Click **Next**, 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.

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 more information about the wizard, see Creating a Policy Object.

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.

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.

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 the Group Object Deprovisioning 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.

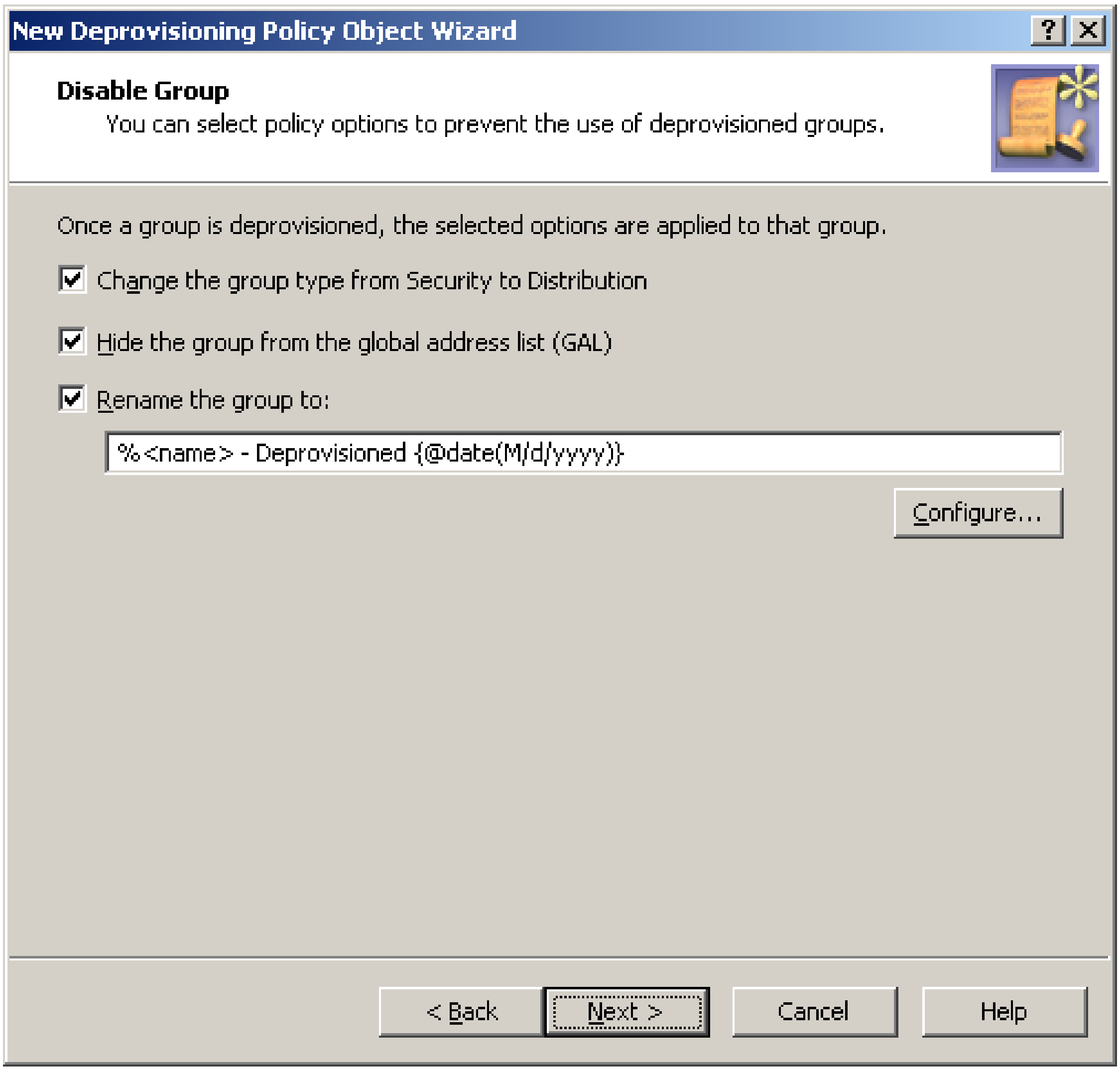
Configuring a Group Object Deprovisioning policy

You can configure a new Group Object Deprovisioning policy with the Active Roles Console.

To configure a Group Object Deprovisioning policy

1. On the **Policy to Configure** page, select **Group Object Deprovisioning**, then click **Next**.

Figure 71: Disable Group



New Deprovisioning Policy Object Wizard

Disable Group

You can select policy options to prevent the use of deprovisioned groups.

Once a group is deprovisioned, the selected options are applied to that group.

☑ Change the group type from Security to Distribution

☑ Hide the group from the global address list (GAL)

☑ Rename the group to:

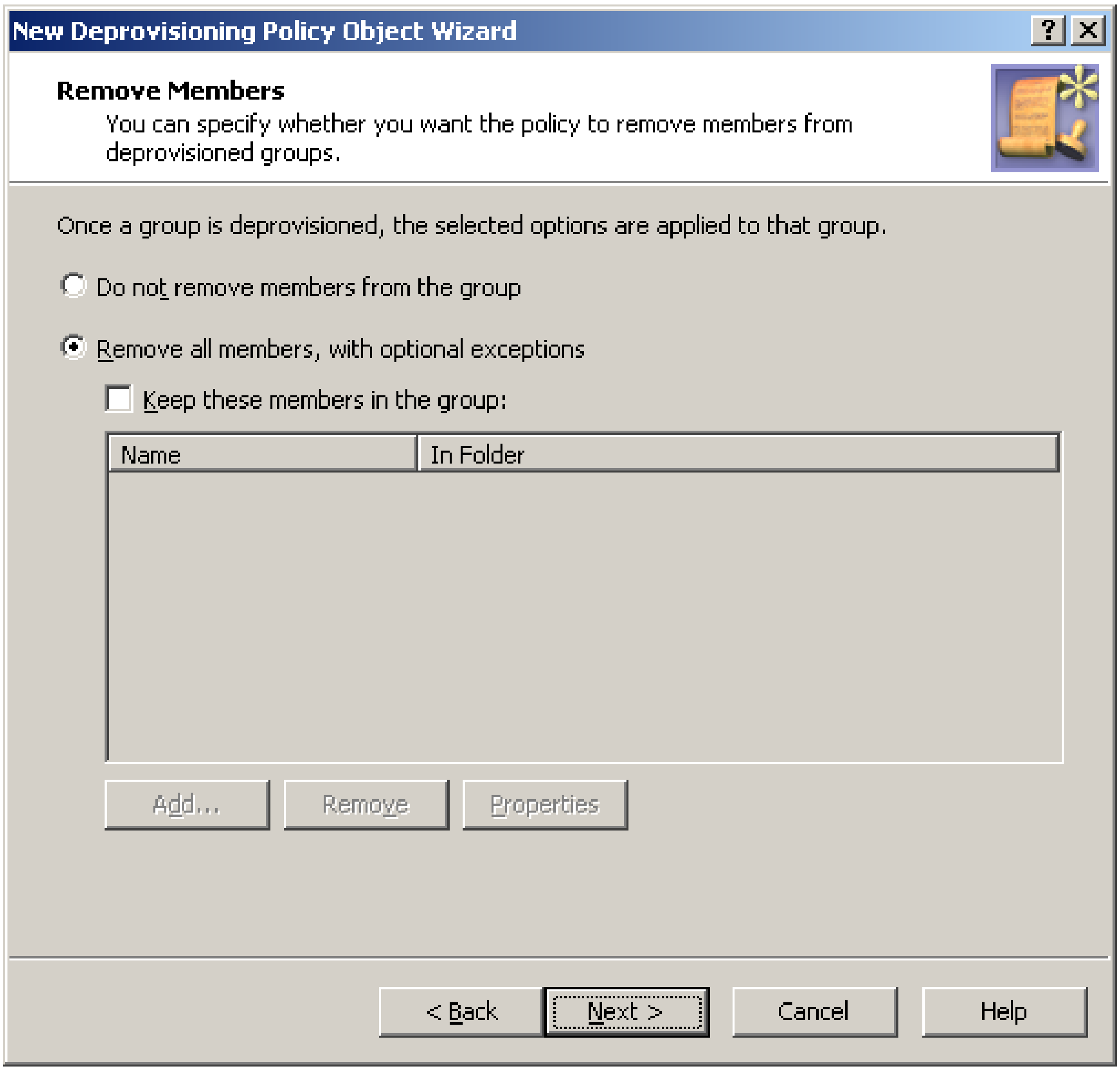
%<name> - Deprovisioned {@date(M/d/yyyy)}

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**: 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.
3. If you selected **Rename the group to**, specify how you want the policy to update the group name when deprovisioning a group. To do so, click **Configure** and

complete the **Configure Value** dialog by using the procedure outlined later in this topic. For more information, see Configuring a property update rule.

4. Click **Next**.

Figure 72: Remove members



New Deprovisioning Policy Object Wizard

Remove Members

You can specify whether you want the policy to remove members from deprovisioned groups.

Once a group is deprovisioned, the selected options are applied to that group.

○ Do not remove members from the group

● Remove all members, with optional exceptions

□ Keep these members in the group:

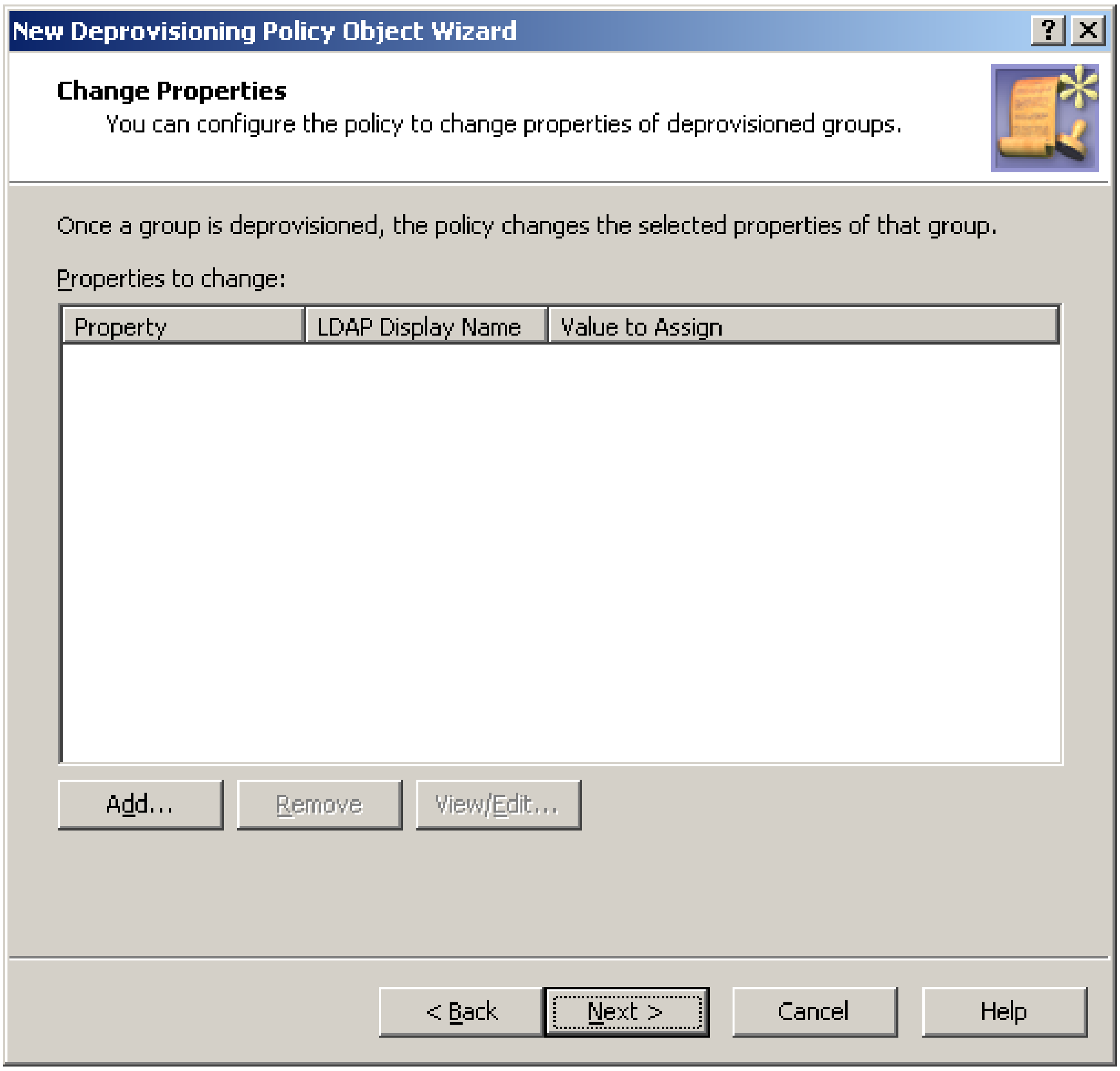
Name	In Folder

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 **Remove all members, with optional exceptions**, 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. Click **Next**.

Figure 73: Change Properties



Change Properties

You can configure the policy to change properties of deprovisioned groups.

Once a group is deprovisioned, the policy changes the selected properties of that group.

Properties to change:

Property	LDAP Display Name	Value to Assign

8. 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.

    Specify how you want the policy to update properties of the group object when deprovisioning a group:

    - Click **Add**, then add property rules by completing the **Select Object Property** dialog.
    - Use **View/Edit** to modify existing rules.
    - Use **Remove** to delete existing rules.

9. Click **Next**.
10. 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 to locate and select the objects you want.

11. Click **Next**, then click **Finish**.

To complete the Configure Value dialog

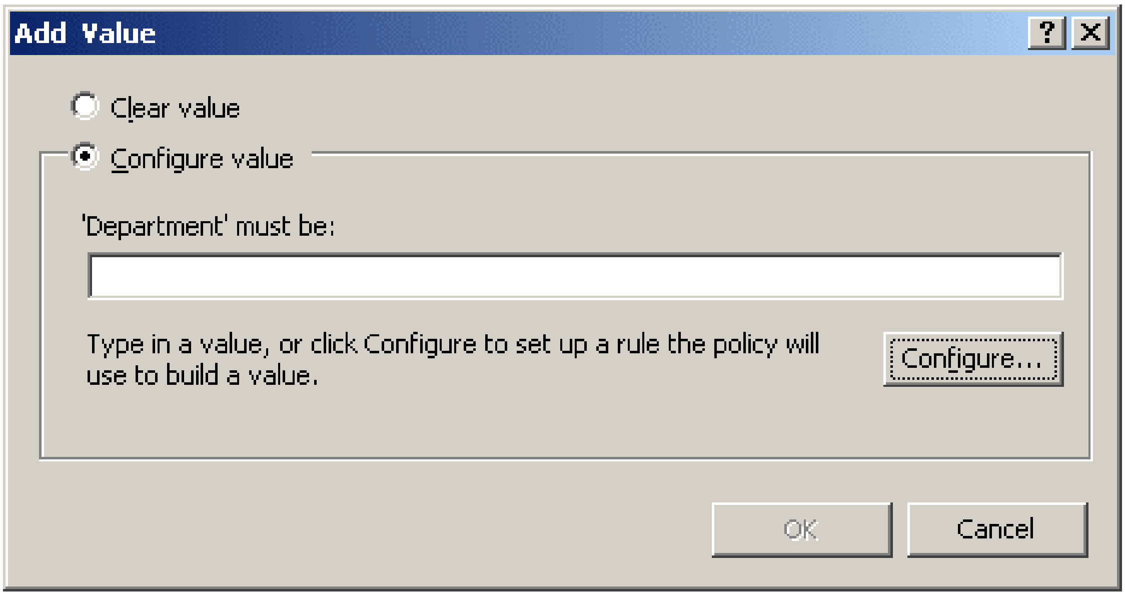
1. Click **Add**.
2. Configure an entry to include in the value. For more information, see Configuring entries.
3. In the **Configure Value** dialog, add more entries, delete or edit existing ones, then click **OK**.

To complete the Select Object Property dialog

1. From the **Object property** list, select an object property, then click **OK**. The **Add Value** dialog appears.

   If you select multiple properties, the **Add Value** dialog 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.

Figure 74: Add value



○ Clear value

● Configure value

'Department' must be:

Type in a value, or click Configure to set up a rule the policy will use to build a value.

2. In the **Add Value** dialog, 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.
3. When you are done configuring a value, click **OK** to close the **Add Value** dialog. 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 **View/Edit** beneath the list of properties. This displays a dialog, similar to the **Add Value** dialog, allowing you to choose a different update option or set up a different value for the **'property' must be** condition.

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, Active Roles disables and renames the group as prescribed by this policy.

Configuring the Group Object Deprovisioning 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.

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, if empty, enter the following name under **Rename the group to**:

```
%<name> - Deprovisioned {@date(M/d/yyyy)}
```

Click **Next** and follow the instructions in the wizard to create the Policy Object.

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.

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, Active Roles automatically adds that group to the Managed Unit you created.

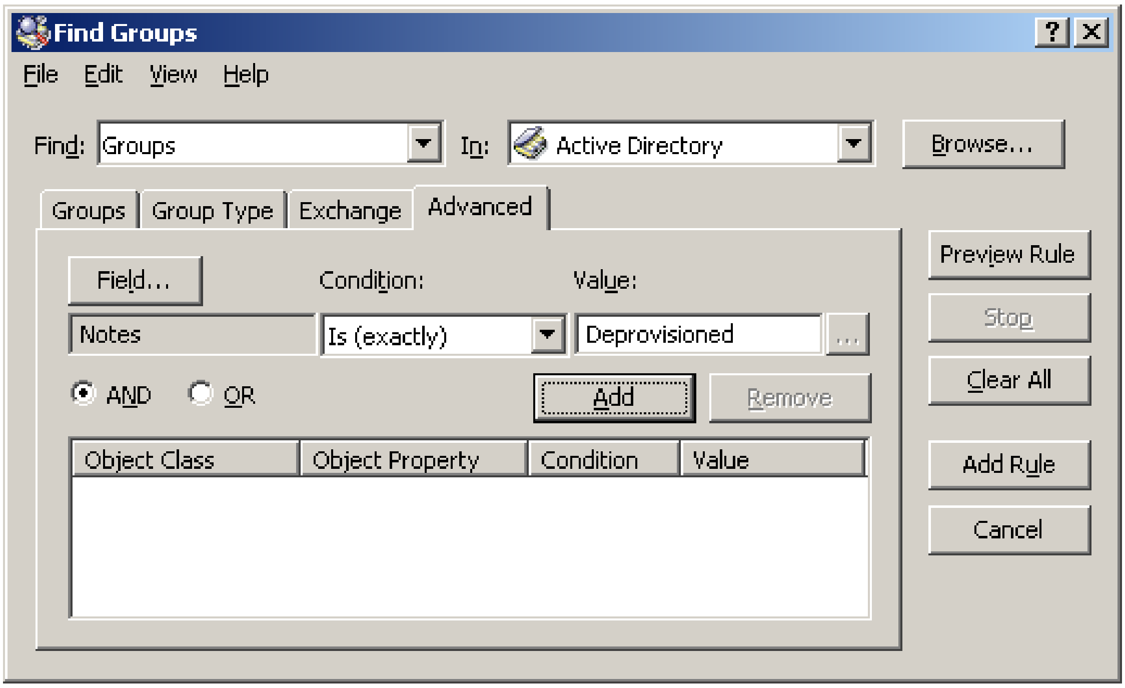
Creating and configuring the Managed Unit for the Group Object Deprovisioning policy

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, 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**.
        - Navigate to **Advanced**.
        - Click **Field**, and then click **Notes**.
        - In **Condition**, click **Is (exactly)**.
        - In **Value**, type `Deprovisioned`.

    At this stage, the window should look as follows.

Figure 75: Find Groups



Find: Groups In: Active Directory

Condition: Is (exactly)

Value: Deprovisioned

Notes

● AND ○ OR

- Click **Add**.
- Click **Add Rule**.

5. On the wizard page, click **Add**.
6. In the **Membership Rule Type** dialog, click **Retain Deprovisioned**, and then click **OK**.
7. Click **Next**, click **Next**, and then click **Finish**.

Configuring the Policy Object for Group Object Deprovisioning

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 group upon deprovisioning.

Display the **Properties** dialog for that Policy Object and navigate to **Policies**. Then, select the policy from the list, and click **View/Edit** to display the **Group Object Deprovisioning Policy Properties** dialog. Navigate to **Change Properties**.

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.
2. Select the check box next to the **Notes** property, and then click **OK**.

3. In the **Add Value** dialog, type `Deprovisioned` in the **'Notes' must be** box, and then click **OK**.

Click **OK** to close the **Group Object Deprovisioning Policy Properties** dialog.

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.

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 the Group Object Relocation 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.

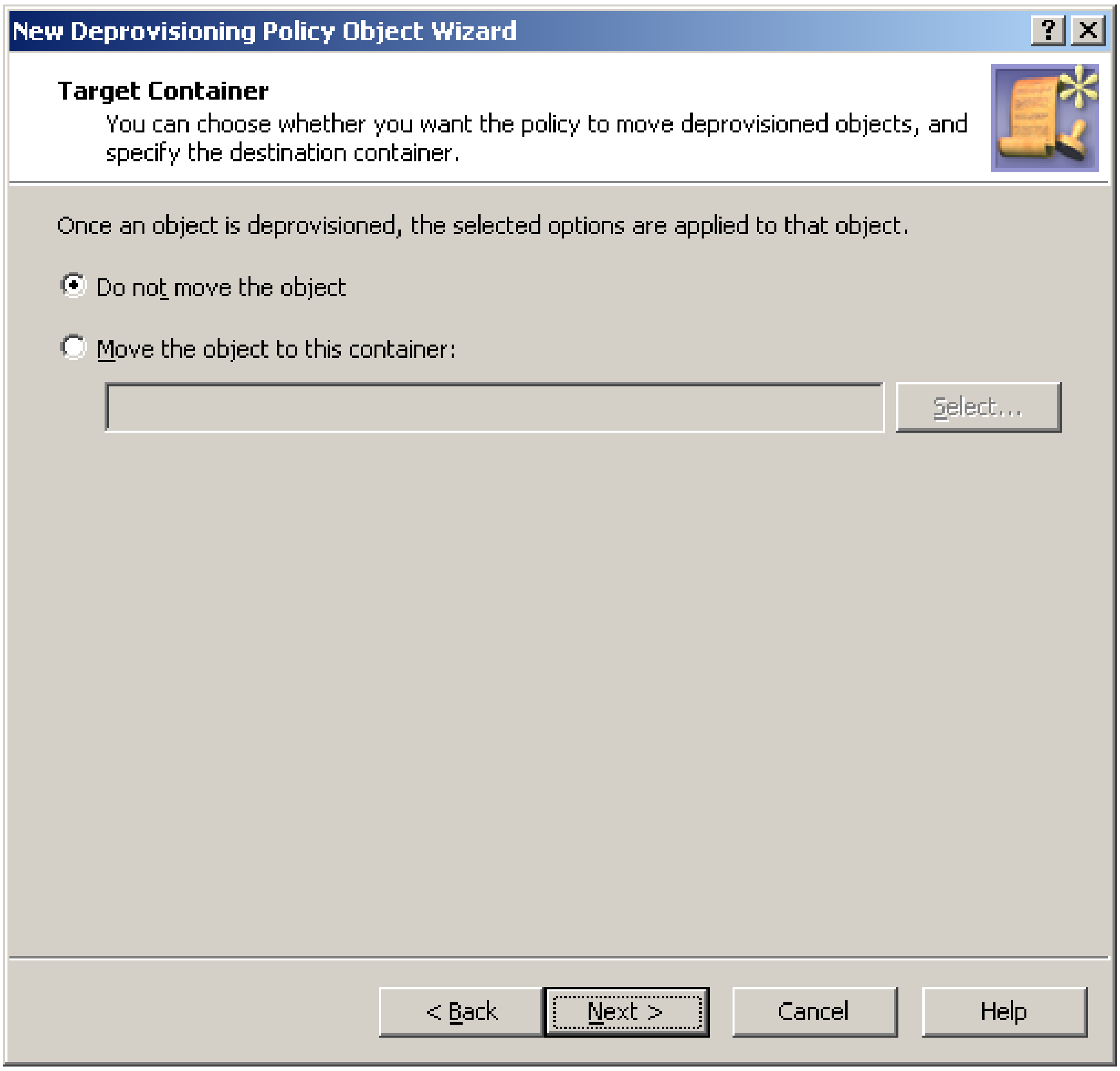
Configuring a Group Object Relocation policy

You can configure a new Group Object Relocation Policy with the Active Roles Console.

To configure a Group Object Relocation policy

1. On the **Policy to Configure** page, select **Group Object Relocation**, then click **Next**.

   Figure 76: Target container

   Target Container

   You can choose whether you want the policy to move deprovisioned objects, and specify the destination container.

   Once an object is deprovisioned, the selected options are applied to that object.

   ● Do not move the object

   ○ Move the object to this container:

2. On the **Target Container** page, do one of the following, 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 to locate and select the objects you want.
4. Click **Next**, 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.

Configuring the Group Object Relocation 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.

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 **Select** to display the **Browse for Container** dialog. 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.

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.

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 the Group Object Permanent Deletion 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.

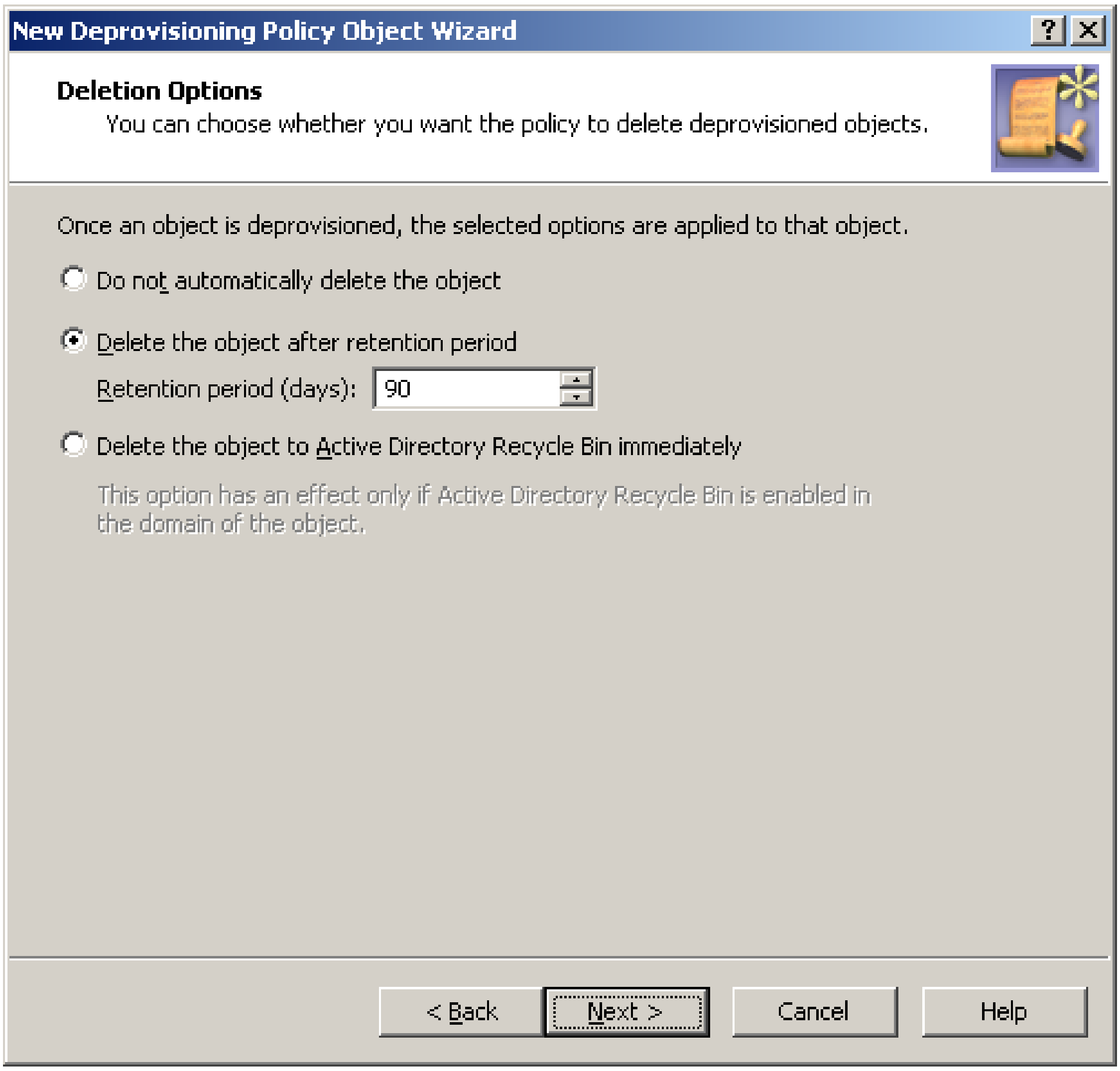
Configuring a Group Object Permanent Deletion policy

You can configure a new Group Object Permanent Deletion Policy with the Active Roles Console.

To configure a Group Object Permanent Deletion policy

1. On the **Policy to Configure** page, select **Group Object Permanent Deletion** and click **Next**.

Figure 77: Deletion Options



New Deprovisioning Policy Object Wizard

Deletion Options

You can choose whether you want the policy to delete deprovisioned objects.

Once an object is deprovisioned, the selected options are applied to that object.

○ Do not automatically delete the object

● Delete the object after retention period

Retention period (days): 90

○ Delete the object to Active Directory Recycle Bin immediately

This option has an effect only if Active Directory Recycle Bin is enabled in the domain of the object.

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.

      > NOTE: If you select the third option, apply this policy to domains that have Active Directory Recycle Bin enabled, or the policy will have no effect.

> With this option, once a group is deprovisioned, Active Roles deletes the deprovisioned group immediately. In a domain where Active Directory Recycle Bin is enabled, this means that the group object will be marked as deleted and moved to a certain container from which it can be restored later without data loss.

   Click **Next**.

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 to locate and select the objects you want.
4. Click **Next**, 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, Active Roles retains the deprovisioned group object for 90 days and then it deletes that object.

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 more information about the wizard, see Creating a Policy Object.

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.

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.

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 the Notification Distribution 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.

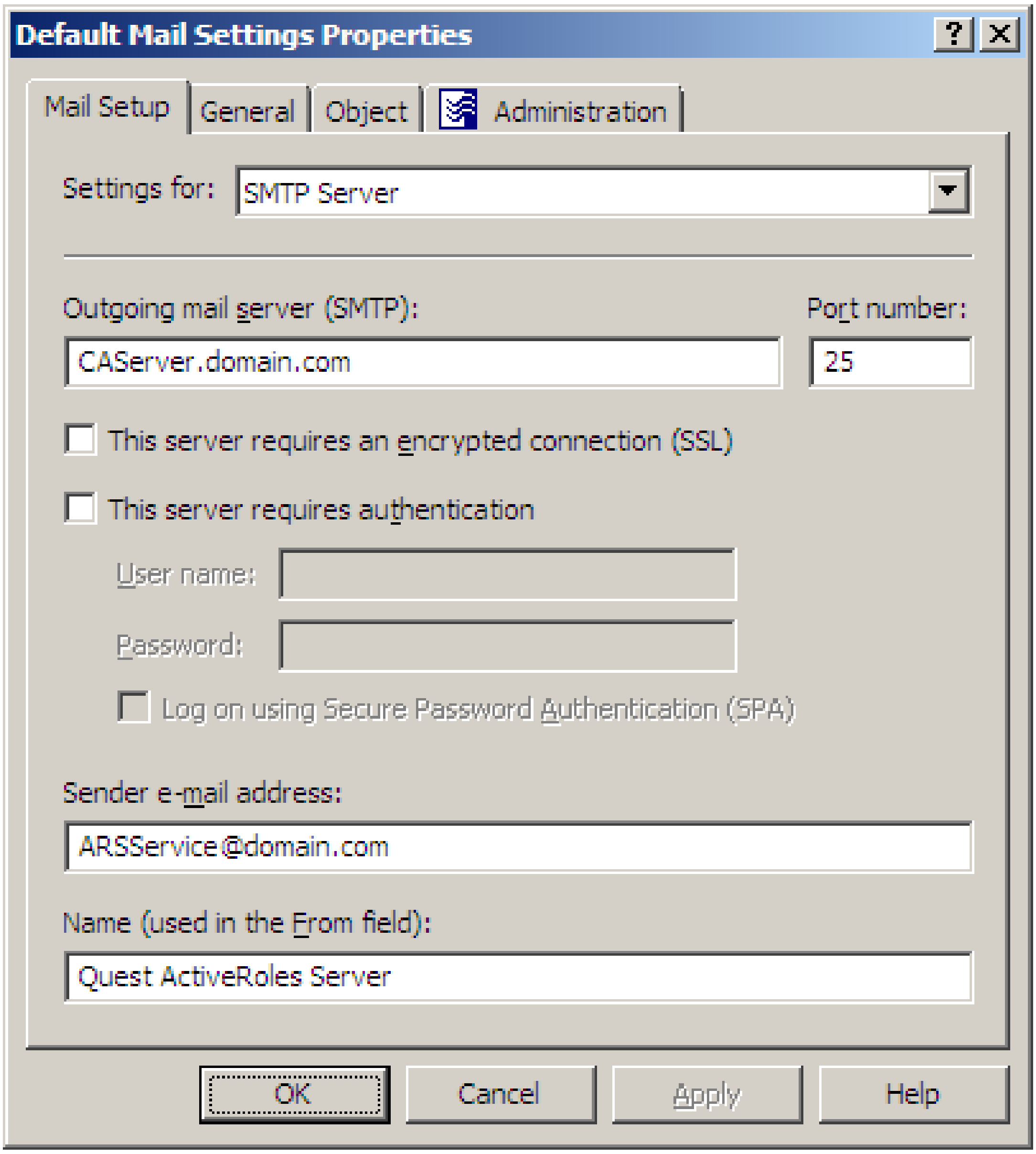
Notification performs on a per-object basis: Each notification message contains information about a request to deprovision one object. When deprovisioning multiple objects, Active Roles sends multiple notification messages, one message per object.

Active Roles sends notification messages via an SMTP server. The policy configuration specifies the outbound SMTP server by using Active Roles email settings that include the name of the SMTP server and information required to connect to the SMTP server.

Configuring email settings

When you click **Settings**, the Console displays the **Properties** dialog for the selected email configuration, with the **Mail Setup** tab that looks like the following figure.

Figure 78: Mail Setup



Default Mail Settings Properties

Settings for: SMTP Server

Outgoing mail server (SMTP): CAServer.domain.com

Port number: 25

□ This server requires an encrypted connection (SSL)

□ This server requires authentication

User name:

Password:

□ Log on using Secure Password Authentication (SPA)

Sender e-mail address: ARSService@domain.com

Name (used in the From field): Quest ActiveRoles Server

On this tab, you can configure the following email 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 email address of the message sender. A valid email 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 email configuration.

When you are done configuring the email server-related settings, click **OK** to close the **Properties** dialog box the email configuration. Then, click **Next** and follow the instructions in the wizard to create the Policy Object.

Configuring a Notification Distribution policy

You can configure a new Notification Distribution policy with the Active Roles Console.

To configure a Notification Distribution policy

1. On the **Policy to Configure** page, select **Notification Distribution Policy**, then click **Next**.
2. On the **Notification Recipients and Message** page, do the following, then click **Next**:
    - Click the button next to **Notification recipients**, and select one or more email 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.

   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 example, before sending a message, Active Roles replaces %<name> with the name of the object to deprovision.

3. On the **Outgoing Mail Server** page, select the email configuration you want the policy to use. In the **Outgoing mail server (SMTP)** list, click the appropriate mail settings.

   > 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** > **Configuration**, right-click **Mail Configuration**, select **New** > **Mail Configuration**, then follow the instructions in the wizard.

4. If you want to view or modify the selected mail settings, click **Settings**, and use the **Mail Setup** tab. For more information, see Configuring email settings.
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 to locate and select the objects you want.
7. Click **Next**, then click **Finish**.

> NOTE: Each email configuration specifies an SMTP server and provides information required to connect to that server. You can view and modify configuration parameters by clicking **Settings**.

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.

Creating the email 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 email address of **ARSService@mycompany.com**.

Create the email 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 email address of the service account: `ARSService@mycompany.com`.
7. In **Name (used in the From field)**, type `Active Roles`.
8. Click **Next**, then click **Finish**.

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.

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. In that dialog, type the administrator's email address, such as `administrator@mydomain.com`, 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**, which is the email configuration you created earlier. Then, click **Next** to display the **Enforce Policy** page.

Add the **Active Directory** folder to the list on the **Enforce Policy** page:

1. Click **Add** to display the **Select Objects** window.
2. In the **Select Objects** window, click **Browse** to display the **Browse for Container** dialog.
3. In the **Browse for Container** dialog, click **Active Directory**, then click **OK**.
4. From the upper list in the **Select Objects** window, select **Active Directory**.
5. Click **Add**, then click **OK** to close the **Select Objects** window.

Click **Next**, 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.

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. For more information, see Report on deprovisioning results.

Reports are delivered via email. 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 the Report Distribution 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 email settings that include the name of the SMTP server and information required to connect to the SMTP server.

Configuring a Report Distribution policy

You can configure a new Report Distribution policy via the Policy to Configure page of the Active Roles Console.

To configure a Report Distribution policy

1. On the **Policy to Configure** page, select **Report Distribution Policy**, then click **Next**.
2. On the **Report Recipients and Message** page, do the following, then click **Next**:
    - Click the button next to **Report recipients**, then select one or more email recipients.
    - In **Message Subject**, type the subject of the message that the specified recipients will receive upon completion of a deprovisioning operation.

      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.

    - 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 email configuration you want the policy to use. In the **Outgoing mail server (SMTP)** list, click the appropriate mail settings.

   This page is similar to the respective wizard page for Notification Distribution policies. For more information, see Configuring a Notification Distribution policy). You can select the email configuration you want the policy to use, and view or modify email settings in the selected configuration.

   First, from the **Outgoing mail server (SMTP)** list, select the email 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**, then follow the instructions in the wizard.

4. If you want to view or modify the selected mail settings, click **Settings**, and use the **Mail Setup** tab. For more information, see Configuring email settings.
5. Click **Next**.
6. On the **Enforce Policy** page, you can specify objects to which this Policy Object is to be applied. To do so, click **Add**, then use the **Select Objects** dialog 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 email 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.

Creating the email configuration

You can use the instructions in see Scenario: Sending deprovisioning notification to create the email configuration. When prompted to specify a name for the new configuration, type `Deprovisioning Report Distribution`.

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.

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. In that dialog, type the administrator’s email 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 email configuration you have created in Step 1, and then click **Next** to display the **Enforce Policy** page.

On the **Enforce Policy** page, click **Add** 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.

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 initiated before or after every action that can happen in the directory. See the following table for a list of these handlers.

Table 15: 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 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

Name	Description
	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.
onPreDeprovision	Receives control upon a request to run 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 run the Deprovision operation. Enables the use of a script for customizing the behavior of the operation.
onPostDeprovision	Receives control after a request to run the Deprovision operation is completed. Enables a script to perform custom actions following the operation.
onPreUnDeprovision	Receives control upon a request to run 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 run the Undo Deprovisioning operation. Enables the use of a script for customizing the behavior of the operation.
onPostUnDeprovision	Receives control after a request to run the Undo Deprovisioning operation is completed. Enables a script to perform custom actions following the operation.
onPreUnDelete	Receives control upon a request to run the Undelete operation. Enables a script to perform custom actions prior to starting the operation.

Name	Description
onPostUnDelete	Receives control after a request to run 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.

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 run 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.

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 for the policy entry), and use the **IEDSRequestParameters** interface in a post-action event handler.

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 email recipient.

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. To do so, 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 by clicking the **properties** link next to the **Policy Object** label. This displays the **Properties** dialog for the Policy Object. For instructions on how to add, modify, or remove policies in the **Properties** dialog, see Adding policies to a Policy Object, Modifying policies in a Policy Object, and Removing policies from a Policy Object.
   - View or modify the properties of the object that violates the policy. To do so, click **Properties** 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). To do so, 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.

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, 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, select one of the acceptable values (**Production** or **Sales**) from the **Department** combo-box.

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 16: Policy options for users: Built-in Policy - User Default Deprovisioning

Policy	Options
User Account Deprovisioning	• Disable the user account. • Set the user 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.

Policy	Options
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 17: 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" 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, 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, 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.

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 email.

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 accordance 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 disabling user accounts.

Report section: User Account Deprovisioning

Table 18: 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.	Failed to change the user name.

Report item (Success)	Report item (Failure)
Original name: `<name>`. New name: `<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 19: Group Membership Removal

Report item (Success)	Report item (Failure)
In accordance 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.	Not applicable
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. • Security groups from which the user is not removed due to an error.
In accordance 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 accordance 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:	Not applicable

Report item (Success)	Report item (Failure)
• Distribution groups and mail-enabled security groups to which the user belongs.	
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 accordance 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 20: Exchange Mailbox Deprovisioning

Report item (Success)	Report item (Failure)
Mailbox deprovisioning is skipped because the user does not have an Exchange mailbox.	Not applicable

Report item (Success)	Report item (Failure)
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 manager of the user is provided with full access to the user mailbox. Manager name: `<name>`	Failed to provide the manager of the user 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.	Not applicable
Automatic replies turned on.	Failed to turn on automatic replies.

Report section: Home Folder Deprovisioning

Table 21: 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

Report item (Success)	Report item (Failure)
The user’s rights on the home folder are removed.	Failed to remove the user’s rights on the home folder.
The manager of the user is provided with read-only access to the user home folder. Manager name: `<name>`.	Failed to provide the manager of the user with read-only access to the user home folder. Manager name: `<name>`.
Not applicable	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 accordance with the policy, the user home folder will be deleted when the user account is deleted. Home folder name: `<name>`.	Not applicable
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 22: User Account Relocation

Report item (Success)	Report item (Failure)
In accordance with the policy, the user account is not moved from its original location: `<container-name>`.	Not applicable
The user account is moved to new location. Original location: `<container-name>`. New location: `<container-name>`.	Failed to move the user account to new location. Original location: `<container-name>`. Failed to move to this location: `<container-name>`.

Report section: User Account Permanent Deletion

Table 23: User Account Permanent Deletion

Report Item (Success)	Report Item (Failure)
In accordance with the 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 24: 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 accordance 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:	Failed to remove some members

Report Item (Success)	Report Item (Failure)
- List of the members removed from the group.	from the group. Details: - List of the members removed from the group. - List of the members that are not removed from the group due to an error.
In accordance 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 25: Group Object Relocation

Report Item (Success)	Report Item (Failure)
In accordance 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: `<container-name>`. New location: `<container-name>`.	Failed to move the group to new location. Original location: `<container-name>`. Failed to move to this location: `<container-name>`.

Report section: Group Object Permanent Deletion

Table 26: Group Object Permanent Deletion

Report Item (Success)	Report Item (Failure)
In accordance 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 27: Notification Distribution

Report Item (Success)	Report Item (Failure)
Deprovisioning notification will be sent to the listed recipients (not sent so far). List: • `<name-of-recipients>`	Not applicable
Deprovisioning notification was sent to the listed recipients. List: • `<name-of-recipients>`	Due to an error, deprovisioning notification was not sent to the listed recipients. List: • `<name-of-recipients>`

Report section: Report Distribution

Table 28: 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: • `<name-of-recipients>`	Not applicable
Deprovisioning report was sent to the listed recipients. List: • `<name-of-recipients>`	Due to an error, deprovisioning report was not sent to the listed recipients. List: • `<name-of-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 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, 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, 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, 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 email.

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 29: 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.	Failed to restore the user name.

Report item (Success)	Report item (Failure)
Old name: `<name>` Restored name: `<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 30: 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-enabled security groups is not restored.	N/A
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 31: 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 email forwarding is restored on the user mailbox.	Failed to restore the original configuration of the email 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 32: 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 33: Undo User Account Relocation

Report item (Success)	Report item (Failure)
No changes to undo.	N/A
The user account is moved to its original	Failed to move the user account to its original

Report item (Success)	Report item (Failure)
location. Former location: `<name of container>` Restored original location: `<name of container>`	location. Current location: `<name of container>` Failed to move to this location: `<name of container>`

Report section: Undo User Account Permanent Deletion

Table 34: Undo User Account Permanent Deletion

Report Item (Success)	Report Item (Failure)
No changes to undo.	N/A
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 35: 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.
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 36: Undo Group Object Relocation

Report Item (Success)	Report Item (Failure)
No changes to undo.	N/A
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 37: Undo Group Object Permanent Deletion

Report Item (Success)	Report Item (Failure)
No changes to undo.	N/A
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 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 to prevent deletion of non-empty Organizational Units.
5. Click **OK** to close the dialogs 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 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 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 downsampling 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 that appears, click the policy in the list, and then click **View/Edit**.
5. In the **Properties** dialog 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 predefined types that are installed with Active Roles. By default, the list of policy types in the Active Roles Console contains only the predefined 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 predefined 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.

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.

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 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 38: 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.
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	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

To change	Do this	Commentary
	appropriate function from the **Function to declare parameters** list.	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**.

6

Using rule-based and role-based tools for granular administration

Although you can cover many administration scenarios with the exclusive use of either rule-based Managed Units (MUs) or role-based Access Templates (ATs), combining MUs and ATs in your administration workflow provides additional flexibility to achieve the highest level of granularity.

This is useful if you want to ensure that certain management resources (for example, departmental administrators or helpdesk agents) can access and administer only a specific set of resources (for example, the resources of a specific department or geography, or specific resource types within a department only).

The following example use cases demonstrate how to configure and delegate such high-granularity permissions to:

- Deny access to certain Azure AD resources.
- Allow access to certain Azure AD resources only.

> NOTE: Active Roles Console supports managing the following Azure AD resources with Managed Units:
>
> - Azure users
> - Azure guest users
> - Azure contacts (if the MU is configured with an **Include Explicitly** rule)
> - Microsoft 365 (M365) groups
> - Azure distribution groups (if the MU is configured with an **Include Explicitly** rule)
> - Azure security groups
>
> However, Managed Units do not support any Azure mailbox types and dynamic distribution groups.

⚠ **CAUTION: Hazard of data loss!**

The combined AT and MU configurations described in these example scenarios are meant to delegate granular access for roles like departmental administrators or helpdesk agents.

Do not delegate such granular permissions to:

- **Administrators with an Active Roles Administration Service account.**
- **Super administrators.**
- **Any other high-level administration personnel.**

Otherwise, you can lose access to the Azure AD resources in the scope of the configured granular access.

Example: Configuring high granularity by hiding a specific Azure group

This scenario describes how to use the Managed Units (MUs) and Access Templates (ATs) of the Active Roles Console together to configure Azure group administration permissions with high granularity. In this example, the MUs and ATs are used to deny the read access of a group of helpdesk users to a specific Azure Microsoft 365 (M365) group. You can achieve this by:

1. Configuring an MU containing the M365 group that the helpdesk users should not access. For more information on this procedure, see Configuring a Managed Unit to hide specific Microsoft 365 groups.
2. Configuring an AT to deny access to that M365 group for the helpdesk users. For more information on this procedure, see Configuring an Access Template to hide Microsoft 365 Groups.

Prerequisites

To configure this example scenario, your organization must meet the following requirements:

- To create MUs and ATs in the Active Roles Console, you must use an Active Roles Administration Service account. For more information, see *Configuring the Administration Service account* in the *Active Roles Quick Start Guide*.
- The organization must already have one or more Azure tenants configured and consented for use with Active Roles. For more information, see Configuring a new Azure tenant and consenting Active Roles as an Azure application.
- To ensure that the **Helpdesk** group receiving the granular read permission can still read other Azure groups, they must have the built-in **Azure Microsoft365 Groups**

- Read All Attribute AT (or a custom AT based on this built-in AT) applied to them, with the affected **Object** being the Azure tenant of the managed Azure AD resources. For more information on how to apply an AT, see Applying Access Templates.

- The users receiving the configured permissions must be on-premises or hybrid Active Directory users. You cannot delegate the configured granular permission to cloud-only Azure users.

Configuring a Managed Unit to hide specific Microsoft 365 groups

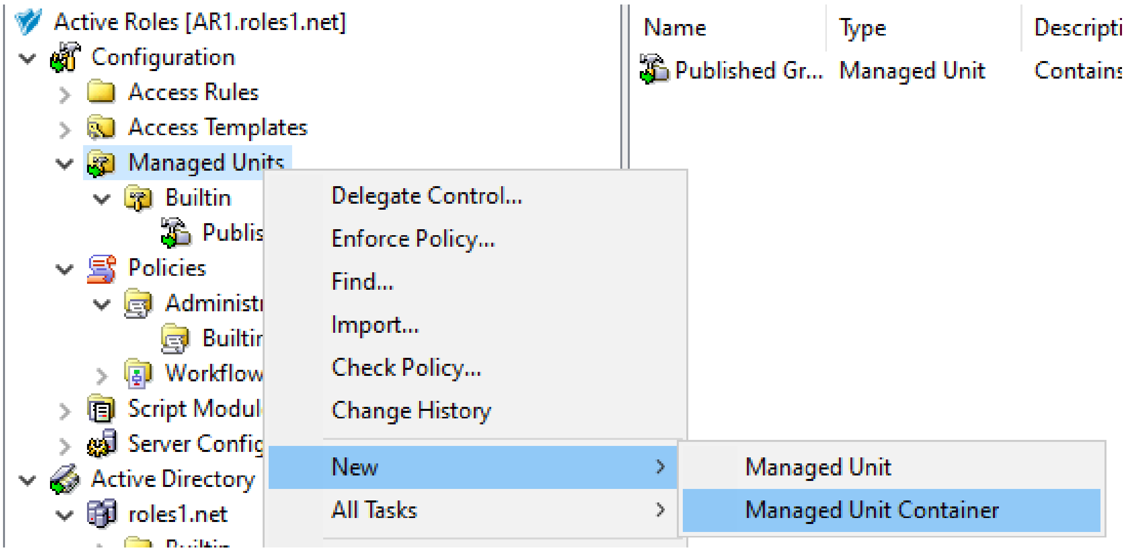
To set up a highly-granular Microsoft 365 (M365) group access logic, first you must configure a Managed Unit (MU) that will contain the M365 group that cannot be read by the affected helpdesk users.

In this example, the MU is configured to explicitly include the **Marketing** M365 group of an example Azure tenant. For more information on the available membership rule options for MUs, see Creating a Managed Unit.

To configure a Managed Unit to hide a specific Microsoft 365 group

1. In the Active Roles Console, on the **Console tree**, navigate to **Configuration** > **Managed Units**.
2. To create a new container for the configured MU, right-click on the **Managed Units** node, then click **New** > **Managed Unit Container**.

Figure 79: Active Roles Console – Launching the Managed Unit Container dialog



3. In the **Managed Unit Container** dialog, specify a **Name**, and optionally, a **Description** for the new MU container.

   This example uses the following container settings:

   - **Name**: `Denied-Azure-Resources`
   - **Description**: `Managed Units for the granular denial of Azure resources.`

4. To create the new container, click **Next** then **Finish**.
5. To start configuring the new MU, right-click the newly-created **Denied-Azure-Resources** container, then click **New** > **Managed Unit**.
6. In the **New Object - Managed Unit** dialog, specify a **Name**, and optionally, a **Description** for the new MU.

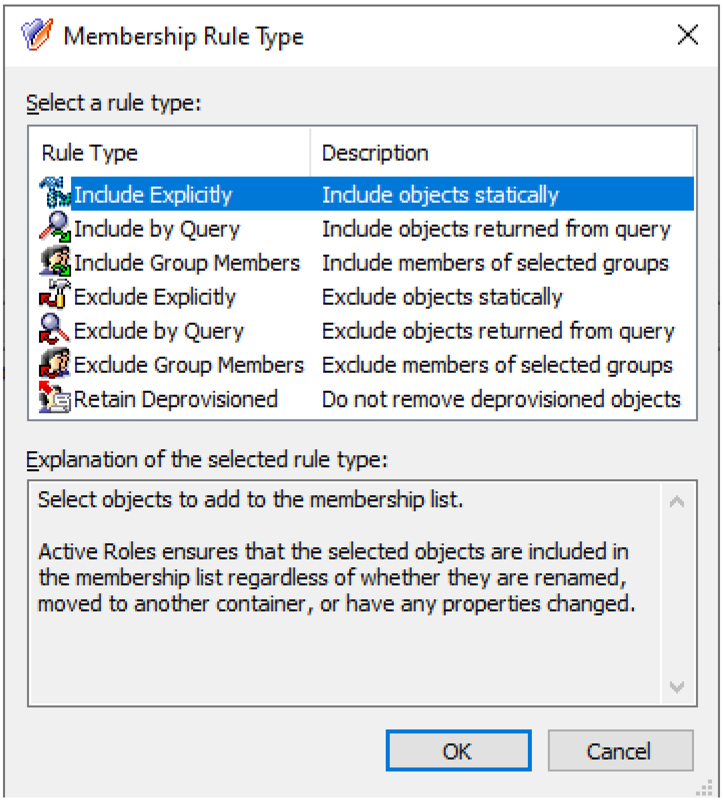
   This example uses the following MU settings:

   - **Name**: `Denied-M365-Groups`
   - **Description**: `Managed Unit for the granular denial of M365 groups.`

   To continue, click **Next**.

7. To specify a new membership rule for the MU, in the **Membership rule** step, click **Add**.
8. In the **Membership Rule Type** dialog, select the rule type used to populate the MU. This example uses the **Include Explicitly** rule type. Select it, then click **Next**.

Figure 80: New Managed Unit – Selecting the Include Explicitly membership rule type



Membership Rule Type

Select a rule type:

Rule Type	Description
Include Explicitly	Include objects statically
Include by Query	Include objects returned from query
Include Group Members	Include members of selected groups
Exclude Explicitly	Exclude objects statically
Exclude by Query	Exclude objects returned from query
Exclude Group Members	Exclude members of selected groups
Retain Deprovisioned	Do not remove deprovisioned objects

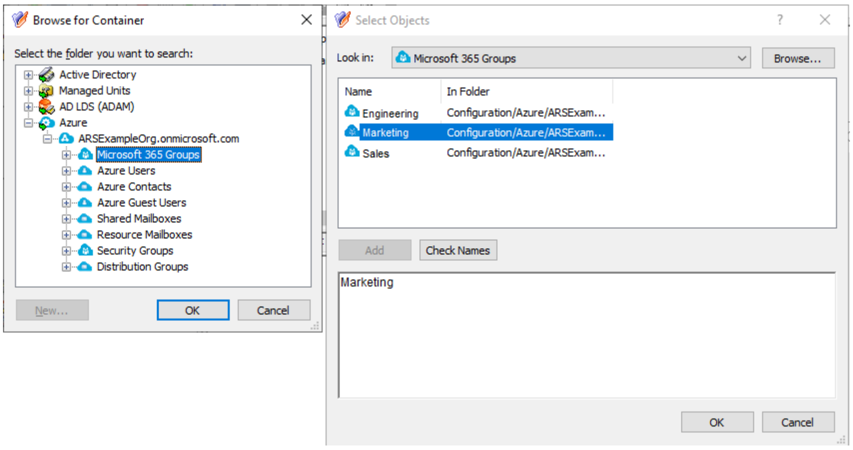
Explanation of the selected rule type:

Select objects to add to the membership list.

Active Roles ensures that the selected objects are included in the membership list regardless of whether they are renamed, moved to another container, or have any properties changed.

9. In the **Select Objects** dialog, select the M365 group whose members you want to add to the MU.

Figure 81: New Managed Unit – Adding an M365 Group to an MU



Browse for Container

Select the folder you want to search:

- Active Directory
- Managed Units
- AD LDS (ADAM)
- Azure
  - ARSExampleOrg.onmicrosoft.com
    - Microsoft 365 Groups
    - Azure Users
    - Azure Contacts
    - Azure Guest Users
    - Shared Mailboxes
    - Resource Mailboxes
    - Security Groups
    - Distribution Groups

Select Objects

Look in: Microsoft 365 Groups

Name	In Folder
Engineering	Configuration/Azure/ARSExam...
Marketing	Configuration/Azure/ARSExam...
Sales	Configuration/Azure/ARSExam...

Marketing

To do so:

a. In the **Select Objects** dialog, click **Browse**.

b. In the **Browse for Container** dialog, expand the **Azure** > **<azure-tenant-name>** node (in this example, the Azure tenant is named **ARSExampleOrg.onmicrosoft.com**).

c. Select the **Microsoft 365 Groups** node, and click **OK**. The M365 groups existing in the Azure tenant will appear in the **Select Objects** dialog.

d. In the **Select Objects** dialog, select the M365 group you want to add to the MU (in this example, the **Marketing** group).

e. To apply the selection, click **Add** and **OK**.

10. To finish creating the MU, click **Next**, then **Next** again in the **Object Security** / **Policy Object** step, and finally **Finish**.

11. To verify that the MU is populated correctly, select the newly-created MU in the Console Tree. The **Marketing** M365 group must appear in the Active Roles Console.

Configuring an Access Template to hide Microsoft 365 Groups

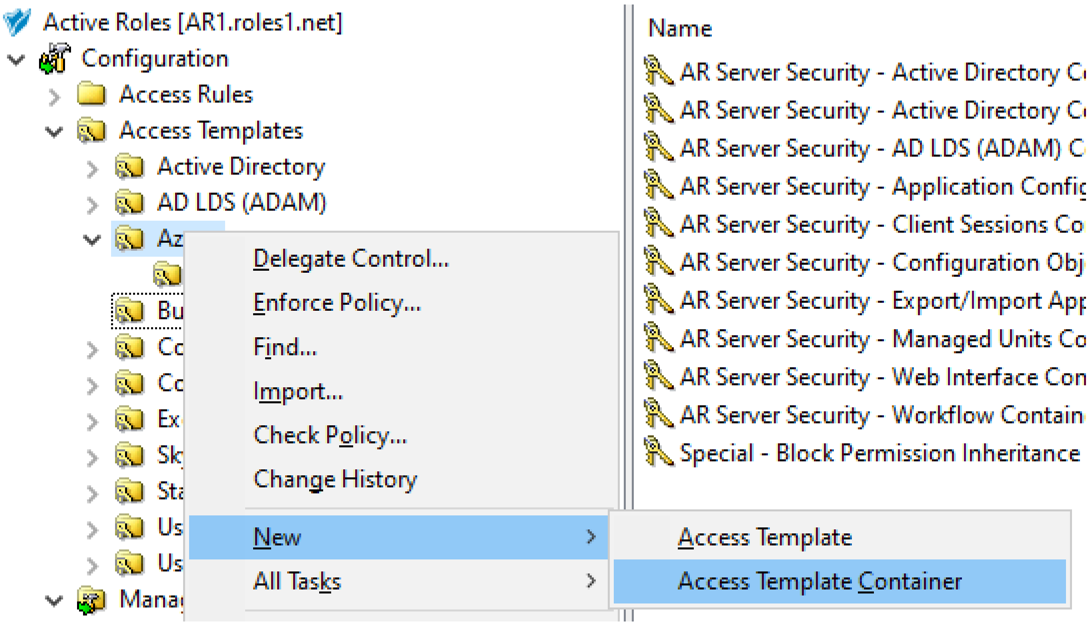
Once you set up the Managed Unit (MU) as described in Configuring a Managed Unit to hide specific Microsoft 365 groups, you must create an Access Template (AT) that denies the read access of the affected helpdesk users to the Microsoft 365 (M365) group included in that MU.

To create the AT, perform the following steps. For more information on creating ATs in general, see Creating an Access Template.

To deny access to the Microsoft 365 group of a Managed Unit with an Access Template

1. In the Active Roles Console, in the Active Directory (AD) tree, navigate to **Configuration** > **Access Templates**.
2. Create a new container where you will store the AT. In this example, the container is created in the **Azure** sub-container of the **Access Templates** node. Right-click **Access Templates** > **Azure**, then click **New** > **Access Template Container**.

Figure 82: Active Roles Console – Launching the Access Templates Container dialog


Active Roles [AR1.roles1.net]
Configuration
Access Rules
Access Templates
Active Directory
AD LDS (ADAM)
Az
Bu
Cc
Cc
Ex
Sk
Sta
Us
Us
Mana
Delegate Control...
Enforce Policy...
Find...
Import...
Check Policy...
Change History
New
All Tasks
Access Template
Access Template Container
Name
AR Server Security - Active Directory C
AR Server Security - Active Directory C
AR Server Security - AD LDS (ADAM) C
AR Server Security - Application Confic
AR Server Security - Client Sessions Co
AR Server Security - Configuration Obj
AR Server Security - Export/Import App
AR Server Security - Managed Units Co
AR Server Security - Web Interface Con
AR Server Security - Workflow Contain
Special - Block Permission Inheritance

3. In the **Access Templates Container** dialog, specify a **Name**, and optionally, a **Description** for the new AT container.

   This example uses the following container settings:

   - **Name**: `Denied-Azure-Resources`
   - **Description**: `Access Templates for the granular access of Azure resources.`

4. To create the new container, click **Next** then **Finish**.

5. To start configuring the new AT, right-click the **Denied-Azure-Resources** container, then click **New** > **Access Template**.

6. In the **New Object - Access Template** dialog, specify a **Name**, and optionally, a **Description** for the new AT.

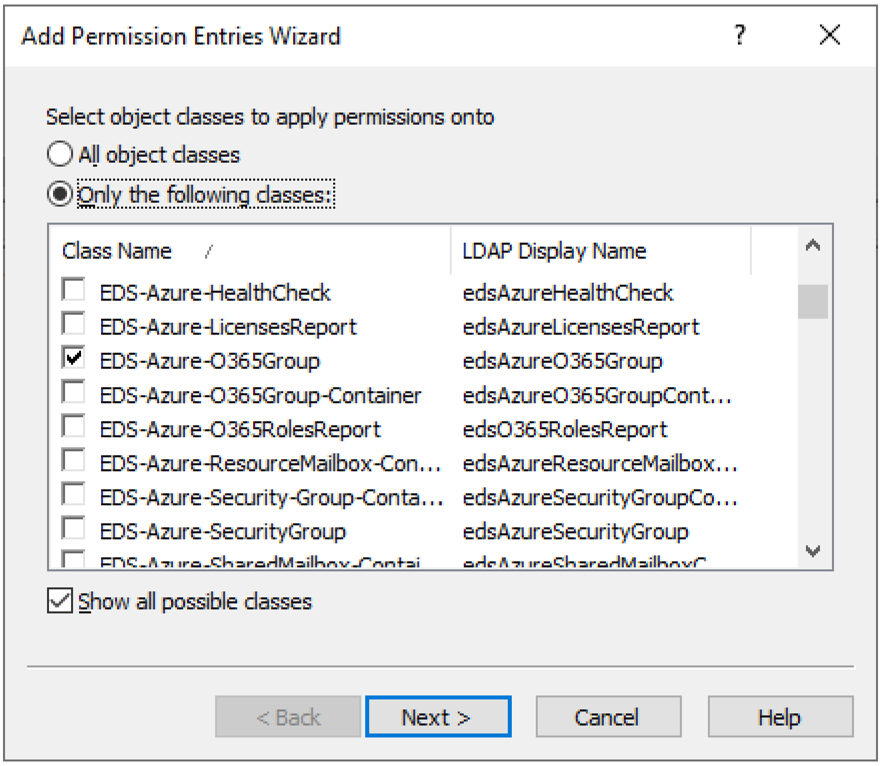
   This example uses the following AT settings:

   - **Name**: `DenyM365Groups`
   - **Description**: `AT to deny access to the specified M365 groups.`

   To continue, click **Next**.

7. In the **Access Template permission entries** step, click **Add**. Then, in the **Add Permission Entries Wizard**, select **Only the following classes**, and select **EDS-Azure-O365Group** from the list. To continue, click **Next**.

Figure 83: New Access Template – Selecting the M365 group object class to deny general access to them



Add Permission Entries Wizard

Select object classes to apply permissions onto

○ All object classes

● Only the following classes:

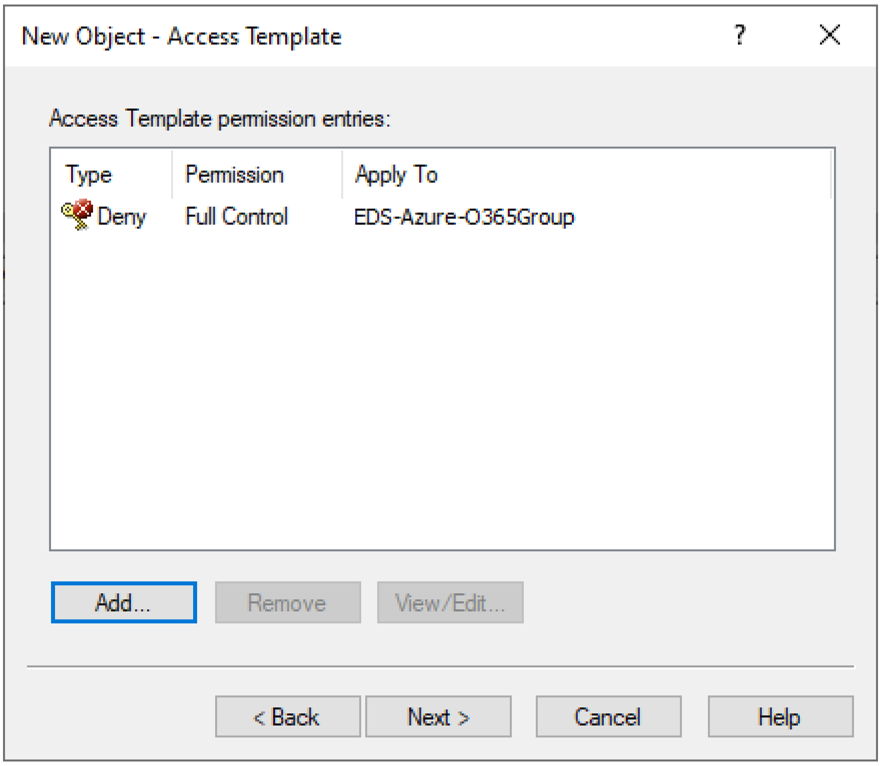
Class Name	LDAP Display Name
□ EDS-Azure-HealthCheck	edsAzureHealthCheck
□ EDS-Azure-LicensesReport	edsAzureLicensesReport
☑ EDS-Azure-O365Group	edsAzureO365Group
□ EDS-Azure-O365Group-Container	edsAzureO365GroupCont...
□ EDS-Azure-O365RolesReport	edsO365RolesReport
□ EDS-Azure-ResourceMailbox-Con...	edsAzureResourceMailbox...
□ EDS-Azure-Security-Group-Conta...	edsAzureSecurityGroupCo...
□ EDS-Azure-SecurityGroup	edsAzureSecurityGroup
□ EDS-Azure-SharedMailbox-Contai	edsAzureSharedMailboxC

☑ Show all possible classes

TIP: If you cannot find the class in the list, select **Show all possible classes**.

8. In the **Select permission category** step, select **Deny permission**, then click **Finish**. The permission then appears in the **Access Template permission entries** step of the **New Object - Access Template** dialog.

Figure 84: New Access Template – Verifying the deny permission



New Object - Access Template

Access Template permission entries:

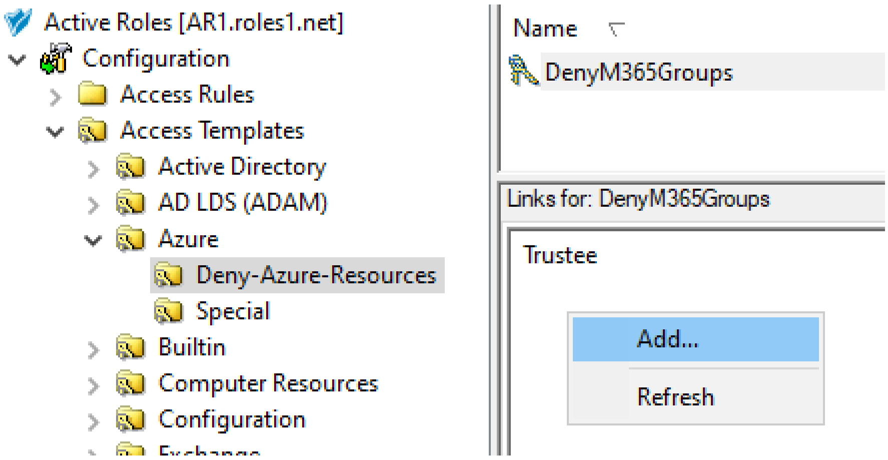
Type	Permission	Apply To
Deny	Full Control	EDS-Azure-O365Group

9. To finish creating the AT, click **Next**, then **Finish**.

10. Assign the newly-created AT to the helpdesk users whose access you want to restrict. To do so, check if the **Advanced Details Pane** option of the Active Roles Console is selected. If not, open **View**, and select **Advanced Details Pane**.

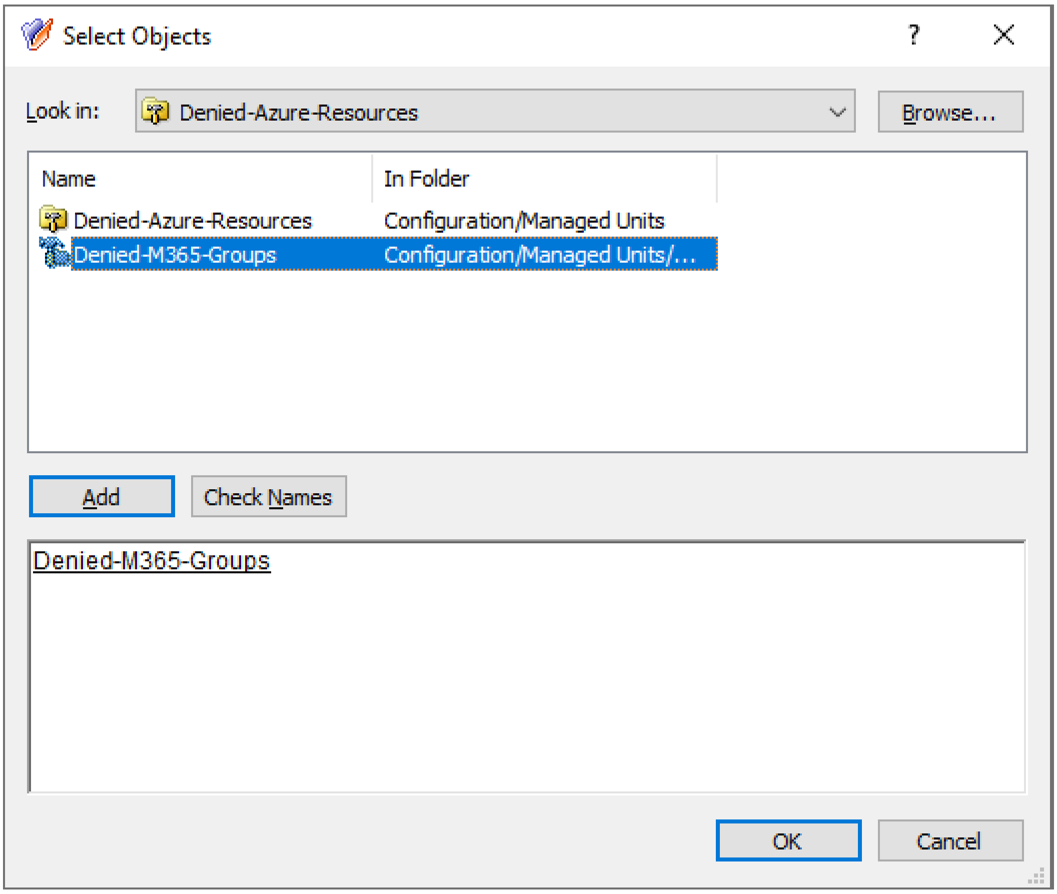
11. To start the **Delegation of Control Wizard**, select the newly-created **DenyM365Groups** AT, then right-click in the Advanced Details Pane, and click **Add**.

Figure 85: Active Roles Console – Launching the Delegation of Control Wizard from the Advanced Details Pane



12. In the **Objects** step of the wizard, click **Add**. Then, in the **Select Objects** dialog, **Browse** for the **Denied-Azure-Resources** Managed Unit Container that you created in Configuring a Managed Unit to hide specific Microsoft 365 groups. To add the **Denied-M365-Groups** MU to the list of managed objects, click **Add**, then click **OK**.

Figure 86: Delegation of Control Wizard – Selecting the Managed Unit as an administered object



Look in: Denied-Azure-Resources

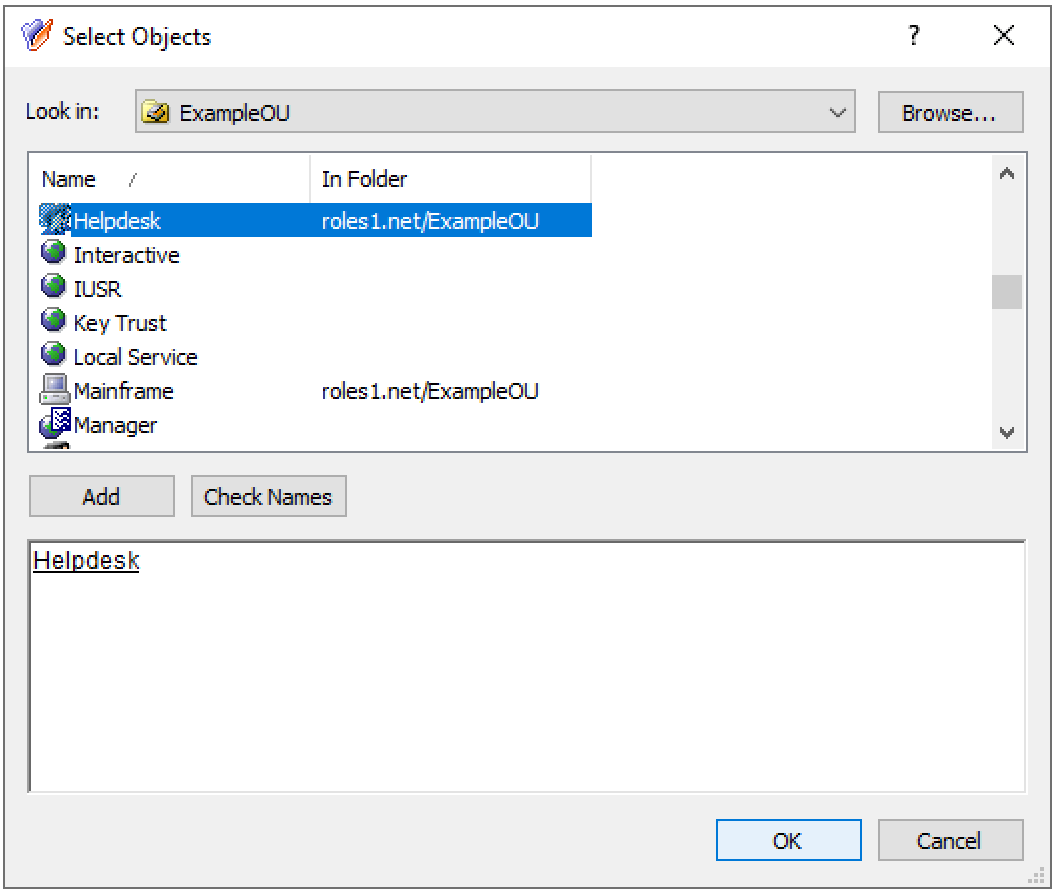
Name	In Folder
Denied-Azure-Resources	Configuration/Managed Units
Denied-M365-Groups	Configuration/Managed Units/...

Denied-M365-Groups

To continue, in the **Objects** step, click **Next**.

13. In the **Users or Groups** step, click **Add**, then select the users to which you want to delegate the permission. In this example, the AT is delegated to the **Helpdesk** group of an example Organizational Unit (OU). To add the group, click **Add**, then click **OK**.

Figure 87: Delegation of Control Wizard – Selecting the Helpdesk group as Trustee



To continue, in the **Users or Groups** step, click **Next**.

14. In the **Inheritance Options** step, make sure that the **This directory object** and **Child objects of this directory object** settings are selected. To continue, click **Next**.
15. In the **Permissions Propagation** step, leave the **Propagate permissions to Active Directory** setting in its default state. To continue, click **Next**.
16. To complete the wizard, click **Finish**.

Enabling or disabling the granular access to Microsoft 365 Groups

Once you configured the Managed Unit (MU) of the Microsoft 365 (M365) group, and set up the Access Template (AT) to deny access to that group, the **Helpdesk** group to which the AT is assigned can no longer see the M365 group included in the MU. Instead:

- If they expand the **Microsoft 365 Groups** node of the Azure tenant on the Active Roles Web Interface, the M365 group included in the MU will not be visible to them.
- If they open the **Azure Member Of** page of any Azure user or Azure guest user who are also members of the affected M365 group, the page will not list the M365 group included in the MU among the group membership of the users.

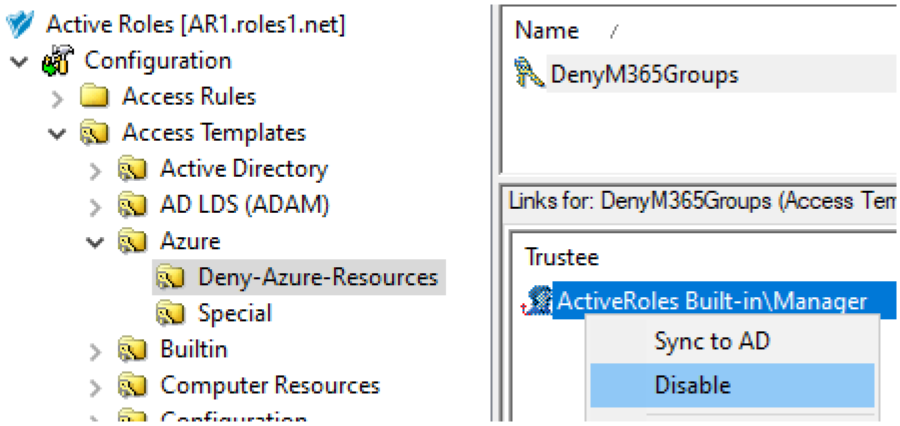
This behavior is dynamic: adding additional M365 groups into the MU in the Active Roles Console will result in those M365 groups also disappearing in the Active Roles Web Interface for the affected helpdesk users once the changes of the Console are synchronized to the Web Interface. Likewise, removing an M365 group from the MU will result in that M365 group appearing again for the affected helpdesk users in the Web Interface.

You can easily enable or disable the configured granular access later for the affected helpdesk users by enabling or disabling the AT.

To enable or disable the configured granular access to Microsoft 365 groups

1. In the Active Roles Console, on the Console Tree, navigate to **Configuration** > **Access Templates** > **Denied Azure Resources**.
2. Select the **DenyM365Groups** AT.
3. In the Advanced Details Pane, right-click the configured link, and click **Disable**.

   Figure 88: Active Roles Console – Disabling the configured Access Template

   TIP: If the Advanced Details Pane does not appear for you, click **View** > **Advanced Details Pane**.

   Once the AT is disabled, the M365 group included in the associated **Denied-M365-Groups** MU will appear in the Web Interface for the users to which the AT is assigned.
4. (Optional) To re-enable the AT, right-click the configured link again, and click **Enable**.

Example: Configuring high granularity by hiding specific Azure users

This scenario describes how to use the Managed Units (MUs) and Access Templates (ATs) of the Active Roles Console together to configure Azure user administration permissions with high granularity. In this example, the MUs and ATs are used to deny the read access of a group of helpdesk users to Azure users reporting to a specific manager. You can achieve this by:

1. Configuring an MU containing all the Azure users that the helpdesk users should not access. For more information on this procedure, see Configuring a Managed Unit to hide specific Azure users.
2. Configuring an AT to deny access to those Azure users for the helpdesk users. For more information on this procedure, see Configuring an Access Template to hide Azure users.

Prerequisites

To configure this example scenario, your organization must meet the following requirements:

- To create MUs and ATs in the Active Roles Console, you must use an Active Roles Administration Service account. For more information, see *Configuring the Administration Service account* in the *Active Roles Quick Start Guide*.
- The organization must already have one or more Azure tenants configured and consented for use with Active Roles. For more information, see Configuring a new Azure tenant and consenting Active Roles as an Azure application.
- To ensure that the **Helpdesk** group receiving the granular read permission can still read other Azure users, the group must have the built-in **Azure Cloud User - Read All Attributes** AT (or a custom AT based on this built-in AT) applied to them, with the affected **Object** being the Azure tenant of the managed Azure AD resources. For more information on how to apply an AT, see Applying Access Templates.
- The users receiving the configured permissions must be on-premises or hybrid Active Directory users. You cannot delegate the configured granular permission to cloud-only Azure users.

Configuring a Managed Unit to hide specific Azure users

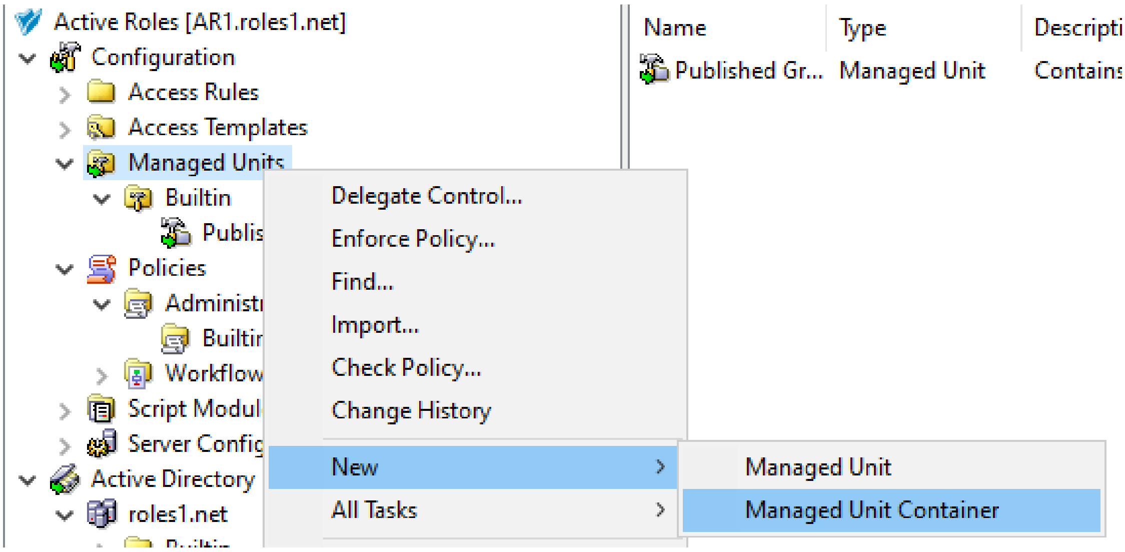
To set up a highly-granular Azure user access logic, first you must configure a Managed Unit (MU) that will contain the Azure users that cannot be read by the affected helpdesk users.

In this example, the membership of the MU is configured via a query, specifying that only Azure users reporting to a specific manager (in this example, **Sam Smith**) are included in the MU. For more information on the available membership rule options for MUs, see Creating a Managed Unit.

To configure a Managed Unit to hide specific Azure users

1. In the Active Roles Console, on the **Console tree**, navigate to **Configuration** > **Managed Units**.
2. To create a new container for the configured MU, right-click on the **Managed Units** node, then click **New** > **Managed Unit Container**.

   Figure 89: Active Roles Console – Launching the Managed Unit Container dialog

3. In the **Managed Unit Container** dialog, specify a **Name**, and optionally, a **Description** for the new MU container.

   This example uses the following container settings:

   - **Name**: `Denied-Azure-Resources`
   - **Description**: `Managed Units for the granular denial of Azure resources.`
4. To create the new container, click **Next** then **Finish**.
5. To start configuring the new MU, right-click the newly-created **Denied-Azure-Resources** container, then click **New** > **Managed Unit**.
6. In the **New Object - Managed Unit** dialog, specify a **Name**, and optionally, a **Description** for the new MU.

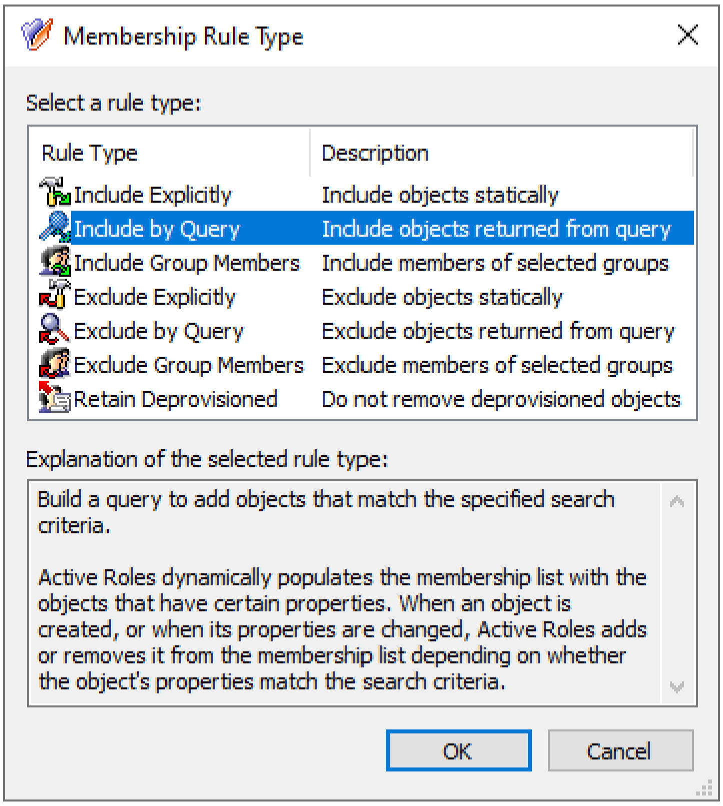
   This example uses the following MU settings:

   - **Name**: `Denied-Azure-Users`
   - **Description**: `Managed Unit for the granular denial of Azure users.`

   To continue, click **Next**.

7. To specify a new membership rule for the MU, in the **Membership rule** step, click **Add**.
8. In the **Membership Rule Type** dialog, select the rule type used to populate the MU. This example uses the **Include by Query** rule type. Select it, then click **Next**.

Figure 90: New Managed Unit – Selecting the Include by Query membership rule type



Membership Rule Type

Select a rule type:

Rule Type	Description
Include Explicitly	Include objects statically
Include by Query	Include objects returned from query
Include Group Members	Include members of selected groups
Exclude Explicitly	Exclude objects statically
Exclude by Query	Exclude objects returned from query
Exclude Group Members	Exclude members of selected groups
Retain Deprovisioned	Do not remove deprovisioned objects

Explanation of the selected rule type:

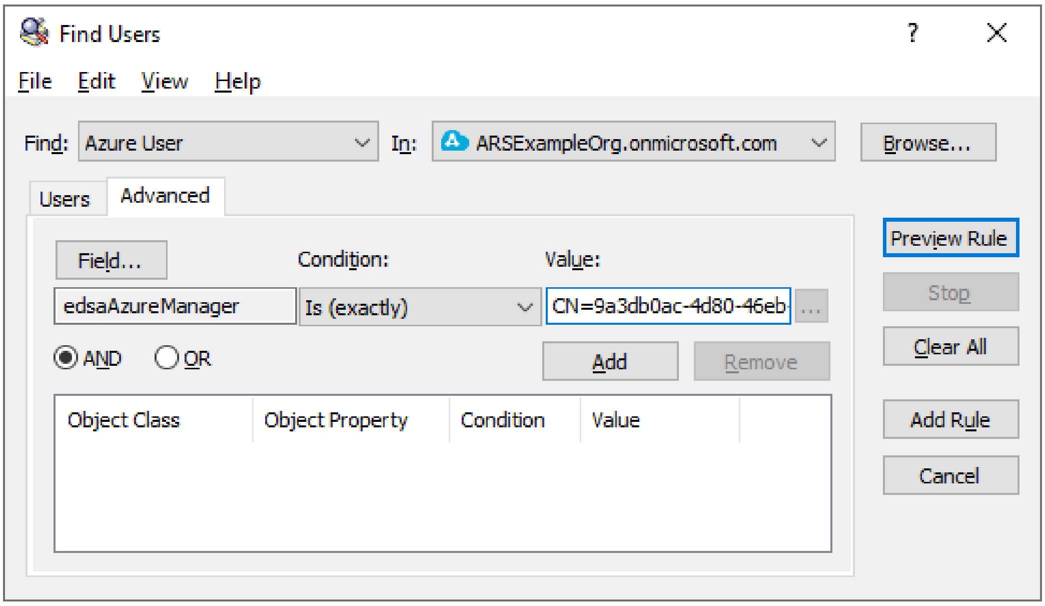
Build a query to add objects that match the specified search criteria.

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 object's properties match the search criteria.

9. In the **Create Membership Rule** dialog, configure the query by which Active Roles will dynamically populate the MU with Azure users. This example uses the following settings:
   - In the **Find** drop-down list, select **Azure User**.
   - Under the **Advanced** tab, click **Field**, and select the **edsaAzureManager** attribute.

     TIP: If you cannot find the attribute in the list, select **Show all possible properties**.
   - In **Condition**, select **Is (exactly)**.
   - In **Value**, specify the manager Azure user (in this example, `Sam Smith`) by clicking the ... (**Browse**) button and selecting it from the **Azure Users** container. Once selected, the distinguished name of the Azure user appears in the **Value** text box.

Figure 91: New Managed Unit – Configuring the Include by Query membership rule type



Find Users

Find: Azure User

In: ARSExampleOrg.onmicrosoft.com

Field...: edsaAzureManager

Condition: Is (exactly)

Value: CN=9a3db0ac-4d80-46eb

● AND ○ OR

10. To verify that the configured rule works properly, click **Preview Rule**. If Active Roles asks if you want to add the current criteria to your search, click **OK**. Active Roles then adds and immediately tests the membership rule for the MU, and the users reporting to **Sam Smith** must appear in the list. If the results look correct, click **OK**.

11. To finish creating the MU, click **Next**, then **Next** again in the **Object Security / Policy Object** step, and finally **Finish**.

12. To verify that the MU is populated correctly, select the newly-created MU in the Console Tree. The Azure users reporting to **Sam Smith** must appear in the Active Roles Console.

Configuring an Access Template to hide Azure users

Once you set up the Managed Unit (MU) as described in Configuring a Managed Unit to hide specific Azure users, you must create an Access Template (AT) that denies the read access of the affected helpdesk users to the Azure users included in that MU.

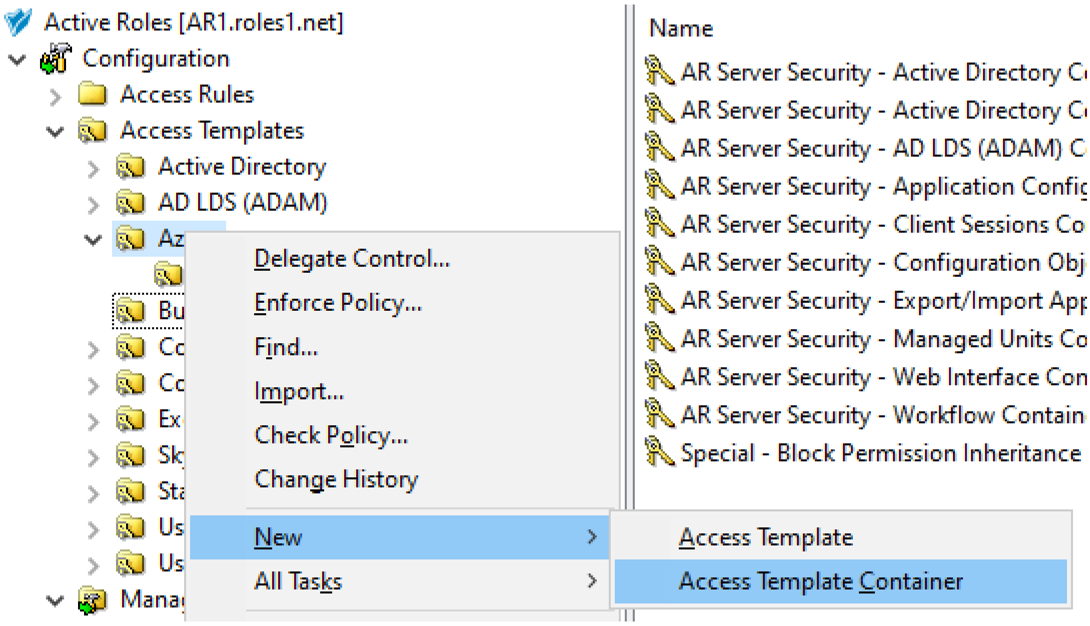
To create the AT, perform the following steps. For more information on creating ATs in general, see Creating an Access Template.

To deny access to the Azure users of a Managed Unit with an Access Template

1. In the Active Roles Console, in the Active Directory (AD) tree, navigate to **Configuration** > **Access Templates**.

2. Create a new container where you will store the AT. In this example, the container is created in the **Azure** sub-container of the **Access Templates** node. Right-click **Access Templates** > **Azure**, then click **New** > **Access Template Container**.

   Figure 92: Active Roles Console – Launching the Access Templates Container dialog

   Active Roles [AR1.roles1.net]
Configuration
Access Rules
Access Templates
Active Directory
AD LDS (ADAM)
Az
Bu
Co
Co
Ex
Sk
Sta
Us
Us
Mana
Delegate Control...
Enforce Policy...
Find...
Import...
Check Policy...
Change History
New
All Tasks
Access Template
Access Template Container
Name
AR Server Security - Active Directory C
AR Server Security - Active Directory C
AR Server Security - AD LDS (ADAM) C
AR Server Security - Application Config
AR Server Security - Client Sessions Co
AR Server Security - Configuration Obj
AR Server Security - Export/Import App
AR Server Security - Managed Units Co
AR Server Security - Web Interface Con
AR Server Security - Workflow Contain
Special - Block Permission Inheritance

3. In the **Access Templates Container** dialog, specify a **Name**, and optionally, a **Description** for the new AT container.

   - **Name**: `Denied-Azure-Resources`
   - **Description**: `Access Templates for the granular access of Azure resources.`

4. To create the new container, click **Next** then **Finish**.

5. To start configuring the new AT, right-click the **Denied-Azure-Resources** container, then click **New** > **Access Template**.

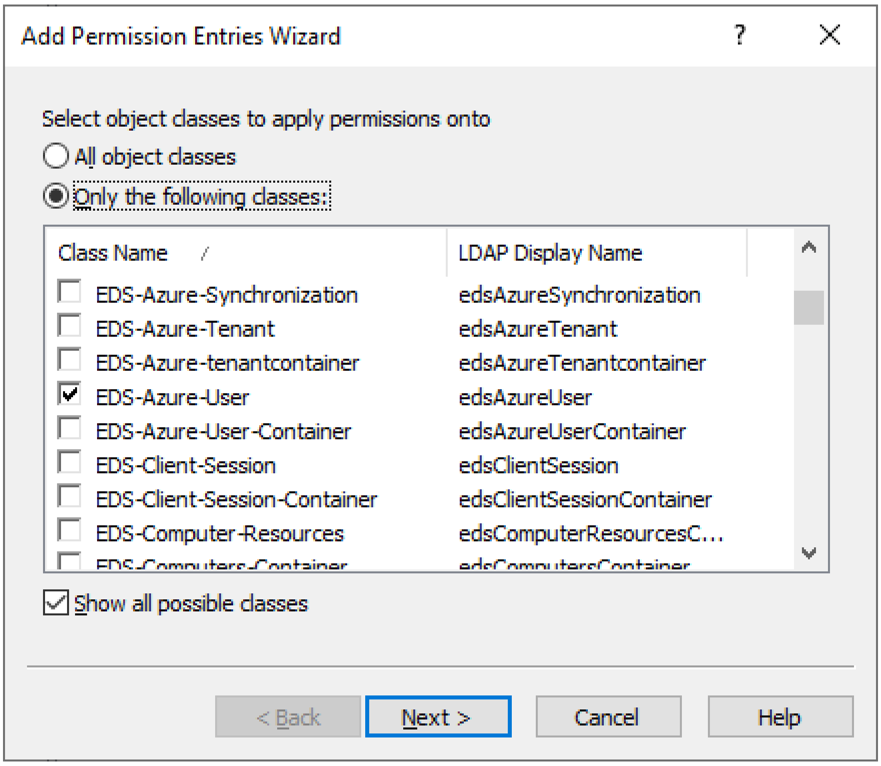
6. In the **New Object - Access Template** dialog, specify a **Name**, and optionally, a **Description** for the new AT.

   - **Name**: `DenyAzureUsers`
   - **Description**: `AT to deny access to the specified Azure users.`

   To continue, click **Next**.

7. In the **Access Template permission entries** step, click **Add**. Then, in the **Add Permission Entries Wizard**, select **Only the following classes**, and select **EDS-Azure-User** from the list. To continue, click **Next**.

Figure 93: New Access Template – Selecting the Azure user object class to deny general access to them



Add Permission Entries Wizard

Select object classes to apply permissions onto

○ All object classes

● Only the following classes:

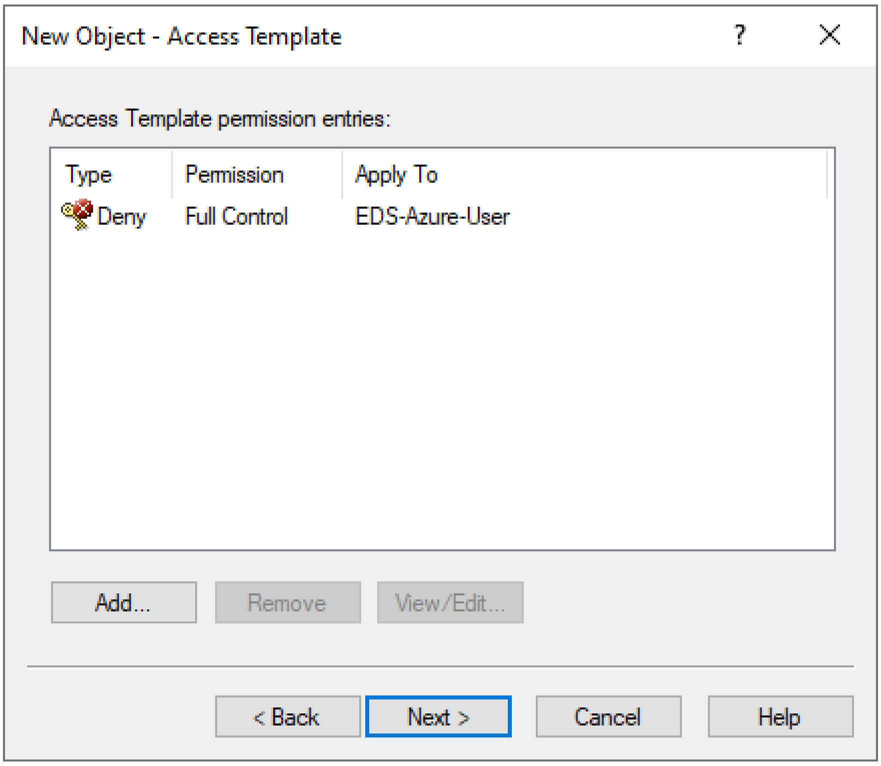
Class Name	LDAP Display Name
□ EDS-Azure-Synchronization	edsAzureSynchronization
□ EDS-Azure-Tenant	edsAzureTenant
□ EDS-Azure-tenantcontainer	edsAzureTenantcontainer
☑ EDS-Azure-User	edsAzureUser
□ EDS-Azure-User-Container	edsAzureUserContainer
□ EDS-Client-Session	edsClientSession
□ EDS-Client-Session-Container	edsClientSessionContainer
□ EDS-Computer-Resources	edsComputerResourcesC...
□ EDS-Computers-Container	edsComputersContainer

☑ Show all possible classes

| TIP: If you cannot find the class in the list, select **Show all possible classes**.

8. In the **Select permission category** step, select **Deny permission**, then click **Finish**. The permission then appears in the **Access Template permission entries** step of the **New Object - Access Template** dialog.

Figure 94: New Access Template – Verifying the deny permission



New Object - Access Template

Access Template permission entries:

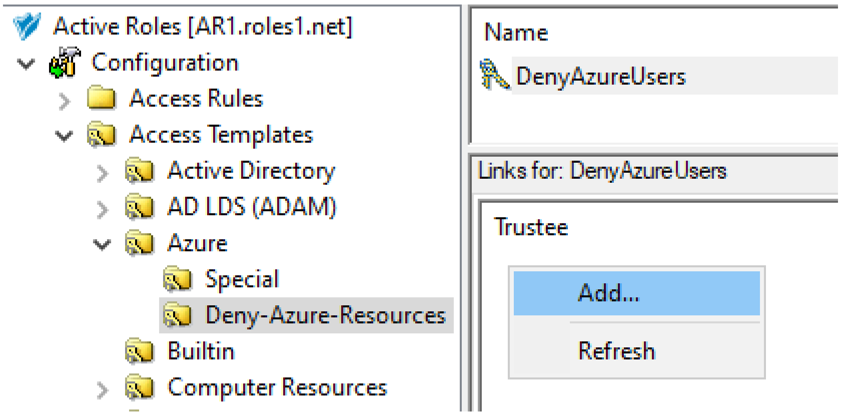
Type	Permission	Apply To
Deny	Full Control	EDS-Azure-User

9. To finish creating the AT, click **Next**, then **Finish**.

10. Assign the newly-created AT to the helpdesk users whose access you want to restrict. To do so, check if the **Advanced Details Pane** option of the Active Roles Console is selected. If not, open **View**, and select **Advanced Details Pane**.

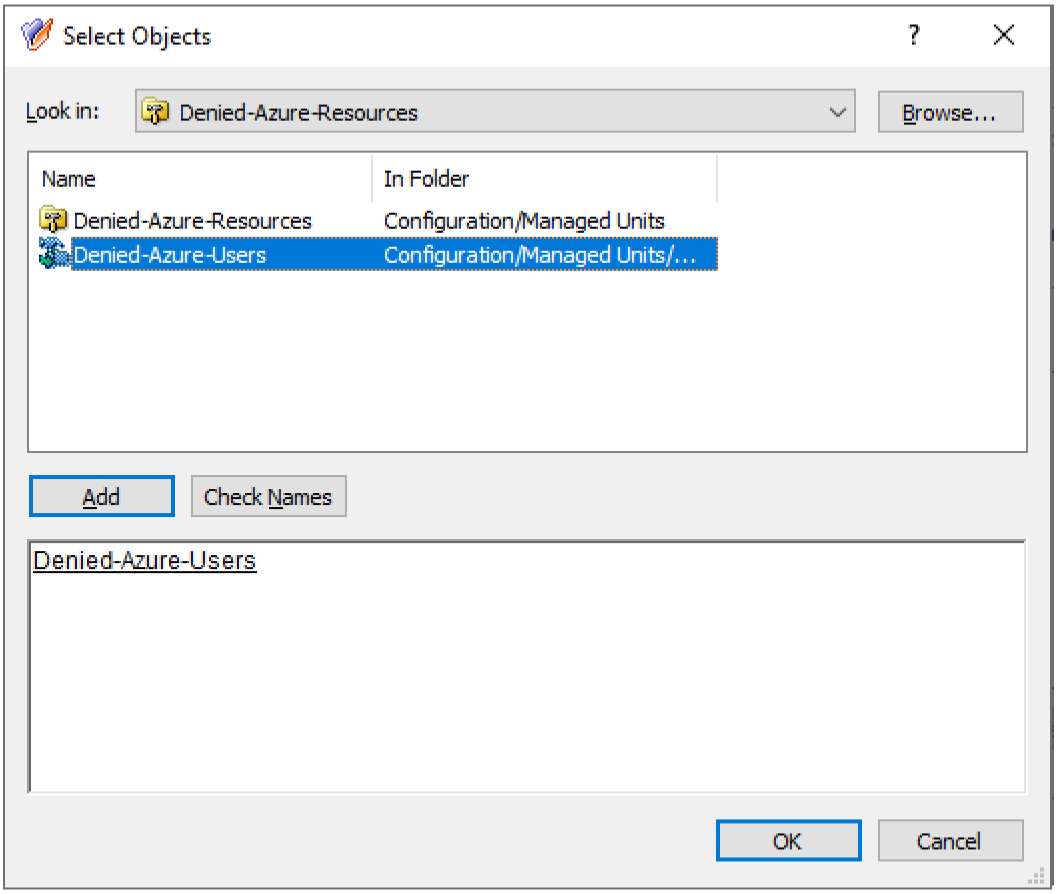
11. To start the **Delegation of Control Wizard**, select the newly-created **DenyAzureUsers** AT, then right-click in the Advanced Details Pane, and click **Add**.

Figure 95: Active Roles Console – Launching the Delegation of Control Wizard from the Advanced Details Pane



12. In the **Objects** step of the wizard, click **Add**. Then, in the **Select Objects** dialog, **Browse** for the **Denied-Azure-Resources** Managed Unit Container that you created in Configuring a Managed Unit to hide specific Azure users, and select the **Denied-Azure-Users** MU as the object managed by the AT. To add the **Denied-Azure-Users** MU to the list of managed objects, click **Add**, then click **OK**.

    Figure 96: Delegation of Control Wizard – Selecting the Managed Unit as an administered object

    Look in: Denied-Azure-Resources

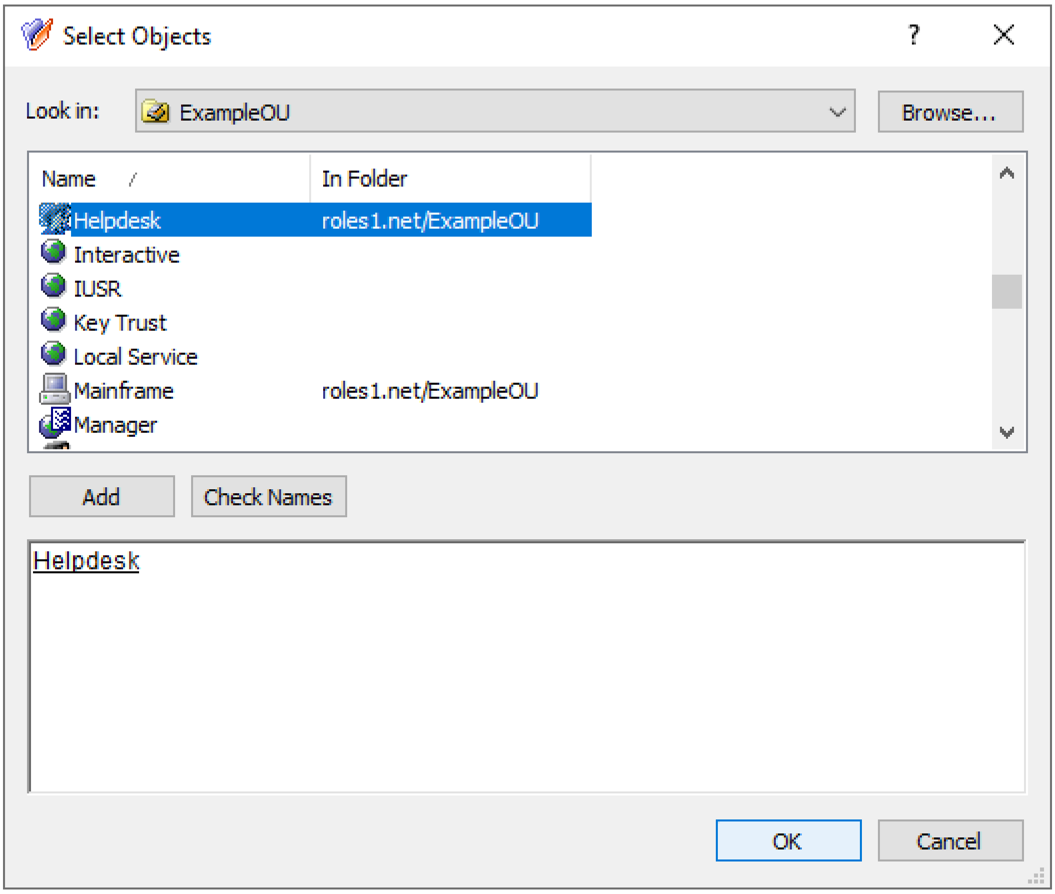
Name	In Folder
Denied-Azure-Resources	Configuration/Managed Units
Denied-Azure-Users	Configuration/Managed Units/...

    Denied-Azure-Users

    To continue, in the **Objects** step, click **Next**.

13. In the **Users or Groups** step, click **Add**, then select the users to which you want to delegate the permission. In this example, the AT is delegated to the **Helpdesk** group of an example Organizational Unit (OU). To add the group, click **Add**, then click **OK**.

Figure 97: Delegation of Control Wizard – Selecting the Helpdesk group as Trustee



To continue, in the **Users or Groups** step, click **Next**.

14. In the **Inheritance Options** step, make sure that the **This directory object** and **Child objects of this directory object** settings are selected. To continue, click **Next**.
15. In the **Permissions Propagation** step, leave the **Propagate permissions to Active Directory** setting in its default state. To continue, click **Next**.
16. To complete the wizard, click **Finish**.

Enabling or disabling the granular access to Azure users

Once you configured the Managed Unit (MU) of the Azure users, and set up the Access Template (AT) to deny access to those Azure users, the **Helpdesk** group to which the AT is assigned can no longer read the Azure users included in the MU. Instead, when opening the list of **Azure Users** on the Active Roles Web Interface, the Azure users included in the MU will be hidden from the **Helpdesk** group members.

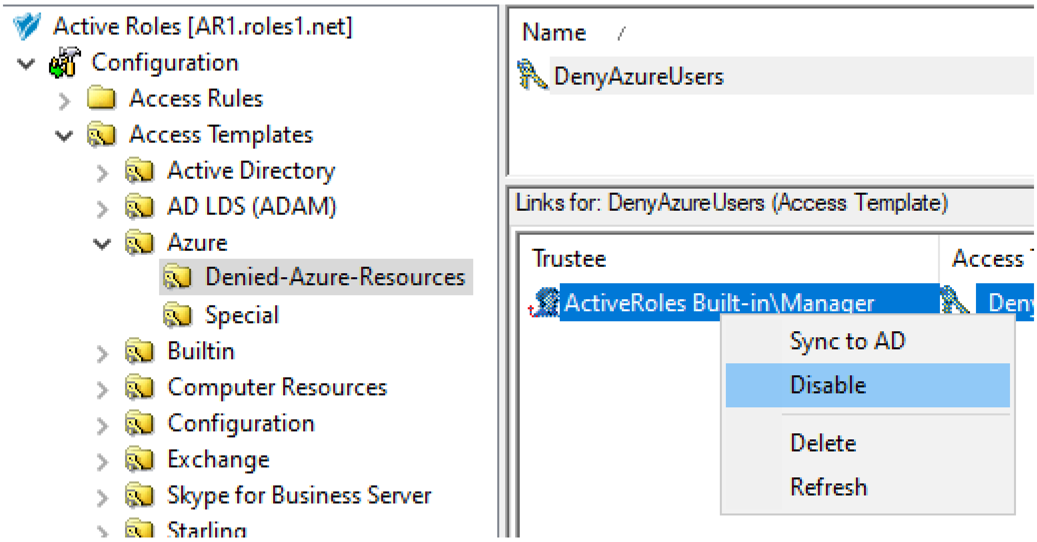
This behavior is dynamic: adding new Azure users into the MU in the Active Roles Console will result in those Azure users disappearing in the Active Roles Web Interface for the affected helpdesk users once the changes of the Console are synchronized to the Web Interface. Likewise, removing an Azure user from the MU will result in that Azure user appearing for the affected helpdesk users in the Web Interface.

You can easily enable or disable the configured granular access later for the affected helpdesk users by enabling or disabling the AT.

To enable or disable the configured granular access to Azure users

1. In the Active Roles Console, on the Console Tree, navigate to **Configuration** > **Access Templates** > **Denied Azure Resources**.
2. Select the **DenyAzureUsers** AT.
3. In the Advanced Details Pane, right-click the configured link, and click **Disable**.

   Figure 98: Active Roles Console – Disabling the configured Access Template

   > TIP: If the Advanced Details Pane does not appear for you, click **View** > **Advanced Details Pane**.

   Once the AT is disabled, the Azure users included in the associated **Denied-Azure-Users** MU will appear in the Web Interface for the users to which the AT is assigned.
4. (Optional) To re-enable the AT, right-click the configured link again, and click **Enable**.

Example: Configuring high granularity by showing only specific Azure users

This scenario describes how to use the Managed Units (MUs) and Access Templates (ATs) of the Active Roles Console together to configure Azure user administration permissions with high granularity. In this example, the MUs and ATs are used to grant a group of helpdesk users read access only to a specific group of Azure users. You can achieve this by:

1. Configuring an MU containing all the Azure users that the helpdesk users should access. For more information on this procedure, see Configuring a Managed Unit for specific Azure users.
2. Configuring an AT to grant access only to those Azure users for the helpdesk users. For more information on this procedure, see Configuring Access Templates to read specific Azure users.

Prerequisites

To configure this example scenario, your organization must meet the following requirements:

- To create MUs and ATs in the Active Roles Console, you must use an Active Roles Administration Service account. For more information, see *Configuring the Administration Service account* in the *Active Roles Quick Start Guide*.
- The organization must already have one or more Azure tenants configured and consented for use with Active Roles. For more information, see Configuring a new Azure tenant and consenting Active Roles as an Azure application.
- The users receiving the configured permissions must be on-premises or hybrid Active Directory users. You cannot delegate the configured granular permission to cloud-only Azure users.

Configuring a Managed Unit for specific Azure users

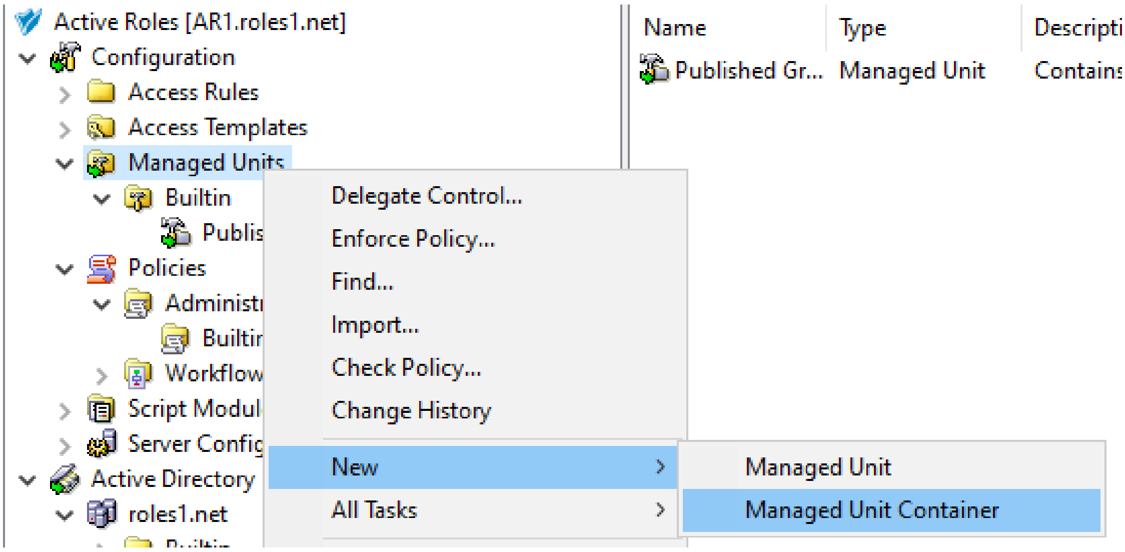
To set up a highly-granular Azure user access logic, first you must configure a Managed Unit (MU) that will contain the Azure users that the affected helpdesk users can read.

In this example, the membership of the MU is configured via group membership, specifying that only Azure users that are members of a specific group (in this example, **Engineering**) are included in the MU. For more information on the available membership rule options for MUs, see Creating a Managed Unit.

To configure a Managed Unit for specific Azure users

1. In the Active Roles Console, on the **Console tree**, navigate to **Configuration** > **Managed Units**.
2. To create a new container for the configured MU, right-click on the **Managed Units** node, then click **New** > **Managed Unit Container**.

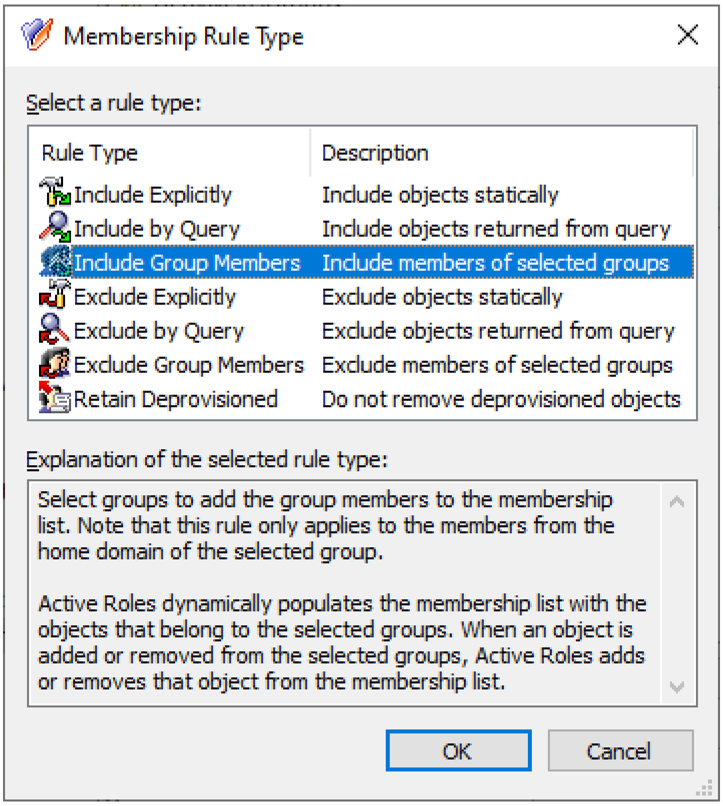
Figure 99: Active Roles Console – Launching the Managed Unit Container dialog



3. In the **Managed Unit Container** dialog, specify a **Name**, and optionally, a **Description** for the new MU container.
   - **Name**: `Allowed-Azure-Resources`
   - **Description**: `Managed Units for the granular access of Azure resources.`
4. To create the new container, click **Next** then **Finish**.
5. To start configuring the new MU, right-click the newly-created **Allowed-Azure-Resources** container, then click **New** > **Managed Unit**.
6. In the **Managed Unit Container** dialog, specify a **Name**, and optionally, a **Description** for the new MU container.
   - **Name**: `Allowed-Azure-Users`
   - **Description**: `Managed Unit for the granular access of Azure users.`

   To continue, click **Next**.
7. To specify a new membership rule for the MU, in the **Membership rule** step, click **Add**.
8. In the **Membership Rule Type** dialog, select the rule type used to populate the MU. This example uses the **Include Group Members** rule type. Select it, then click **Next**.

Figure 100: New Managed Unit – Selecting the Include Group Members rule type



Select a rule type:

Rule Type	Description
Include Explicitly	Include objects statically
Include by Query	Include objects returned from query
Include Group Members	Include members of selected groups
Exclude Explicitly	Exclude objects statically
Exclude by Query	Exclude objects returned from query
Exclude Group Members	Exclude members of selected groups
Retain Deprovisioned	Do not remove deprovisioned objects

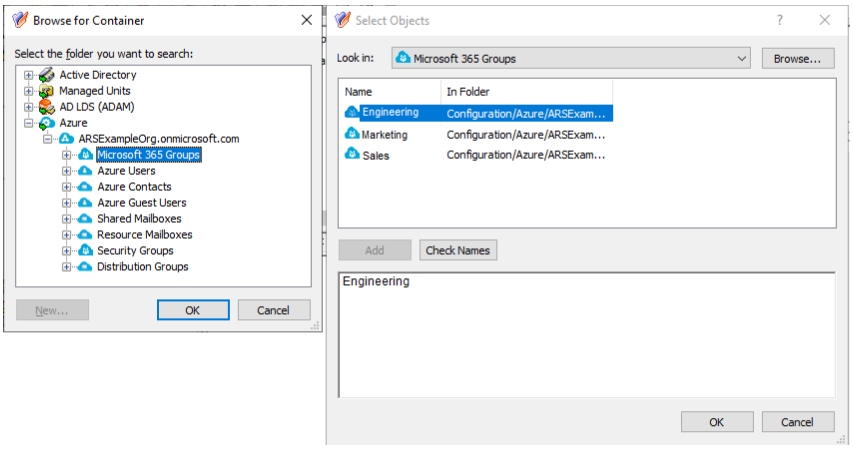
Explanation of the selected rule type:

Select groups to add the group members to the membership list. Note that this rule only applies to the members from the home domain of the selected group.

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.

9. In the **Select Objects** dialog, select the M365 group whose members you want to add to the MU.

Figure 101: New Managed Unit – Adding the members of an M365 Group to an MU



Name	In Folder
Engineering	Configuration/Azure/ARSExam...
Marketing	Configuration/Azure/ARSExam...
Sales	Configuration/Azure/ARSExam...

Engineering

To do so:

a. In the **Select Objects** dialog, click **Browse**.

b. In the **Browse for Container** dialog, expand the **Azure** > **<azure-tenant-name>** node (in this example, the Azure tenant is named **ARSExampleOrg.onmicrosoft.com**).

c. Select the **Microsoft 365 Groups** node, and click **OK**. The M365 groups existing in the Azure tenant will appear in the **Select Objects** dialog.

d. In the **Select Objects** dialog, select the M365 group you want to add to the MU (in this example, the **Engineering** group).

e. To apply the selection, click **Add** and **OK**.

10. To finish creating the MU, click **Next**, then **Next** again in the **Object Security** / **Policy Object** step, and finally **Finish**.

11. To verify that the MU is populated correctly, select the newly-created MU in the Console Tree. The members of the **Engineering** M365 group must appear in the Active Roles Console.

Configuring Access Templates to read specific Azure users

Once you set up the Managed Unit (MU) as described in Configuring a Managed Unit for specific Azure users, you must create two Access Templates (ATs) so that the affected helpdesk users:

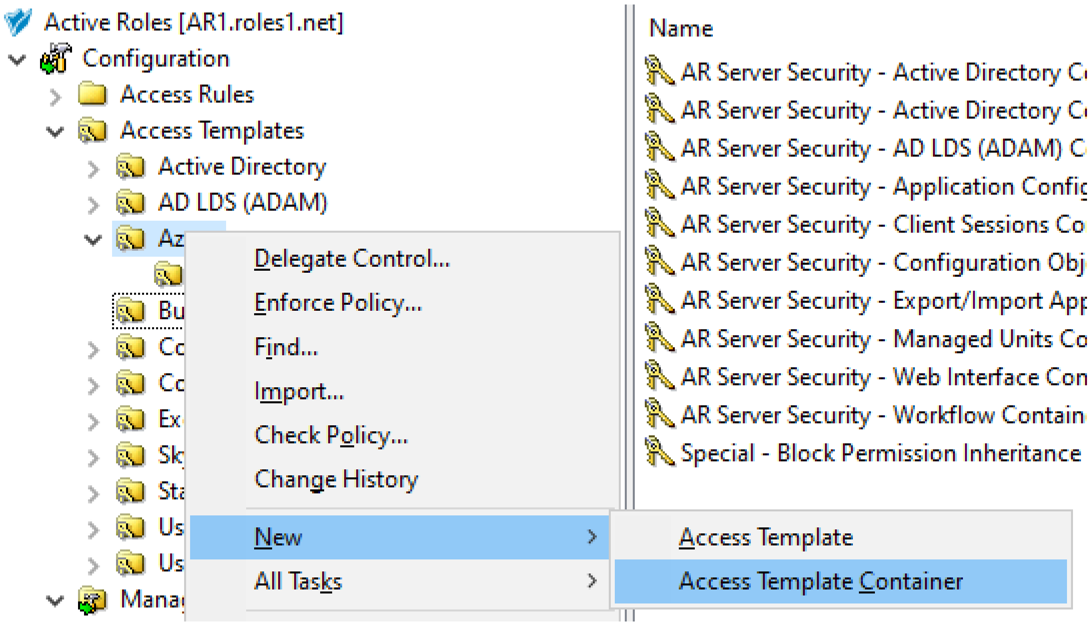
- Can read the Azure users of the configured MU.
- Cannot read any other Azure users in your organization.

To create these ATs, perform the following steps. For more information on creating ATs in general, see Creating an Access Template.

To provide read access to the Azure user object class

1. In the Active Roles Console, in the Active Directory (AD) tree, navigate to **Configuration** > **Access Templates**.
2. Create a new container where you will store the AT. In this example, the container is created in the **Azure** sub-container of the **Access Templates** node. Right-click **Access Templates** > **Azure**, then click **New** > **Access Template Container**.

Figure 102: Active Roles Console – Launching the Access Templates Container dialog

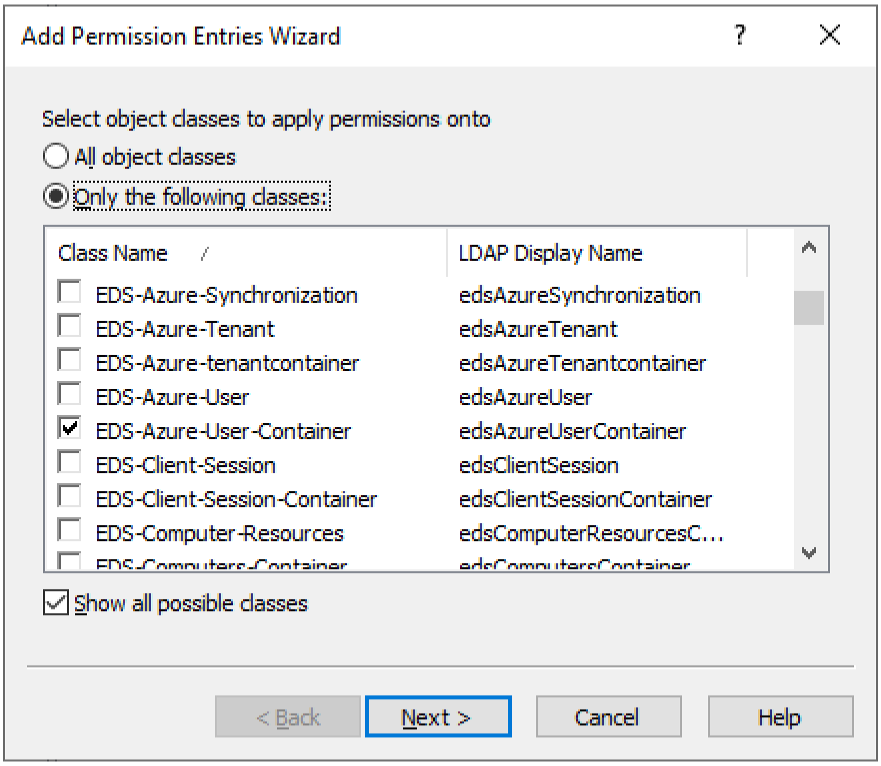

Active Roles [AR1.roles1.net]
Configuration
Access Rules
Access Templates
Active Directory
AD LDS (ADAM)
Az
Bu
Cc
Cc
Ex
Sk
Sta
Us
Us
Mana
Delegate Control...
Enforce Policy...
Find...
Import...
Check Policy...
Change History
New
All Tasks
Access Template
Access Template Container
Name
AR Server Security - Active Directory C
AR Server Security - Active Directory C
AR Server Security - AD LDS (ADAM) C
AR Server Security - Application Confic
AR Server Security - Client Sessions Co
AR Server Security - Configuration Obj
AR Server Security - Export/Import App
AR Server Security - Managed Units Co
AR Server Security - Web Interface Con
AR Server Security - Workflow Contain
Special - Block Permission Inheritance

3. In the **Access Templates Container** dialog, specify a **Name**, and optionally, a **Description** for the new AT container.
   - **Name**: `Allowed-Azure-Resources`
   - **Description**: `Access Templates for the granular access of Azure resources.`
4. To create the new container, click **Next** then **Finish**.
5. To start configuring the new AT, right-click the **Allowed-Azure-Resources** container, then click **New** > **Access Template**.
6. In the **New Object - Access Template** dialog, specify a **Name**, and optionally, a **Description** for the new AT.
   - **Name**: `ReadAzureUserObject`
   - **Description**: `AT to read cloud-only Azure user objects.`

   To continue with specifying the required permissions, click **Next**.
7. In the **Access Template permission entries** step, click **Add**. Then, in the **Select object classes to apply permissions onto** dialog, select **Only the following classes**, and the **EDS-Azure-User-Container** class from the list.

   TIP: If you cannot find the class in the list, select **Show all possible classes**.

Figure 103: New Access Template – Selecting the Azure Users container class to allow reading Azure users



Add Permission Entries Wizard

Select object classes to apply permissions onto

○ All object classes

● Only the following classes:

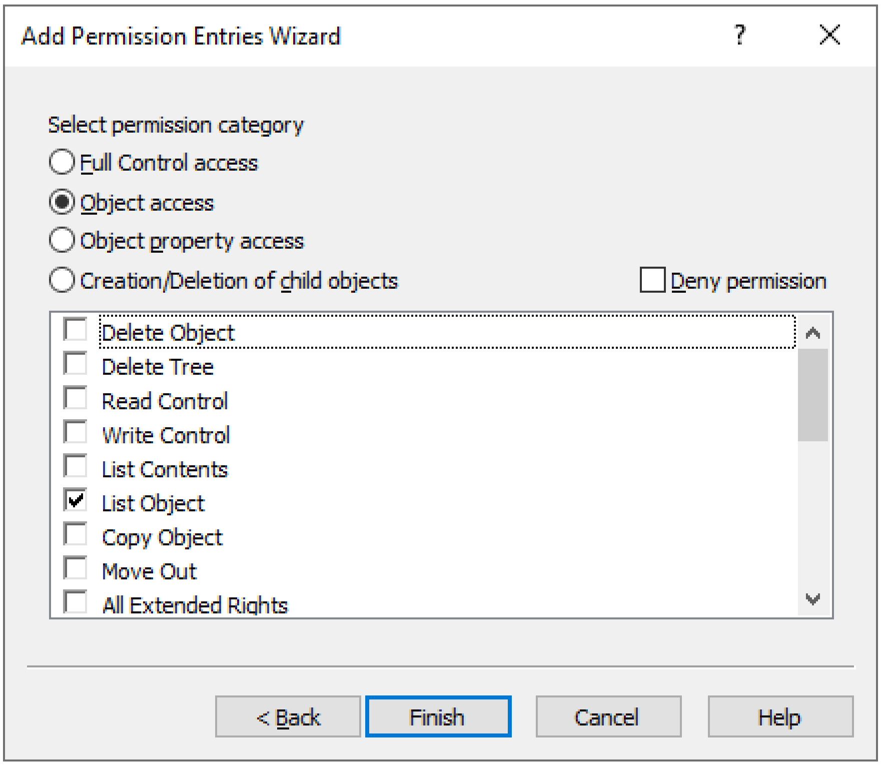
Class Name	LDAP Display Name
□ EDS-Azure-Synchronization	edsAzureSynchronization
□ EDS-Azure-Tenant	edsAzureTenant
□ EDS-Azure-tenantcontainer	edsAzureTenantcontainer
□ EDS-Azure-User	edsAzureUser
☑ EDS-Azure-User-Container	edsAzureUserContainer
□ EDS-Client-Session	edsClientSession
□ EDS-Client-Session-Container	edsClientSessionContainer
□ EDS-Computer-Resources	edsComputerResourcesC...
□ EDS-Computers-Container	edsComputersContainer

☑ Show all possible classes

To continue, click **Next**.

8. In the **Select permission category** step, select **Object access**, then select the **List Object** access permission from the list.

Figure 104: New Access Template – Specifying the permission to read allowed objects in the Azure Users container



Add Permission Entries Wizard

Select permission category

○ Full Control access

● Object access

○ Object property access

○ Creation/Deletion of child objects

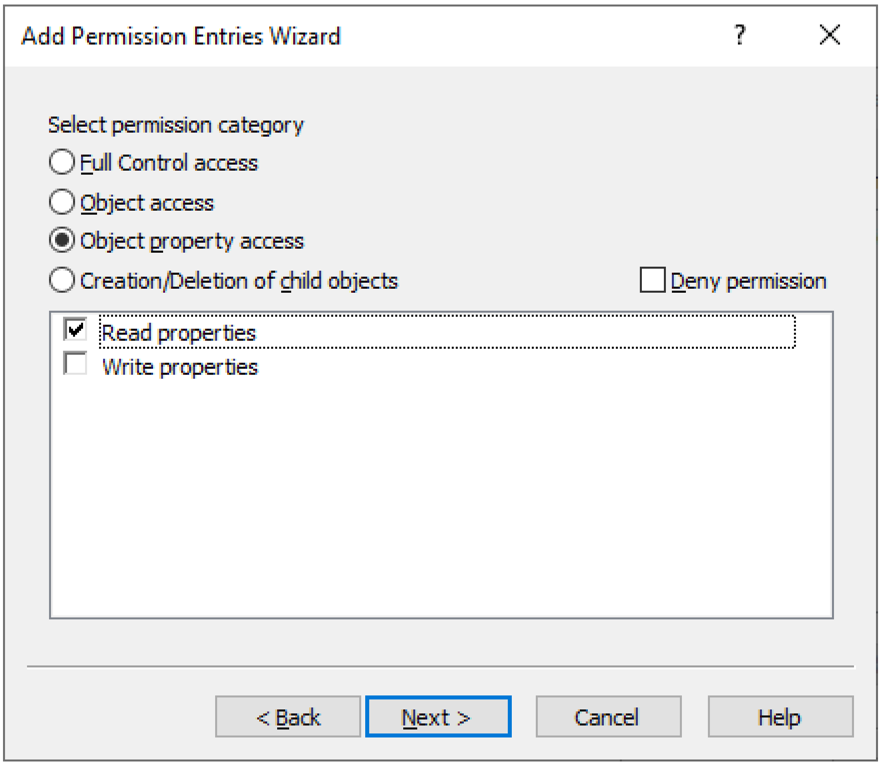
□ Deny permission

□ Delete Object
□ Delete Tree
□ Read Control
□ Write Control
□ List Contents
☑ List Object
□ Copy Object
□ Move Out
□ All Extended Rights

To finish configuring the permission, click **Finish**. Then, in the **Access Template permission entries step**, click **Add** again.

9. In the **Select object classes to apply permissions onto** dialog, select **Only the following classes**, then the **EDS-Azure-User-Container** class from the list again. To continue, click **Next**.
10. In the **Select permission category** step, select **Object property access**, then select the **Read properties** access permission from the list.

Figure 105: New Access Template – Specifying the permission to read the properties of the Azure Users container



Add Permission Entries Wizard

Select permission category

○ Full Control access

○ Object access

● Object property access

○ Creation/Deletion of child objects

□ Deny permission

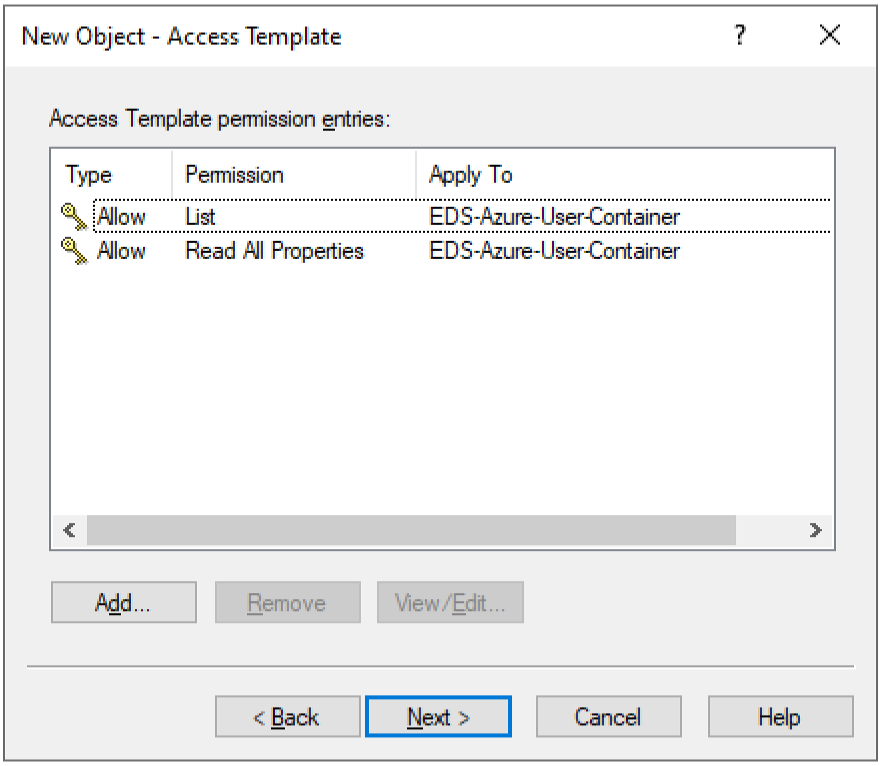
☑ Read properties

□ Write properties

To continue, click **Next**.

11. In the **Select object properties** step, leave the **All properties** option selected, then click **Finish.** The two permissions configured in the previous steps then appear in the **Access Template permission entries** step.

Figure 106: New Access Template – Listing the permissions to properly read the Azure Users container

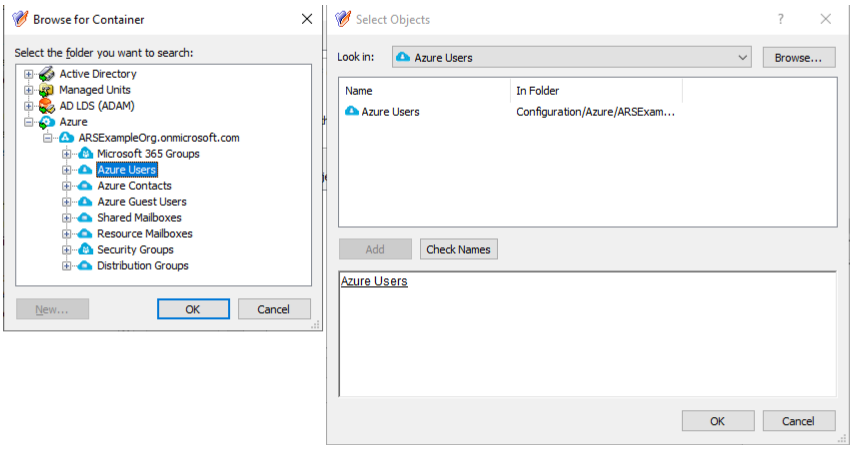


Access Template permission entries:

Type	Permission	Apply To
Allow	List	EDS-Azure-User-Container
Allow	Read All Properties	EDS-Azure-User-Container

12. To finish configuring the permissions of the AT, click **Next**, then **Finish**.
13. In the **Create in** step, select **Display the object properties when this wizard closes**, and click **Finish**.
14. To assign the AT to the helpdesk users and the Azure user container of the Azure tenant, in the **Properties** page that appears, click **Administration** > **Links**.
15. In the **Links** dialog, click **Add**, then specify the **Azure Users** container as the directory object managed by this AT.

Figure 107: New Access Template – Specifying the Azure Users container as the directory object in scope



To do so:

a. In the **Select Objects** dialog, click **Browse**.

b. In the **Browse for Container** dialog, expand the **Azure** > **<azure-tenant-name>** node (in this example, the Azure tenant is named **ARSExampleOrg.onmicrosoft.com**).

c. Select the **Azure Users** node, and click **OK**. The **Azure Users** container and the users contained in it will appear in the **Select Objects** dialog.

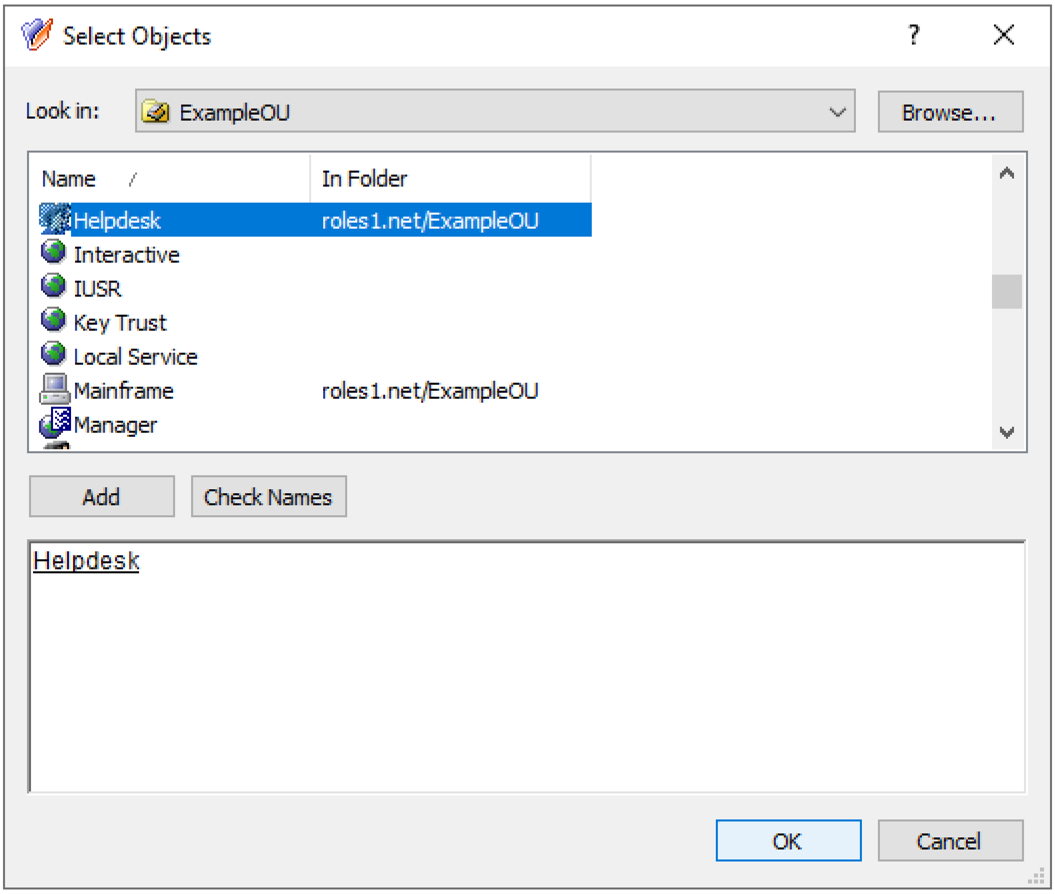
d. In the **Select Objects** dialog, select the **Azure Users** container.

e. To apply the selection, click **Add** and **OK**.

The **Azure Users** container then appears in the **Objects** step. To continue configuring the AT, click **Next**.

16. In the **Users or Groups** step, click **Add**, then select the users to which you want to delegate the permission. In this example, the AT is delegated to the **Helpdesk** group of an example Organizational Unit (OU). To add the group, click **Add**, then click **OK**.

Figure 108: Delegation of Control Wizard – Selecting the Helpdesk group as Trustee



Select Objects

Look in: ExampleOU

Name	In Folder
Helpdesk	roles1.net/ExampleOU
Interactive	
IUSR	
Key Trust	
Local Service	
Mainframe	roles1.net/ExampleOU
Manager	

Helpdesk

To continue, in the **Users or Groups** step, click **Next**.

17. In the **Inheritance Options** step, make sure that the **This directory object** and **Child objects of this directory object** settings are selected. To continue, click **Next**.
18. In the **Permissions Propagation** step, leave the **Propagate permissions to Active Directory** setting in its default state. To continue, click **Next**.
19. To apply your changes, click **Apply** and **OK**.

To restrict read access to the Azure users of a specific Managed Unit

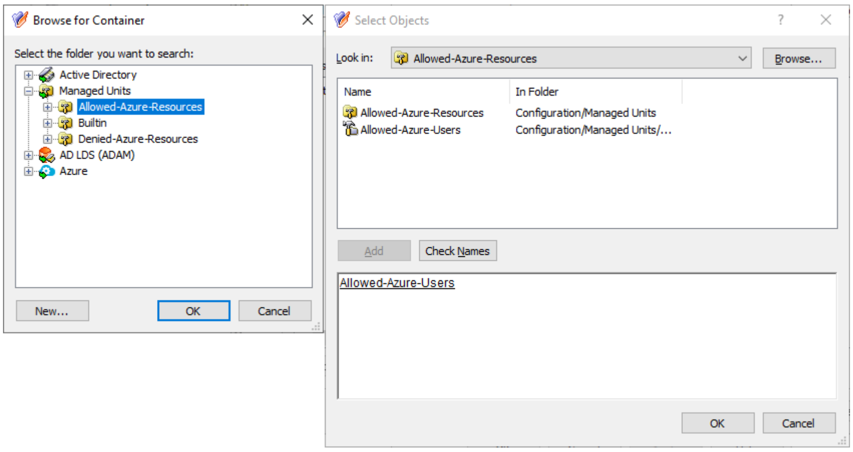
1. In the Active Roles Console, in the Active Directory (AD) tree, navigate to **Configuration** > **Access Templates**.
2. Right-click the **Azure Cloud User - Read All Attributes** built-in AT, and select **Copy**.
3. In the **Copy Object - Access Template** wizard, specify a **Name** and optionally, a **Description** for the new AT. This example uses the following values:

- **Name**: `AllowAzureUsers`
- **Description**: `AT to grant read access to the specified Azure users.`

To continue, click **Next**.

4. In the **Create in** step, select **Display the object properties when this wizard closes**, and click **Finish**.
5. To assign the AT to the helpdesk users and the Azure user container of the Azure tenant, in the **Properties** page that appears, click **Administration** > **Links**.
6. In the **Links** dialog, click **Add**, then specify the **Allowed Azure Users** MU as the directory object managed by this AT.

Figure 109: New Access Template – Specifying the Allowed Azure Users MU as the directory object in scope



Look in: Allowed-Azure-Resources

Name	In Folder
Allowed-Azure-Resources	Configuration/Managed Units
Allowed-Azure-Users	Configuration/Managed Units/...

Allowed-Azure-Users

To do so:

a. In the **Select Objects** dialog, click **Browse**.

b. In the **Browse for Container** dialog, select the **Managed Units** > **Allowed-Azure-Resources** node, and click **OK**.

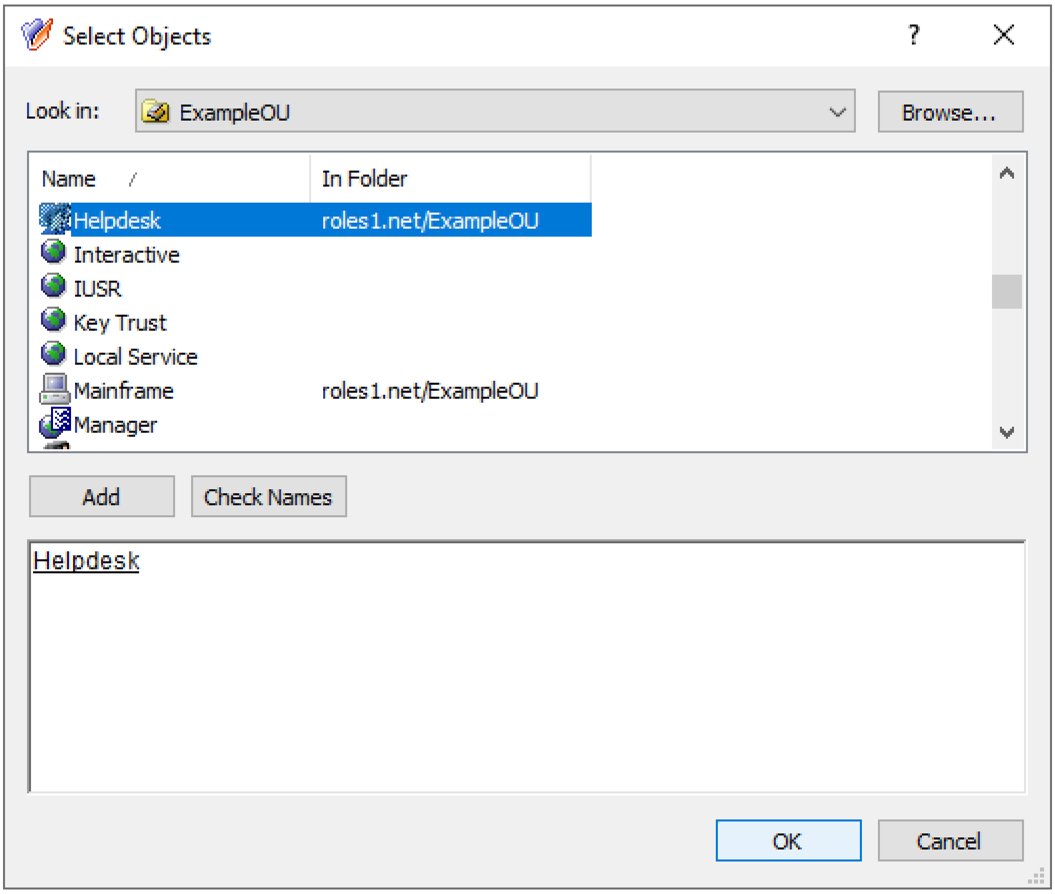
c. In the **Select Objects** dialog, select the **Allowed-Azure-Users** MU.

d. To apply the selection, click **Add** and **OK**.

The **Allowed-Azure-Users** MU then appears in the **Objects** step. To continue configuring the AT, click **Next**.

7. In the **Users or Groups** step, click **Add**, then select the users to which you want to delegate the permission. In this example, the AT is delegated to the **Helpdesk** group of an example Organizational Unit (OU). To add the group, click **Add**, then click **OK**.

Figure 110: Delegation of Control Wizard – Selecting the Helpdesk group as Trustee



To continue, in the **Users or Groups** step, click **Next**.

8. In the **Inheritance Options** step, make sure that the **This directory object** and **Child objects of this directory object** settings are selected. To continue, click **Next**.
9. In the **Permissions Propagation** step, leave the **Propagate permissions to Active Directory** setting in its default state. To continue, click **Next**.
10. To complete the configuration of the AT, click **Finish**. Then, in the **Links** dialog, click **OK**.
11. To apply your changes, click **Apply** and **OK**. Active Roles will create the copied AT in the **Configuration** > **Access Templates** > **Azure** container.
12. Move the AT to the **Configuration** > **Access Templates** > **Azure** > **Allowed-Azure-Resources** container. To do so, right-click the AT and click **Move**. Then, in the **Move** dialog, navigate to the **Allowed-Azure-Resources** container, select it, and click **OK**.

Enabling or disabling the granular access to specific Azure users

Once you configured the Managed Unit (MU) of the Azure users, and set up the Access Templates (ATs) to allow access to those Azure users only, the **Helpdesk** group to which the ATs are assigned can only read the Azure users included in the MU. When opening the list of **Azure Users** on the Active Roles Web Interface, all other Azure users included in the Azure tenant will be hidden from the **Helpdesk** group members.

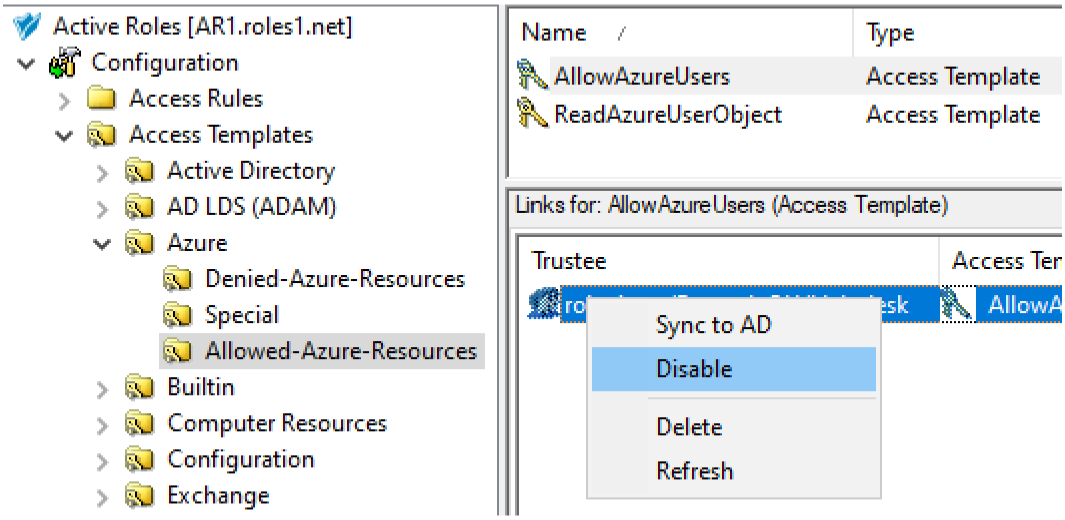
This behavior is dynamic: adding new Azure users into the MU in the Active Roles Console will result in those Azure users appearing in the Active Roles Web Interface for the affected helpdesk users once the changes of the Console are synchronized to the Web Interface. Likewise, removing an Azure user from the MU will result in that Azure user disappearing for the affected helpdesk users in the Web Interface.

You can easily enable or disable the configured granular access later for all affected helpdesk users by enabling or disabling the **AllowAzureUsers** and **ReadAzureUserObject** ATs.

To enable or disable the configured granular access to specific Azure users

1. In the Active Roles Console, on the Console Tree, navigate to **Configuration** > **Access Templates** > **Allowed-Azure-Resources**.
2. Select the **AllowAzureUsers** AT.
3. In the Advanced Details Pane, right-click the configured link, and click **Disable**.

Figure 111: Active Roles Console – Disabling the configured Access Template



Name	Type
AllowAzureUsers	Access Template
ReadAzureUserObject	Access Template

Links for: AllowAzureUsers (Access Template)

Trustee	Access Ter
ro esk	AllowA

TIP: If the Advanced Details Pane does not appear for you, click **View** > **Advanced Details Pane**.

4. Select the **ReadAzureUserObject** AT, and disable it as you did with the **AllowAzureUsers** AT.

   Once both ATs are disabled, the users of the **Helpdesk** group can no longer read the users included in the configured **Allowed-Azure-Resources** MU, and can no longer see the **Azure Users** container in the Active Roles Web Interface either.

5. (Optional) To re-enable the granular access, select one of the ATs, right-click the configured link, and click **Enable**. Then, enable the other AT similarly.

6. (Optional) To provide general read access to the entire **Azure Users** container of the Azure tenant instead of the configured granular access, assign the built-in **Azure Cloud User - Read All Attributes** AT (or a custom AT based on this built-in AT) to the **Helpdesk** group. For more information, see Applying Access Templates on a user or group.

7

Workflows

Active Roles provides a rich workflow system for directory data management automation and integration, allowing you to create, view, update or delete automation and approval workflows.

For more information on workflows in general, see *Workflows* in the *Active Roles Feature Guide*.

Key workflow 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 running activities and the relationship 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 initialized. 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 the Microsoft 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.

Email notifications

Users are notified via email 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.

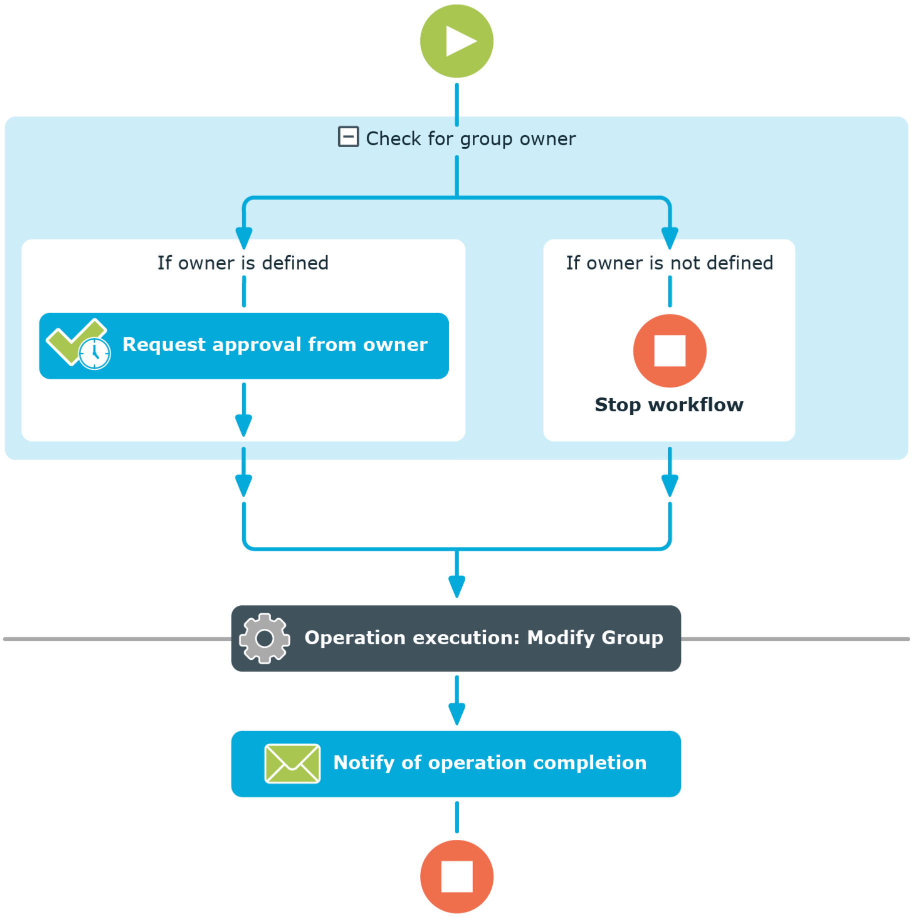
About workflow processes

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 email notification to the person who requested the changes, and then finishes.

Figure 112: Workflow process in Active Roles


Check for group owner
If owner is defined
Request approval from owner
If owner is not defined
Stop workflow
Operation execution: Modify Group
Notify of operation completion

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 be 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-run, run, and post-run. 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-run 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-run activities are completed so that the operation is not rejected, the Administration Service runs the pre-run 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-run 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-run polices finish running, the Administration Service runs the post-run 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 emails 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-run 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-run activities included in those workflows. Then, during the post-run phase, the Administration Service runs all the post-run 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 run the workflows. The activities of the workflow with a lower value of that attribute are initiated prior to the activities of the workflow with a higher value of that attribute. The workflows with the same priority value are initiated 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 workflow start conditions

To deploy a workflow in Active Roles, you create a workflow definition, configure the start conditions for that workflow, then 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 a set of activities organized in a process diagram. When you construct a workflow using the Workflow 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.

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.

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 click **Escalate** 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 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 emails, and set up email transport. Approval rules provide email 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 emails that include hypertext links to the approval-related section in the Web Interface.

Workflow notification recipients

Notification recipients are the users or groups to which the activity sends emails. 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.

Workflow 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.

Email server for workflow notifications

The email server setting determines the name and other parameters of the email 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 running the **Notification** activity. When configured to notify of this event, the activity creates and instantly sends an email message about initiating the **Notification** activity. Notification of this event is normally intended to inform that the workflow initiation 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 the running 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 the

running 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 notification event and recipients. When run 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 once that event occurs. The configurable settings of a **Notification** activity are similar to the notification settings of an **Approval** activity.

Workflow notification recipients

Notification recipients are the users or groups to which the activity sends emails. 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.

Workflow 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.

Email server for workflow notifications

The email server setting determines the name and other parameters of the email 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 Module** can freely be added and the **Script** contained in a **Script** can be developed and revised as necessary. This provides a mechanism for creating custom functions, enabling the extensibility of actions performed by a workflow.

The **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.

Notification – Script activity

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 email if no significant errors occurred during the run of this activity.
- **Activity encountered an error**: When configured to notify of this event, the activity causes Active Roles to send a notification email if any significant errors occurred during the run of this activity.

The notification settings specify the notification events and recipients. When run 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 once that event occurs. The notification settings are similar to the notification settings of a **Notification** activity. For more information, see Notification activity.

Error handling – Script activity

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 that condition is met, Active Roles runs the activities of that branch; otherwise, Active Roles evaluates the condition on the next branch (from left to right), and so on.

When configuring If-Else branch conditions, consider the following:

- Active Roles runs only the first branch whose condition is evaluated to **TRUE**.
- An **If-Else** activity can finish without any of its branches being initiated, if the condition of each branch is evaluated as **FALSE**.

If no condition is defined for a branch, Active Roles considers that branch to be always **TRUE**. Therefore, the final (rightmost) branch normally must have no condition, so that it is always evaluated as **TRUE**. This way, the final branch acts as the **Else** branch that runs if the conditions on the other branches are not fulfilled.

> TIP: To ensure that at least one activity is run from a branch, make sure that you define no running condition for the last branch of an **If-Else** activity.

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 runs 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: Click the name of the group, point to **Convert condition group to**, and then click the option appropriate to the desired logical operator.

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 evaluates whether Active Roles encountered an error when running a certain activity. You can select the activity when you configure a condition.

  > NOTE: This option requires the activity configuration to allow the workflow to continue even if the activity encounters an error. For more information, see Configuring error handling for a CRUD activity.

- **Workflow parameter value**: This option is intended to evaluate the value of a certain parameter of the workflow. You can select the 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 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 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 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 **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 39: Comparison value options

Condition to evaluate	Comparison value options
• 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	• 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 initialization status	• Not initiated • 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 – If-Else activity

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 stop 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 – Search activity
- "Run as" options
- Additional settings – Search activity
- 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 attribute of the object (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 you can select the object type varies depending on the search scenario you have selected.

Table 40: Search activity: Object type

Search scenario	Object types to search for
Search in the Organizational Unit or container.	• Users • 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

Search scenario	Object types to search for
	• 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 41: Search activity: Search scope

Search scenario	Search scope settings available
Search in the Organizational Unit (OU) 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.

Search scenario	Search scope settings available
	- **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 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

Search scenario	Search scope settings available
	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** 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 42: Search activity: Search options

Search scenario	Search options available
Search in the Organizational Unit (OU) 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 **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 43: Search 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 that have not been used to log on for more than a certain number of days, have old passwords, or are expired. • **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.

Objects to search for	Search options available
Printers	- **Printer features**: Search for printers with particular features, such as the printer model, paper size, print resolution, print speed, and other capabilities, for example printing double-sided or colored documents, or stapling pages.
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 have not 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 specific user or computer successfully logged in. Active Directory updates that attribute only periodically, rather than every time that a user or computer logs in. 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 login time is later than `lastLogonTimeStamp`. Hence, it is advisable to choose the login 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 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 specified 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 must 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 do not 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 44: 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	The text specified by the comparison value does not occur at the

Comparison operator	Indicates that
with	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.
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 regular expression.

The comparison values from which you can choose when configuring a filter condition are as follows.

Table 45: Comparison values

Comparison value	Description
Text string	A literal string of characters. You can type the 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 property (typically, a string-value property) when you configure a filter condition.
Property of workflow initiator	The value of a certain property of the user whose request started the workflow. You can select the property (typically, a string-value property) when you configure a filter condition.
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 property (typically, a string-value property) when you configure a filter condition.
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

Comparison value	Description
	a filter condition in a **Search** activity, you can choose the 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 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

Comparison value	Description
	particular number of days before or after the **Search** activity run. This comparison value is applicable to filter 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 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 email if no significant errors occurred during the run of this activity.
- **Activity encountered an error**: When configured to notify of this event, the activity causes Active Roles to send a notification email if any significant errors occurred during the run 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. For more information, see Notification activity.

Error handling – Search activity

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 stop 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 – Search activity

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).

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 email about whether or not the activity has encountered an error condition at run time.
- Error handling – CRUD activities: 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 – CRUD activities: Some advanced configuration options that allow you to adjust the processing of the operation requested by the activity.

Create activity

The **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 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 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 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 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 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 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 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 property when you configure a **Create** activity.
  - **Workflow parameter value**: Use the value of a certain parameter of the workflow. You can choose the 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 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 rule expression when you configure a **Create** activity.

The **Create** 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 – CRUD activities: 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 – CRUD activities: Some advanced configuration options that allow you to adjust the processing of the operation requested by the activity.

Update activity

The **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. For more information, see Activity target.
- **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 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 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 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 property when you configure an **Update** activity.
  - **Workflow parameter value**: Use the value of a certain parameter of the workflow. You can choose the 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 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 running the workflow. You can create the rule expression when you configure an **Update** activity.

The **Update** 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 – CRUD activities: 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 – CRUD activities: Some advanced configuration options that allow you to adjust the processing of the operation requested by the activity.

Add to group activity

The **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. For more information, see Activity target.
- **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.

The **Add to group** 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 – CRUD activities: 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 – CRUD activities: Some advanced configuration options that allow you to adjust the processing of the operation requested by the activity.

Remove from group activity

The **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. For more information, see Activity target.
- **Groups - Remove the object from all groups**: This options configures the activity to remove the object from all groups in Active Directory.

  > NOTE: The **Remove from group** activity cannot remove the object from its primary group. It can only remove the object from all other groups.
- **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.

The **Remove from group** 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 – CRUD activities: 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 – CRUD activities: Some advanced configuration options that allow you to adjust the processing of the operation requested by the activity.

Move activity

The **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. For more information, see Activity target.
- **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 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 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 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 rule expression when you configure a **Move** activity.

The **Move** 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 – CRUD activities: 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 – CRUD activities: Some advanced configuration options that allow you to adjust the processing of the operation requested by the activity.

Deprovision activity

The **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. For more information, see Activity target.

The **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 – CRUD activities: 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 – CRUD activities: Some advanced configuration options that allow you to adjust the processing of the operation requested by the activity.

Undo deprovision activity

The **Undo deprovision** activity restores the user or group that was deprovisioned with Active Roles. The activity causes Active Roles to roll back the changes 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 were 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. For more information, see Activity target.

The **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 – CRUD activities: 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 – CRUD activities: Some advanced configuration options that allow you to adjust the processing of the operation requested by the activity.

Delete activity

The **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. For more information, see Activity target.

The **Delete** 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 – CRUD activities: 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 – CRUD activities: 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 46: 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 email if no significant errors occurred during the run of this activity.
- **Activity encountered an error**: When configured to notify of this event, the activity causes Active Roles to send a notification email if any significant errors occurred during the run of this activity.

The notification settings specify the notification event and its recipients. When run 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 when the event occurs. The notification settings are similar to the notification settings of a **Notification** activity. For more information, see Notification activity.

Error handling – CRUD activities

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 stop 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, Active Roles runs CRUD activities under the user account specified with the **Workflow options and start conditions** > **Configure** > **"Run as" options** setting. This user account can 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" options** 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 – CRUD activities

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

The **Save Object Properties** activity is used 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 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 running the workflow. You can specify which object you want the activity to select at the time of running the workflow.
    - **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 running the workflow. You can create the 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 email if no significant

errors occurred during the run of this activity.

- **Activity encountered an error**: When configured to notify of this event, the activity causes Active Roles to send a notification email if any significant errors occurred during the run of this activity.

The notification settings specify the notification events and recipients. When run 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 once that event occurs. The notification settings are similar to the notification settings of a **Notification** activity. For more information, see Notification activity.

- **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 activity 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 must specify an attribute listed in the **Target properties** setting of the Save Object Properties**Save Object Properties** activity; otherwise, this expression returns no property value when running the workflow.

- By adding the **Workflow - Saved Object Properties** token to the notification message template. For more information, see Events, recipients, messages.
- To add the token:
  1. In the **Insert Token** dialog, click **Workflow - Saved Object Properties** in the list of tokens, and then click **OK**.
  2. In the dialog that appears, select the name of the **Save Object Properties** activity and the saved property you want the token to retrieve.

     You must 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 when running the workflow.
- 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:
  1. In the **Object Property** dialog, click the link in the **Target object** field, and then click **More choices**.
  2. In the dialog 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**.
  3. In the **Object Property** dialog, click the link in the **Target property** field, and select the property you want.

     You must 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 when running the workflow.

Modify Requested Changes activity

The **Modify Requested Changes** 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 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 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 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 property from a list of object properties.

  - **Workflow parameter value**: Use the value of a certain parameter of the workflow. You can choose the 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 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 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 email if no significant errors occurred during the run of this activity.
  - **Activity encountered an error**: When configured to notify of this event, the activity causes Active Roles to send a notification email if any significant errors occurred during the run of this activity.

  The notification settings specify the notification event and recipients. When run 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. For more information, see Notification activity.
- **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 stop 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 the *Active Roles SDK* documentation.

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.

Creating a workflow definition for 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 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. For more information, see Automation workflow.

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 in the Details pane, representing the workflow definition as a process diagram.
2. In the Details pane, click **Workflow options and start conditions** to expand the area above the process diagram, and then click **Configure**.
3. Click the **Conditions** tab in the **Change Workflow Options and Start Conditions** dialog.

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, navigate to **Conditions**, 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, 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, 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: Click the name of the group, point to **Convert condition group to**, and then click the option appropriate to the desired logical operator.

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.

Configuring filtering conditions

The **Change Workflow Options and Start Conditions** dialog 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 perform 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, click **Add entry** and then click **Value generated by script**.
4. Use the **Configure Entry** dialog to select the appropriate script module and script function.
5. Click **OK** to close the **Configure Entry** dialog.
6. Click **OK** to close the **Configure Rule Expression** dialog.
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, under **Text string**, type `TRUE`.
10. Click **OK** to close the **Configure Entry** dialog.
11. Click **OK** to close the **Change Workflow Options and Start Conditions** dialog.
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.

"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 runs 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 Active Roles, 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 in the Details pane, representing the workflow definition as a process diagram.

2. In the Details pane, click **Workflow options and start conditions** to expand the area above the process diagram, and then click **Configure**.
3. Click the **Parameters** tab in the dialog that opens.
4. On the **Parameters** page, click **Add** to open the **Parameter Definition** dialog.
5. In the **Parameter Definition** dialog, 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.
   - (Optional) **Description**: Use this box to type a description of the parameter.
   - **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 **View or change parameter value**.

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 to configure the activity.

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-running phase of operation processing. For more information, see Workflow processing overview. 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-running phase of operation processing. Certain activities, such as an **Approval** activity, which are intended to run in the pre-running phase, cannot be added to the lower part of the diagram.

In the **Properties** dialog, 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 the following configuration procedures:

- Configuring approvers
- Configuring escalation
- Configuring request for additional information
- Configuring request for review
- Customizing the header of the approval task page
- Customizing approval action buttons

For more information on how to configure notification settings, see Configuring a Notification activity.

Configuring 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 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, navigate to the **Approvers** tab.
4. Verify that the **Initial approver - level 0** item is selected in the **Select approver level to configure** box.
5. Click **Designate approvers**.
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 Configuring escalation), then you have to specify approvers for escalation level 1 (the escalation level subsequent to the initial approver level). Active Roles supports 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 **Designate approvers**.
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.

Configuring 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 called 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 Configuring approvers).
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 Configuring approvers).

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 Configuring approvers).

> NOTE: Each approver level has separate configuration, so the escalation options of a specific level apply only to that level. Therefore, 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 that level are allowed to escalate the approval task manually.

Configuring 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. Navigate to the **Request for information** tab in the **Properties** dialog 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 for the **Approval** activity, under the **Show this instruction to the approver** heading.

Configuring 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. Navigate to the **Request for information** tab in the **Properties** dialog for the **Approval** activity.
2. Select the **Show the original request to the approver** check to enable the approver to review the properties submitted for approval.
3. (Optional) 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.

> TIP: 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.

Customizing 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 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. Navigate to the **Customization** tab in the **Properties** dialog 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 **Add** and **Remove** 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.

> TIP: 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.

> NOTE: 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 do not affect the tasks created earlier.

Customizing 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 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, 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: This option is unavailable for the **Escalate** and **Delegate** action types. The Web Interface displays the **Escalate** or **Delegate** button only 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. Navigate to the **Customization** tab in the **Properties** dialog for the **Approval** activity.
2. Click **Customize action buttons**.
3. Click **Add**.
4. In the **Action Button Properties** dialog, 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.

> TIP: When adding a custom action button, One Identity recommends including instructions explaining the meaning and purpose of the custom action. You can enter these instructions in the **Properties** > **Customization** > **Customize action buttons** > **Show this instruction for action buttons** field of 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 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 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 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. Navigate to the **Notification** tab in the **Properties** dialog.

The page for configuring notifications includes three areas:

- Events, recipients, 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 URL in Notifications: This area is used to specify the address (URL) of the Active Roles Web Interface, for constructing hyperlinks in the notification messages.
- Email server settings: In this area you can view or change the name and other settings of the email server that is used for delivery of notification messages.

Events, recipients, messages

To add a notification

1. In the **Events, recipients and messages** area, click **Add**.
2. In the **Notification Settings** dialog, 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 email addresses of the recipients you select on the **Notification Recipients** tab appear in the **To** field of the notification email messages. To add recipient addresses to the **Cc** or **Bcc** field, click **Cc Recipients** or **Bcc Recipients**, 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.

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 email 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 **Insert Token** 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 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 URL in Notifications

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.

Email server settings

The settings specified in this area determine the server to use for notification delivery.

To configure email server settings

1. In the **E-mail server settings** area, click **Properties**.
2. Use the **Properties** dialog to view or change the email server settings.

To select a different email 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.

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 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. Navigate to the **General** tab in the **Script Activity Properties** dialog.
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 **Browse** 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) Navigate to the **Notification** tab in the **Script Activity Properties** dialog, 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 email if no significant errors occurred during the run of this activity.
   - **Activity encountered an error**: When configured to notify of this event, the activity causes Active Roles to send a notification email if any significant errors occurred during the run of this activity.

8. (Optional) Navigate to the **Error handling** tab in the **Script Activity Properties** dialog, 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 stop 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 Configuring 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 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-running phase of operation processing. For more information, see Workflow processing overview.

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-run phase of operation processing. Certain activities, such as an **Approval** activity, which are intended to run in the pre-running 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**.

For more information on how to configure conditions for an **If-Else Branch**, see Configuring conditions for an If-Else branch.

Configuring 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. Navigate to the **Error handling** tab in the **If-Else Activity Properties** dialog, 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 stop 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.

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 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 **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 **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 workflows.
   - **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 workflows.

- **Activity execution status**: Evaluate whether Active Roles encountered an error when running a certain activity. The condition builder prompts you to select the desired activity.

  > NOTE: 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 workflows.
- **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 47: Comparision value options

Condition to evaluate	Comparison value options
• Property of workflow target object • Property of workflow initiator • Changed value of workflow target	• Text string • Property of workflow target object • Property of workflow initiator

Condition to evaluate	Comparison value options
object property • Workflow parameter value • Property of object from workflow data context • 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 run status	• Not initialized • Completed successfully • Encountered an error
Approver action choice	• The name of an action button • Value generated by script

For more information on 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, 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, click **Add entry** and then click **Value generated by script**.
4. Use the **Configure Entry** dialog to select the appropriate **Script Module** and script function.
5. Click **OK** to close the **Configure Entry** dialog.
6. Click **OK** to close the **Configure Rule Expression** dialog.

7. In the **If-Else Branch Activity Properties** dialog, 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, under **Text string**, type `TRUE`.
9. Click **OK** to close the **Configure Entry** dialog.
10. Click **OK** to close the **If-Else Branch Activity Properties** dialog.
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 the **Configure Entry** dialog returns TRUE at workflow run time.

Configuring a Stop/Break activity

When configuring a **Stop/Break** activity, you can specify the text of an information message. The activity stops 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 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 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.
   - **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 **X** 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 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:

- Configuring Scope and filter settings
- Configuring a notification for a Search activity
- Configuring error handling for a Search activity
- Configuring "Run as" options for a Search activity
- Configuring additional settings for a Search activity

Configuring Scope and filter settings

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.

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.
   - **Contacts**: 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 Recipients**: Search for mailboxes or mail-enabled users, groups, or contacts.
   - **Inactive Accounts**: Search for users computers that have not logged in 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**. To see what object types are available for a given search scenario option, see Object type.
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 48: Configure scope and 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 running 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 running the workflow. You can create the desired rule expression when

Search scenario	“Find-in” options available
	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 running 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 running 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

Search scenario	“Find-in” options available
	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 running 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 running 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 running 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 running 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 contacts.
- 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 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 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 email addresses, mail-enabled groups, contacts with external email addresses, mail-enabled Public Folders, Query-based Distribution Groups, room mailboxes, equipment mailboxes, linked mailboxes, or shared mailboxes. In the dialog 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 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 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. (Optional) Configure a filter to further refine your search, as described in Configuring a search filter.

Configuring a search filter

Search filters allow 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 must use to determine whether a specific object is to be included in the search results. The Workflow Designer provides a condition builder for configuring filter conditions, located in the **"Search" Activity Properties** > **Scope and filter** > **Search options** setting when selecting a **Search** activity.

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 **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 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 originally selected.
3. Click **Define value to compare to**, and then choose an option to specify the desired comparison value. The following options are available:

Table 49: 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 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.

Option	Description
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 running 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 running 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.
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 run 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.

Configuring a notification for a Search activity

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 email if no significant errors occurred during the run of this activity.
- **Activity encountered an error**: When configured to notify of this event, the activity causes Active Roles to send a notification email if any significant errors occurred during the run 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, 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 initiated 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 for a Search activity

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, 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 stop 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.

Configuring “Run as” options for a Search activity

By default, the **Search** activity is started 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.
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.

Configuring additional settings for a Search activity

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.
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 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.

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.

- 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 a notification for a CRUD activity: Active Roles can notify via email about whether or not the activity has encountered an error condition at run time.
- Configuring error handling for a CRUD activity: 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 for a CRUD activity: Determines the user account under which to run the activity.
- Configuring additional settings for a CRUD activity: Some advanced configuration options that allow you to adjust the processing of the operation requested by the activity.

Configuring a 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. Navigate to the **Container** tab in the **“Create” Activity Properties** dialog.
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. Navigate to the **Object name** tab in the **"Create" Activity Properties** dialog.
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. Navigate to the **Object properties** tab in the **"Create" Activity Properties** dialog.
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 running the workflow. You can create the desired rule expression when you configure the activity.

11. View or change notification settings. For more information, see Configuring a notification for a CRUD activity.
12. View or change error handling settings. For more information, see Configuring error handling for a CRUD activity.
13. View or change the **"Run as" options**. For more information, see Configuring "Run as" options for a CRUD activity.
14. View or change additional settings. For more information, see Configuring additional settings for a CRUD activity.

Configuring an 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 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. Navigate to the **Activity target** tab in the **"Update" Activity Properties** dialog.
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 running 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 running the workflow. You can create the desired rule expression when you configure the activity.

5. Navigate to the **Target properties** tab in the **"Update" Activity Properties** dialog.

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 **Delete** 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 running 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 running the workflow. You can create the desired rule expression when you configure the activity.

9. View or change notification settings. For more information, see Configuring a notification for a CRUD activity.
10. View or change error handling settings. For more information, see Configuring error handling for a CRUD activity.
11. View or change the **"Run as" options**. For more information, see Configuring “Run as” options for a CRUD activity.
12. View or change additional settings. For more information, see Configuring additional settings for a CRUD activity.

Configuring an “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 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. Navigate to the **Activity target** tab in the **“Add to Group” Activity Properties** dialog.
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. Navigate to the **Groups** tab in the **"Add to Group" Activity Properties** dialog.
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 more information, see Configuring a notification for a CRUD activity.
8. View or change error handling settings. For more information, see Configuring error handling for a CRUD activity.
9. View or change the **"Run as" options**. For more information, see Configuring "Run as" options for a CRUD activity.
10. View or change additional settings. For more information, see Configuring additional settings for a CRUD activity.

Configuring a 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 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. Navigate to the **Activity target** tab in the **"Remove from Group" Activity Properties** dialog.
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. Navigate to the **Groups** tab in the **"Remove from Group" Activity Properties** dialog.
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: You cannot remove objects from their primary groups. Instead, the activity will remove the object from all groups except its 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 runtime.
   - **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 runtime.
   - **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 running 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 **X** on the right side of the list item representing that group.

8. View or change notification settings. For more information, see Configuring a notification for a CRUD activity.
9. View or change error handling settings. For more information, see Configuring error handling for a CRUD activity.
10. View or change the **"Run as" options**. For more information, see Configuring “Run as” options for a CRUD activity.
11. View or change additional settings. For more information, see Configuring additional settings for a CRUD activity.

Configuring a 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 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. Navigate to the **Activity target** tab in the **"Move" Activity Properties** dialog.
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. Navigate 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 more information, see Configuring a notification for a CRUD activity.
8. View or change error handling settings. For more information, see Configuring error handling for a CRUD activity.
9. View or change the **"Run as" options**. For more information, see Configuring “Run as” options for a CRUD activity.
10. View or change additional settings. For more information, see Configuring additional settings for a CRUD activity.

Configuring a 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 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.

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 running 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 more information, see Configuring a notification for a CRUD activity.
6. View or change error handling settings. For more information, see Configuring error handling for a CRUD activity.
7. View or change the **"Run as" options**. For more information, see Configuring “Run as” options for a CRUD activity.
8. View or change additional settings. For more information, see Configuring additional settings for a CRUD activity.

Configuring an 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 were 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 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.
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 running the workflow. You can create the desired rule expression when you configure the **Undo deprovision** activity.
5. View or change notification settings. For more information, see Configuring a notification for a CRUD activity.
6. View or change error handling settings. For more information, see Configuring error handling for a CRUD activity.
7. View or change the **"Run as" options**. For more information, see Configuring “Run as” options for a CRUD activity.
8. View or change additional settings. For more information, see Configuring additional settings for a CRUD activity.

Configuring a 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 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.
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 more information, see Configuring a notification for a CRUD activity.
6. View or change error handling settings. For more information, see Configuring error handling for a CRUD activity.
7. View or change the **"Run as" options**. For more information, see Configuring "Run as" options for a CRUD activity.
8. View or change additional settings. For more information, see Configuring additional settings for a CRUD activity.

Configuring a notification for a CRUD activity

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 email if no significant errors occurred during the run of this activity.
- **Activity encountered an error**: When configured to notify of this event, the activity causes Active Roles to send a notification email if any significant errors occurred during the run of this activity.

To configure a 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, and configure the notification settings. For more information, see Configuring a Notification activity.

The notification settings specify the event to notify of, and notification recipients. When initiated 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 for a CRUD activity

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, 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 stop 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 for a CRUD activity

By default, CRUD activities are executed under the user account specified by the **“Run as” options** 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 **“Run as” options** at the bottom of the **Properties** dialog.

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 for a CRUD activity

By using additional settings, you can override the default operation reason text, and add request controls to modify the 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.
3. In the **Additional Settings** dialog, 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**: When selected, requests created by the activity can start new instances of the workflow containing the activity.

  > TIP: One Identity recommends leaving this setting clear in most cases, so that you can prevent the recurrent initialization of the activity if the operation requested by the activity within a specific workflow matches the start conditions of that same workflow.

  > NOTE: Selecting this setting may result in a loop of workflow instances that repeatedly initializes the same activity, eventually resulting in an overflow.

- **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.

To add or change a control, click **Add** or **Change**, and then, in the dialog 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 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. Navigate to the **Activity target** tab in the **"Save Object Properties" Activity Properties** dialog.
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 running 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 running the workflow. You can create the desired rule expression when you configure the activity.
5. Navigate to the **Target properties** tab in the **"Save Object Properties" Activity Properties** dialog.
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. Navigate to the **Notification** tab in the **"Save Object Properties" Activity Properties** dialog to view or change notification settings. For more information, see Configuring a notification for a CRUD activity.
8. Navigate to the **Error handling** tab in the **"Save Object Properties" Activity Properties** dialog to view or change error handling settings. For more information, see Configuring error handling for a CRUD activity.

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 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. Navigate to the **Target changes** tab in the **"Modify Requested Changes" Activity Properties** dialog.
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 **Delete** 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. Navigate to the **Notification** tab in the **"Modify Requested Changes" Activity Properties** dialog to view or change notification settings. For more information, see Configuring a notification for a CRUD activity.
8. Navigate to the **Error handling** tab in the **"Modify Requested Changes" Activity Properties** dialog to view or change error handling settings. For more information, see Configuring error handling for a CRUD activity.
9. Click the **Additional settings** link at the bottom of the **"Modify Requested Changes" Activity Properties** dialog.
10. In the **Additional Settings** dialog 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 running 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 running 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 running 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 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.

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. In the Active Roles Console tree, select the workflow definition 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 run 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

- In the Active Roles Console tree, right-click the workflow definition 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 running 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 initalization of the workflow instance.

When you configure a workflow, you can specify PowerShell commands you want the workflow runtime engine to run 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 cmdlets 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 the 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 in the Details pane, representing the workflow definition as a process diagram.

2. In the Details pane, click **Workflow options and start conditions** to expand the area above the process diagram, and then click **Configure**.
3. Click the **Initialization script** tab in the dialog 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.

Approval workflow

Approval workflows complement 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.

An **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 the kind of operations that start the workflow, and also 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. 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, you can configure an **Approval** workflow so that when creating a user account via Active Roles, the user is only created if the action is approved beforehand.
- 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 – Approval workflow

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 email 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 the Approval workflow works

An **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).

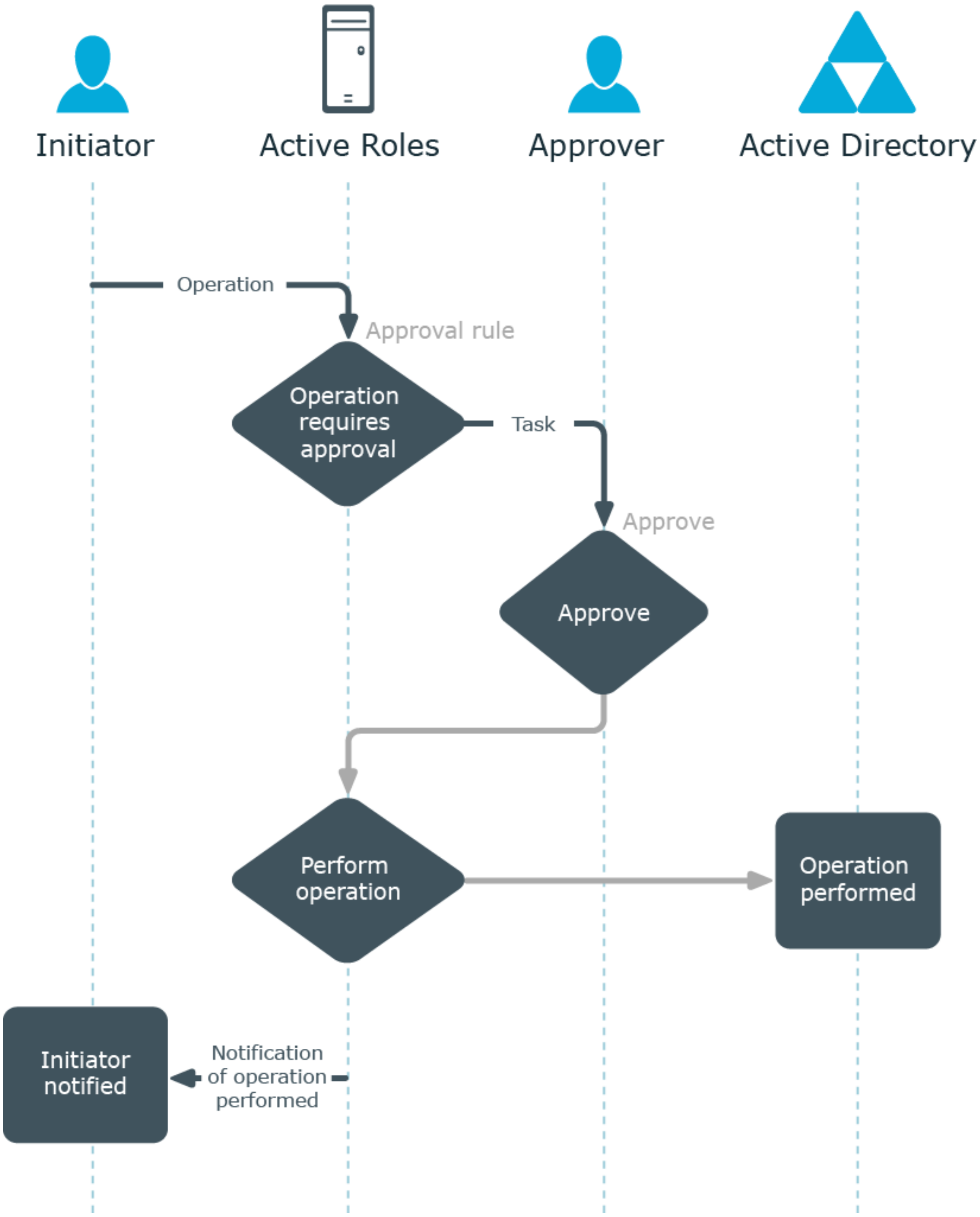
When an Active Roles user requests an operation, Active Roles checks to see whether the operation meets the start conditions of any workflow, and starts the workflow whose conditions are met. An **Approval rule** included in the workflow then generates an approval task and assigns the task to the approvers 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.

Approve action

If the approver applies the Approve action to the task, Active Roles allows the operation to be performed.

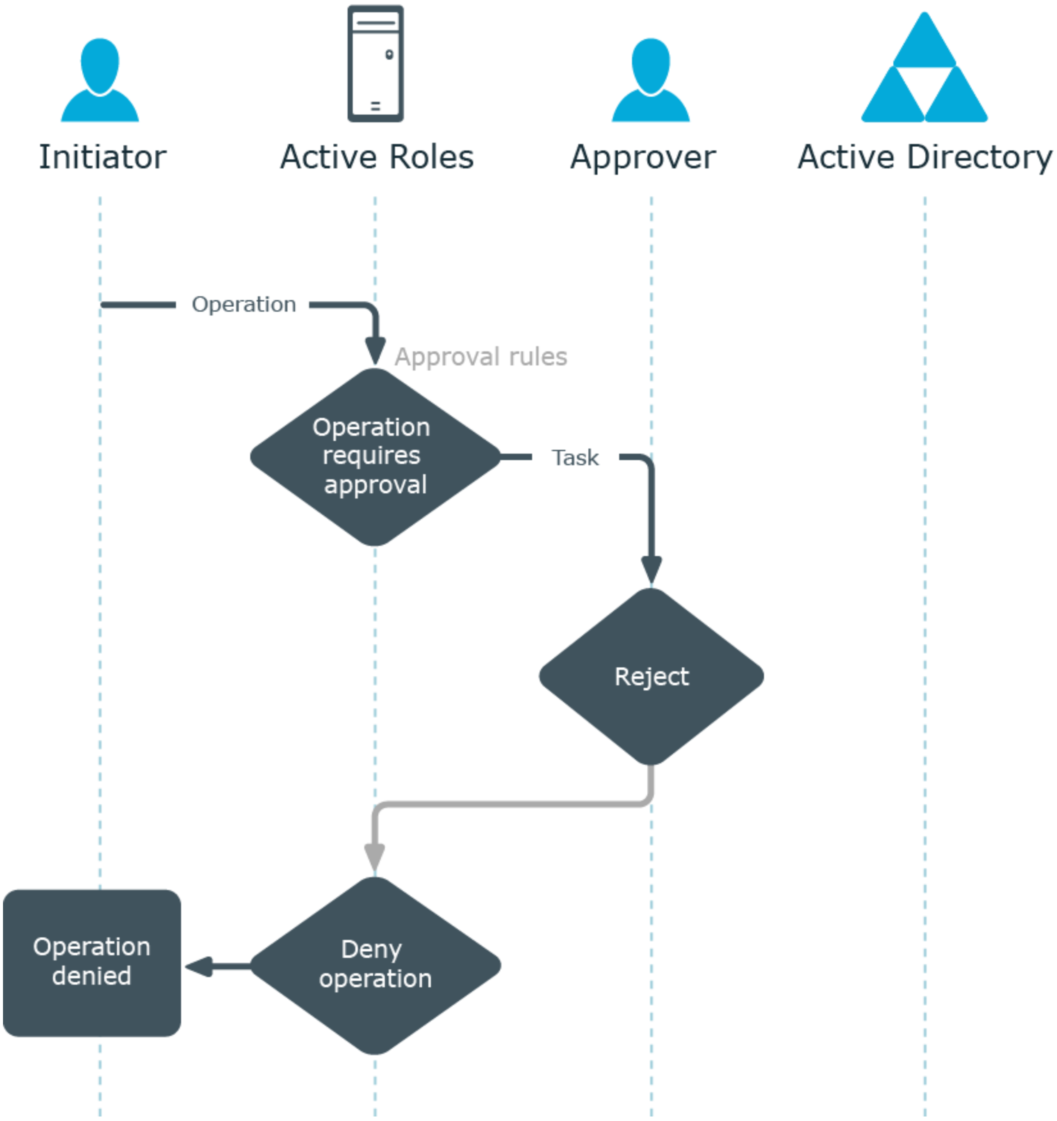
Figure 113: Approve action



Reject action

If the approver applies the **Reject** action to the task, Active Roles cancels the operation.

Figure 114: Reject action

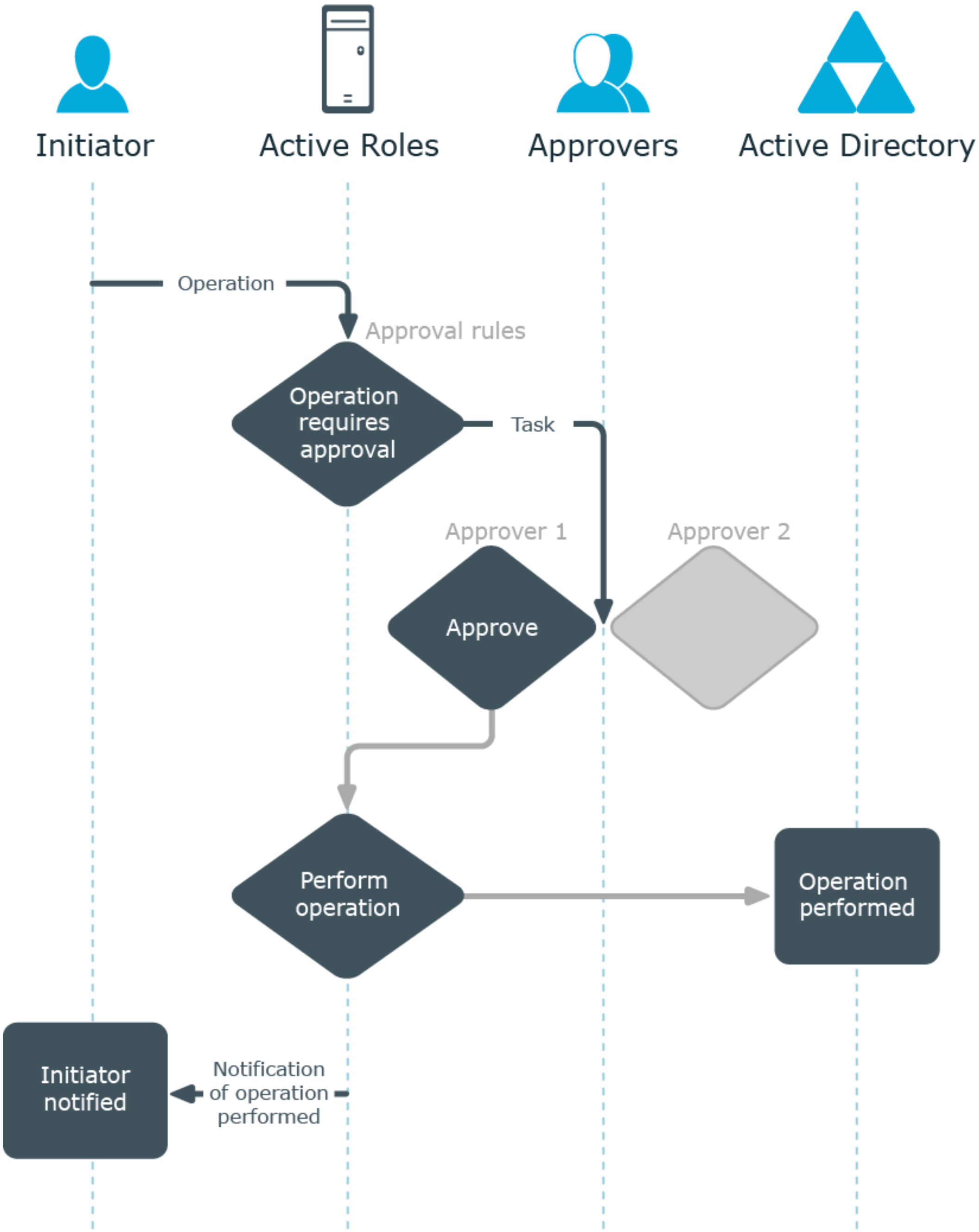


Multiple approvers

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. See the figure for an example of multiple approvers in a workflow.

Figure 115: Multiple approvers



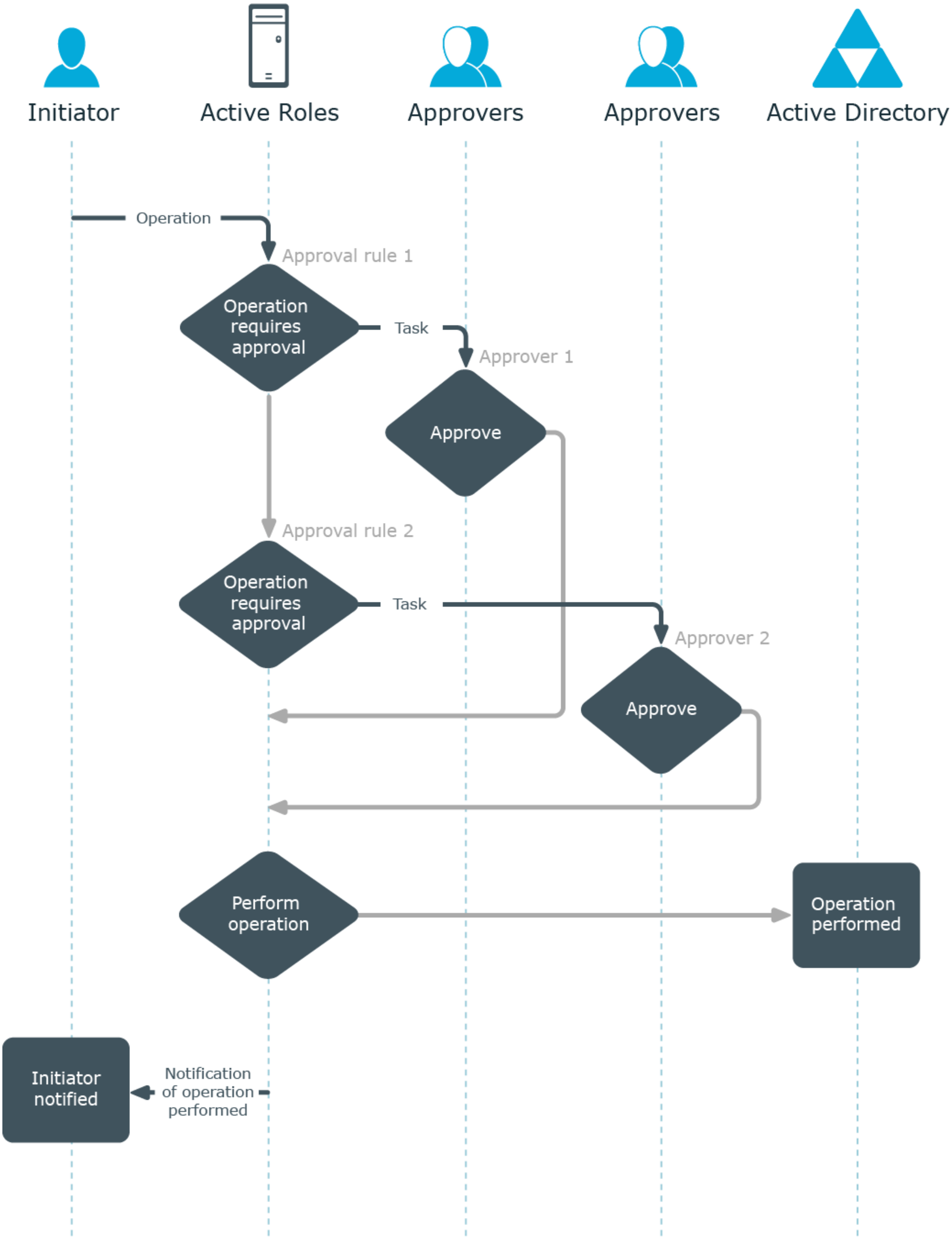
Multiple tasks

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 workflow 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.

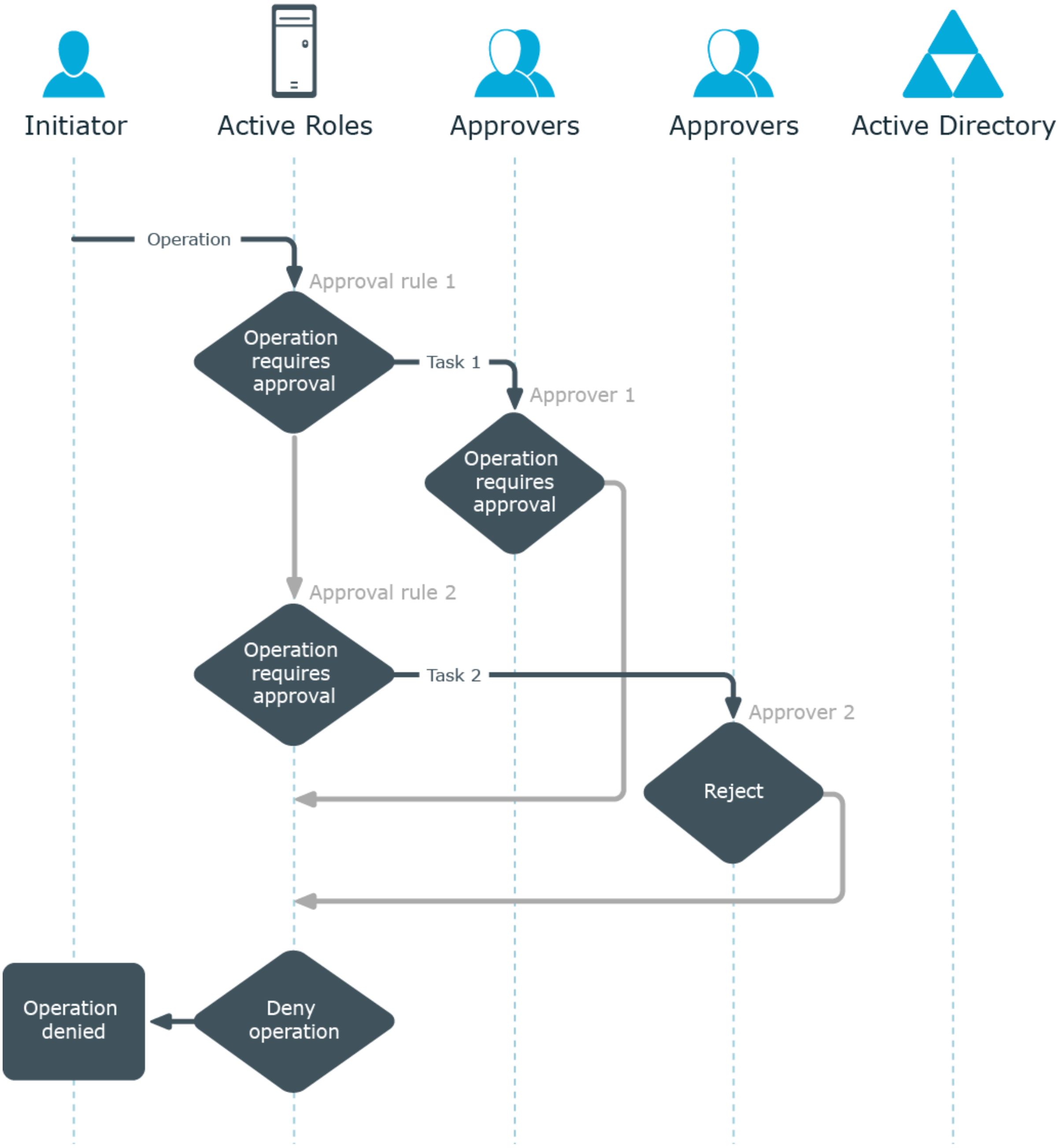
If each of the tasks receives the **Approve** action, Active Roles allows the operation to be performed.

Figure 116: Multiple tasks



If at least one of the tasks receives the **Reject** action, Active Roles cancels the operation.

Figure 117: 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 an object in that container or requests the 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 for 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 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. For more information, see Automation workflow.

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**.

Specifying workflow start conditions for an Approval workflow

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 more information, see Configuring workflow start conditions.

For 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 for an approval workflow

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.

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 more information, see Configuring an Approval activity.

Configuring notifications for an approval workflow

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.

Approval workflow 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.

Approval workflow 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 email server are specified as part of the notification settings on each approval rule.

Approval workflow notification message template

Notification messages are based on a message template that determines the format and contents of an email 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.

In the previous example, you could configure the approval activity so that the approver would receive an email 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 selected, and select the appropriate recipient(s) under **Notification recipients**.

For step-by-step instructions, see Configuring a Notification activity.

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 emails, 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 a version of Microsoft Exchange Server supported by Active Roles, and use a Microsoft-supported version of Outlook as their standard email 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 email 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 email messages, and can be approved or rejected directly from the notification email message, without having to use the Active Roles Web Interface. In every email 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.

For more information on the Microsoft Exchange Server versions supported by Active Roles, see *System requirements* in the *Active Roles Release Notes*.

Software and configuration requirements for Microsoft Outlook integration

Integration with Microsoft Office Outlook requires the following software and configuration prerequisites:

- **A supported Microsoft Exchange Server version**: Integration with Outlook requires at least one server running an Exchange Server version supported by Active Roles. The Exchange server must hold the Client Access server role and the Mailbox server role, to be deployed in your Exchange organization. For more information on the Microsoft Exchange Server versions supported by Active Roles, see *System requirements* in the *Active Roles Release Notes*.
- **A supported Microsoft Office Outlook version**: The approvers must use a Microsoft Outlook version that is still supported by Microsoft as their email 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.

- (Optional) **Approver mailboxes**: The mailboxes by which requests can be approved or rejected must be located on a mailbox server running an Exchange server version supported by Active Roles.

  > TIP: Although setting up approver mailboxes are not mandatory, One Identity highly recommends configuring them.

- **Active Roles mailbox**: A mailbox reserved for the exclusive use of Active Roles. This mailbox must be on a mailbox server running an Exchange server version supported by Active Roles.
- **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 the Active Roles mailbox.

Integration with non-Outlook email clients

For organizations that have deployed Microsoft Exchange Server, but use an email client application other than Outlook, 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 email 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 for non-Outlook integration

The ability to manage approvals from non-Outlook email clients calls for the same software and configuration prerequisites as Outlook integration (see Integration with Microsoft Outlook), with the following exceptions and additions:

- The email 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 email applications on mobile devices.
- Active Roles Add-in for Outlook does not need to be installed on the computer running the email 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 email configuration that is used by the approval rule (this is the default setting).

Email transport via Exchange Web Services

Active Roles can use Exchange Web Services (rather than an 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 email 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 has the following prerequisites:

- A supported Exchange Server installation, with Exchange Web Services deployed on it with the Client Access server role. For more information on the Microsoft Exchange Server versions supported by Active Roles, see *System requirements* in the *Active Roles Release Notes*.
- A dedicated mailbox hosted on Exchange Server. The mailbox must be reserved for the exclusive use of Active Roles.

Configuration settings for email transport

This section describes the available configuration settings with the Exchange Web Services option for email 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`

Authentication type

This setting specifies the authentication method of the Exchange Web Service.

> NOTE: Basic authentication is only available for on-premises Exchange Server services, Exchange Online mail resources should be configured with Modern authentication, as Microsoft does not support Basic authentication in Exchange Online mail resources.

Active Roles' mailbox credentials

This setting specifies the user name and password of the mailbox through which Active Roles will send and receive email. The mailbox must be located on a supported Exchange Server installation, and must be reserved for the exclusive use of Active Roles. For more information on the Microsoft Exchange Server versions supported by Active Roles, see *System requirements* in the *Active Roles Release Notes*.

> IMPORTANT: This mailbox must only be accessible by Active Roles. Providing access to any other application (for example, Microsoft Outlook) to process email messages in this mailbox can negatively impact the operation 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 email 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 email application. When an approver chooses one of the links provided in a notification message to approve or reject a request, the email application replies with an email 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 email application to open a Web Interface page for performing the approval task. The page may not open as expected if the email application does not support HTML format or an appropriate web browser does not exist on the device running the email application.

Configuring 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 Active Roles 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, 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. From the **Authentication type** drop-down, select the authentication method you want to use.

    > NOTE: Basic authentication is only available for on-premises Exchange Server services, Exchange Online mail resources should be configured with Modern authentication, as Microsoft does not support Basic authentication in

Exchange Online mail resources.

d. Under **Active Roles' mailbox credentials**:

   i. For an on-premises Exchange mailbox with Basic authentication, specify the user name and password of the mailbox through which Active Roles will send and receive email.

   ii. For a cloud Exchange Online mailbox or an on-premises Exchange mailbox with Modern authentication, specify the Azure user credentials of the Azure mailbox:

      - **Tenant ID**: The ID of the Azure tenant. To check the ID, on the Azure Portal, navigate to **Azure Active Directory** > **Overview**.
      - **Client ID**: The application client ID. To check the ID, on the Azure Portal, navigate to **App registrations** > **All applications** > **ActiveRoles**.
      - **Certificate thumbprint**: The most recent certificate thumbprint. To check the thumbprint, on the Azure Portal, navigate to **Certificates & secrets** > **Certificates**.
      - **Impersonated mailbox**: The mailbox that appears to be the sender of the email.

   This mailbox must be created on a server running a supported Exchange Server version, reserved for the exclusive use of Active Roles. For more information on the Microsoft Exchange Server versions supported by Active Roles, see *System requirements* in the *Active Roles Release Notes*.

e. Verify the settings you have configured. Click **Verify Settings**, supply a valid email address, and then click **Send**.

This causes Active Roles to send a diagnostic email 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 email** 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.

Automation workflow

An Active Roles "Workflow" is 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 also perform routine administrative tasks either on a scheduled basis or on user demand. In these cases, the 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 workflows 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 of the Active Roles Console.

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.

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 a new instance is started on demand, run it in parallel**
- **Do not start a new instance**

The first option allows starting a new instance of the workflow on demand, even if the workflow is already running. This option works only if the workflow is started on demand. If the workflow is performing a scheduled run, Active Roles allows only one instance of the workflow to run at a time.

“Run as” options for an automation workflow

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 runs 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 Active Roles, 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. You can override this setting on a per-activity basis.

Additional settings for an automation workflow

The additional settings specify whether to stop 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 run, causing an inconvenience to the user.

Parameters for an automation workflow

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 **View or change parameter value**. 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 **Edit**, 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 **Edit**, and then use the options in the **Parameter Definition** dialog.

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 more information about workflow parameters, see Configuring workflow parameters.

Initialization script for an automation workflow

When you configure an automation workflow, you can specify PowerShell commands you want the workflow run-time engine to run 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 more information, see Using the initialization script.

Using automation workflows

This section contains information and step-by-step instructions that explain how to use Active Roles 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 the run history of an automation workflow
- Stopping a running automation workflow
- Blocking an automation workflow from running
- Unblocking 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 **Workflow options and start conditions** to expand the area above the process diagram, and then click **Configure**.

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.
- 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.
- 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 for an automation workflow.
- 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 for an automation workflow.

- 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, navigate to the **Parameters** tab, select the parameter from the list, and then click **View or change parameter value**. For more information, see Parameters for an automation workflow.

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 to configure the activity.

The steps for configuring an activity depend upon the type of the activity. For more information on how to configure each activity type, see Configuring a workflow.

In the **Properties** dialog, 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 by right-clicking 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 can 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 Unblocking an automation workflow to run. For more information 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 running the workflow. The report displays the workflow run status along with information about the activities performed during it. For a workflow that is in progress you have the option to cancel running it by clicking **Terminate**.

Viewing the 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. For more information, see Workflow activity report sections.

To view the 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 the 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.

Stopping a running automation workflow

You can stop a running automation workflow to prevent it from completing its actions.

To stop a running automation workflow

- Click **Terminate** on the page that displays the automation workflow’s run history.

  For more information on how to access run history, see Viewing the 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 stop a running instance of an automation workflow, you may experience a delay (up to several minutes) before the workflow shuts down.

Stopping 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.

Blocking 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 Unblocking an automation workflow to run.

To block an automation workflow from running

1. In the Active Roles **Console tree**, navigate to **Configuration** > **Policies** > **Workflow**.
2. Right-click the automation workflow you want to block, then click **Disable Workflow**.

Unblocking an automation workflow to run

When an automation workflow is blocked, which prevents the workflow from running, you can unblock the workflow so that it can be run on demand or when it is scheduled to run.

To unblock an automation workflow to run

1. In the Active Roles **Console tree**, navigate to **Configuration** > **Policies** > **Workflow**.
2. Right-click the automation workflow you want to unblock, then 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 workflows:

- **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 the 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 the run history reports on the 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, 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 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.

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 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 Active Roles Console tree, right-click the desired container under **Configuration** > **Policies** > **Workflow**, and then click **Delegate Control.**
2. In the **Active Roles Security** dialog, 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 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.

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 Active Roles 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 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 workflows

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 Active Roles Console tree, right-click the desired Workflow container under **Configuration** > **Policies** > **Workflow**, then click **Delegate Control**.
2. In the **Active Roles Security** dialog, 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 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.

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 Active Roles 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 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 the run history of automation workflows

You can authorize users or groups to view the 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 Active Roles Console tree, right-click the desired container under **Configuration** > **Policies** > **Workflow**, and then click **Delegate Control**.
2. In the **Active Roles Security** dialog, 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 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.

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 Active Roles 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 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 convert the on-premises user to a hybrid Azure user with an Office 365 automation workflow and a hybrid migration script, based on the built-in **Sample Azure Hybrid Migration** script.

The Sample Azure Hybrid Migration script is available at the following location in the Active Roles Console:

Configuration > **Script Modules** > **Builtin** > **Sample Azure Hybrid Migration**

The remote mailbox workflow, on the other hand, is available at the following location in the Active Roles Console:

Configuration > **Policies** > **Workflow** > **Builtin** > **Sample Azure Hybrid Migration**

Prerequisites

To create remote mailboxes via hybrid migration with the Sample Azure Hybrid Migration script, your organization must meet the following requirements:

- To enable remote mailboxes, the Exchange management tools of an on-premises Microsoft Exchange installation must be available. For more information on the Microsoft Exchange Server versions supported by Active Roles, see *System requirements* in the *Active Roles Release Notes*.
- The Active Roles service account must be a part of the **Recipient Management** management role group to run Exchange hybrid commands.

To configure and run the Sample Azure Hybrid Migration script

1. Depending on whether you want to enable or disable remote mailboxes, use one of the following functions:
   - **EnableRemoteMailBox**: Use this function to enable remote mailboxes for the users in the workflow scope. Select **EnterExchangeCreds_params** as the function to declare parameters in the script, then provide the Exchange user name and password to run the **EnableRemoteMailBox** function in workflow.

    - **DisableRemoteMailBox**: Use this function to disable remote mailboxes for the users in the workflow scope. Select **EnterExchangeCreds_params** as the function to declare parameters in the script, then provide the Exchange user name, password and Exchange recipient type details to run the **DisableRemoteMailBox** function in workflow.

2. Make sure to specify the Fully Qualified Domain Name (FQDN) of the Exchange Server in the script, and that you modify the required code blocks. For more information on declaring script parameters, see Script activity.
3. After the script is modified, enable or copy the default **Sample Azure Hybrid Migration** workflow and run it.

By default, a remote mailbox is created for users with a valid Exchange Online license and who have no on-premises Exchange mailbox. For more information on creating a remote mailbox for new users, see *Creating a new hybrid Azure user with the Active Roles Web Interface* in the *Active Roles Web Interface User Guide*.

> NOTE: One Identity provides the `Remote mailbox migration (RemoteMailbox.ps1)` script as a sample script to illustrate the required steps of creating remote mailboxes.
>
> Do not use the script in a production environment without the required modifications and enhancements. Using security credentials within a script in clear text is never secure. When testing the script, consider the appropriate authentication and use of credentials. After testing, do not leave any credentials in clear text in the script.
>
> For more information, see Knowledge Base Article 310525.

Managing remote mailboxes

After creating a remote mailbox, you can manage it through the Active Roles Console and the Web Interface. Console supports the following administration actions:

- **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 email addresses
    - View or change the default reply address for each address type

  - View or change the external email address
  - Set the option to update email addresses based on email address policy
- **Mail flow Settings**
  - View or change message size restrictions and message delivery restrictions

For more information on Exchange Online properties, see *Viewing or modifying the Exchange Online properties of a remote mailbox* in the *Active Roles Web Interface User Guide*.

Microsoft 365 automation workflow

To import Azure or M365 Windows PowerShell modules, and run their corresponding M365 services within existing Active Roles workflows, configure M365 automation workflows. These workflows support running scripts from the following Windows PowerShell modules:

- Az.Accounts
- Az.Resources
- Exchange Online Management
- Microsoft.Graph

Creating a new M365 automation workflow has the following steps:

1. In the **Configuration** > **Script Modules** node of the Active Roles Console (also known as the MMC Interface), create the new M365 script that you want to run with the new M365 automation workflow.
2. In the **New Workflow** wizard, configure the new M365 automation workflow.
3. With the **O365 script execution configuration** activity of the Workflow Designer, specify the Azure tenant to which the configured workflow will apply.
4. Import the new M365 script into the workflow created in the first step.

> NOTE: By default, Active Roles does not select any Azure tenants automatically after you configured a new workflow with the **New Workflow** wizard. After the workflow is created, configure one in the Workflow Editor, otherwise the workflow will fail with the following error message:
>
> ```
> Select a configured Azure tenant from the Select a Tenant to configure O365
> Services drop-down list. Alternatively, under Parameter values, provide a
> valid Tenant ID, Tenant Name, Application (Client) ID and Application (Client)
> Certificate Thumbprint to override Azure tenant details from the workflow.
> ```

For more information on how to configure an M365 automation workflow, see Creating a Microsoft 365 automation workflow. For a list of sample M365 workflow scripts, see Sample Office 365 workflow scripts.

Creating a Microsoft 365 automation workflow

To import Azure or Microsoft 365 Windows PowerShell modules, and run their scripts within existing Active Roles workflows, configure a Microsoft 365 (M365) automation workflow.

Prerequisites

Before starting the configuration of an M365 automation workflow, make sure that the following conditions are met:

1. The following Windows PowerShell modules are installed on the system running Active Roles:
   - Az.Accounts
   - Az.Resources
   - Exchange Online Management
   - Microsoft.Graph

   If these PowerShell modules are not installed, Active Roles cannot run workflows that include M365 PowerShell script execution activities.

   > NOTE: Consider the following when planning to use the Exchange Online Management module:
   >
   > - To run a Sample Azure Hybrid Migration script, an on-premises Microsoft Exchange deployment must be available.
   > - As Exchange Online is connected to Exchange Online PowerShell, make sure that the `https://outlook.office365.com/powershell-liveid/` URL is not blocked in your organization domain, and that network connectivity is available.

2. You already created the M365 script module to use as a script activity with the M365 automation workflow. For more information, see Script activity.

To create a Microsoft 365 automation workflow

1. In the Active Roles Console (also known as the MMC Interface), expand **Configuration** > **Policies**.
2. To launch the **New Workflow** wizard, right-click **Workflow**, and select **New** > **Workflow** in the context menu.
3. On the **Name and Description** page, enter a **Name** and optionally, a **Description** for the new workflow.
4. On the **Workflow Type** page, under **This workflow is intended to start**, select **On user demand or on a scheduled basis (automation workflow)**.
5. On the **Completion** page, click **Finish**.
6. To configure the Azure tenant connection settings of the new M365 automation workflow, double-click the workflow to open the Workflow Designer, then click **Basic Activities** > **O365 script execution configuration**.

7. Specify the Azure tenant with one of the available methods:

    - Under **Select a Tenant to configure O365 Services**, select the Azure tenant you want to use with the automation workflow. This setting lists all Azure tenants that are configured in the Active Roles Configuration Center, as described in Configuring a new Azure tenant and consenting Active Roles as an Azure application.
    - Alternatively, to provide the Azure tenant connection details manually, click the parameters under **Parameter values**, and specify the **Tenant ID**, **Tenant Name**, **Application (Client) ID**, and **Application (Client) Certificate Thumbprint** of the Azure tenant as they appear on the Azure portal.

        > NOTE: Providing the Azure tenant details manually overrides the selection of the **Select a Tenant to configure O365 Services** drop-down list.

    To apply your changes, click **OK**.
8. To specify the M365 script to use in the workflow, click **Basic Activities** > **Script**.
9. In the **Script Activity** window, in the **General** tab, specify the **Name** (and optionally, the **Description**) of the M365 script.
10. To select the M365 script to use in the automation workflow, click **Script to use** > **Browse**, then select your M365 script in the **Script Modules** tree.
11. To apply your changes, click **OK**.

> NOTE: The configured workflow will run successfully only if the specified script is well-formed and complete.

Sample Office 365 workflow scripts

This section contains Microsoft 365 (M365) workflow script samples for reference.

$context.O365ImportModules(@(array-of-modules))

The `O365ImportModules` function lets you load an array of Azure and M365 Windows PowerShell modules. The function supports loading the following modules:

- Az.Accounts
- Az.Resources
- Exchange Online Management
- Microsoft.Graph

Once the modules are loaded, the function creates a connection to the specified modules with the connection details specified in the **O365 script execution configuration** workflow activity. For more information, see Creating a Microsoft 365 automation workflow.

Example: Importing all supported Azure and M365 Windows PowerShell modules

In this example, the `O365ImportModules` function is used to import all Windows PowerShell modules that M365 automation workflows support. After that, one command is invoked for each imported PowerShell module, respectively.

```
function Microsoft365ScriptTest() {

       $context.O365ImportModules(@("Az.Accounts", "Az.Resources",
"ExchangeOnlineManagement", "Microsoft.Graph"))

       $context.O365ExecuteScriptCmd("Get-Module | Select-Object -
Property ModuleType,Version,Name | Out-File -FilePath
C:\WS\Files\ImportedModulesInnerRunspace.txt")

       Get-AzContext | ConvertTo-Json | Out-File -FilePath
C:\WS\Files\Az.txt

       Get-EXOMailbox -Identity user | ConvertTo-Json | Out-File -
FilePath C:\WS\Files\ExchangeOnlineManagement.txt

       Get-MgUser -UserId "e38349d9-978a-4e4c-809b-189b68fe713a" |
ConvertTo-Json | Out-File -FilePath C:\WS\Files\Microsoft.Graph.txt

       Get-Module | Select-Object -Property ModuleType,Version,Name |
Out-File -FilePath C:\WS\Files\ImportedModulesOuterRunspace.txt

 }
```

$context.O365ImportModule (module)

The `O365ImportModule` function lets you load a single M365 or Azure Windows PowerShell module. If you have multiple versions of the specified module installed, you can also specify the module version to load.

> NOTE: The `O365ImportModule` function supports specifying major module versions only (such as version `2.x`).

Example: Importing the Azure Az PowerShell module

In this example, the `O365ImportModule` function is used to import version 2.x of the Microsoft Azure Az Windows PowerShell module.

```
function  TestImportTeamsModule()  {
     $context.O365ImportModule("AzureAz",  2)
 }
```

$context.O365ExecuteScriptCmd(string-or-cmd)

The `O365ExecuteScriptCmd` function passes any string or command specified in the script, then runs and returns the results as a PSObject.

$context.O365RemoveAllModulesSessions()

The `O365RemoveAllModulesSessions` script disconnects all PSSessions and removes all modules from the PowerShell pool, allowing Active Roles to import new modules again.

Example: Removing all Windows PowerShell module sessions

In this example, the `O365RemoveAllModulesSessions` function is used to disconnect the PSSession related to a previously loaded Azure Az module, and then remove the Azure Az module from the PowerShell pool.

```
#Get a list of disabled users and Directory Roles available
        $_usersinroles= @()
        $_default_log = "C:\temp\Roles.csv"
        $context.O365ImportModule("Microsoft.Graph", 1)
        $context.O365ExecuteScriptCmd("Get-MgUser -filter 'accountEnabled
eq false'" +" | Export-Csv " +"c:\temp\DisabledUsers.csv" +" -
NoTypeInformation")
        $context.O365ExecuteScriptCmd("Get-MgDirectoryRole | Export-csv
"+$_default_log )
        $context.O365RemoveAllModulesSessions()
```

Creating Office 365 shared mailboxes

To create new Office 365 shared mailboxes, use the **Create Office 365 Shared Mailboxes** built-in workflow. This workflow uses two other built-in resources:

- The **O365 script execution configuration** activity.
- The **Create Office 365 Shared Mailboxes** script.

By default, the **Create Office 365 Shared Mailboxes** workflow is disabled, as One Identity recommends using it as a template for custom workflows that uses the required values in the script, such as **Mailbox name**, **Mailbox display name**, **Alias**, and recipients to grant the **Send As** permission.

The **Create Office 365 Shared Mailboxes** workflow is located in the **Configuration** > **Policies** > **Workflow** > **Builtin** container of the Active Roles Console (also known as the MMC interface). The required **Create Office 365 Shared Mailboxes** script is located in the **Configuration** > **Policies** > **Script Modules** > **Builtin** container.

Enabling Azure Roles

To enable an existing directory role in Azure AD, use the **Enabling Azure Roles** built-in workflow. This workflow uses two other built-in resources:

- The **O365 script execution configuration** activity.
- The **Enabling Azure Roles** script.

By default, the **Enabling Azure Roles** workflow is disabled, as One Identity recommends using it as a template for custom workflows that would use the required values in the script, such as the directory role display name.

The **Enabling Azure Roles** workflow is located in the **Configuration** > **Policies** > **Workflow** > **Builtin** container of the Active Roles Console (also known as the MMC interface). The required **Enabling Azure Roles** script is located in the **Configuration** > **Policies** > **Script Modules** > **Builtin** container.

Activity extensions

In Active Roles, administrators can configure workflow activities of the predefined types that are installed with Active Roles. By default, the list of activities in the Workflow Designer contains only the predefined 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 for activity extension

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 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 is 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.

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.

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 50: 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.
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**, navigate to 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**, navigate to 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**, navigate to 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**,	Changing this setting changes the list of the activity parameters specific to this activity type. The changes do not affect the parameters of the

To change	Do this	Commentary
	navigate to the **Script** tab, and then choose the appropriate function from the **Function to declare parameters** list.	existing activities of this type. When you add a new activity of this type, the list of the activity parameters is built using the new function to declare parameters.
Policy Type icon	Right-click the object, click **Properties**, navigate to 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 on 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, 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**.

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, click **Add**.
3. In the **Select Objects** dialog, click **Temporal Membership Settings**.

4. In the **Temporal Membership Settings** dialog, 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, 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 for the group.

> NOTE: Consider the following when adding temporal members of a group:
>
> - To add temporal members of a group, you must be authorized to add or remove members from the group. You can get the appropriate authorization by applying the **Groups - Add/Remove Members** Access Template.
> - You can make an object a temporal member of particular groups by managing the object properties rather than the group properties. Open the **Properties** dialog for that object, and then, on the **Member Of** tab, click **Add**. In the **Select Objects** dialog, specify the temporal membership settings and enter the names of the groups according to your needs.

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, then click **Properties**.
2. Examine the list on the **Members** tab in the **Properties** dialog:
    - 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 group, then click **Properties**.
2. Examine the list on the **Member Of** tab in the **Properties** dialog:
    - 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 a specific date and time or an indication that the object should be added to the group immediately.

The end time setting specifies when the object is to be removed from the group. This can be a 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, click the member and then click **Temporal Membership Settings**.
3. Use the **Temporal Membership Settings** dialog to view or modify the start or end time settings.

The **Temporal Membership Settings** dialog 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: Consider the following when rescheduling temporal group memberships:
>
> - 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 for that object, and then, on the **Member Of** tab, select the group for which you want to manage the start or end time setting of the object, 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, 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, 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 for that object, and then, on the **Member Of** tab, select the group from the list and click **Remove**.

9

Group Family

With 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 for that object, and then, on the **Member Of** tab, select the group for which you want to manage the start or end time setting of the object 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, 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 of Group Family

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 Group Family 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.

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 on 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. For more information, 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, click **Policies**, select the policy, and click **View/Edit**.
4. In the **Policy Properties** dialog that appears, click **Policy Settings**.

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

Creating a Group Family has two steps:

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.

> NOTE: 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.
>
> Groups created through Group Family does not support group name with special characters, such as, `/\[]:;|=*?<>"`.

To create the Group Family configuration and run the Group Family

1. To start the **New Group Family Wizard**, 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**.
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, then click **Next**:
   - To use a preconfigured grouping criterion, click **Pre-configured grouping by**, then select a criterion from the list.
   - To configure a custom grouping criterion, click **Custom Grouping**.
5. On the **Location of Managed Objects** page, do one of the following, then click **Next**:
   - To assemble objects into groups, click **Add**, then select a container that holds the objects.
   - To remove a selected container from the **Containers** list, click **Remove**.
6. On the **Selection of Managed Objects** page, configure the object type and/or filtering rule for group family membership, then click **Next**:
   - To choose an object by type, click one of the four topmost options. Alternatively, click **Other**, then click **Specify** to choose an object type from the

Object Types list.

- To filter objects, click **Filter**, and complete the **Filter** dialog.

When ready, to view the list of objects that meet your specified conditions, click **Preview**.

7. On the **Group-by Properties** page, Click **Add**, then select an object property from the **Object property** list. To continue, click **Next**.
8. On the **Capture Existing Groups Manually** page, select **Skip this step, without capturing groups manually**, then click **Next**.
9. On the **Group Naming Rule** page, configure the naming rule as applicable, and click **Next**:
   - To configure a value, click **Configure**, and complete the **Configure Value** dialog.
   - To fine-tune group naming with a specific rule, click **Fine-tune Naming Rule**, and complete the **Fine-tune Naming Rule** dialog.
10. On the **Group Type and Scope** page, configure the group settings as applicable, 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, 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**, then click **Select** to choose the OU.
12. On the **Exchange-related Settings** page, 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. To continue, click **Next**.
13. On the **Group Family Scheduling** page, configure the scheduling options as you need, then click **Next**.
    - If you want the Group Family to run once you completed the wizard, select **Run Group Family once after completing this page**.
    - If you want the Group Family to run on a scheduled basis, select **Schedule Group Family to run**, 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

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)**, enter a value for the selected property.
4. Click **Add** to add the filter condition that you just specified, to the **Conditions** list.
5. To add multiple filter conditions, repeat the previous steps.

To complete the Configure Value dialog

1. Click **Add**.
2. In the **Add Entry** dialog, do one of the following, 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**. 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

1. Select the check box and click **Configure** next to the naming property that you want to configure, then complete the **Configure Value** dialog by using the procedure outlined above.
2. Click **OK**.

Starting the New Group Family Wizard

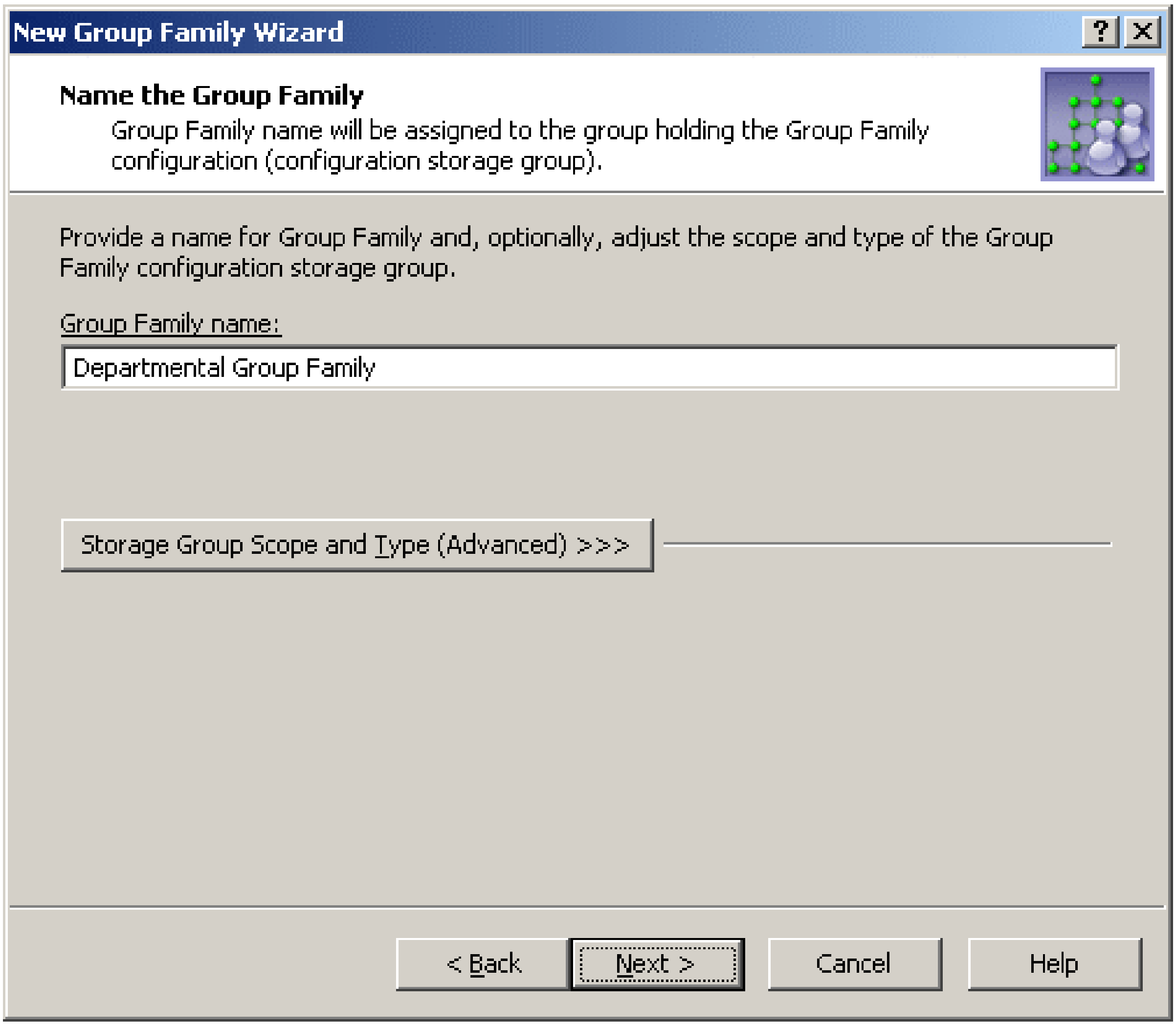
You can start the **New Group Family Wizard** in the Active Roles Console by using the **New** > **Group Family** command on the Organizational Unit in which you want to place the configuration storage group.

Naming 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 do not need to be changed.

Figure 118: Group Family name



Name the Group Family

Group Family name will be assigned to the group holding the Group Family configuration (configuration storage group).

Provide a name for Group Family and, optionally, adjust the scope and type of the Group Family configuration storage group.

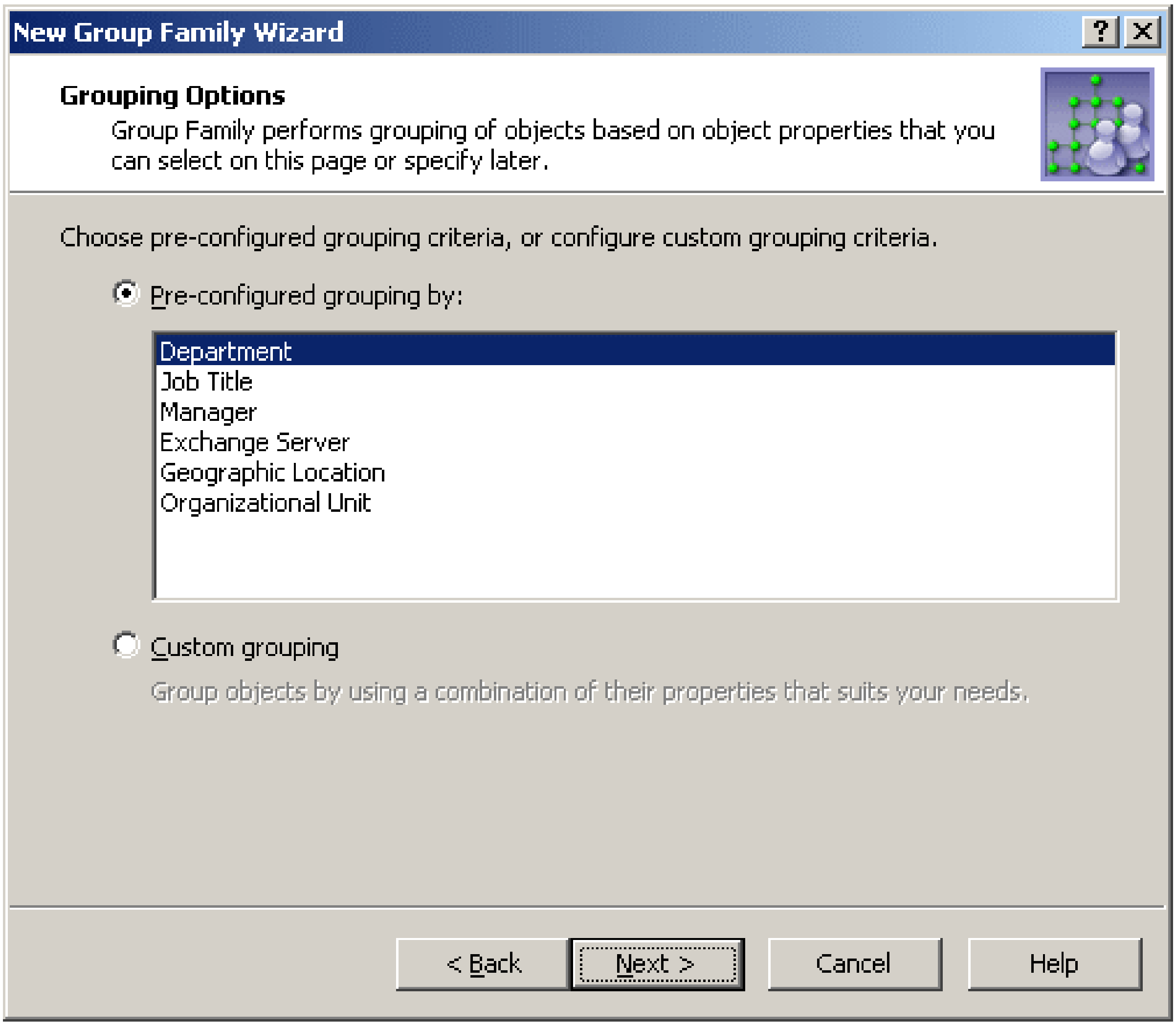
Group Family name: Departmental Group Family

Type in a Group Family name, 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 119: Grouping options



New Group Family Wizard

Grouping Options

Group Family performs grouping of objects based on object properties that you can select on this page or specify later.

Choose pre-configured grouping criteria, or configure custom grouping criteria.

● Pre-configured grouping by:

- Department
- Job Title
- Manager
- Exchange Server
- Geographic Location
- Organizational Unit

○ Custom grouping

Group objects by using a combination of their properties that suits your needs.

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.

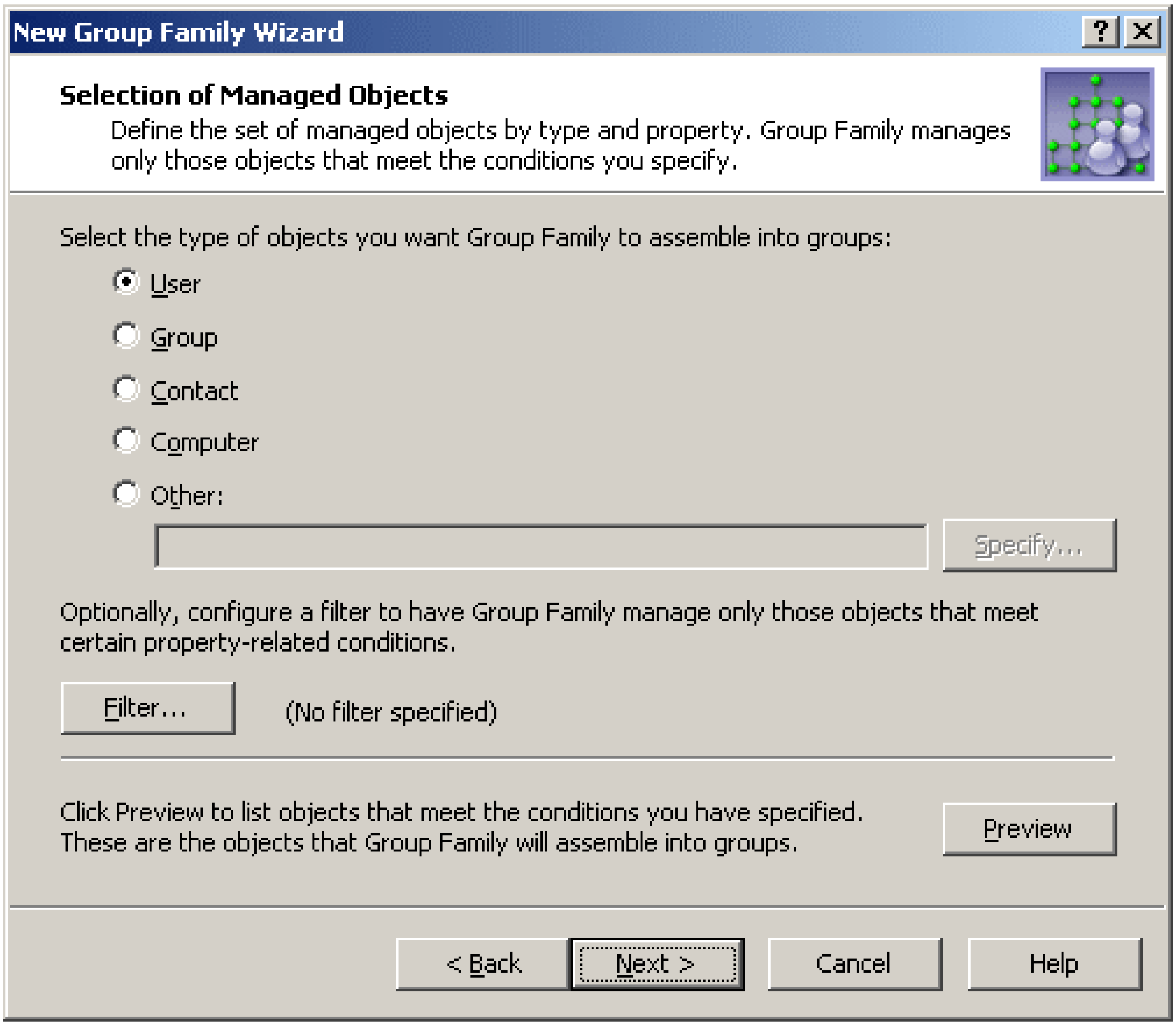
To remove containers from the list, select them and click **Remove**. This will cause the Group Family scope to no longer include the objects held in those containers.

To view or 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 120: Selection of managed objects



New Group Family Wizard

Selection of Managed Objects

Define the set of managed objects by type and property. Group Family manages only those objects that meet the conditions you specify.

Select the type of objects you want Group Family to assemble into groups:

● User

○ Group

○ Contact

○ Computer

○ Other:

Optionally, configure a filter to have Group Family manage only those objects that meet certain property-related conditions.

(No filter specified)

Click Preview to list objects that meet the conditions you have specified. These are the objects that Group Family will assemble into groups.

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: 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 51: 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.

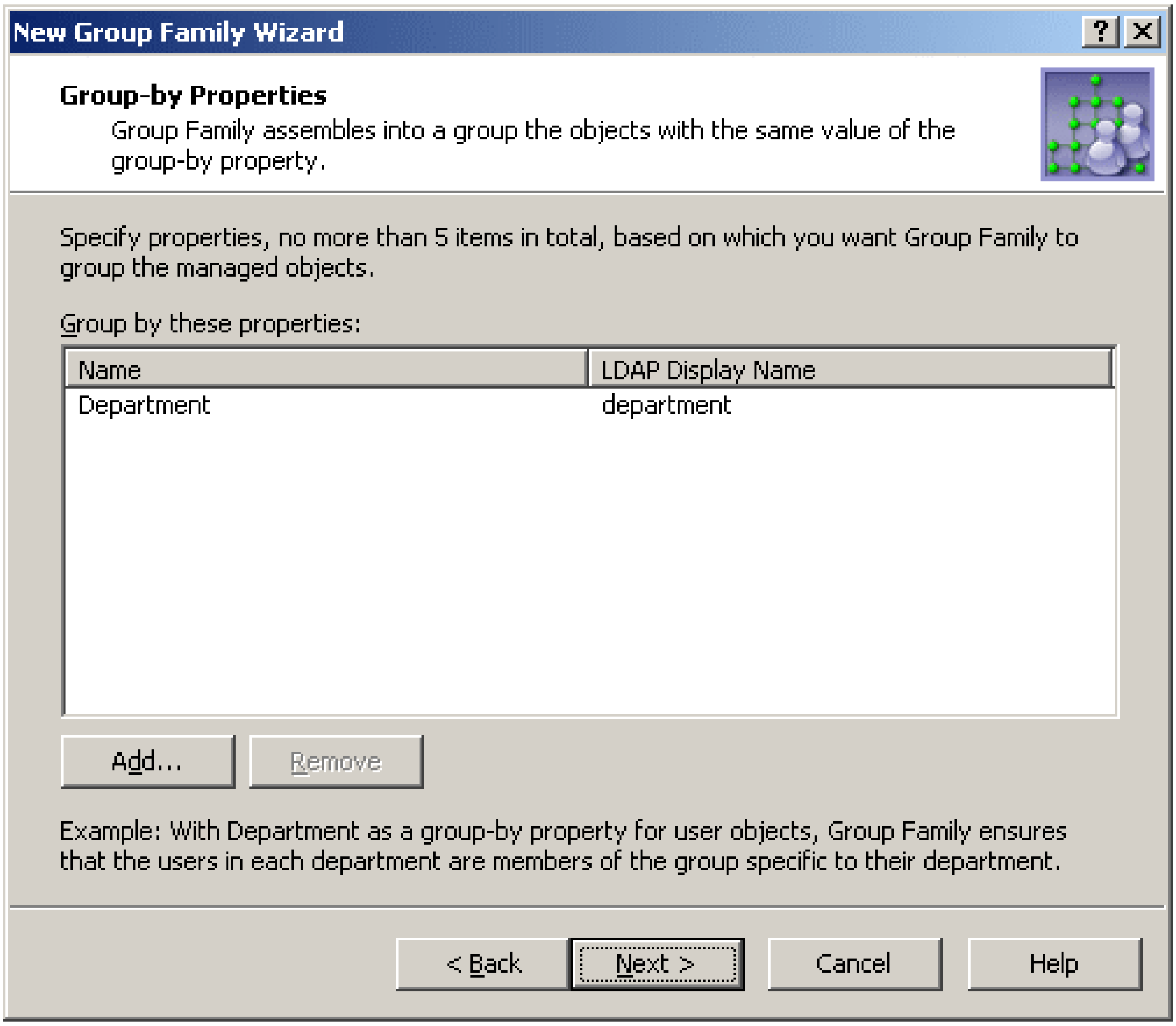
In addition, you have the option to configure an advanced filter by entering an appropriate LDAP query. To do so, click **Advanced** in the **Filter** window. Note that the basic and advanced filter options are mutually exclusive. If you have applied an advanced filter, the basic filter settings are disregarded. To return to the basic filter option, click **Basic** 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 121: Group by properties



New Group Family Wizard

Group-by Properties

Group Family assembles into a group the objects with the same value of the group-by property.

Specify properties, no more than 5 items in total, based on which you want Group Family to group the managed objects.

Group by these properties:

Name	LDAP Display Name
Department	department

Example: With Department as a group-by property for user objects, Group Family ensures that the users in each department are members of the group specific to their department.

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

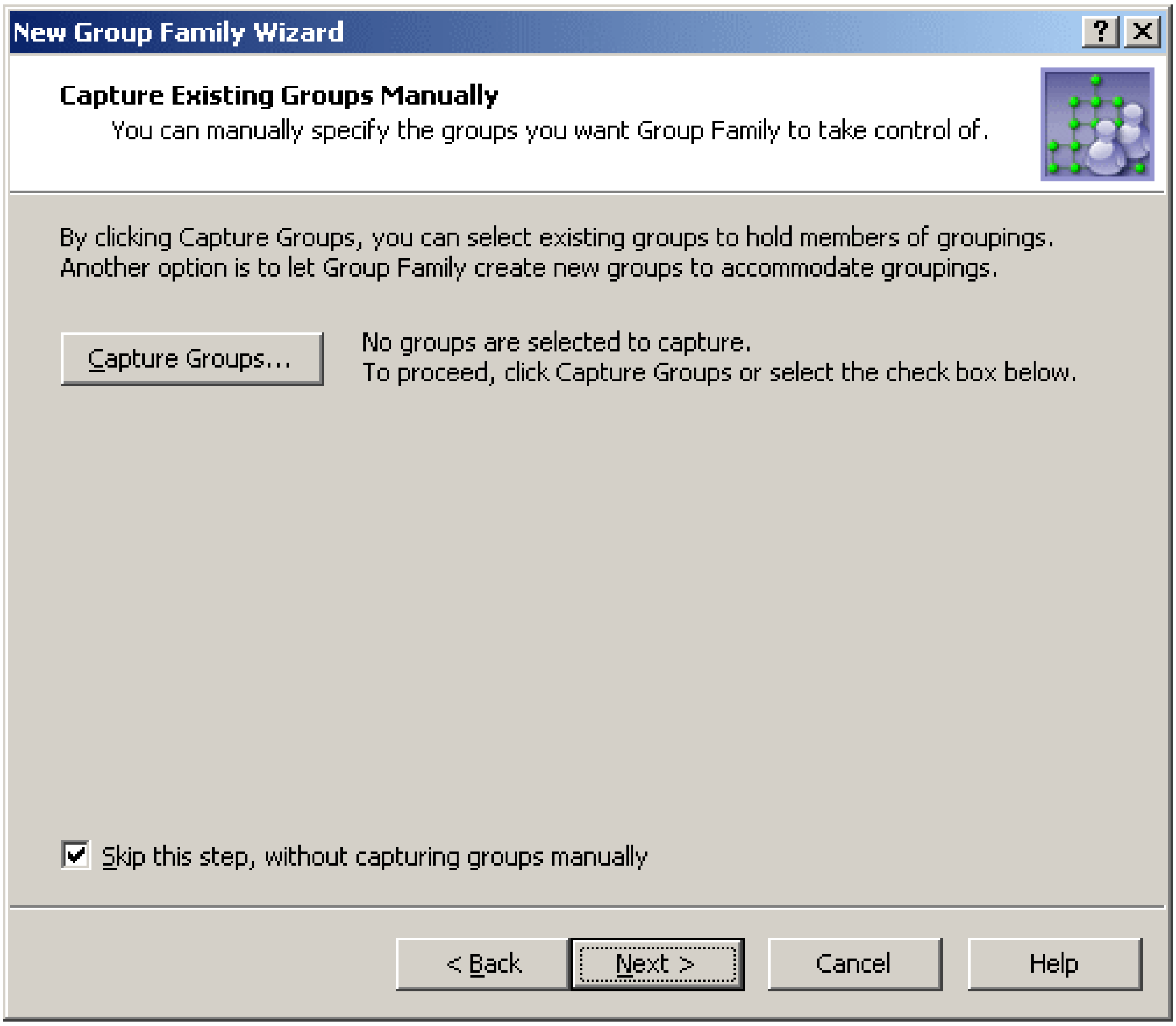
You can configure Group Family so that Active Roles will create 3 groups, each corresponding to one of the three keywords, and populate the groups as follows:

- 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 122: Capture existing groups manually



Capture Existing Groups Manually

You can manually specify the groups you want Group Family to take control of.

By clicking Capture Groups, you can select existing groups to hold members of groupings. Another option is to let Group Family create new groups to accommodate groupings.

No groups are selected to capture.
To proceed, click Capture Groups or select the check box below.

☑ Skip this step, without capturing 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:

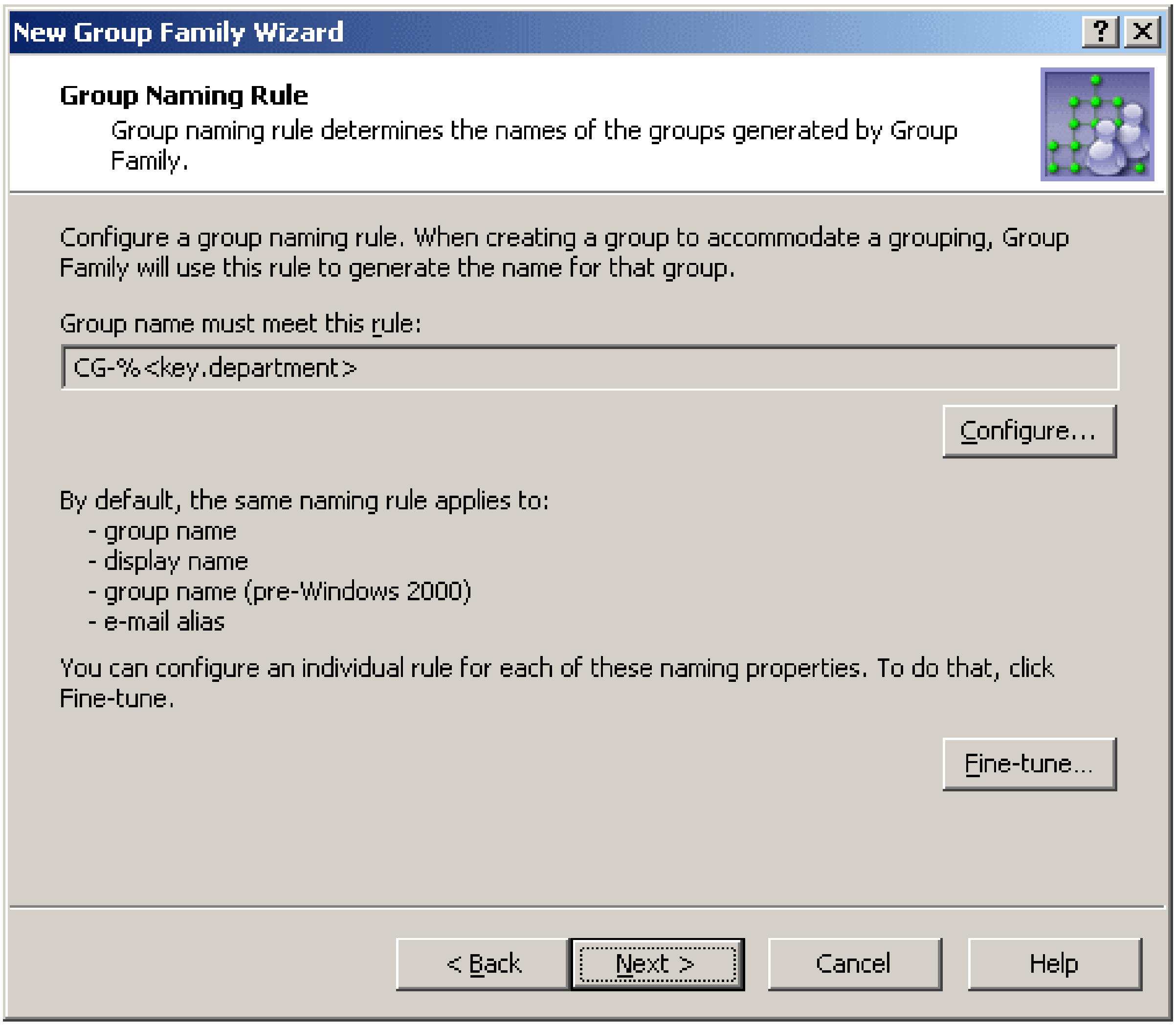
- **Add**: Opens a window where you can select a group and specify a grouping. To specify a grouping, you need to enter a certain value of each of the group-by properties. The result is that the group you select is linked to the grouping identified by the combination of values you have entered.
- **Edit**: 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, see Configuring a Property Generation and Validation policy.

Figure 123: Group naming rule



Group Naming Rule

Group naming rule determines the names of the groups generated by Group Family.

Configure a group naming rule. When creating a group to accommodate a grouping, Group Family will use this rule to generate the name for that group.

Group name must meet this rule:

CG-%<key.department>

By default, the same naming rule applies to:

- group name
- display name
- group name (pre-Windows 2000)
- e-mail alias

You can configure an individual rule for each of these naming properties. To do that, click Fine-tune.

By default, the Group Family generates the group naming properties based on the following syntax: `CG-%<key.property1>-%<key.property2>`, and so on. 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 **Configure**. This displays the **Configure Value** dialog. For more information, see Configuring a Property Generation and Validation policy. You can use that dialog 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 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:

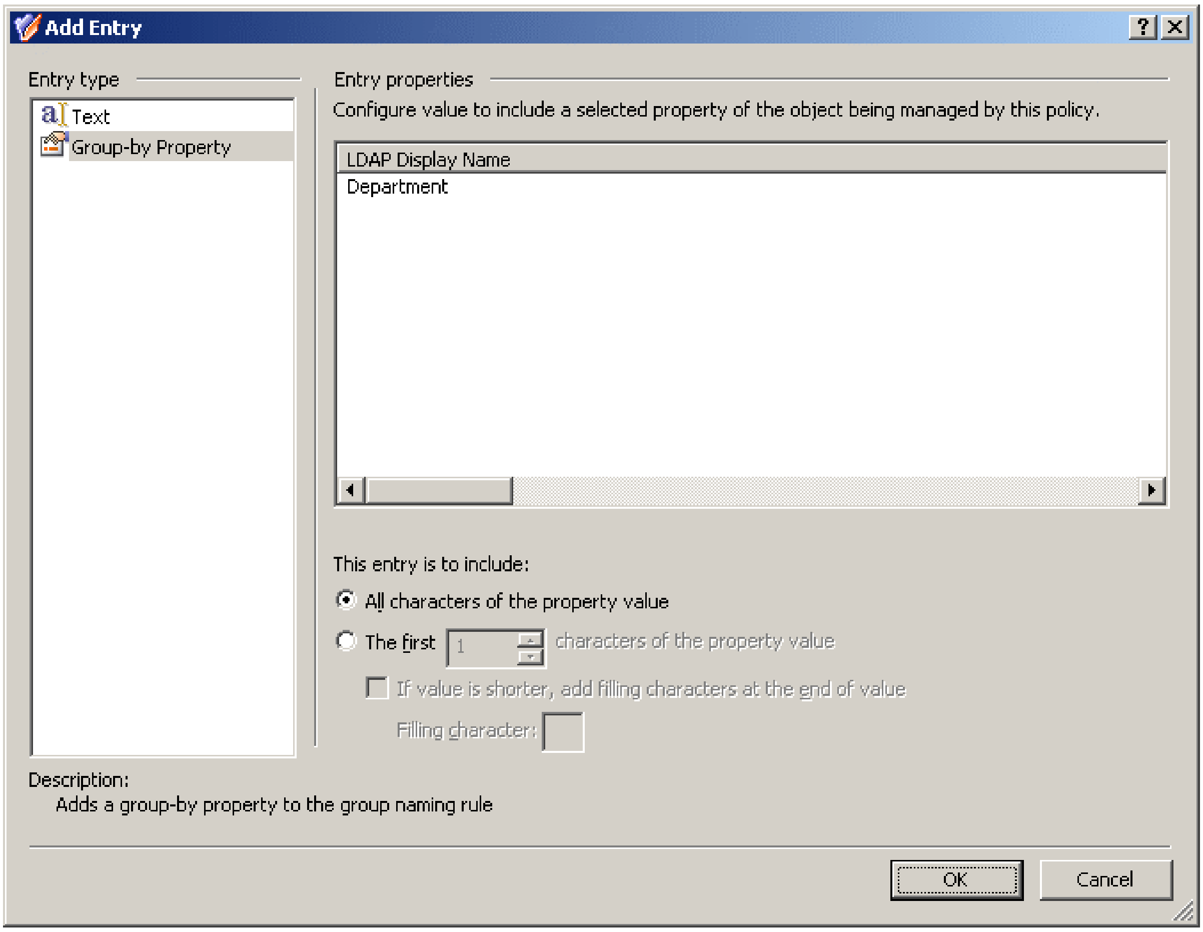
- **Text**: Adds a text string to the group naming rule.
- **Group-by Property**: Adds a group-by property or a part of a group-by property to the group naming rule.

To add a text string, you simply type a text in **Add Entry** window. The next subsection elaborates on the **Group-by Property** entry.

Group-by Property entry type

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 124: Group-by property



Add Entry

Entry type: Group-by Property

Entry properties

Configure value to include a selected property of the object being managed by this policy.

LDAP Display Name
Department

This entry is to include:

● All characters of the property value

○ The first 1 characters of the property value

□ If value is shorter, add filling characters at the end of value

Filling character:

Description:
Adds a group-by property to the group naming rule

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. When you have completed the list of entries, click **OK** to close that dialog.

NOTE: The naming rule must include an entry for each of the group-by properties.

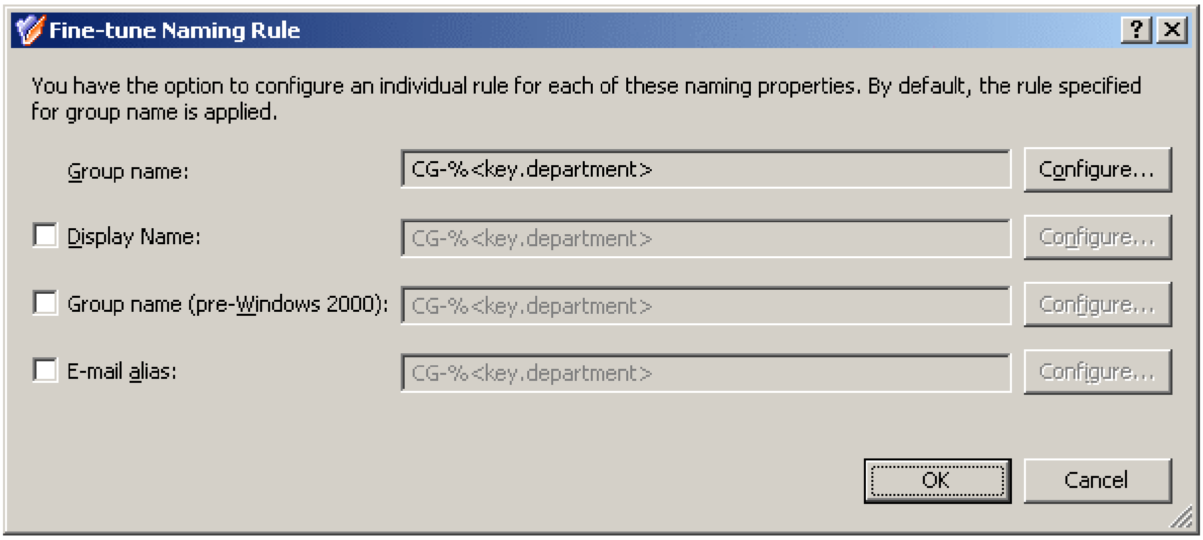
Separate rule for each naming property

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. For more information, see Exchange-related settings.)

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 125: Fine-tune naming rule



You have the option to configure an individual rule for each of these naming properties. By default, the rule specified for group name is applied.

Group name: CG-%<key.department> Configure...

□ Display Name: CG-%<key.department> Configure...

□ Group name (pre-Windows 2000): CG-%<key.department> Configure...

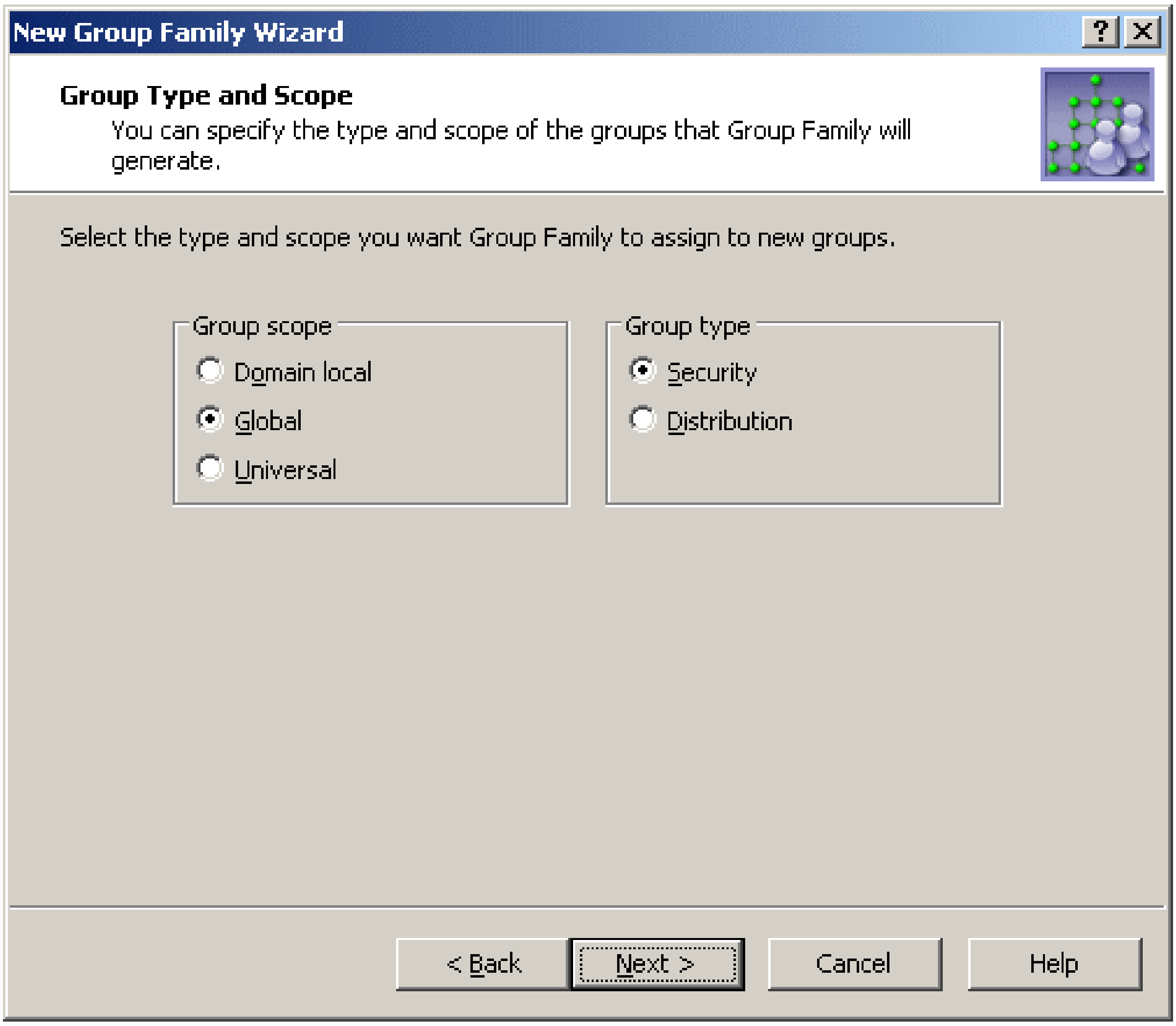
□ E-mail alias: CG-%<key.department> Configure...

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 126: Group type and scope



New Group Family Wizard

Group Type and Scope

You can specify the type and scope of the groups that Group Family will generate.

Select the type and scope you want Group Family to assign to new groups.

Group scope

○ Domain local

● Global

○ Universal

Group type

● Security

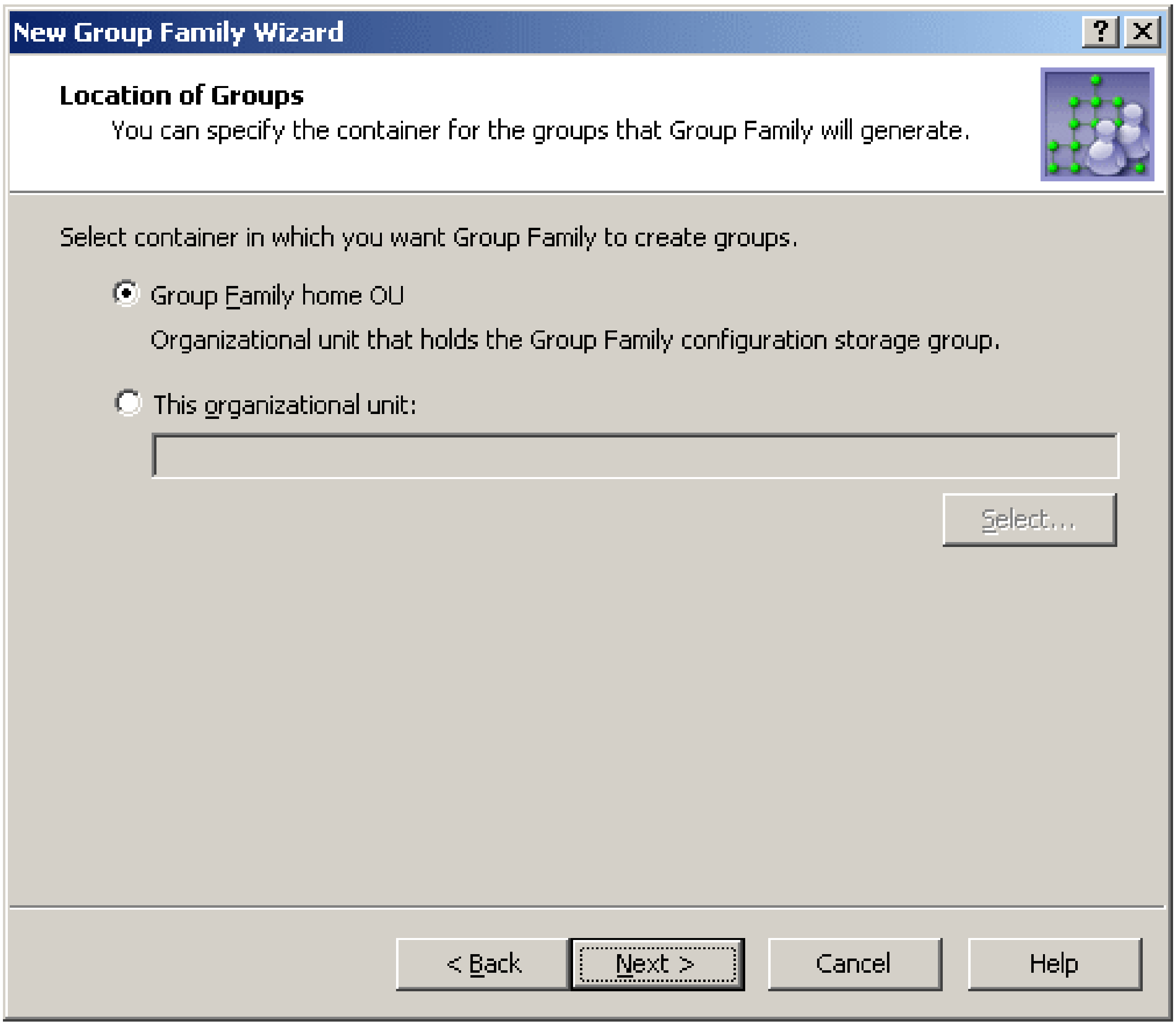
○ Distribution

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 127: Location of groups



New Group Family Wizard

Location of Groups

You can specify the container for the groups that Group Family will generate.

Select container in which you want Group Family to create groups.

● Group Family home OU

Organizational unit that holds the Group Family configuration storage group.

○ This organizational unit:

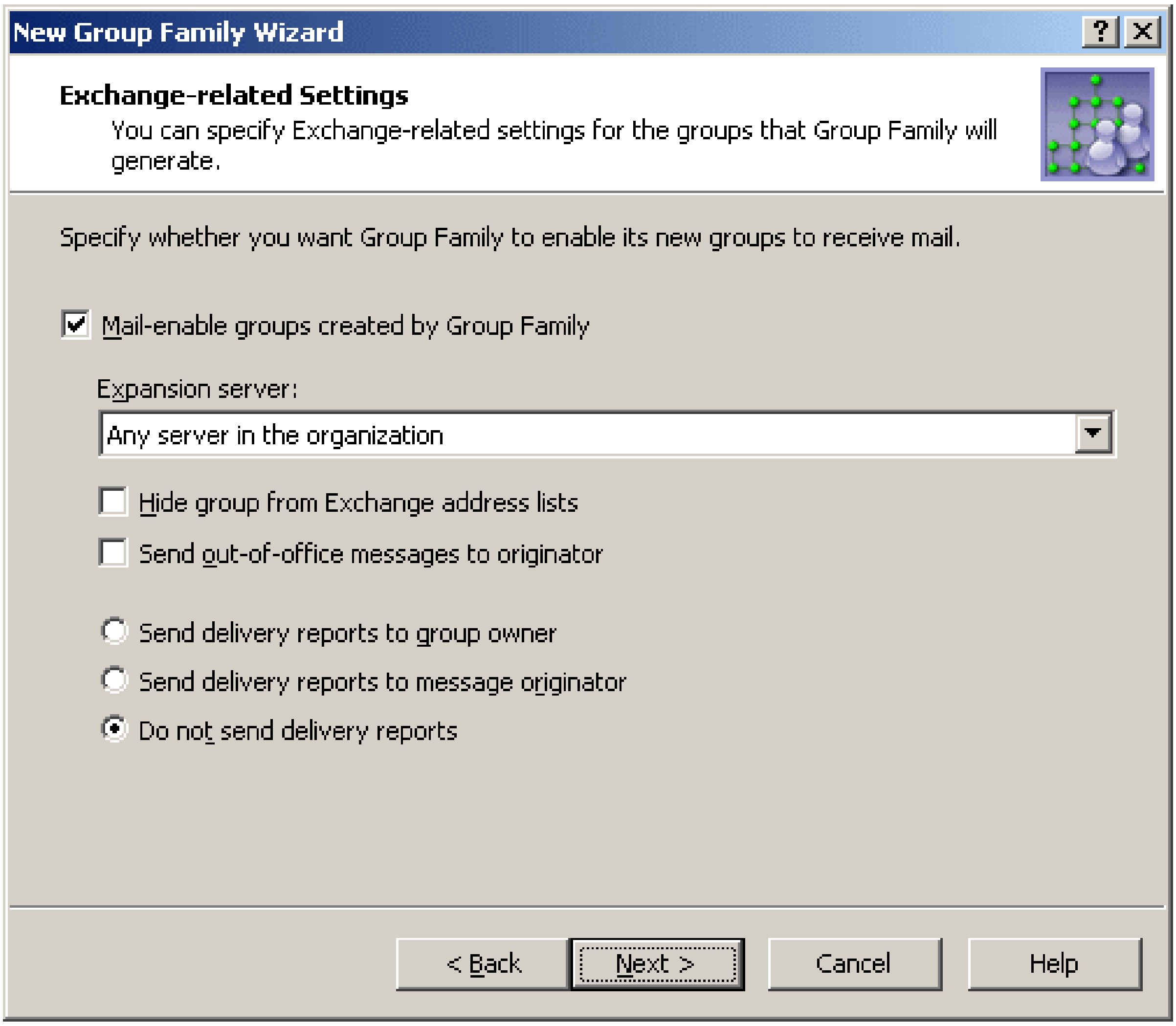
You can choose one of these options:

- **Group Family home OU**: The Group Family creates groups in the container that holds the configuration storage group for that Group Family. For more information, see Starting the New Group Family Wizard.
- **This Organizational Unit**: The Group Family creates groups in the specified container. 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 128: Exchange-related settings



Exchange-related Settings

You can specify Exchange-related settings for the groups that Group Family will generate.

Specify whether you want Group Family to enable its new groups to receive mail.

☑ Mail-enable groups created by Group Family

Expansion server: Any server in the organization

□ Hide group from Exchange address lists

□ Send out-of-office messages to originator

○ Send delivery reports to group owner

○ Send delivery reports to message originator

● Do not send delivery reports

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.

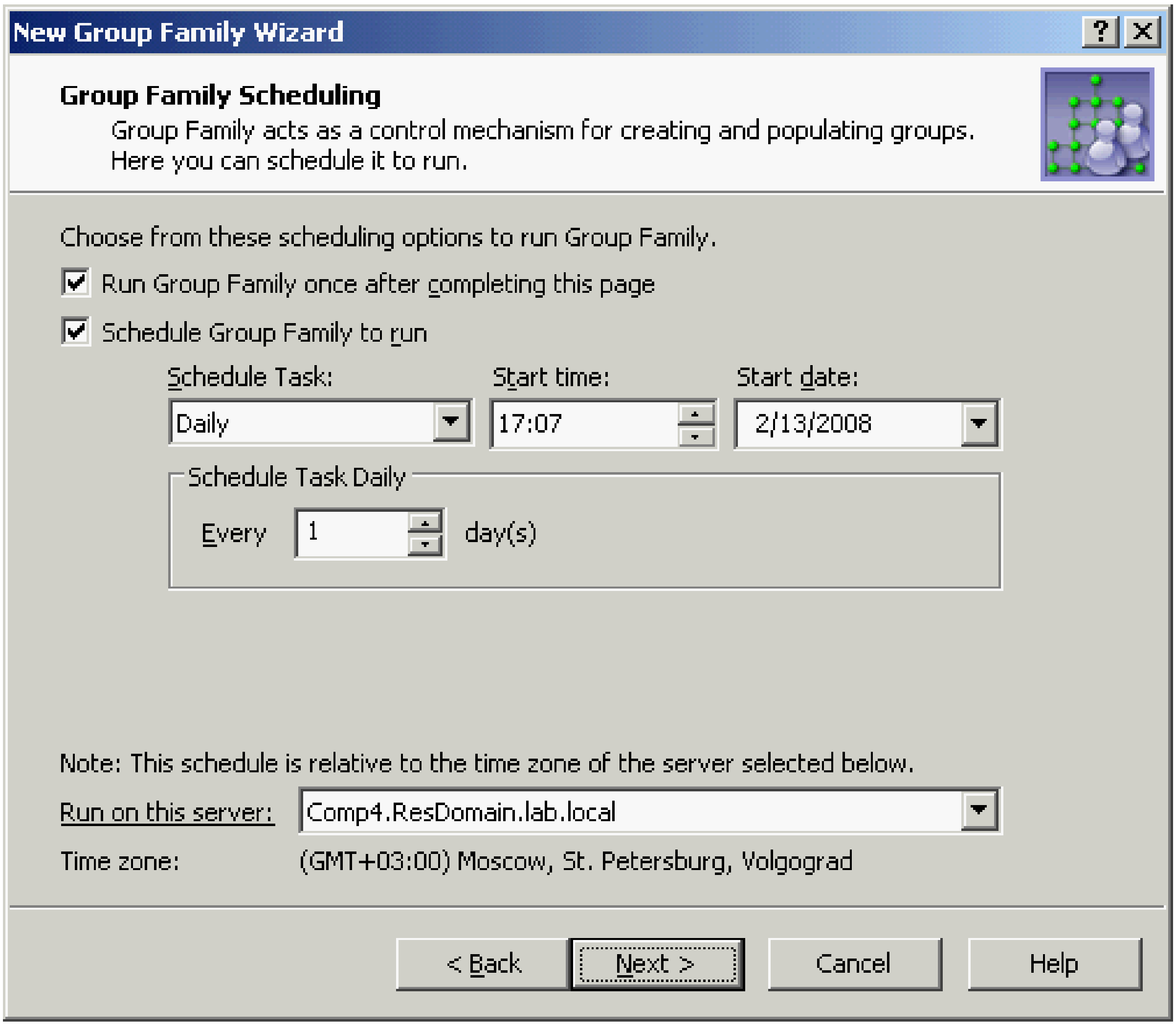
- **Send delivery reports to group owner**: Use this option if you want delivery reports to be sent to the group owner, when a message sent to a Group Family group is not delivered. This lets the group owner know that the message was not delivered.
- **Send delivery reports to message originator**: Use 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 Group Family works.

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 129: Group family scheduling



Group Family Scheduling

Group Family acts as a control mechanism for creating and populating groups.
Here you can schedule it to run.

Choose from these scheduling options to run Group Family.

☑ Run Group Family once after completing this page

☑ Schedule Group Family to run

Schedule Task: Daily

Start time: 17:07

Start date: 2/13/2008

Schedule Task Daily

Every 1 day(s)

Note: This schedule is relative to the time zone of the server selected below.

Run on this server: Comp4.ResDomain.lab.local

Time zone: (GMT+03:00) Moscow, St. Petersburg, Volgograd

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. For more information, see Administering Group Family.

Select the **Schedule Group Family to run** 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.

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.

When you create a Group Family, a group is created to store the Group Family configuration. The group is assigned the name you provided for the Group Family, and marked with the (**Group Family**) icon.

To facilitate Group Family administration, the **Properties** dialog 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.

> 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.

To view or modify grouping rules

1. Open the property sheet for the Group Family.
2. Click the **Groupings** tab, then click **Configure**.
3. Follow Steps 5 through 7 of the procedure for creating a Group Family. For more information, see 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.
2. Click the **Controlled Groups** tab, then click **Manage Rules**.
3. Follow Steps 9 through 12 of the procedure for creating a Group Family. For more information, see 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.
2. Click the **Controlled Groups** tab, then click **Capture Groups**.
3. In the **Capture Groups** window, click **Add**.
   a. In the **Assign Group to Grouping** dialog, do the following, then click **OK**:
   b. Click **Select**, 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.
2. Click the **Controlled Groups** tab, then click **Capture Groups**.
3. In the **Capture Groups** window, select the group you want to remove from the Group Family, click **Remove**, 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.
2. Click the **Schedule** tab, then click **Configure**.
3. On the **Group Family Scheduling** page, do the following, then click **Finish**:
   a. Select **Schedule Group Family to run**, 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.
2. Click the **Action Summary** tab, then click **View Log**.

To delete a Group Family

1. In the Active Roles Console, navigate to the Group Family you want to delete.
2. Right-click the Group Family configuration storage group, 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.

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 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 **Properties**.

There are two ways to access the **Properties** dialog of the Group Family configuration storage group:

- On the **Controlled By** tab in the **Properties** dialog 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 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 **Storage Group Scope and Type (Advanced)**, 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 52: 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 **Edit**, 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 **Manage Rules** on the **Controlled Groups** tab, in the **Properties** dialog 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 more information, see Group naming rule.
    - **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 more information, see Location of groups.
    - **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 more information, see Exchange-related settings.

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, and completes the task of managing the group creation rules. The changes are applied when you click **OK** or **Apply** in the **Properties** dialog, 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 **Configure**, 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 more information, see Location of managed objects.
- **Selection of Managed Objects**: The rules that determine what objects are to be managed by this Group Family. For more information, see Selection of managed objects.
- **Group-by Properties**: The list of properties based on which the Group Family calculates groupings. For more information, see Group-by properties.

  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. For more information, see Group Family scheduling. View or modify the schedule settings on that page, and click **Finish** to commit your changes to the **Properties** dialog. The changes are applied when you click **OK** or **Apply**, 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 **View Log**.

Action summary log

Clicking **View Log** 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.

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 present 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

Active Directory allows groups (herein called basic groups) to include members statically—select objects and add them to groups. Active Roles provides a flexible, rule-based mechanism for populating groups. Once set up, the process automatically adds and removes members from groups.

Active Roles provides rule-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 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:

- A rule-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 only be defined 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. During this conversion, the group only loses the membership rules.

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. As a result, 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. For more information, see Dynamic groups policy options.

Dynamic groups policy options

The behavior of dynamic groups is defined by the policy held in the built-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 Active Roles group membership lists are determined by membership rules.

To view or modify the policy, display the **Properties** dialog for the **Built-in Policy - Dynamic Groups** Policy Object (located in container **Configuration/Policies/Administration/Builtin**), navigate to the **Policies** tab, select the policy, and click **View/Edit**. This displays the **Policy Properties** dialog.

On the **Policy Settings** tab in the **Policy Properties** dialog, 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.

  > NOTE: Enabling cross-domain membership adds an increased load to the Dynamic Group and Group Family processing. If you want to enable cross-domain membership only on a small subset of groups, enable the virtual attribute **edsvaDGCrossDomainMembershipEnabled** on those groups.

- To enable the virtual attribute **edsvaDGCrossDomainMembershipEnabled** on a group, set its value to `TRUE`.
- To disable the virtual attribute (and cross-domain membership) on a group, either set the value to `FALSE` or clear the value.

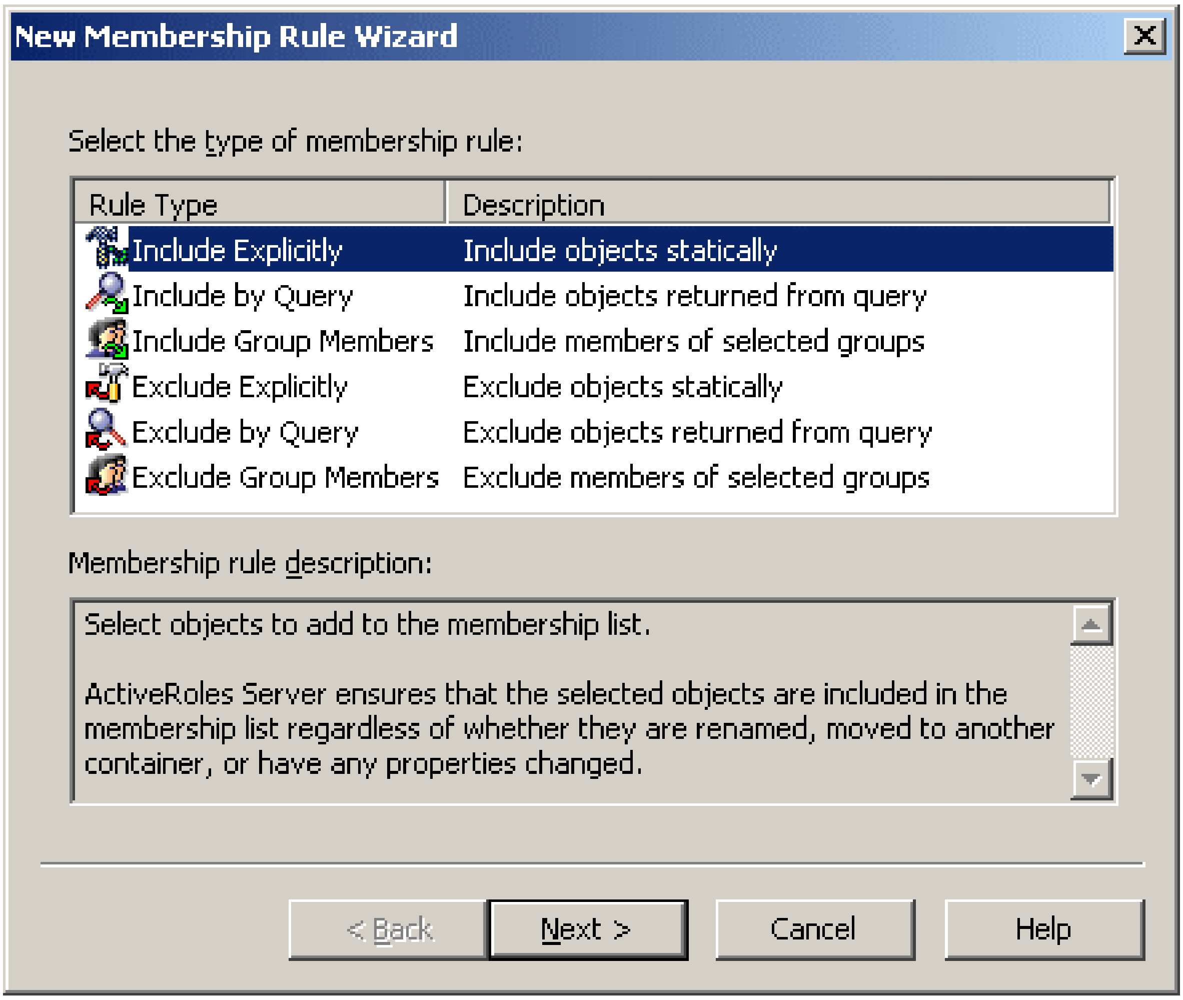
- **Receive directory changes from DirSync control**: Ensures that the policy correctly populates membership lists regardless of what tools are used to manage Active Directory. When this check box is not selected, some rule-based membership lists may be incompatible with membership rules. In this case, 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**: Adds the message text to the **Notes** property of every dynamic group. (The **Notes** property is displayed in the group's **Properties** > **General** tab.)

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 decide to clear this policy option later, the dynamic groups configured to include or exclude objects from external domains will no longer 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.

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 following figure illustrates the first page of the wizard.

Figure 130: Convert to dynamic group



Select the type of membership rule:

Rule Type	Description
Include Explicitly	Include objects statically
Include by Query	Include objects returned from query
Include Group Members	Include members of selected groups
Exclude Explicitly	Exclude objects statically
Exclude by Query	Exclude objects returned from query
Exclude Group Members	Exclude members of selected groups

Membership rule description:

Select objects to add to the membership list.

ActiveRoles Server ensures that the selected objects are included in the membership list regardless of whether they are renamed, moved to another container, or have any properties changed.

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 with the 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. The **Exclude Explicitly** rule takes precedence over all other types of rule. As a result, the selected objects will be excluded from the group even if a different rule states that they should be included.

The **Exclude by Query** membership rule allows you to define the criteria that 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).

By using the **Exclude Group Members** membership rule, you can select which groups' 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 the **Exclude Explicitly** rule type, you are presented with the **Select Objects** dialog that lists users, groups, contacts, and computers. Select the objects you want to include or exclude from the dynamic group, click **Add**, then click **OK**.

If you have selected the **Include Group Members** or **Exclude Group Members** rule type, the **Select Objects** dialog appears. The list of objects in that dialog 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, similar to the **Find** dialog, is displayed. In that dialog, 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 the properties of the group.

If you add several membership rules and some of them conflict with each other, 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. 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, 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. To add a membership rule to a dynamic group, you can also use the **Membership Rules** tab in the **Properties** dialog.

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**, 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, 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 appears. Otherwise, the **Select Objects** dialog appears.
5. Complete the **Create Membership Rule** or **Select Objects** dialog using the procedures outlined below in this section.
6. Click **Finish** to close the wizard.

To complete the Create Membership Rule dialog

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.
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. (Optional) To view a list of objects that match the criteria you defined, click **Preview Rule**.
5. Click **Add Rule**.

To complete the Select Objects dialog

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: Consider the following when adding a membership rule to a dynamic group:
>
> - 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 is similar to the **Find** dialog 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 directory objects* in the *Active Roles Console User Guide*.
> - 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 *Using advanced search options* and *Building a custom search* in the *Active Roles Console User Guide*.

Removing a membership rule from a dynamic group

To remove a membership rule from a dynamic group, open the **Properties** dialog 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.

> 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 includes the **Membership Rules** tab if the selected group is a dynamic group. If you do not see the **Membership Rules** tab, the selected group is a basic group.

Converting a dynamic group to a basic group

When converting a dynamic group to a basic group, only the membership rules 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.
- 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.

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.

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`**.

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**, then click **Next**.

On the second page, click **Add** to display the **Create Membership Rules** dialog. 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 the user accounts of the employees.
3. Click the **Advanced** tab.
4. Click **Field**, click **City**, then click **OK** in the **Select Object Property** dialog.
5. In the **Condition** list, click **Is (exactly)**.
6. In the **Value** box, type **`Seattle`**.
7. Click **Add**, then click **Add Rule**.

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

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 email 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 users' properties, groups and other directory objects, group membership lists, and the 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.

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.

Collector to prepare data for reports

The Active Roles Collector allows you to collect data from computers running the Administration Service and store them 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 are 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 **Active Roles** > **Active Roles Collector and Report Pack** on the **Apps** page or select **All Programs** > **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

You can use the Active Roles Collector to prepare data for reporting. The data you will prepare for reporting are stored in the database you specify. To make the data available for the report server, you have to configure the data source on the report server to connect to the database that stores the data.

This section describes how to prepare report data. For more information on how to configure the data source for the Active Roles Report Pack, see Configuring the data source.

To collect data from the network, start and complete the Collector wizard, and complete the wizard pages as follows. For more information, see Starting the Active Roles Collector wizard.

To configure data collection with the Active Roles Collector wizard

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. To initially specify a database, or choose a different database, click the button next to the **Database** box, then use the dialog that appears to specify the required database type, database, and authentication option for connection to database server.
- The computer running the Administration Service. To do so, in **Active Roles Service**, specify the full name of the computer running the Administration Service from which you want to collect information.
- The credentials to log in to that computer. To do so, under **Log on as**, click one of these options:
  - **Current user**: Connect to the Administration Service with the user account under which the Collector is running.
  - **Specified user**: 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 are 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.

   > 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.
   >
   > The wizard only displays the **Data to Collect** page if you select the **Active Directory** check box on the **Data Collection Tasks** page.

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 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 initialization immediately or schedule the task to run at a convenient time:

   - To start the collection process immediately, click **Now**, then click **Next**.
   - To schedule the task, select **On a schedule**, then click **Next**.

   TIP: You can also disable SID resolving for faster data collection.

7. If you selected the **On a schedule** option, on the **Schedule** page, specify the task schedule and login 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 configured. 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.

Once configured, 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 properties (such as task name, description, security options, triggers, conditions, and settings).

You can also view the task history with its properties. Task Scheduler tracks the task history by events that are raised when the task is started, run, finished running, and at other times as needed to track the task history. Errors related to the task are also tracked in the task 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 typing `Taskschd.msc` into a command prompt (for example, `cmd.exe`).

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 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

You can process the gathered events with the Collector wizard. For instructions on how to start the wizard, see Starting the Active Roles Collector wizard.

To process gathered events

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. Click **Next**.
4. On to the **Source database** page, click **Specify**, and enter 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.
5. Click **Next**.
6. On to the **Target Database** page, click **Specify** box, and enter 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.
7. 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.

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 more information, see Configuring the data source.

To import events from the database of an earlier Collector version, start the Collector wizard, and complete the wizard pages as follows. For more information, see Starting the Active Roles Collector wizard.

1. On the **Select Task** page, select the **Import events from an earlier database version** option.
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, and complete the wizard pages as follows. For more information, see Starting the Active Roles Collector wizard.

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`.

   > NOTE: 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 Configure Report Server URLs (SSRS Configuration Manager).
3. (Optional) On the **Data Source** page, configure the data source for the Active Roles reports:
   a. Click **Configure Data Source**.
   b. Use the **Configure Data Source** dialog 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 more information, see Configuring the data source.

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 more information, see Generating and viewing a report.

Working with reports

You use the Active Roles Collector to prepare data for reporting. The data are stored in the database you specify when configuring the data collection job. For more information, see Collector to prepare data for reports. 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. For more information, see Deploying reports to the Report Server. If you have not configured the data source, or need to change the data source, 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 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, 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:

- You have report data prepared, by using the Active Roles Collector.
- The data source on the report server is configured to connect to the database containing 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 **View Report** 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 subsections. Each subsection heading identifies the path to a certain report folder, with the reports contained in that folder being listed under the subsection 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 email.

- **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 due to 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 with passwords past their 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 with expired passwords), 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
  - 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 as the creation date, status (`enabled` | `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 the accounts that are disabled, expired, or not used for logon during a certain time period), or the accounts with expired passwords. Also allows you to examine each account in detail.

Active Directory Assessment/Potential Issues/

- **Cycled Groups**: Lists the Active Directory groups that are members of themselves (if any exists). You can use this report to determine if your Active Directory domain has any group configured to contain itself as a member (for example, group A is a member of group B, which in turn is a member of group A).

  > NOTE: Consider that having Cycled Groups in your 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, the name of the person who deprovisioned user accounts, and the 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:
  - Create
  - Modify (change properties)
  - Add to groups
  - Remove from groups

- Add in place of current group members
- 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 the events), by computer (where the 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, the name of the object to which the Access Template is linked, the time period when the link was created, and the 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, the name of the object to which the Policy Object is linked, the time period when the link was created, and the 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 the name of the person who approved or rejected requests (approver), the name of the person whose requests were subject to approval (initiator), the approval decision (approved or rejected requests), and the 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 the operation that started workflow or by the 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 due to group memberships.
- **Control delegation by Trustee**: Lists Active Directory security principals (Trustees) with 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 determining 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 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), the number of directory objects to which the Policy Object is linked (reference number), the total number of individual policies defined in the Policy Object (entry number), and the 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. Also, it 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. Also, it 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 that have policy rules 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

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 Service (specifically the instances that share a common database). As the Active Roles Console and the Web Interface automatically select 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 Service instances may not be present on other Administration Service instances. 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 discussed in Management History configuration.

Active Roles also includes reports to examine Management History by collecting and analyzing event log records. For more information, see Active Roles Reporting. However, the process of retrieving and consolidating records from the event log may be time-consuming and inefficient.

Management History 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 Service, 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.

IMPORTANT: Consider the following when migrating the Management History and Configuration databases:

- The Management History Migration Wizard is designed to perform one-to-one database migration during Active Roles updates, to speed up the upgrade process. This is because migrating the Management History database can take a long time, depending on the size of the deployment history and the particularities of the Active Roles environment.
- The Management History Migration Wizard was not tested to migrate and merge several Management History databases to a single database instance. Such scenarios are not supported.
- However, you can re-run the Management History Migration Wizard several times from the same source database. In such cases, the Wizard can merge the changes that occurred in the source database since the last import to the target database.
- Importing a Configuration database results in the source configuration replacing the target Configuration database, overwriting the current settings of the target system. As Active Roles stores its configuration data in the Configuration SQL database, One Identity strongly recommends backing up the target Configuration database before import.
- For more information on the supported upgrade paths of Active Roles 8.1.3, see *Upgrade and installation instructions* in the *Active Roles Release Notes*. For more information on supported upgrade paths in general, see Knowledge Base Article Active Roles upgrade paths in the One Identity support portal.

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 for the **Built-in Policy - Change Tracking** Policy Object (located in container **Configuration/Policies/Administration/Builtin**), navigate to the **Policies** tab, select the policy, and click **View/Edit**. This displays the **Policy Properties** dialog. The **Object Types and Properties** in that dialog 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 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 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 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. For more information, see Management History considerations and best practices.

> 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 Configuring replication, 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 Service (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):

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: The **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 Configuring replication: 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 Configuring replication. 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 Service 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. 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 Service 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.

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 Roles 8.1.3 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.

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.

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 **Details**, 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 ran 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 initiated; 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 report section

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 report section

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 report section

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 report section

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 report section

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 task was created>
- Status: <indicates if 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 report section

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 task was created>
- 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 if 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 report section

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 task was created>
- Status: <indicates if 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 report section

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 task was created>
- Status: <indicates if 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 report section

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 or group)
- Name: <name of the object>
- Moved to: <identifies the move destination container>
- Operation ID: <number>
- Requested: <date and time that the task was created>
- Status: <indicates if 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 report section

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 task was created>
- Status: <indicates if 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 report section

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, group or computer)
- Name: <name of the object>
- Operation ID: <number>
- Requested: <date and time that the task was created>
- Status: <indicates if 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 report section

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 task was created>
- Status: <indicates if 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 running 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.

Policy report sections

User Logon Name Generation policy

Table 53: User Logon Name Generation policy

Report Item (Success)	Report Item (Failure)
The user logon name (pre-Windows 2000) is set to `value`.	Not applicable

E-mail Alias Generation policy

Table 54: 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

Exchange Mailbox AutoProvisioning policy

Table 55: 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

Group Membership AutoProvisioning policy

Table 56: 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

Home Folder AutoProvisioning policy

Table 57: 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

Report Item (Success)	Report Item (Failure)
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 folder.	Unable to set user permissions on home folder name on the file server. 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 name on the file server.	Not applicable
Home folder name is renamed to name on the file server.	Unable to rename home folder name to 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

Property Generation and Validation policy

Table 58: 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

Running policy script <name>

Table 59: 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 email 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.

Active Roles internal policy report sections

Creating user mailbox

Table 60: 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`. 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

Creating linked mailbox

Table 61: 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

Creating equipment mailbox

Table 62: 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
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 63: 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 descriptions

Creating shared mailbox

Table 64: 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

Report Item (Success)	Report Item (Failure)
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

Moving mailbox

Table 65: Moving mailbox

Report Item (Success)	Report Item (Failure)
The following items apply to the mailbox move operation.	
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

Deleting mailbox

Table 66: Deleting mailbox

Report Item (Success)	Report Item (Failure)
Mailbox `name` is deleted.	Not applicable

Removing Exchange attributes

Table 67: Removing Exchange attributes

Report Item (Success)	Report Item (Failure)
The following Exchange attributes are removed from `name`. List: Attribute names	Not applicable

Enabling mailbox for Unified Messaging

Table 68: Enabling mailbox for Unified Messaging

Report Item (Success)	Report Item (Failure)
Mailbox `name` is enabled for Unified Messaging.	Not applicable
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

Disabling Unified Messaging for mailbox

Table 69: Disabling Unified Messaging for mailbox

Report Item (Success)	Report Item (Failure)
Unified Messaging is disabled for mailbox `name`.	Not applicable

Resetting Unified Messaging PIN

Table 70: Resetting Unified Messaging PIN

Report Item (Success)	Report Item (Failure)
The Unified Messaging PIN is reset for mailbox `name`.	Not applicable

Establishing email address for group

Table 71: Establishing email 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

Creating query-based distribution group

Table 72: 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

Establishing e-mail address for user

Table 73: 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

Establishing e-mail address for contact

Table 74: Establishing e-mail address for contact

Report Item (Success)	Report Item (Failure)
An e-email 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
E-mail alias is set to alias.	Not applicable
The following properties of the contact are set. List: Property names and values	Not applicable

Deleting e-mail address for group

Table 75: 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

Deleting e-mail address for user

Table 76: 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

Deleting e-mail address for contact

Table 77: 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

Converting user mailbox to linked mailbox

Table 78: 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

Converting linked mailbox to user mailbox

Table 79: Converting linked mailbox to user mailbox

Report Item (Success)	Report Item (Failure)
Linked mailbox `name` is converted to a user mailbox.	Not applicable

Report Item (Success)	Report Item (Failure)
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: For more information, see Change-tracking policy.

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. For more information, see Viewing change history.

13

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 80: 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 more information on how Active Roles applies entitlement rules, see About entitlement profile build process.

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: 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 predefined 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 Predefined 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 to specify your search criteria. You can specify search criteria the same way you do when using the **Find** dialog. 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**.

10. Click **Next**.

11. 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**.

12. Type the name of the resource type to be displayed in the entitlement profile.

13. 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.

14. Click **Next**.
15. Set up the list of the resource-related attributes that will be displayed in the entitlement profile:
    - Use **Add** or **Remove** 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.
16. 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 81: 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.

To change	Do this	Commentary
Entitlement type	Right-click the object, click **Properties**, click the **Type** tab, and then select the appropriate option.	The entitlement type specifies how the user is entitled to the resource. You 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 to specify rule conditions. You can do this the same way you use the **Find** dialog 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 **Remove** to remove a rule of any type. For more information, see Step 6 in 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	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

To change	Do this	Commentary
	type, and the resource naming attribute.	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.

Predefined specifiers

Active Roles comes with a collection of predefined 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 predefined specifier (see Changing entitlement profile specifiers) or you can apply the **Disable** command for the specifier to have no effect.

NOTE: Predefined specifiers cannot be deleted.

The predefined 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 predefined specifier and a customer-created specifier, the latter specifier is applied. The priority of specifiers is governed by the **edsaPriority** attribute setting. For more information, see About entitlement profile build process.

The following table provides information about the predefined specifiers. For each specifier, the table lists the specifier’s name, description, entitlement type and rules, and resource display settings.

Table 82: Predefined specifiers

Name and Description	Type and Rules	Resource Display Settings
Name: Self - Exchange Mailbox	**Type**: Personal resource	**Resource type name**: Exchange Mailbox **Resource naming attribute**: mail

Name and Description	Type and Rules	Resource Display Settings
Description: Specifies user entitlement to Exchange mailbox.	entitlement **Rules**: Entitlement target object is an Exchange mailbox enabled user account.	**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 Office Communications Server enabled account.	**Type**: Personal resource entitlement **Rules**: Entitlement target object has the `msRTCSIP-UserEnabled` attribute set to **TRUE**.	**Resource type name**: Enabled for Office Communications Server **Resource naming attribute**: msRTCSIP-PrimaryUserAddress **Other resource-related attributes**: • msRTCSIP-PrimaryUserAddress • edsva-OCS-Pool
Name: Membership - Member of Security Group	**Type**: Shared resource entitlement	**Resource type name**: Member of Security Group **Resource naming attribute**: name

Name and Description	Type and Rules	Resource Display Settings
Description: Specifies entitlement to a resource via membership in a security group.	**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.	**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: 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

Name and Description	Type and Rules	Resource Display Settings
		• 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 owner role for a room, equipment, or shared mailbox.	**Type**: Managed resource entitlement **Rules**: Entitlement target object is a user account associated with a room, equipment or shared mailbox.	**Resource type name**: Owner of Resource Exchange Mailbox **Resource naming attribute**: displayName **Other resource-related attributes**: • displayName • edsva-MsExch-MailboxTypeDescription • mail • description • homeMDB • edsvaParentCanonicalName
Name: Managed By - Owner of Exchange Contact **Description**: Specifies entitlement to the	**Type**: Managed resource entitlement **Rules**: Entitlement target object is an	**Resource type name**: Owner of Exchange Contact **Resource naming attribute**: displayName **Other resource-related attributes**: • displayName

Name and Description	Type and Rules	Resource Display Settings
owner role for an Exchange mail contact.	Exchange mail contact.	- 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 owner role.	**Type**: Managed resource entitlement **Rules**: No rules specified, which 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.	**Resource type name**: Owner of <target object class display name> **Resource naming attribute**: name **Other resource-related attributes**: - name - 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 **Entitlement Profile** on the **Managed Resources** tab in the **Properties** dialog 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 83: 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

Resource Type	Resource Name	Resource Details
		- 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 SharePoint Site	Group name	- Group name - 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

Resource Type	Resource Name	Resource Details
		• 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)
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

Resource Type	Resource Name	Resource Details
		- 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

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* in the *Microsoft Windows Server 2008 documentation*.

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, 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 *Using advanced search options* and *Building a custom search* in the *Active Roles Console User Guide*.
4. Click **Find Now** to start the search.

When the search completes, the **Find** dialog 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 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 **Clear Search**.

> 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.
- A list of search results prepared using the **Deleted Objects** search category in the **Find** dialog.
- 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, click the deleted object and then click **Restore**.

   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, and then click **OK** to start the restore process.

The **Restore Object** dialog 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 by the **deleted from:** label, is displayed beneath the name of the deleted object on the **General** tab in the **Properties** dialog. 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, 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.

Although it is possible to delegate the operation of restoring deleted objects in any managed domain, OU or MU, 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 more information on how to enable Recycle Bin, see Active Directory Recycle Bin Step-by-Step Guide.

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 email.

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, click **Add**.
3. In the **Select Policy Objects** dialog, 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.

For more information on configuring and applying Policy Objects, see Applying Policy Objects.

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 **Workflow options and start conditions** 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 Workflows.

15

AD LDS data management

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.

To add or remove members to or from an AD LDS group

1. In the Console tree, under **AD LDS (ADAM)**, locate and select the container that holds the group.
2. In the details pane, right-click the group, and click **Properties**.
3. On the **Members** tab in the **Properties** dialog, click **Add**.
4. Use the **Select Objects** dialog to locate and select the security principals that you want to add to the group. When finished, click **OK**.
5. On the **Members** tab, select the group members that you want to remove from the group, and then click **Remove**.
6. After making the changes that you want to the group, click **OK** to close the **Properties** dialog.

When using the **Select Objects** dialog 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 more information, see Setting or modifying the password of an AD LDS user.

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, 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.

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 OU 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 OU by using the **New** > **User** or **New** > **Group** command on that OU, as discussed earlier in this section.

You can move an existing AD LDS user or group to an OU 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 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 for the user account, navigate 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.

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 Configuring rule-based administrative views.

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, click **Include by Query**, and then click **OK**.
4. Use the **Create Membership Rule** dialog 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, 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 **Field**, 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 **Add** 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 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. Further steps for configuring a membership rule are all about using either the **Create Membership Rule** dialog to set up a certain query or the **Select Objects** dialog 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 preconfigured Access Templates to facilitate delegation of AD LDS data management tasks. For a list of the AD LDS-specific Access Templates, refer to the *Active Roles Built-in Access Templates Reference Guide*. 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 more information on how to create, configure and apply Access Templates, see Configuring role-based administration.

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, click **Security**.
4. In the **Active Roles Security** dialog, 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.
6. Click **OK** to close the **Properties** dialog 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, 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, 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 more information on how to create, configure and apply Policy Objects, see Rule-based autoprovisioning and deprovisioning.

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, click **Policy**.

4. In the **Active Roles Policy** dialog, 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.
6. Click **OK** to close the **Properties** dialog for the AD LDS object.

In the **Active Roles Policy** dialog, 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, 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

Active Roles 8.1.3 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 Identity Analytics and Risk Intelligence, and Starling Connect.

To start the wizard, in the Active Roles Configuration Center, click **Dashboard** > **Starling** > **Configure**.

To join One Identity Starling to Active Roles, in the Active Roles Configuration Center, navigate to **Starling** and click **Join One Identity Starling**. 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: Consider the following when configuring Active Roles to join One Identity Starling:
   >
   > - 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** appears in Active Roles with a progress message indicating the Starling joining progress. A join confirmation page appears with the name of the Active Roles instance that will be joined to Starling.

After the operation is completed successfully, the **Starling** tab appears with **Account Joined** success message.

Disconnecting One Identity Starling from Active Roles

After you configure Active Roles to join Starling, in case you want to disconnect from Starling, on the **Starling** tab on the **Starling** page, click **Unjoin One Identity Starling**. The **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.

17

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 Roles 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 the **Command** pane, click **Undo Deprovisioning**.

   The **Password Options** dialog 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: Consider the following when configuring notifications for Starling operations:
>
> - 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 the managed environment.
> - 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. 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` and the notifications do not expire. 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 Active Roles website over HTTPS:
>
> - Import a valid certificate into Trusted Root Certificate Authority in the machine where Active Roles 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`.
> - Run 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.

Table 84: 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

Table 85: SCIM attribute mapping with Active Directory for Groups

SCIM	Active Directory
displayName	cn
members	member
email	mail
manager	managedBy

18

Configuring linked mailboxes with Exchange Resource Forest Management

The Exchange Resource Forest Management (ERFM) feature of Active Roles allows you to automate mailbox provisioning for on-premises users in environments where the mailboxes and the user accounts are managed in different Active Directory (AD) forests. Such multi-forest environments are based on the resource forest model, and mailboxes provisioned in such environments are called linked mailboxes.

Multi-forest AD deployments have higher administrative and support costs. However, they offer the highest level of security isolation between AD objects and the Exchange service. As such, One Identity recommends configuring the resource forest model for use with Active Roles in organizations that:

- Aim for an extra layer of data security.
- Frequently experience organizational changes (for example, buying companies, or consolidating and breaking off branch companies, departments and other business units).
- Abide by certain legal or regulatory requirements.

AD deployments following the resource forest model use two types of AD forests:

- **Account forests**: These AD forests store the user objects. Organizations can use one or more account forests in the resource forest model.
- **Resource forest**: This AD forest contains the Exchange server and stores the mailboxes of the user objects.

For more details on ERFM, see *Exchange Resource Forest Management* in the *Active Roles Feature Guide*.

Prerequisites of configuring linked mailboxes

To use linked mailboxes with Exchange Resource Forest Management (ERFM) in Active Roles for your organization, your deployment must meet the following requirements.

Multi-forest deployment

Your organization must have at least two Active Directory (AD) forests:

- **Account forest**: One or more forests that contain the user accounts.
- **Resource forest**: A forest that contains the Exchange server and will store the mailboxes and the shadow accounts connecting the linked mailboxes to the user objects. ERFM requires a supported version of Exchange Server installed in the resource forest. For more information on the Microsoft Exchange Server versions supported by Active Roles, see *System requirements* in the *Active Roles Release Notes*.

For more information on planning and configuring multi-forest AD deployments, see Setting up a forest trust to support linked mailboxes and Plan a multi-forest deployment in the *Microsoft documentation*.

For more information on the Microsoft Exchange Server versions Active Roles supports, see *Supported platforms* in the *Active Roles Release Notes*.

Two-way trust relation

The resource and account forests must identify each other as trusted domains (that is, they must be in a two-way trust relation).

- For more information on forest trust in general, see One-way and two-way trusts in the *Microsoft documentation*.
- For more information on how to set up a forest trust, see Create a Forest Trust in the *Microsoft documentation*.

Registering the forests in Active Roles

You must register the resource and account forests in Active Roles via the Active Roles Console. For more information, see Registering the resource and account forests in Active Roles.

Applying the ERFM - Mailbox Management built-in policy

You must apply the **ERFM - Mailbox Management** built-in policy (or a copy of it) on the Organizational Unit (OU) whose users will use linked mailboxes. For more information, see Applying the ERFM Mailbox Management policy to an OU.

(Optional) Modifying the ERFM scheduled task

Once the **ERFM - Mailbox Management** built-in policy is configured for an OU, Active Roles synchronizes the properties of every managed master user account to the corresponding shadow account with the **ERFM - Mailbox Management** built-in scheduled task.

By default, the scheduled task runs on a daily basis, and normally, you do not need to modify its settings. To change the default ERFM scheduling (for example, because of organizational reasons), or run it manually so that you can immediately identify master accounts in your OU, see Configuring the ERFM Mailbox Management scheduled task.

(Optional) Changing the default location of the shadow accounts

By default, the **ERFM - Mailbox Management** built-in policy saves shadow accounts in the **Users** container of the resource forest. If your organization stores other users as well in the **Users** container, then One Identity recommends changing the container for storing the shadow accounts for clarity.

For more information, see Changing the location of the shadow accounts.

(Optional) Modifying the synchronized properties of the master account

By default, ERFM synchronizes a pre-defined set of user and mailbox properties between the master accounts and shadow accounts. If you need to modify and/or expand the default set of synchronized properties (for example, because of organizational reasons), open and update the applicable **ERFM - Mailbox Management** policy settings.

For the list of default synchronized properties and more information on changing them, see Configuring the synchronized, back synchronized or substituted properties of linked mailboxes.

(Optional) Delegating Exchange Access Templates

If you want to manage linked mailboxes with non-administrator users, you must assign one or more of the following Exchange Access Templates (ATs) to them in the Active Roles Console:

- `Exchange - Manage Resource, Linked and Shared Mailboxes`
- `Exchange - Convert Linked Mailbox to User Mailbox`
- `Exchange - Convert User Mailbox to Linked Mailbox`
- `Exchange - Create Linked Mailboxes`
- `Exchange - Read ERFM Attributes`
- `Exchange - Recipients Full Control`

> TIP: To provide full control for a user to create, view, or change linked mailboxes in the Exchange forest, assign the `Exchange - Recipients Full Control` AT to them.

For more information on how to apply ATs, see Applying Access Templates.

Registering the resource and account forests in Active Roles

To provision linked mailboxes in the resource forest model with Exchange Resource Forest Management (ERFM), you must register the resource forest and the account forest(s) of your organization in Active Roles as managed domains.

Prerequisites

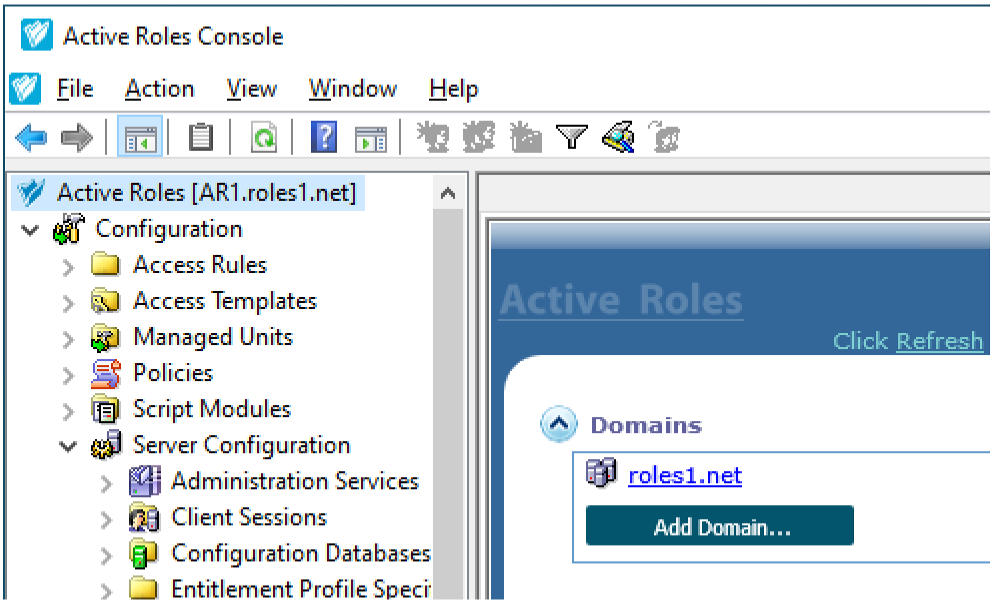
To register the forests, you must have access to administrator accounts with sufficient rights in the account forest(s) and the resource forest.

- To register the **account forest(s)**, you must use the Active Roles Administration Service account.
- To register the **resource forest**, you must use a Microsoft Exchange administrator account of the resource forest. Specifically, this Exchange administrator account must have the following rights and permissions:
  - It must be a member of the **Account Operators** domain security group.
  - It must have read access to Exchange configuration data in the resource forest. For more information on how to configure read access, see *Permission to read Exchange configuration data* in the *Active Roles Quick Start Guide*.

To register the account forest(s) in Active Roles

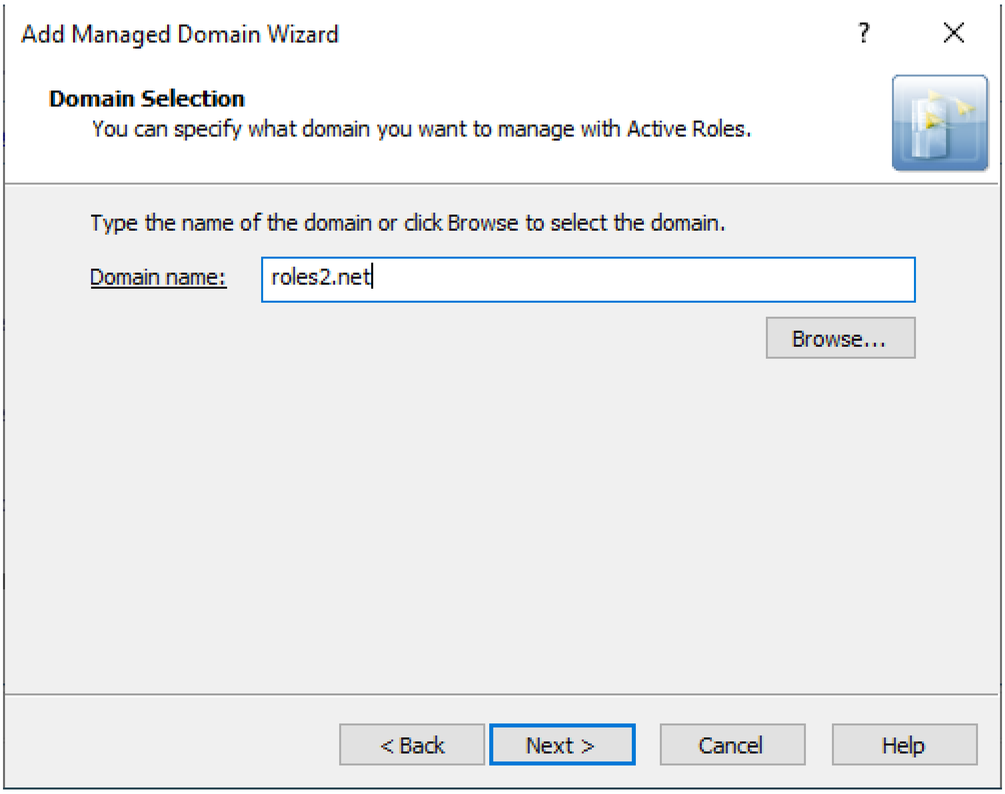
1. In the Active Roles Console, open the **Add Managed Domain Wizard**. To do so, open the Active Roles main page by clicking the top Active Roles node, then click **Domains** > **Add Domain**.

Figure 131: Active Roles Console – Add Domain setting in the main node



2. In the **Domain Selection** step, either enter the `domain name` of the forest, or click **Browse** to select it.

Figure 132: Add Managed Domain Wizard > Domain Selection – Specifying an account forest


Add Managed Domain Wizard
?
Domain Selection
You can specify what domain you want to manage with Active Roles.
Type the name of the domain or click Browse to select the domain.

Domain name: roles2.net

Browse...
< Back
Next >
Cancel
Help

3. In the **Active Roles Credentials** step, under **Access the domain using**, select **The service account information the Administration Service uses to log on**.
4. To apply your changes, click **Finish**.

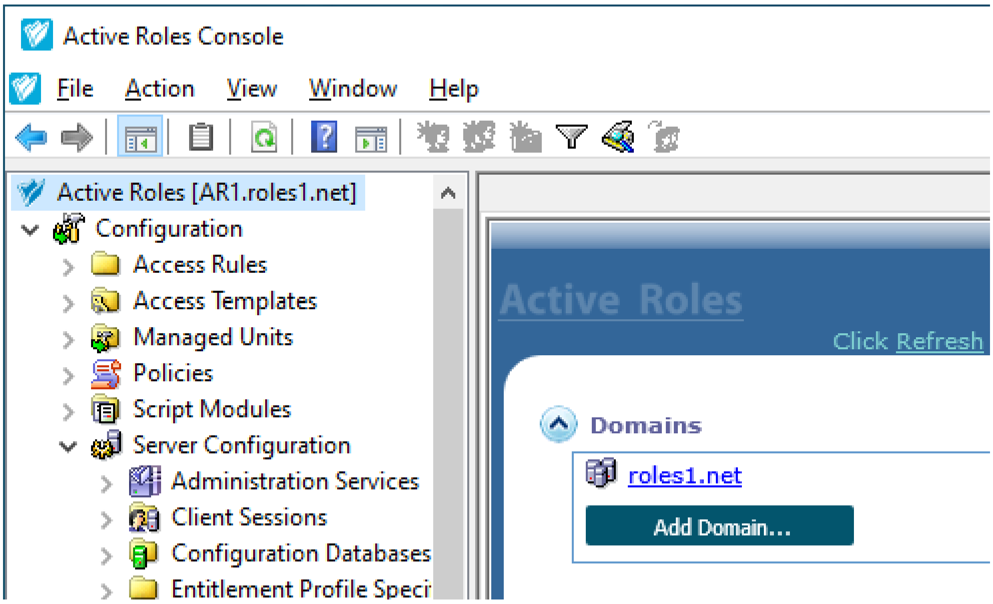
Active Roles then establishes the connection to the configured forest, indicated with the `Domain information is being loaded` message on the main page. Once Active Roles connected to the domain, the Active Roles Console will indicate it with the `Available for management` message.

> TIP: To check the current domain connection status, use the **click to update the display** link. The link is replaced with the `Available for management` feedback once Active Roles finishes connecting to the forest.

To register the resource forest (Exchange forest) in Active Roles

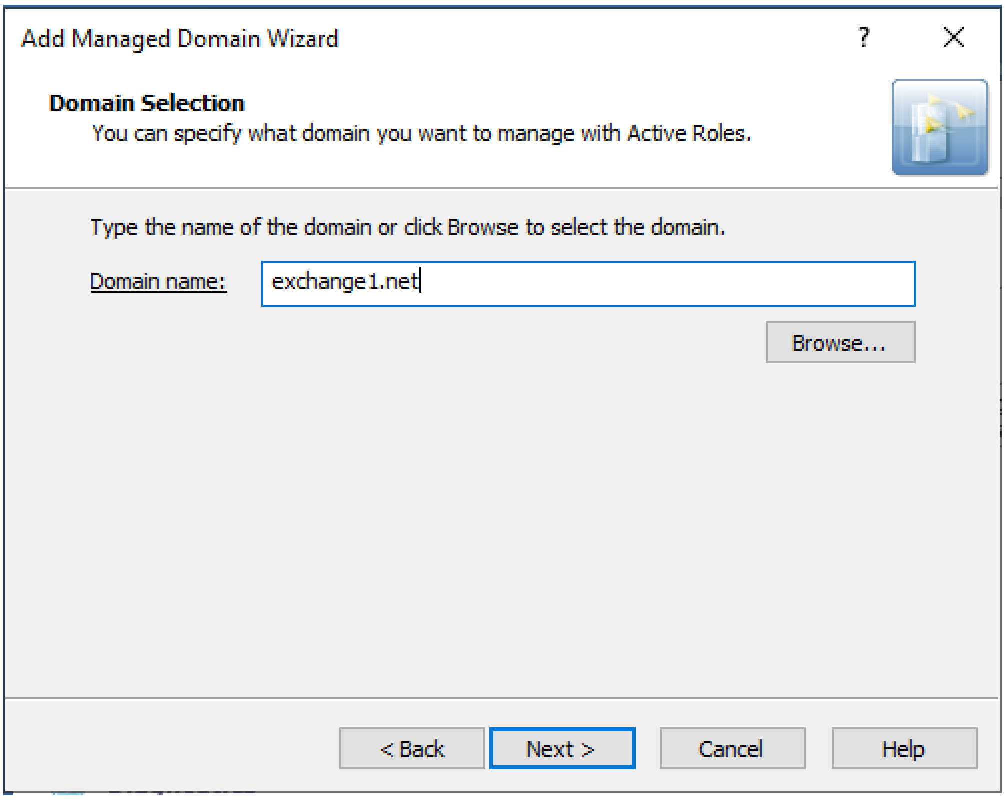
1. In the Active Roles Console, open the **Add Managed Domain Wizard**. To do so, open the Active Roles main page by clicking the top Active Roles node, then click **Domains** > **Add Domain**.

Figure 133: Active Roles Console – Add Domain setting in the main node



2. In the **Domain Selection** step, either enter the `domain name` of the forest, or click **Browse** to select it.

Figure 134: Add Managed Domain Wizard > Domain Selection – Specifying the resource forest



Add Managed Domain Wizard

Domain Selection

You can specify what domain you want to manage with Active Roles.

Type the name of the domain or click Browse to select the domain.

Domain name: exchange1.net

3. In the **Active Roles Credentials** step, under **Access the domain using**, select **The Windows user account information specified below**, and provide the **User name**, **Password** and **User domain** of the resource forest administrator account.

   > NOTE: Make sure that you specify a valid resource forest administrator user in this step, instead of the Active Roles service account used for registering the account forest(s).
   >
   > Using your Active Roles administrator account in this step can result in Active Roles being unable to create the shadow accounts later in the resource forest.

4. To apply your changes, click **Finish**.

Active Roles then establishes the connection to the configured forest, indicated with the `Domain information is being loaded` message on the main page. Once Active Roles connected to the domain, the Active Roles Console will indicate it with the `Available for management` message.

> TIP: To check the current domain connection status, use the **click to update the display** link. The link is replaced with the `Available for management` feedback once Active Roles finishes connecting to the forest.

Applying the ERFM Mailbox Management policy to an OU

Active Roles can provision linked mailboxes automatically for users only if the **ERFM - Mailbox Management** built-in policy is applied to the Organizational Unit (OU) of the users in the Active Roles Console.

Prerequisites

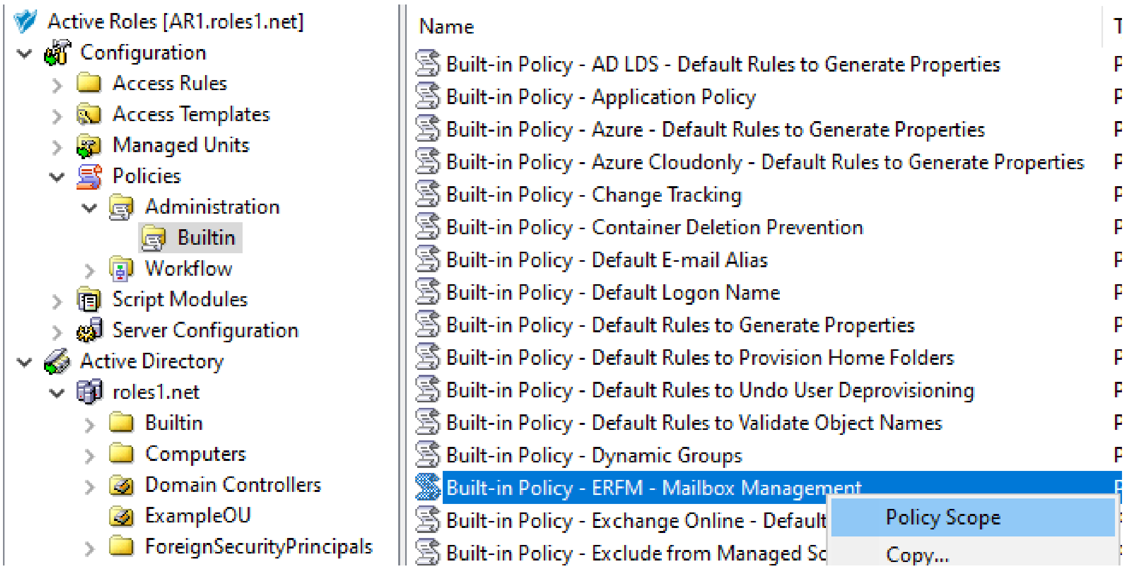
Before applying the **ERFM - Mailbox Management** policy to an OU in the Active Roles Console, make sure that the account forest(s) and the resource forest are already registered in Active Roles as managed domains.

For more information, see Registering the resource and account forests in Active Roles.

To apply the ERFM - Mailbox Management policy to an OU

1. In the Active Roles Console, in the Active Directory (AD) tree, navigate to **Configuration** > **Policies** > **Administration** > **Builtin**.
2. To open the **Scope** tab of the **ERFM - Mailbox Management** policy, right-click **Built-in Policy - ERFM - Mailbox Management**, then in the context menu, click **Policy Scope**.

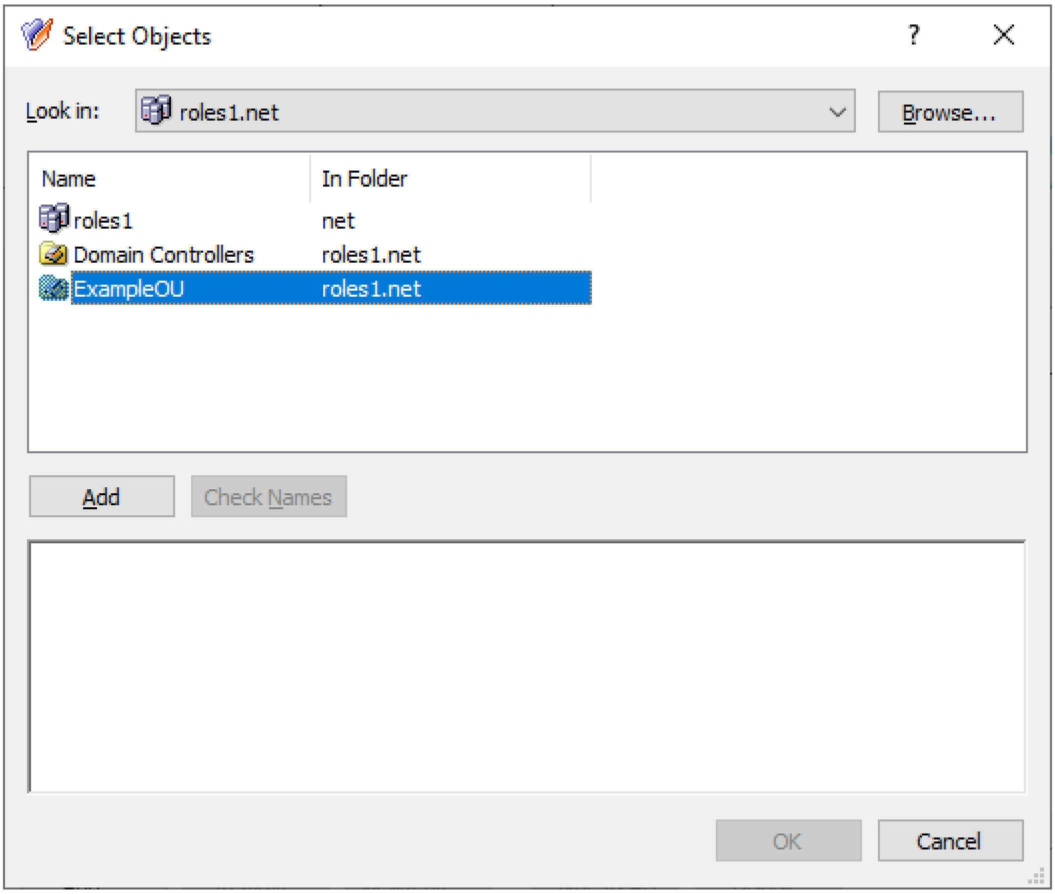
   Figure 135: Active Roles Console – Opening the Policy Scope settings of the ERFM - Mailbox Management built-in policy


Active Roles [AR1.roles1.net]
Configuration
Access Rules
Access Templates
Managed Units
Policies
Administration
Builtin
Workflow
Script Modules
Server Configuration
Active Directory
roles1.net
Builtin
Computers
Domain Controllers
ExampleOU
ForeignSecurityPrincipals
Name
Built-in Policy - AD LDS - Default Rules to Generate Properties
Built-in Policy - Application Policy
Built-in Policy - Azure - Default Rules to Generate Properties
Built-in Policy - Azure Cloudonly - Default Rules to Generate Properties
Built-in Policy - Change Tracking
Built-in Policy - Container Deletion Prevention
Built-in Policy - Default E-mail Alias
Built-in Policy - Default Logon Name
Built-in Policy - Default Rules to Generate Properties
Built-in Policy - Default Rules to Provision Home Folders
Built-in Policy - Default Rules to Undo User Deprovisioning
Built-in Policy - Default Rules to Validate Object Names
Built-in Policy - Dynamic Groups
Built-in Policy - ERFM - Mailbox Management
Built-in Policy - Exchange Online - Default
Built-in Policy - Exclude from Managed Sc
Policy Scope
Copy...

3. To enable linked mailboxes for an OU, in the **Active Roles Policy Scope for Built-in Policy** window, select the OU to which you want to apply the policy. Click **Add**, select the OU in the **Select Objects** window, click **Add**, then click **OK**.

TIP: If the **Select Objects** window lists no objects, use the **Click here to display objects** link.

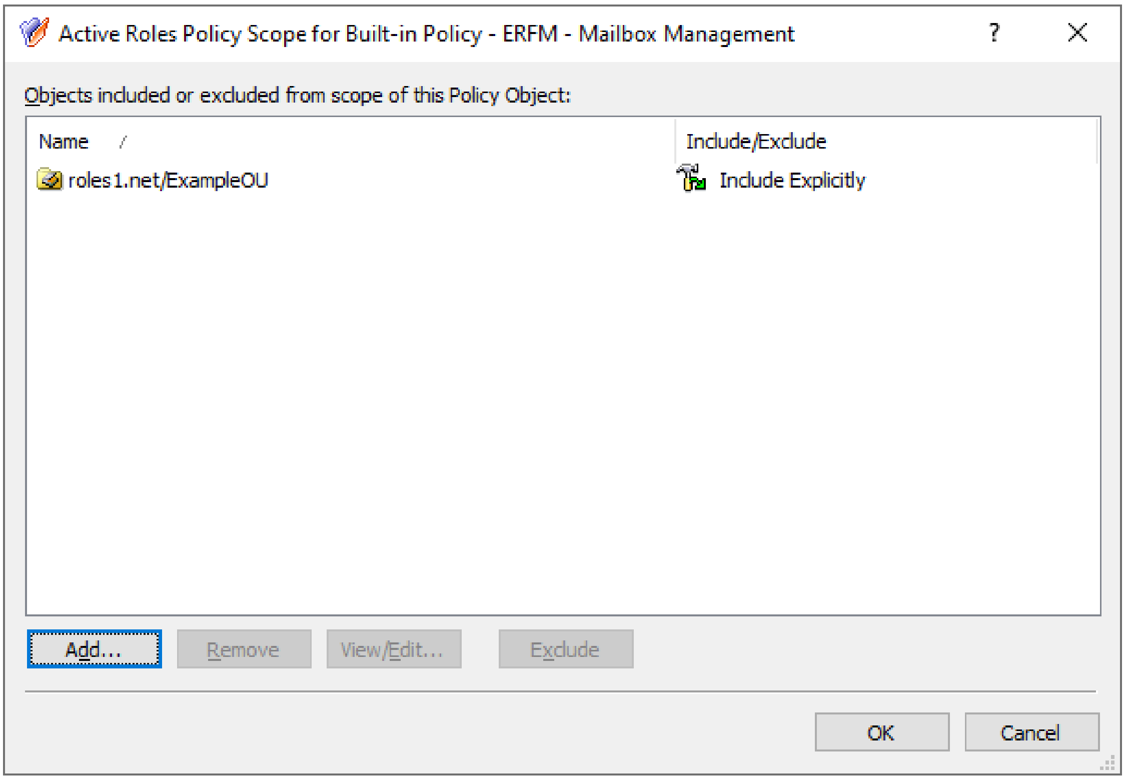
Figure 136: Active Roles Console– Selecting the OU for the ERFM - Mailbox Management policy



Name	In Folder
roles1	net
Domain Controllers	roles1.net
ExampleOU	roles1.net

4. To apply your changes, click **OK**.

Figure 137: Active Roles Console– Applying the OU to the scope of the ERFM - Mailbox Management policy


Active Roles Policy Scope for Built-in Policy - ERFM - Mailbox Management
?
Objects included or excluded from scope of this Policy Object:
Name
Include/Exclude
roles1.net/ExampleOU
Include Explicitly
Add...
Remove
View/Edit...
Exclude
OK
Cancel

After the policy is applied, creating a new on-premises user in the OU with the **Create an Exchange Mailbox** setting enabled will automatically result in the following provisioning steps:

1. Active Roles creates the master user account of the user on the account forest.
2. Active Roles then creates the linked mailbox of the user in the Exchange server of the resource forest, and a shadow user account connected to the master user account.

NOTE: Consider the following when using the **ERFM - Mailbox Management** policy:

- If you registered the forest root domain of the resource forest to Active Roles as a managed domain, then Active Roles will create shadow accounts in that domain. Otherwise, Active Roles creates shadow accounts in the domain that is listed first in the ordered list of the resource forest managed domains.
- After the policy is configured, linked mailboxes will only be available for users in the OU who were created after applying the policy, and for existing users with no mailboxes. For more information on configuring a linked mailbox for existing users, see Creating a linked mailbox for an existing user with no mailbox.

Configuring the ERFM Mailbox Management scheduled task

Once the **ERFM - Mailbox Management** built-in policy is configured for an OU, Active Roles synchronizes the properties of every managed master user account to the corresponding shadow account with the **ERFM - Mailbox Management** built-in scheduled task.

By default, the scheduled task runs on a daily basis, and normally you do not need to modify its settings. However, it can happen that you need to:

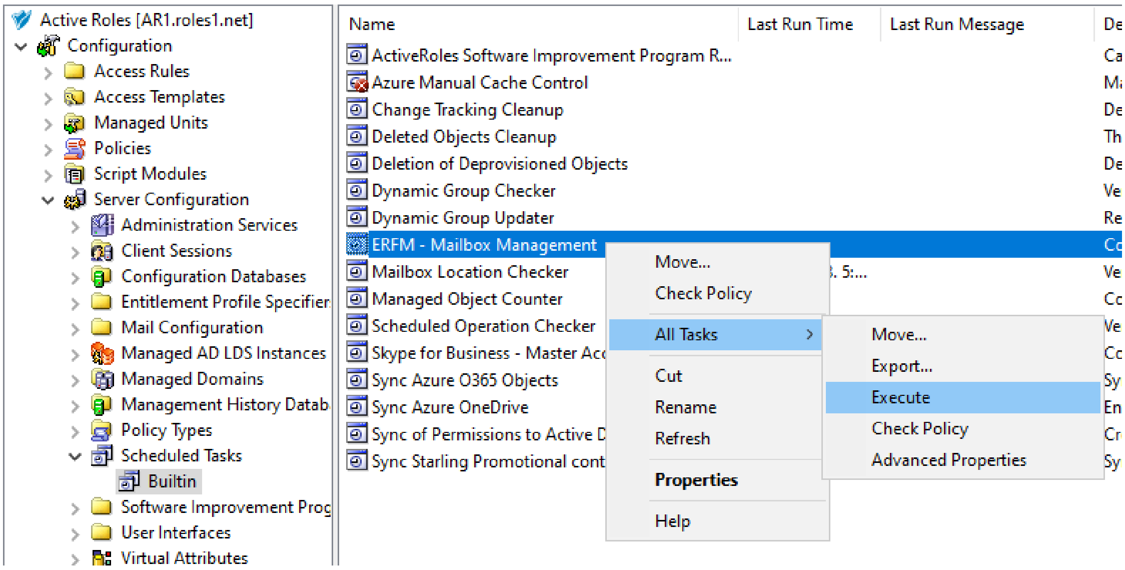
- Change the default ERFM scheduling, for example, because of organizational reasons.
- Run the scheduled task manually to make Active Roles immediately identify the existing master accounts of your Organizational Unit (OU), without waiting for its scheduled run to complete.

> NOTE: The **ERFM - Mailbox Management** scheduled task affects only user accounts whose OU is in the scope of the **ERFM - Mailbox Management** built-in policy, or a copy of that policy.

To run the ERFM - Mailbox Management built-in schedule manually

1. In the Active Roles Console, in the Active Directory (AD) tree, navigate to **Configuration** > **Server Configuration** > **Scheduled Task** > **Builtin**.
2. Right click the scheduled task **ERFM - Mailbox Management**, then click **All Tasks** > **Execute**.

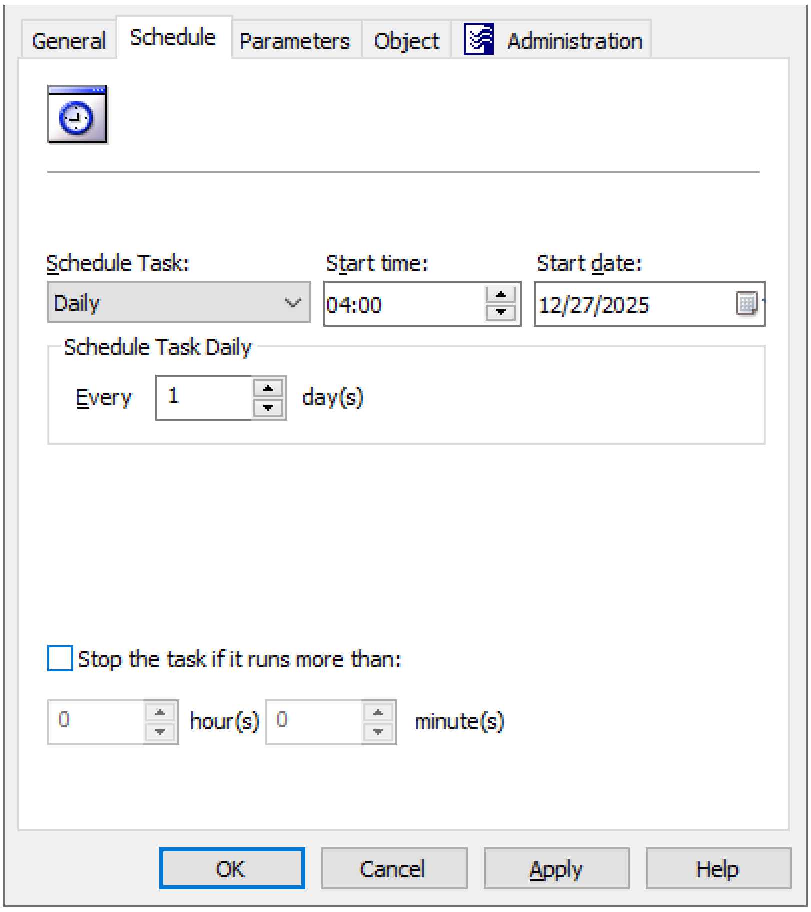
Figure 138: Active Roles Console– Running the ERFM Mailbox Management scheduled task



To modify the settings of the ERFM - Mailbox Management built-in schedule

1. In the Active Roles Console, in the Active Directory (AD) tree, navigate to **Configuration** > **Server Configuration** > **Scheduled Task** > **Builtin**.
2. Open the scheduling properties of the **ERFM - Mailbox Management** built-in scheduled task. To do so, either:
   - Double-click **ERFM - Mailbox Management**, then in the **Properties** window, open the **Schedule** tab.
   - Right-click **ERFM - Mailbox Management**, then click **Properties** > **Schedule**.

Figure 139: Active Roles Console– Scheduling properties of the scheduled task



Schedule Task: Daily

Start time: 04:00

Start date: 12/27/2025

Schedule Task Daily

Every 1 day(s)

□ Stop the task if it runs more than:

0 hour(s) 0 minute(s)

3. To change the default scheduling settings of the task for your needs, modify the options of the **Schedule** tab accordingly:
   - **Schedule Task**: Specifies how frequently Active Roles runs the task (each hour, every day, or on a weekly/monthly basis). By default, tasks are run on a daily basis.

- **Start time** and **Start date**: These settings specify the time and date of the first scheduled task run. These settings are not available if **Schedule Task** is set to **Once** or **When Service starts**.
- **Schedule Task Hourly** / **Daily** / **Weekly** / **Monthly**: These settings specify the time interval of repeating the configured task.

  For example, setting **Schedule Task** to **Hourly** lets you specify the time interval between two task runs in hours and minutes, while setting it to **Weekly** lets you specify not just the number of weeks between two task runs, but also the days of the week on which Active Roles must run the task.

  > NOTE: This setting is not available if **Schedule Task** is set to **Once** or **When Service starts**.
- **Stop the task if it runs more than**: When selected, this setting sets a timeout (in hours and minutes) after which the task stops if it runs longer than the specified interval.

4. To save your settings, click **Apply**, then **OK**.

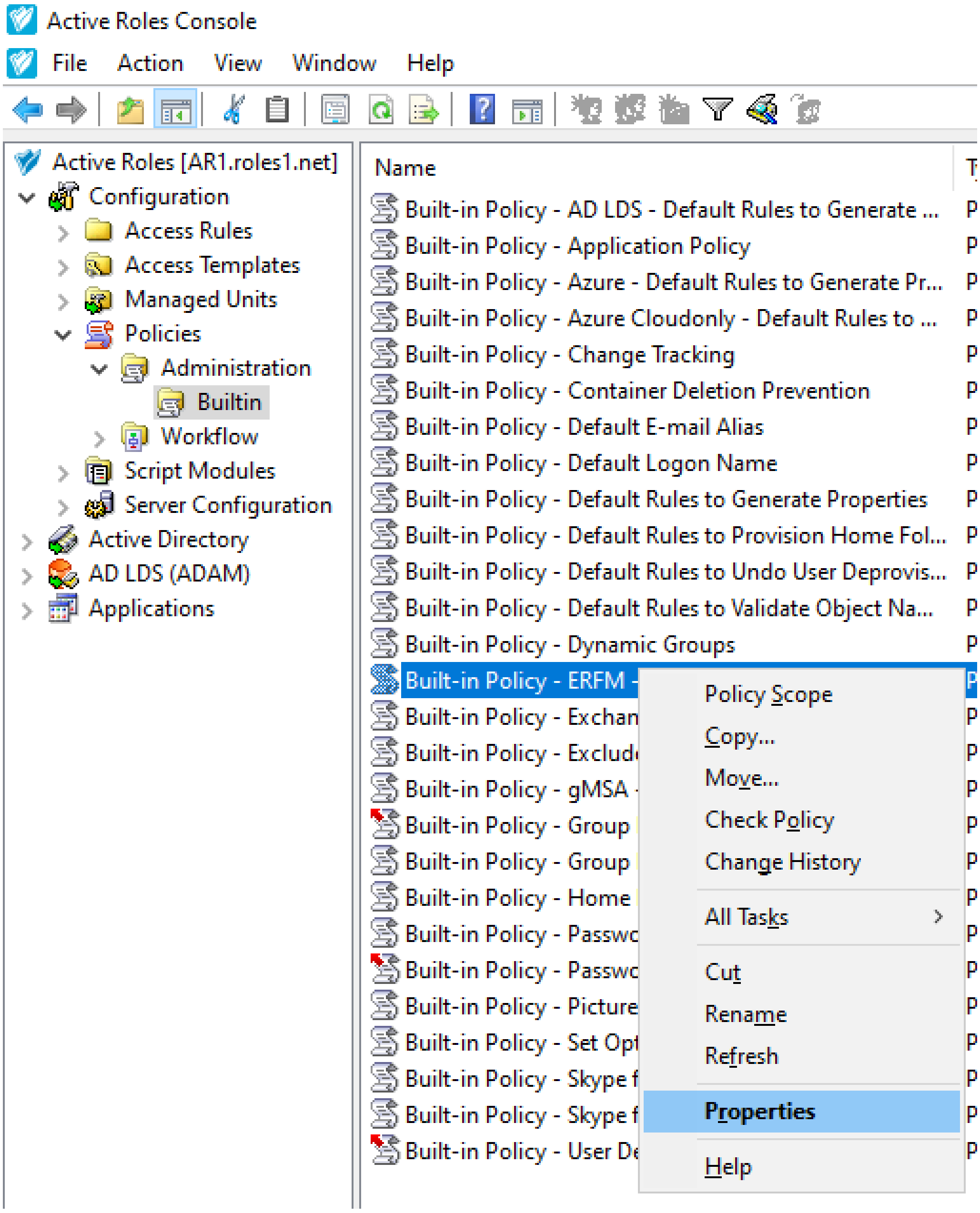
Changing the location of the shadow accounts

By default, the **ERFM - Mailbox Management** built-in policy saves shadow accounts in the **Users** container of the resource forest. If your organization stores other users as well in the **Users** container, then One Identity recommends changing the container for storing the shadow accounts for clarity.

To configure the location of the shadow accounts

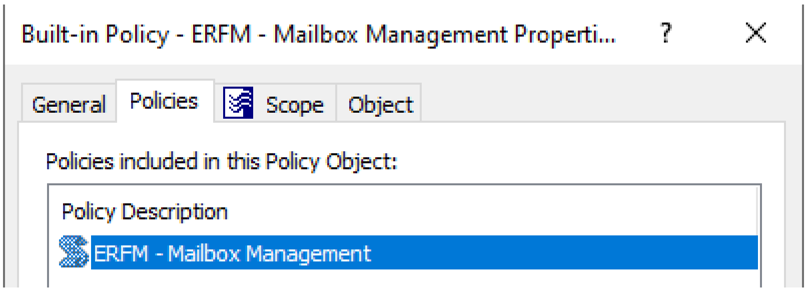
1. In the Active Roles Console, in the Active Directory (AD) tree, navigate to **Configuration** > **Policies** > **Administration** > **Builtin**.
2. To open the **Properties** of the **ERFM - Mailbox Management** policy, in the list of policies, double-click **Built-in Policy - ERFM - Mailbox Management**. Alternatively, right-click the policy, then click **Properties** in the context menu.

Figure 140: Active Roles Console– Opening the Properties of the ERFM - Mailbox Management built-in policy


Active Roles Console
File
Action
View
Window
Help
Active Roles [AR1.roles1.net]
Configuration
Access Rules
Access Templates
Managed Units
Policies
Administration
Builtin
Workflow
Script Modules
Server Configuration
Active Directory
AD LDS (ADAM)
Applications
Name
Built-in Policy - AD LDS - Default Rules to Generate ...
Built-in Policy - Application Policy
Built-in Policy - Azure - Default Rules to Generate Pr...
Built-in Policy - Azure Cloudonly - Default Rules to ...
Built-in Policy - Change Tracking
Built-in Policy - Container Deletion Prevention
Built-in Policy - Default E-mail Alias
Built-in Policy - Default Logon Name
Built-in Policy - Default Rules to Generate Properties
Built-in Policy - Default Rules to Provision Home Fol...
Built-in Policy - Default Rules to Undo User Deprovis...
Built-in Policy - Default Rules to Validate Object Na...
Built-in Policy - Dynamic Groups
Built-in Policy - ERFM -
Built-in Policy - Exchan
Built-in Policy - Exclud
Built-in Policy - gMSA -
Built-in Policy - Group
Built-in Policy - Group
Built-in Policy - Home
Built-in Policy - Passwo
Built-in Policy - Passwo
Built-in Policy - Picture
Built-in Policy - Set Opt
Built-in Policy - Skype f
Built-in Policy - Skype f
Built-in Policy - User De
Policy Scope
Copy...
Move...
Check Policy
Change History
All Tasks
Cut
Rename
Refresh
Properties
Help

3. To open the policy settings, in the **Policies** tab, click **Policy Description** > **ERFM - Mailbox Management**.

Figure 141: Active Roles Console– Opening the policy settings of the ERFM - Mailbox Management built-in policy



Built-in Policy - ERFM - Mailbox Management Properti... ? ×

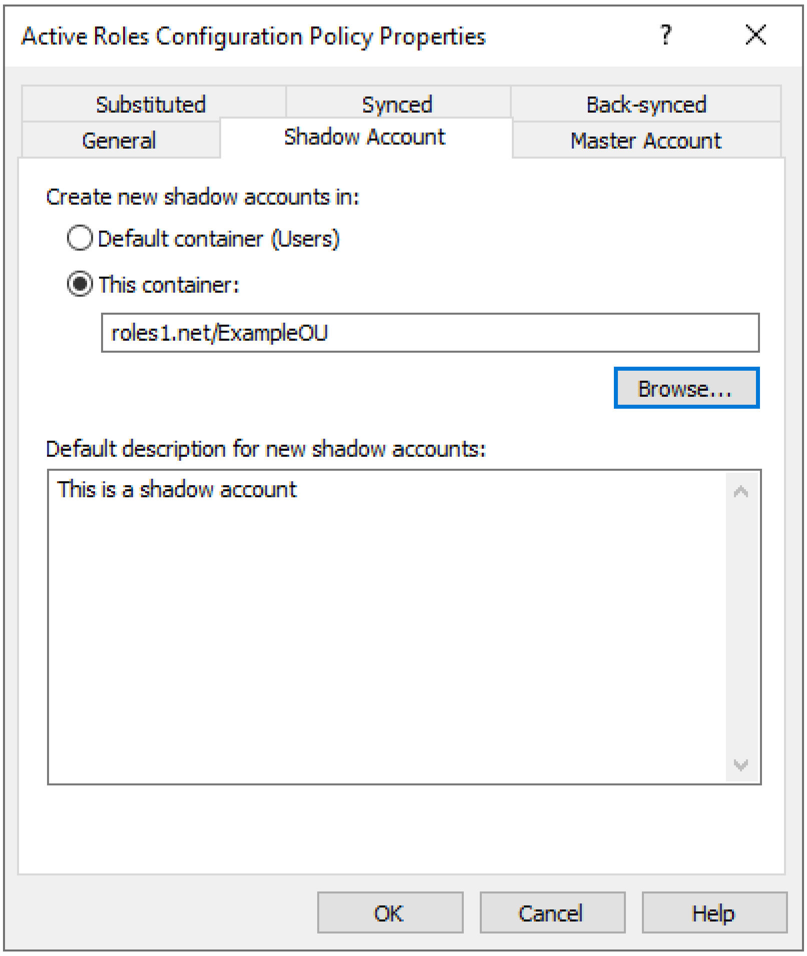
Policies included in this Policy Object:

Policy Description
ERFM - Mailbox Management

4. Under **Shadow Account**, select **This container**, then **Browse** for the container in the resource forest where you want to store the shadow accounts of the linked mailboxes.

   TIP: You can also modify the default shadow account description (`This is a shadow account`).

Figure 142: Active Roles Console– Specifying the container for the shadow accounts



Active Roles Configuration Policy Properties ? ×

Create new shadow accounts in:

○ Default container (Users)

● This container:

roles1.net/ExampleOU

Default description for new shadow accounts:

This is a shadow account

5. To apply your changes, click **OK**.

Configuring the synchronized, back synchronized or substituted properties of linked mailboxes

By default, ERFM synchronizes a pre-defined set of user and mailbox properties between the master accounts and shadow accounts. If you need to modify and/or expand the default set of synchronized properties (for example, because of organizational reasons), open and update the applicable **ERFM - Mailbox Management** policy settings.

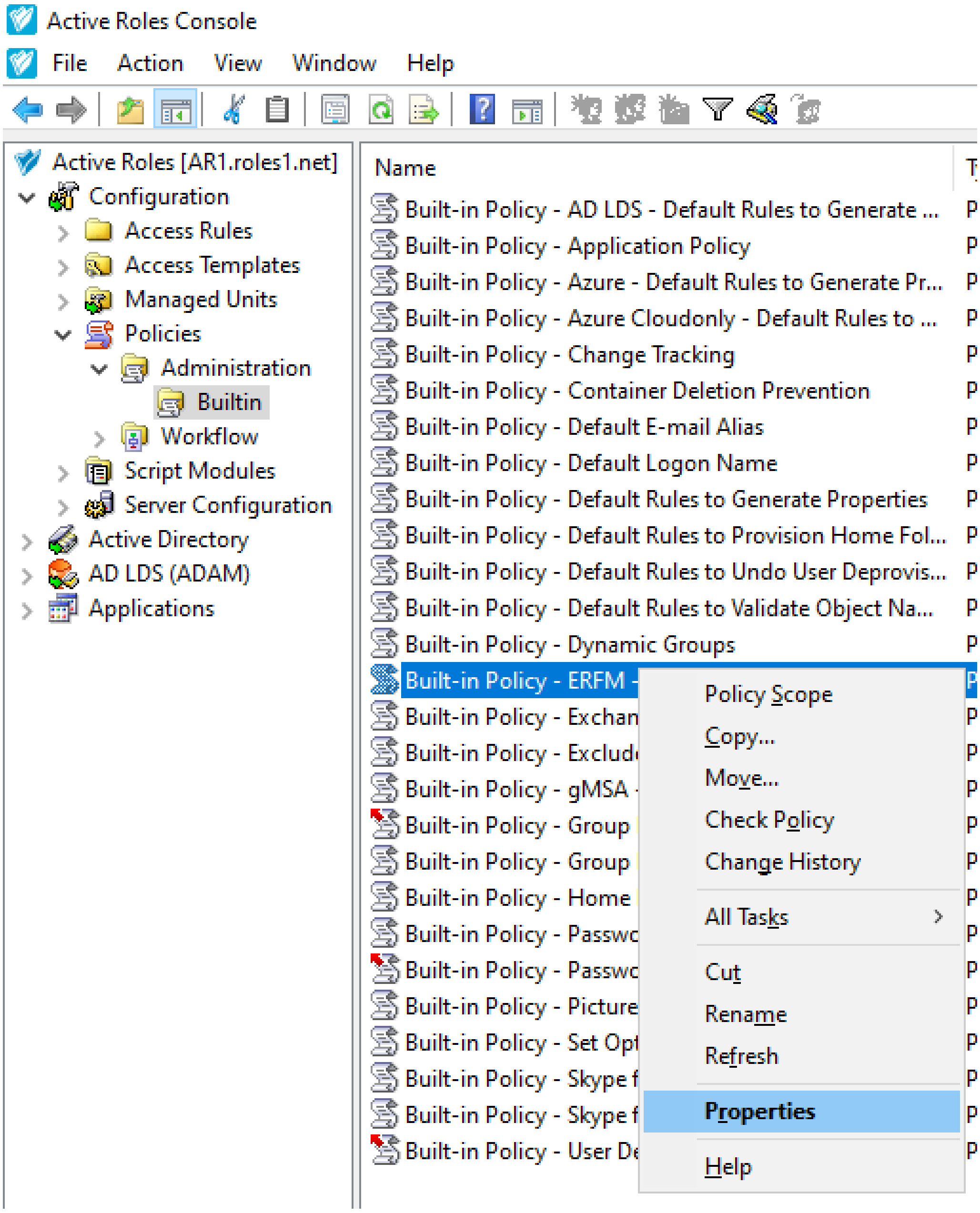
ERFM synchronizes three types of properties:

- **Synchronized properties**: Active Roles updates these properties of the master account in both the master account and its shadow account whenever they are modified. Such properties include, for example, the personal (**First Name**, **Last Name**, and so on), geographical (**Office Location**, **City**), organizational (**Company**, **Department**, and so on) or contact (**Home Phone**, **Mobile Number**) information of the user.
- **Substituted properties**: Active Roles updates these properties in the shadow account in the resource forest, even if you modify them in the master account in the account forest. Substituted properties include all Exchange recipient properties of the mail-enabled user.
- **Back synchronized properties**: Active Roles copies these properties from the shadow account to the master account. By default, this category includes a single property, **E-Mail Address (mail)**.

To view or modify the synchronized, back synchronized or substituted properties of linked mailboxes

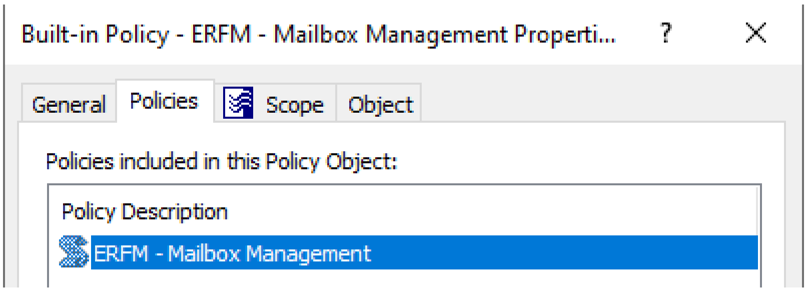
1. In the Active Roles Console, in the Active Directory (AD) tree, navigate to **Configuration** > **Policies** > **Administration** > **Builtin**.
2. To open the **Properties** of the **ERFM - Mailbox Management** policy, in the list of policies, double-click **Built-in Policy - ERFM - Mailbox Management**. Alternatively, right-click the policy, then click **Properties** in the context menu.

Figure 143: Active Roles Console– Opening the Properties of the ERFM - Mailbox Management built-in policy


Active Roles Console
File
Action
View
Window
Help
Active Roles [AR1.roles1.net]
Configuration
Access Rules
Access Templates
Managed Units
Policies
Administration
Builtin
Workflow
Script Modules
Server Configuration
Active Directory
AD LDS (ADAM)
Applications
Name
Built-in Policy - AD LDS - Default Rules to Generate ...
Built-in Policy - Application Policy
Built-in Policy - Azure - Default Rules to Generate Pr...
Built-in Policy - Azure Cloudonly - Default Rules to ...
Built-in Policy - Change Tracking
Built-in Policy - Container Deletion Prevention
Built-in Policy - Default E-mail Alias
Built-in Policy - Default Logon Name
Built-in Policy - Default Rules to Generate Properties
Built-in Policy - Default Rules to Provision Home Fol...
Built-in Policy - Default Rules to Undo User Deprovis...
Built-in Policy - Default Rules to Validate Object Na...
Built-in Policy - Dynamic Groups
Built-in Policy - ERFM -
Built-in Policy - Exchan
Built-in Policy - Exclud
Built-in Policy - gMSA -
Built-in Policy - Group
Built-in Policy - Group
Built-in Policy - Home
Built-in Policy - Passwo
Built-in Policy - Passwo
Built-in Policy - Picture
Built-in Policy - Set Opt
Built-in Policy - Skype f
Built-in Policy - Skype f
Built-in Policy - User De
Policy Scope
Copy...
Move...
Check Policy
Change History
All Tasks
Cut
Rename
Refresh
Properties
Help

3. To open the policy settings, in the **Policies** tab, click **Policy Description** > **ERFM - Mailbox Management**.

Figure 144: Active Roles Console– Opening the policy settings of the ERFM - Mailbox Management built-in policy



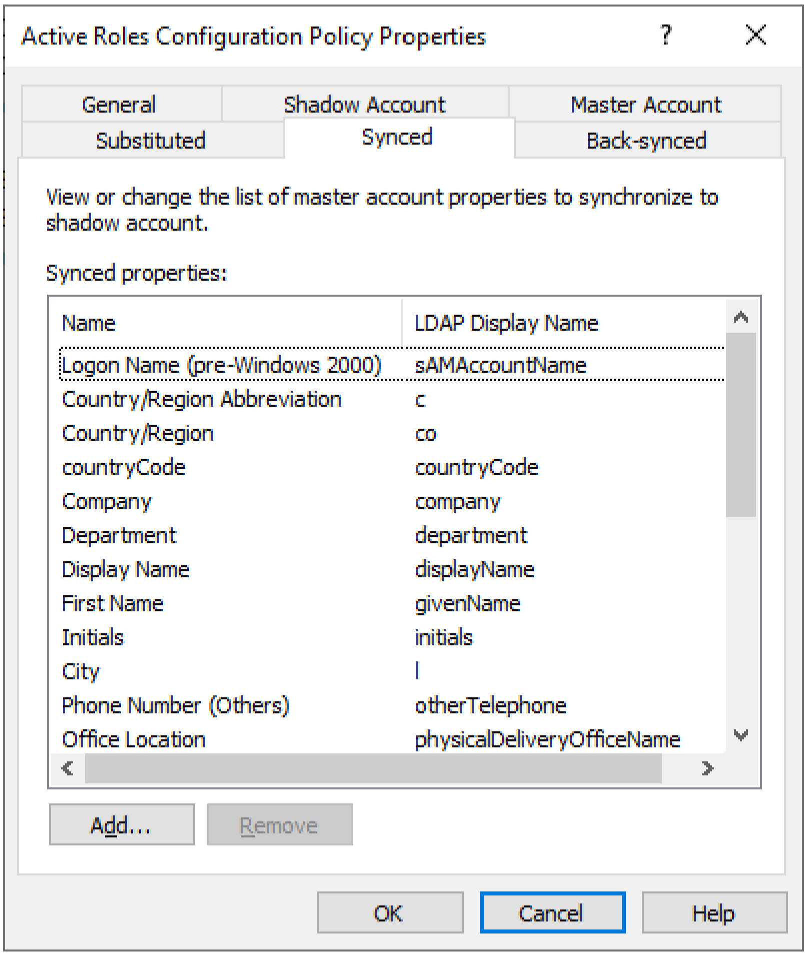
Built-in Policy - ERFM - Mailbox Management Properti...

Policies included in this Policy Object:

Policy Description
ERFM - Mailbox Management

4. (Optional) To view or modify the list of properties synchronized by the **ERFM - Mailbox Management** policy, click **Synced**.

Figure 145: Active Roles Console– Viewing or modifying the synchronized properties of linked mailboxes



Active Roles Configuration Policy Properties

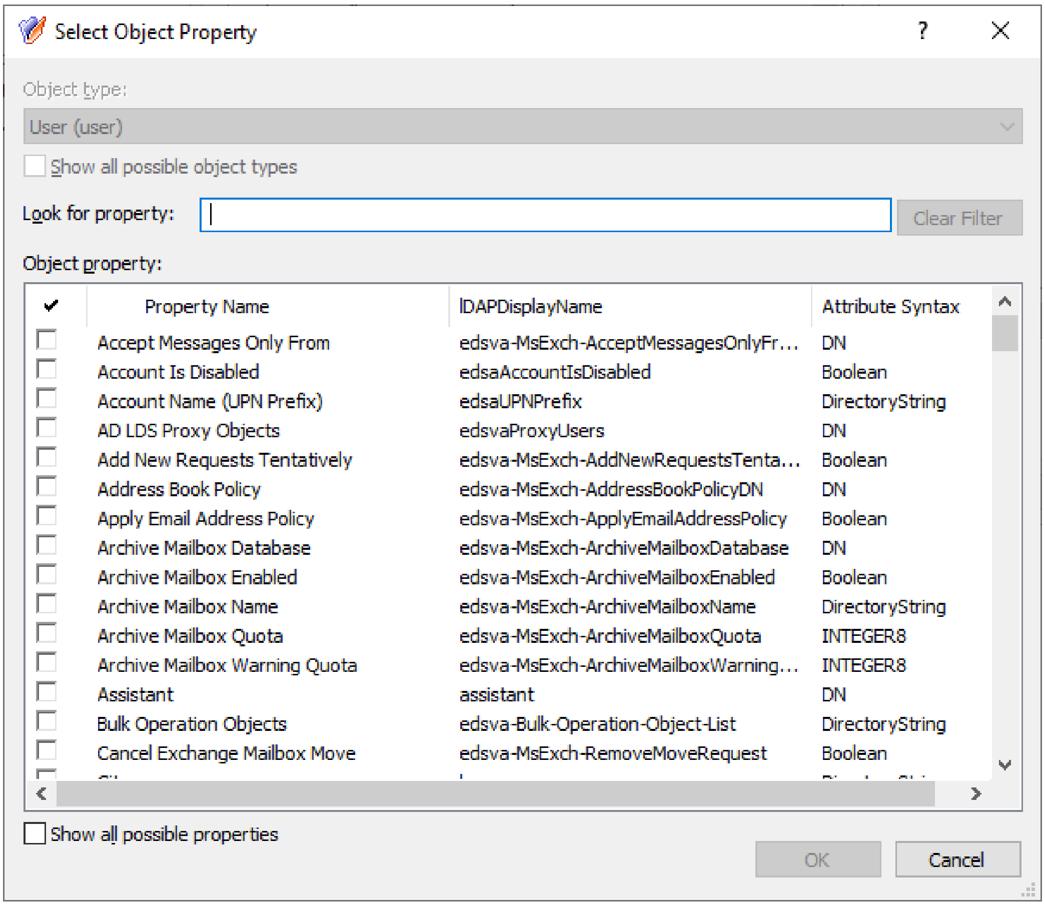
View or change the list of master account properties to synchronize to shadow account.

Synced properties:

Name	LDAP Display Name
Logon Name (pre-Windows 2000)	sAMAccountName
Country/Region Abbreviation	c
Country/Region	co
countryCode	countryCode
Company	company
Department	department
Display Name	displayName
First Name	givenName
Initials	initials
City	l
Phone Number (Others)	otherTelephone
Office Location	physicalDeliveryOfficeName

- To add a new property to the list, click **Add**. Then, in the **Select Object Property** window, select the property (or properties) you wish to add, and click **OK**.

Figure 146: Active Roles Console– Adding or removing synchronized properties for linked mailboxes



Object type:

User (user)

□ Show all possible object types

Look for property:

Object property:

✓	Property Name	lDAPDisplayName	Attribute Syntax
□	Accept Messages Only From	edsva-MsExch-AcceptMessagesOnlyFr...	DN
□	Account Is Disabled	edsaAccountIsDisabled	Boolean
□	Account Name (UPN Prefix)	edsaUPNPrefix	DirectoryString
□	AD LDS Proxy Objects	edsvaProxyUsers	DN
□	Add New Requests Tentatively	edsva-MsExch-AddNewRequestsTenta...	Boolean
□	Address Book Policy	edsva-MsExch-AddressBookPolicyDN	DN
□	Apply Email Address Policy	edsva-MsExch-ApplyEmailAddressPolicy	Boolean
□	Archive Mailbox Database	edsva-MsExch-ArchiveMailboxDatabase	DN
□	Archive Mailbox Enabled	edsva-MsExch-ArchiveMailboxEnabled	Boolean
□	Archive Mailbox Name	edsva-MsExch-ArchiveMailboxName	DirectoryString
□	Archive Mailbox Quota	edsva-MsExch-ArchiveMailboxQuota	INTEGER8
□	Archive Mailbox Warning Quota	edsva-MsExch-ArchiveMailboxWarning...	INTEGER8
□	Assistant	assistant	DN
□	Bulk Operation Objects	edsva-Bulk-Operation-Object-List	DirectoryString
□	Cancel Exchange Mailbox Move	edsva-MsExch-RemoveMoveRequest	Boolean

□ Show all possible properties

TIP: If you cannot find the property you are looking for, select **Show all possible properties** to list all available properties.

- To remove a property (or properties) from the list, select the property (or properties), click **Remove**, and confirm the removal.
- To apply your changes, click **OK**.

5. (Optional) To view or modify the list of back synchronized properties, click **Back-synced**.

   - To add a new property to the list, click **Add**. Then, in the **Select Object Property** window, select the property (or properties) you wish to add, and click **OK**.

     TIP: If you cannot find the property you are looking for, select **Show all possible properties** to list all available properties.

   - To remove a property (or properties) from the list, select the property (or properties), click **Remove**, and confirm the removal.

- To apply your changes, click **OK**.

6. (Optional) To view or modify the list of substituted properties, click **Substituted**.

   - To add a new property to the list, click **Add**. Then, in the **Select Object Property** window, select the property (or properties) you wish to add, and click **OK**.

     > TIP: If you cannot find the property you are looking for, select **Show all possible properties** to list all available properties.

   - To remove a property (or properties) from the list, select the property (or properties), click **Remove**, and confirm the removal.

   - To apply your changes, click **OK**.

Creating a linked mailbox for a new user

After Exchange Resource Forest Management (ERFM) is set up for your organization, you can configure linked mailboxes for new users in the Active Roles Web Interface.

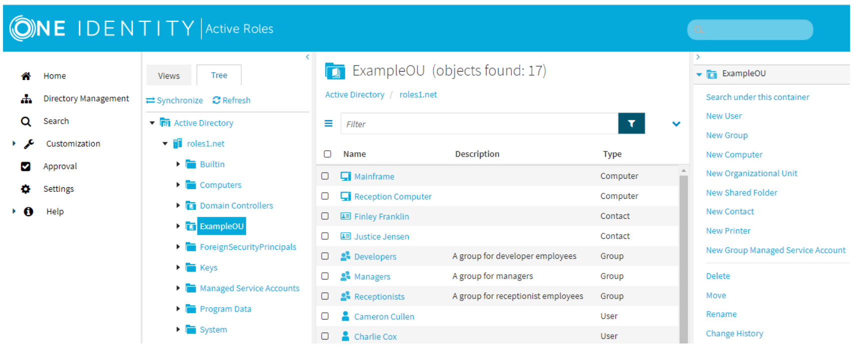
Prerequisites

Make sure that all mandatory requirements listed in Prerequisites of configuring linked mailboxes have been performed in your organization. Otherwise, linked mailboxes will not be available for your users.

To create a new user with a linked mailbox

1. In the Active Roles Web Interface, under **Directory Management** > **Tree** > **Active Directory**, navigate to the OU for which ERFM is configured.

   Figure 147: Active Roles Web Interface – Navigating to the OU supporting linked mailboxes



□	Name	Description	Type
□	Mainframe		Computer
□	Reception Computer		Computer
□	Finley Franklin		Contact
□	Justice Jensen		Contact
□	Developers	A group for developer employees	Group
□	Managers	A group for managers	Group
□	Receptionists	A group for receptionist employees	Group
□	Cameron Cullen		User
□	Charlie Cox		User

2. In the list of actions available for the selected OU, click **New User**.

3. In the **General** step, specify the following information as required by your organization:

   - **First name**: The first name of the user.
   - **Last name**: The last name of the user.
   - (Optional) **Initials**: The initials of the user.
   - **Name**: The fully-qualified user name of the user. By default, Active Roles automatically fills this property based on the specified **First name**, **Last name**, and **Initials**.
   - **Display name**: The name of the user as it will appear in Active Directory. By default, Active Roles automatically fills this property based on the specified **Name**.
   - **User logon name**: The user name used to log in to the domain. The **User logon name** also contains a user principal name (UPN) suffix. To configure the appropriate UPN suffix, use the drop-down button and select the appropriate domain for the user.

     > NOTE: The list contains:
     >
     > - The full DNS name of the current domain.
     > - The full DNS name of the root domain of the current forest.
     > - Any alternative UPN suffixes created via the Active Directory Domains and Trusts console.

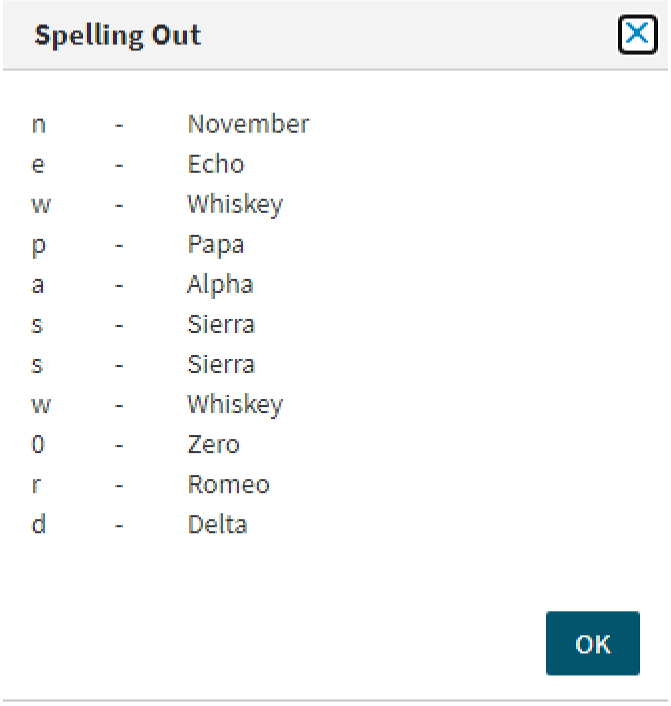
   - (Optional) **User logon name (pre-Windows 2000)**: The user name used to log in to the domain, following the pre-Windows 2000 logon name format: `<domain-name>\<user-name>`. By default, Active Roles automatically fills this property based on the specified **User logon name**.

4. In the **Account** step, specify the security settings of the user:

   - **Password** and **Confirm password**: The initial password of the user and the corresponding password confirmation field. You can specify the password either manually, or **Generate** one with Active Roles that follows the password policy requirements of your organization.

     To clear the specified password, click **Clear**. To spell out each character of the password for clarification, click **Spell out**.

Figure 148: Active Roles Web Interface – Spelling out the characters of the generated or specified password



Spelling Out

n	-	November
e	-	Echo
w	-	Whiskey
p	-	Papa
a	-	Alpha
s	-	Sierra
s	-	Sierra
w	-	Whiskey
0	-	Zero
r	-	Romeo
d	-	Delta

- **Account options**: Use these options to specify additional security settings for the user (for example, to have them change the configured password during their next login attempt, or have the configured password expire after some time). If you want to enable the created user account later for increased security (for example, because the new user joins later to your organization), select **Account is disabled**.

5. In the **Create Mailbox** step, configure the following settings:
   - **Create an Exchange mailbox**: Make sure that this setting is selected.
   - **Alias**: Specify the Microsoft Exchange alias of the new mailbox. By default, Active Roles generates the mailbox alias from the value specified for the **General** > **User logon name** property of the user.
   - **Mailbox database**: If all the mandatory prerequisites of this procedure are met, Active Roles must indicate in this field the default mailbox database of the Microsoft Exchange server deployed in the resource forest.

     If this field does not point to the Exchange server of the resource forest for any reason, click **Browse** and select the Exchange server of the resource forest.
6. (Optional) **Retention policy**: If your organization has any retention policies configured for user mailboxes as part of its messaging records management (MRM) strategy, apply them to the new mailbox by selecting this setting and clicking **Browse** to select the appropriate policy or policies.
7. (Optional) **Exchange ActiveSync mailbox policy**: If your organization has any Exchange ActiveSync mailbox policies configured for mobile devices, then apply them

to the new mailbox by selecting this setting and clicking **Browse** to select the appropriate policy or policies.

8. (Optional) **Address book policy**: If your organization has any address book policies configured for global address list (GAL) segmentation, apply them to the new mailbox by selecting this setting and clicking **Browse** to select the appropriate policy or policies.
9. (Optional) To open the settings of the new user immediately after finishing the procedure, select **Open properties for this object when I click Finish**.
10. To apply your changes, click **Finish**.

Active Roles then creates the new user with the following resources:

- A new master user account in the OU of the account forest you navigated to at the beginning of this procedure.
- A new shadow account and a linked mailbox in the resource forest, either in the default **Users** container or in the container you manually specified in Changing the location of the shadow accounts.

Creating a linked mailbox for an existing user with no mailbox

After Exchange Resource Forest Management (ERFM) is set up for your organization, you can configure linked mailboxes for existing users without mailboxes in the Active Roles Web Interface.

> NOTE: If your organization has any existing users whose user mailboxes were created before configuring linked mailboxes, you cannot configure new linked mailboxes for those users. Instead, you must convert their existing user mailboxes to linked mailboxes. For more information, see Converting a user mailbox to a linked mailbox.

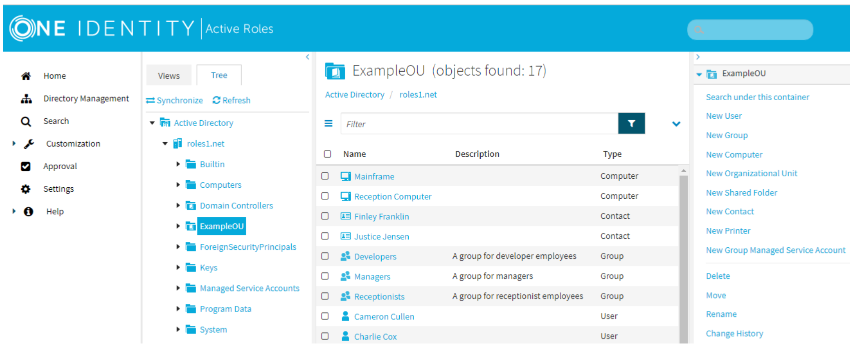
Prerequisites

Make sure that all mandatory requirements listed in Prerequisites of configuring linked mailboxes have been performed in your organization. Otherwise, linked mailboxes will not be available for your users.

To create a linked mailbox for an existing user

1. In the Active Roles Web Interface, under **Directory Management** > **Tree** > **Active Directory**, navigate to the OU for which ERFM is configured.

Figure 149: Active Roles Web Interface – Navigating to the OU supporting linked mailboxes



Name	Description	Type
Mainframe		Computer
Reception Computer		Computer
Finley Franklin		Contact
Justice Jensen		Contact
Developers	A group for developer employees	Group
Managers	A group for managers	Group
Receptionists	A group for receptionist employees	Group
Cameron Cullen		User
Charlie Cox		User

2. Select the user account for which you want to configure the linked mailbox.
3. To start configuring the mailbox for the user, click **New User Mailbox**.
4. In the **Mailbox Settings** step, configure the following settings:
    - **Alias**: Specify the Microsoft Exchange alias of the new mailbox. By default, Active Roles generates the mailbox alias from the value specified for the **General** > **User logon name** property of the user.
    - **Mailbox database**: If all the mandatory prerequisites of this procedure are met, Active Roles must indicate in this field the default mailbox database of the Microsoft Exchange server deployed in the resource forest.

      If this field does not point to the Exchange server of the resource forest for any reason, click **Browse** and select the Exchange server of the resource forest.
    - (Optional) **Retention policy**: If your organization has any retention policies configured for user mailboxes as part of its messaging records management (MRM) strategy, apply them to the new mailbox by selecting this setting and clicking **Browse** to select the appropriate policy or policies.
    - (Optional) **Exchange ActiveSync mailbox policy**: If your organization has any Exchange ActiveSync mailbox policies configured for mobile devices, then apply them to the new mailbox by selecting this setting and clicking **Browse** to select the appropriate policy or policies.
    - (Optional) **Address book policy**: If your organization has any address book policies configured for global address list (GAL) segmentation, apply them to the new mailbox by selecting this setting and clicking **Browse** to select the appropriate policy or policies.
5. To apply your changes, click **Finish**.

Active Roles then creates a new shadow account and a linked mailbox in the resource forest, either in the default **Users** container or in the container you manually specified in Changing the location of the shadow accounts.

Modifying the Exchange properties of a linked mailbox

After Exchange Resource Forest Management (ERFM) is set up for your organization, you can modify the Exchange properties of an existing linked mailbox in the Active Roles Web Interface by selecting the master user account in the account forest, and opening the **Exchange Properties** window. This is typically required in case of organizational or employment status changes.

When you modify the available Exchange properties this way, Active Roles redirects the change requests of the Exchange mailbox properties from the master account that you have opened to the shadow user account in the Exchange forest.

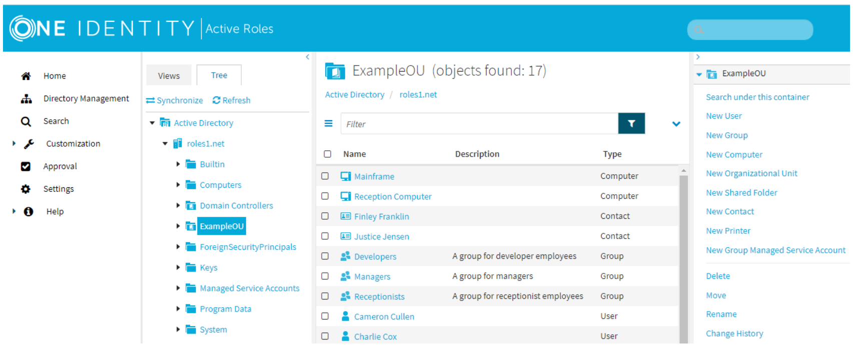
> TIP: For more information on the Exchange properties synchronized between the master and shadow accounts by ERFM, and how to modify the list of synchronized properties, see Configuring the synchronized, back synchronized or substituted properties of linked mailboxes.

> NOTE: If your environment has a large number of Microsoft Exchange mailboxes (or a complex Microsoft Exchange deployment), Active Roles may retrieve the properties of users with Exchange mailboxes slower than for users without Exchange mailboxes.
>
> To solve this problem, enable a performance fix by creating a new registry key as described in Knowledge Base Article 4336544:
>
> 1. On the machine(s) running the Administration Service and the Web Interface, launch the Windows Registry Editor.
> 2. In the Registry Editor, navigate to the following registry path:
>
>   `HKEY_LOCAL_ MACHINE\SOFTWARE\One Identity\Active Roles\Configuration`
> 3. Create a new **DWORD (32-bit) Value** named `PerformanceFlag`.
> 4. Double-click the new **PerformanceFlag** DWORD, and set its **Value data** to `1`.
> 5. To apply the fix, restart the Active Roles Administration Service and IIS. If the fix is enabled successfully, the following Active Roles event log with Event ID 2508 will appear in the Event Viewer:
>
>    ```
>    Performance flag value set to 1.
>    ```
> 6. (Optional) To deactivate the fix later, set the **Value data** of the **PerformanceFlag** DWORD to 0.
>
> The **PerformanceFlag** registry key accepts only a value of `1` (to activate the fix) or `0` (to deactivate it).

To view or modify the Exchange properties of a linked mailbox

1. In the Active Roles Web Interface, under **Directory Management** > **Tree** > **Active Directory**, navigate to the OU for which ERFM is configured.

   Figure 150: Active Roles Web Interface – Navigating to the OU supporting linked mailboxes



Name	Description	Type
Mainframe		Computer
Reception Computer		Computer
Finley Franklin		Contact
Justice Jensen		Contact
Developers	A group for developer employees	Group
Managers	A group for managers	Group
Receptionists	A group for receptionist employees	Group
Cameron Cullen		User
Charlie Cox		User

2. Select the master user account whose Exchange properties you want to modify and click **Exchange Properties**.
3. View or change the following mailbox settings as you need:
   - (Optional) **General**: View and configure the general email settings, for example the **First name**, **Last name**, or **User logon name**.
   - (Optional) **E-mail Addresses**: View and configure email addresses for the selected user.
   - (Optional) **Mailbox Features**: View and configure various Exchange mailbox features for the user, for example, mobile device synchronization features, web application access, or email messaging protocols.
   - (Optional) **Mail Flow Settings**: View and configure rules for the emails that the user sends or receives via the Exchange server of your organization, for example, message size restrictions or delivery and forwarding settings.
   - (Optional) **Mailbox Settings**: View and configure Messaging Records Management (MRM) settings for the user.
4. To apply your changes, click **Save**.

After you save your changes, Active Roles applies the modifications on the shadow user account associated with the master user account.

> TIP: To verify if your changes have already been synchronized, in the Active Roles Web Interface, under **Directory Management** > **Tree** > **Active Directory**, navigate to the resource forest, select the shadow account associated with the master account you modified, and click **Change History**.

Configuring a user with a linked mailbox for managing mail-enabled groups

With Active Roles, you can specify a user for group membership management tasks by selecting the group in the Active Roles Web Interface, and clicking **General Properties** > **Managed by**. However, both the group and the user specified as the group manager must be in the same Active Directory (AD) forest.

If Exchange Resource Forest Management (ERFM) is configured for your organization, user accounts and mail-enabled groups may be located in different forests: the user accounts are stored in the account forest, while the mail-enabled groups are in the resource forest. In such cases, you can assign a user for group management by specifying the shadow account of the user as the group manager instead of their master user account.

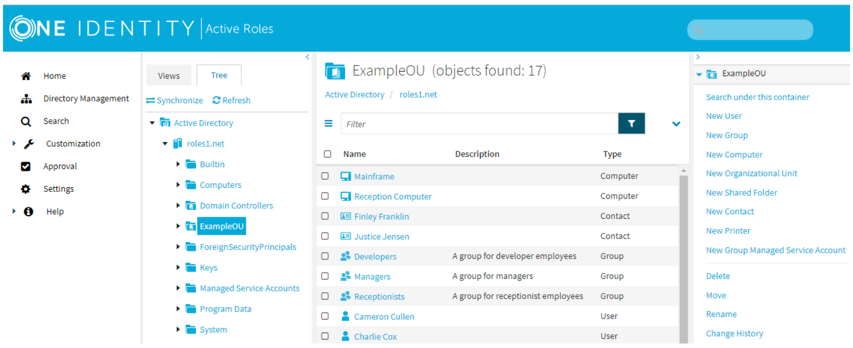
Doing so will result in Active Roles synchronizing the group management settings of the shadow account to the master account, allowing the master account to add or remove members from the specified group, even if it is located in a different forest.

> NOTE: If your environment has a large number of Microsoft Exchange mailboxes (or a complex Microsoft Exchange deployment), Active Roles may retrieve the properties of users with Exchange mailboxes slower than for users without Exchange mailboxes.
>
> To solve this problem, enable a performance fix by creating a new registry key as described in Knowledge Base Article 4336544:
>
> 1. On the machine(s) running the Administration Service and the Web Interface, launch the Windows Registry Editor.
> 2. In the Registry Editor, navigate to the following registry path:
>
>   `HKEY_LOCAL_ MACHINE\SOFTWARE\One Identity\Active Roles\Configuration`
> 3. Create a new **DWORD (32-bit) Value** named `PerformanceFlag`.
> 4. Double-click the new **PerformanceFlag** DWORD, and set its **Value data** to `1`.
> 5. To apply the fix, restart the Active Roles Administration Service and IIS. If the fix is enabled successfully, the following Active Roles event log with Event ID 2508 will appear in the Event Viewer:
>
>    ```
>    Performance flag value set to 1.
>    ```
> 6. (Optional) To deactivate the fix later, set the **Value data** of the **PerformanceFlag** DWORD to 0.
>
> The **PerformanceFlag** registry key accepts only a value of `1` (to activate the fix) or `0` (to deactivate it).

To configure a user with a linked mailbox for group membership management

1. In the Active Roles Web Interface, under **Directory Management** > **Tree** > **Active Directory**, navigate to the OU for which ERFM is configured.

   Figure 151: Active Roles Web Interface – Navigating to the OU supporting linked mailboxes

Name	Description	Type
Mainframe		Computer
Reception Computer		Computer
Finley Franklin		Contact
Justice Jensen		Contact
Developers	A group for developer employees	Group
Managers	A group for managers	Group
Receptionists	A group for receptionist employees	Group
Cameron Cullen		User
Charlie Cox		User

2. In the container of your users, select the user you want to assign as a group manager.

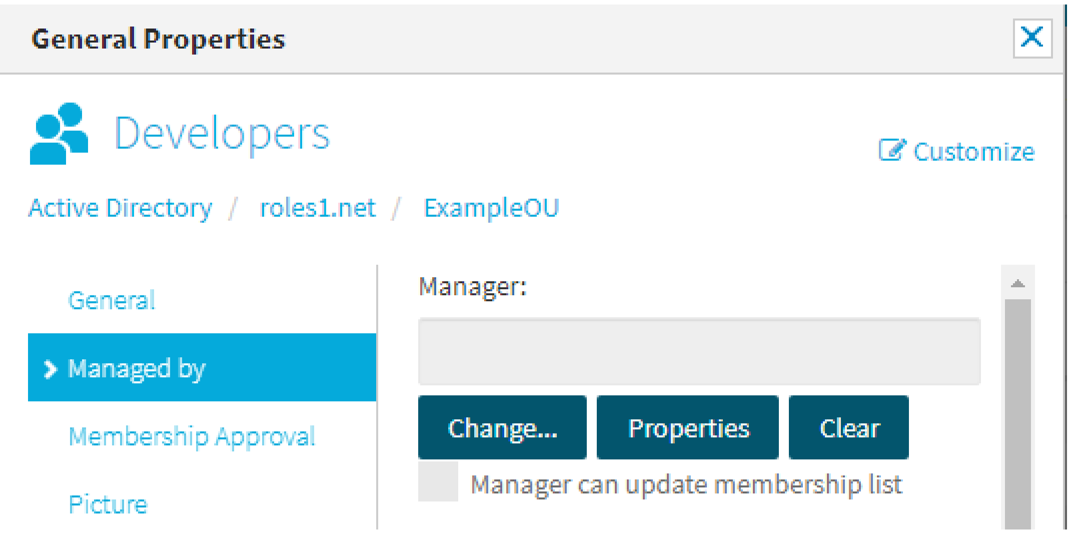
3. To view the general Exchange settings of the user, click **Exchange Properties** > **Shadow Account** > **Properties**.

4. Open the **General Properties** > **Account** tab, and take note of the **User logon name (pre-Windows 2000)** value of the shadow account. You will need to specify this user logon name for the group later in this procedure.

5. In the Active Roles Web Interface, under **Directory Management** > **Tree** > **Active Directory**, navigate to your resource forest containing the Exchange server and the shadow accounts.

6. Select the group whose management settings you want to configure. Then, to open the group management settings, click **General Properties** > **Managed by**.

Figure 152: Active Roles Web Interface – Opening the group management settings of a group via General Properties > Managed by



7. To specify a new group manager, click **Change**. This opens the **Select Object** dialog, allowing you to specify the manager account.
8. In the **Select Object** dialog, specify the **User logon name** of the shadow account that you have noted down earlier in the procedure, then click the search button. Once the dialog lists the user, select it and click **OK**.

   > TIP: If your search returns no results, then double check that the specified user logon name is correct, and make sure that the **Search in** drop-down list is set to the resource forest where the shadow account is stored.

9. After the user is displayed in the **Manager** text box, click **Save**. Then, to make sure that the user receives all group management permissions, select **Manager can update membership list** and click **Save** again.

> NOTE: The master account of the specified user will receive the configured group administration permissions during the next run of the **ERFM - Mailbox Management** scheduled task. To make sure that the group management permissions of the shadow account are immediately synchronized to its master account, run the scheduled task manually. For more information, see Configuring the ERFM Mailbox Management scheduled task.

Converting a user mailbox to a linked mailbox

Once Exchange Resource Forest Management (ERFM) is set up for your organization, you can convert the existing user mailboxes of your users to linked mailboxes. This is typically required if your organization had already contained users with regular Exchange user mailboxes before configuring linked mailboxes with ERFM.

To convert a user mailbox to a linked mailbox

1. In the Active Roles Web Interface, under **Directory Management** > **Tree** > **Active Directory**, navigate to your resource forest containing the Exchange server and the shadow accounts.
2. In the container of your users, select the user whose mailbox you want to convert.
3. To start the mailbox conversion, in the list of actions available for the selected mailbox, click **Convert to Linked Mailbox**.
4. Under **Linked master account**, click **Change** and select the user in the account forest whose mailbox you are converting. To do so, specify the account forest in the **Search in** field, then enter the name of the user in the **Search** field. Once the **Select Object** window lists the user, select it and click **OK**.
5. To apply your changes, click **Finish**.

Active Roles then performs the following actions:

1. It changes the specified user mailbox to a linked mailbox.
2. It specifies the user selected in the account forest as the master user account.
3. It changes the user associated with the mailbox in the resource forest to a shadow account.

Converting a linked mailbox to a user mailbox

You can convert existing linked mailboxes configured with Exchange Resource Forest Management (ERFM) to user mailboxes. This is typically required during organizational changes or IT infrastructure migrations.

When you convert an existing linked mailbox to a user mailbox, Active Roles performs the following changes:

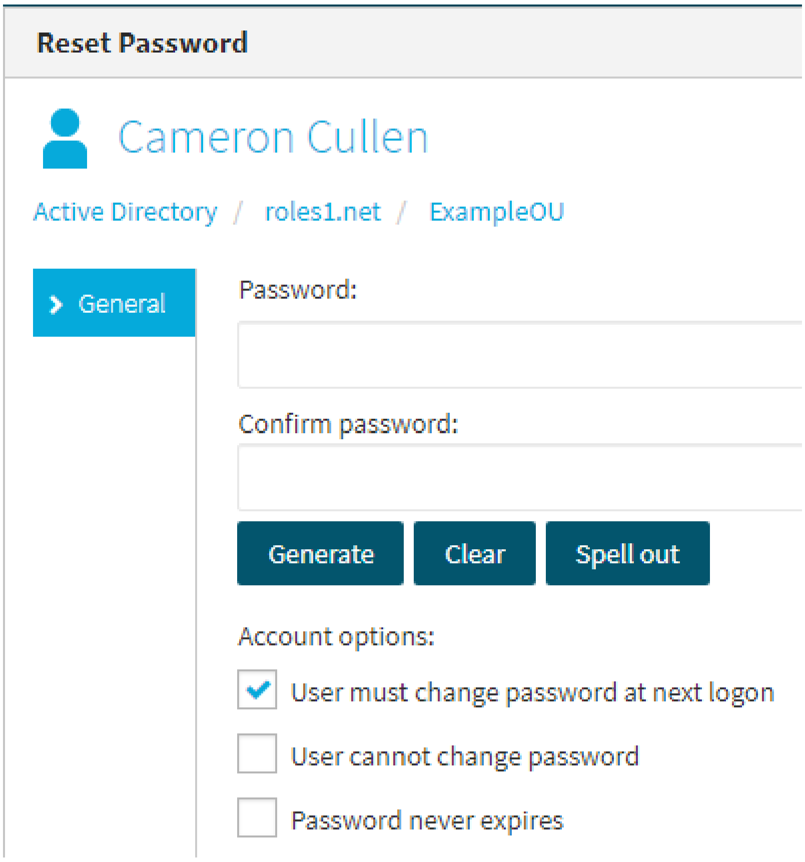
1. The former master user account in the account forest becomes an external user, and can no longer access the mailbox.
2. The former shadow account becomes the new user account associated with the mailbox in the resource forest.

To convert a linked mailbox to a user mailbox

1. In the Active Roles Web Interface, under **Directory Management** > **Tree** > **Active Directory**, navigate to your resource forest containing the Exchange server and the shadow accounts.
2. In the container of your users, select the user whose mailbox you want to convert.

3. To start the mailbox conversion, in the list of actions available for the selected mailbox, click **Convert to User Mailbox**.
4. To apply your changes, click **OK**.
5. Following the mailbox conversion, the user mailbox will be in a disabled state. To enable it, in the list of actions available for the selected mailbox, click **Enable Account**.
6. After the account is enabled, you must also reset the account password. To do so, in the list of actions available for the selected mailbox, click **Reset Password**.
7. In the **Reset Password** window, configure the following settings:

Figure 153: Active Roles Web Interface – Resetting the password of a converted user mailbox


Reset Password

Cameron Cullen

Active Directory / roles1.net / ExampleOU

Password:

Confirm password:

Account options:

☑ User must change password at next logon

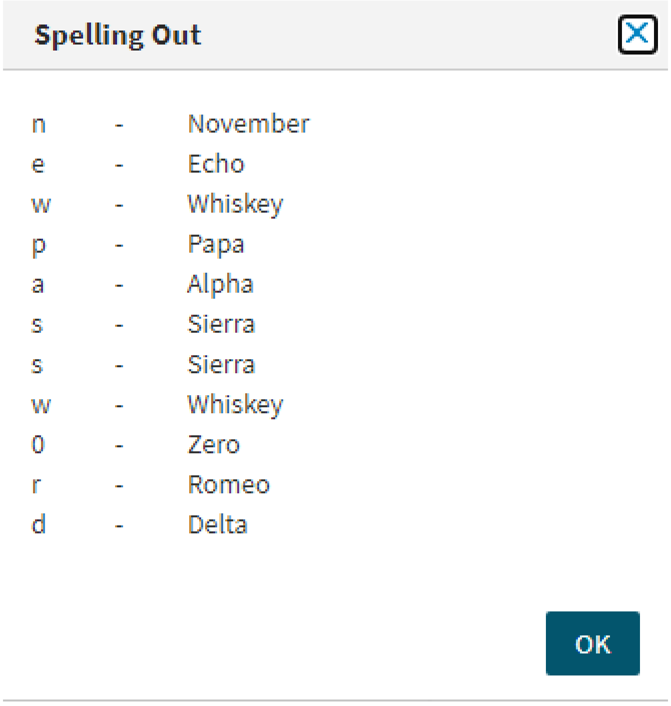
□ User cannot change password

□ Password never expires

- **Password** and **Confirm password**: The initial password of the user and the corresponding password confirmation field. You can specify the password either manually, or **Generate** one with Active Roles that follows the password policy requirements of your organization.

  To clear the specified password, click **Clear**. To spell out each character of the password for clarification, click **Spell out**.

Figure 154: Active Roles Web Interface – Spelling out the characters of the generated or specified password



Spelling Out

n	-	November
e	-	Echo
w	-	Whiskey
p	-	Papa
a	-	Alpha
s	-	Sierra
s	-	Sierra
w	-	Whiskey
0	-	Zero
r	-	Romeo
d	-	Delta

- **Account options**: Use these options to specify additional security settings for the user (for example, to have them change the configured password during their next login attempt, or have the configured password expire after some time).

8. To apply your changes, click **Finish**.

Deprovisioning a user with a linked mailbox

You can deprovision users with linked mailboxes by using the **Deprovision** action of the Active Roles Web Interface. When doing so, Active Roles, by default:

- Disables the user account, and resets the user password to a random value.
- Removes the user from all assigned security and distribution groups.
- Disables the linked mailbox.
- Disables the home folder of the user.

Optionally, deprovisioning also lets you relocate deprovisioned users to a specific folder, and even schedule them for deletion after some time.

One Identity typically recommends deprovisioning users instead of deleting them and their mailboxes, if the user is affected by an organizational change, suspension, or longer periods of time off work. You can undo the effects of deprovisioning later and reinstate the user with the **Undo Deprovisioning** action of the Active Roles Web Interface.

When a user with a linked mailbox configured via Exchange Resource Forest Management (ERFM) is deprovisioned, Active Roles runs all deprovisioning policies applied to the Active Directory (AD) container holding the shadow account, including any mailbox deprovisioning policies in effect in your organization.

> TIP: Besides deprovisioning, you can also disable users by using the **Disable Account** action. Disabling a user account with a linked mailbox prevents the user from logging in and accessing their resources, but it does not remove the user from their groups, and does not disable the mailbox and the user home folder. One Identity recommends disabling user accounts instead of completely deprovisioning them if the organization still needs to access the user resources (such as the home folder or the mailbox).
>
> To disable a user account, in the Active Roles Web Interface, navigate to the OU where your user is stored in the **Directory Management** > **Tree** > **Active Directory** node, select the user, and in the list of actions available for the selected user, click **Disable Account**.

Prerequisites

To deprovision users with linked mailboxes configured via ERFM, make sure that the mailbox deprovisioning policies of your organization (for example, the built-in **Exchange Mailbox Deprovisioning** policy) are applied to the container that holds the shadow accounts in the resource forest, instead of the container of the master user accounts in the account forest. By default, the deprovisioning workflow runs the following built-in policies for users with linked mailboxes:

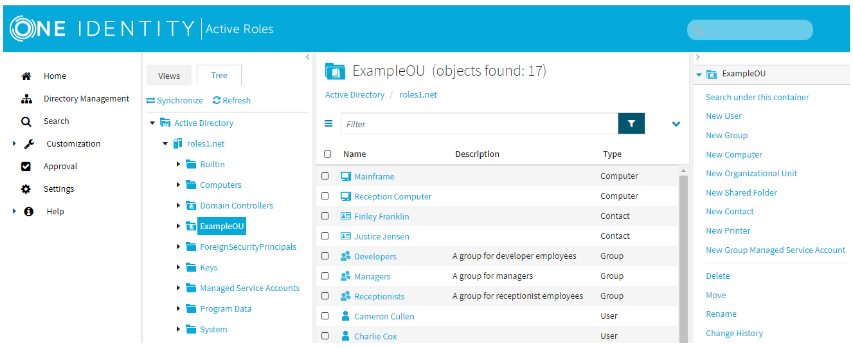
- Built-in policy - User Default Deprovisioning
- Built-in policy - ERFM - Mailbox Management

For more information on deprovisioning policies, see Deprovisioning Policy Objects.

To deprovision a user with a linked mailbox

1. In the Active Roles Web Interface, under **Directory Management** > **Tree** > **Active Directory**, navigate to the OU for which ERFM is configured.

Figure 155: Active Roles Web Interface – Navigating to the OU supporting linked mailboxes



ExampleOU (objects found: 17)

Active Directory / roles1.net

Name	Description	Type
Mainframe		Computer
Reception Computer		Computer
Finley Franklin		Contact
Justice Jensen		Contact
Developers	A group for developer employees	Group
Managers	A group for managers	Group
Receptionists	A group for receptionist employees	Group
Cameron Cullen		User
Charlie Cox		User

2. Select the master user account that you want to deprovision, and in the list of available actions, click **Deprovision**.
3. To confirm deprovisioning, click **OK**.

Active Roles then performs deprovisioning of the master user account and its associated shadow account. After the process is completed, it displays the operation summary of deprovisioning.

> TIP: To verify that Active Roles also deprovisioned the shadow account, in the Active Roles Web Interface, navigate to the user container of your shadow accounts in the **Directory Management** > **Tree** > **Active Directory** node of the resource forest, select the shadow account, and from the list of actions available for the shadow account, click **Deprovisioning Results**.

Undo deprovisioning for a user with a linked mailbox (re-provisioning)

You can undo the deprovisioning of users with linked mailboxes by using the **Undo Deprovisioning** action of the Active Roles Web Interface. When re-provisioning a user, Active Roles rolls back the changes of the deprovisioning policies in effect in your organization by:

- Restoring access to the user account.
- Reassigning the user to all security and distribution groups it was originally a member of.
- Re-enabling the linked mailbox.
- Re-enabling the home folder of the user.

Re-provisioning a deprovisioned user is typically required if the person is reinstated in your organization: for example, their suspension is lifted or they are returning to work from an extended leave.

When re-provisioning a user with a linked mailbox, Active Roles first re-provisions the master account, then re-provisions the shadow account. After the shadow account is re-provisioned, the linked mailbox also returns to its original provisioned state.

Prerequisites

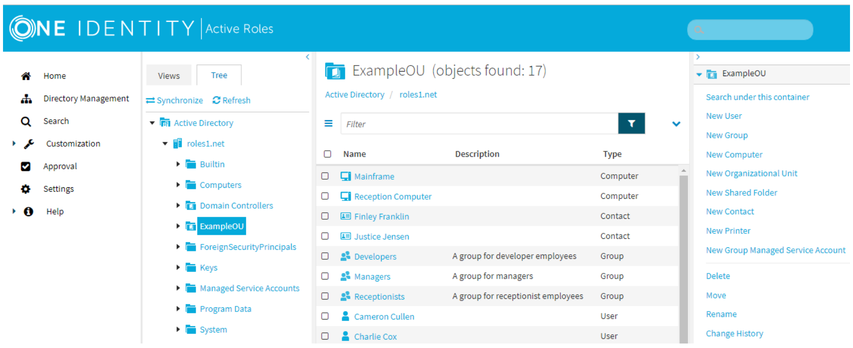
Active Roles can perform the **Undo Deprovisioning** action on the shadow account of a re-provisioned master account only if the Active Directory (AD) container holding the deprovisioned master accounts is in the scope of the **Built-in Policy - ERFM - Mailbox Management** policy, or a copy of that policy.

Therefore, if the deprovisioning workflow of your organization moves deprovisioned master accounts to a container separate from provisioned master accounts, make sure that the **Built-in Policy - ERFM - Mailbox Management** policy is also applied to the container where the deprovisioned master accounts are stored. For more information on configuring the policy, see Applying the ERFM Mailbox Management policy to an OU.

To undo the deprovisioning of a user with a linked mailbox

1. In the Active Roles Web Interface, under **Directory Management** > **Tree** > **Active Directory**, navigate to the OU for which ERFM is configured.

   Figure 156: Active Roles Web Interface – Navigating to the OU supporting linked mailboxes



ExampleOU (objects found: 17)

Active Directory / roles1.net

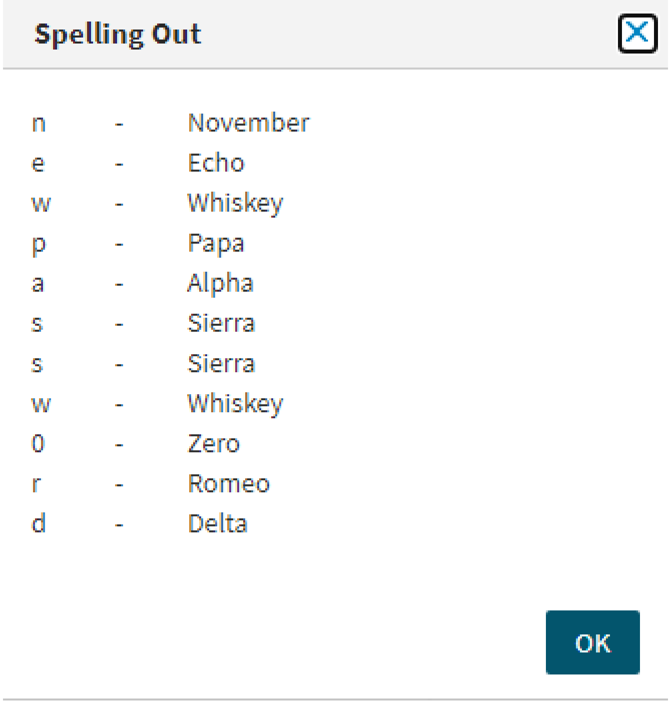
Name	Description	Type
Mainframe		Computer
Reception Computer		Computer
Finley Franklin		Contact
Justice Jensen		Contact
Developers	A group for developer employees	Group
Managers	A group for managers	Group
Receptionists	A group for receptionist employees	Group
Cameron Cullen		User
Charlie Cox		User

2. Select the deprovisioned master user account for which you want to undo deprovisioning. Then, in the list of available actions, click **Undo Deprovisioning**.
3. To confirm the restoration of the user account, click **OK**.
4. In the **Password Options** dialog, configure the password settings of the restored user:
   - **Leave the password unchanged**: The user account will be re-provisioned with its original password. Select this option if the user password will be reset by an organizational workflow outside the scope of Active Roles (for example by helpdesk, or another password management solution).
   - **Reset the password**: Select this option to immediately change the password of the re-provisioned user in Active Roles, either by specifying a new password

manually, or generating one that meets the password policy requirements of your organization.

To clear the specified password, click **Clear**. To spell out each character of the password for clarification, click **Spell out**.

Figure 157: Active Roles Web Interface – Spelling out the characters of the generated or specified password



Spelling Out

n	-	November
e	-	Echo
w	-	Whiskey
p	-	Papa
a	-	Alpha
s	-	Sierra
s	-	Sierra
w	-	Whiskey
0	-	Zero
r	-	Romeo
d	-	Delta

- **Account options**: Use these options to specify additional security settings for the user (for example, to have them change the configured password during their next login attempt, or have the configured password expire after some time).

5. To apply your changes, click **OK**.

Active Roles then re-provisions the master user account, the shadow user account and the linked mailbox.

Deleting a user with a linked mailbox

You can delete users with linked mailboxes by using the **Delete** action of the Active Roles Web Interface. When doing so, Active Roles deletes the master account, then disables the linked mailbox of the corresponding shadow account.

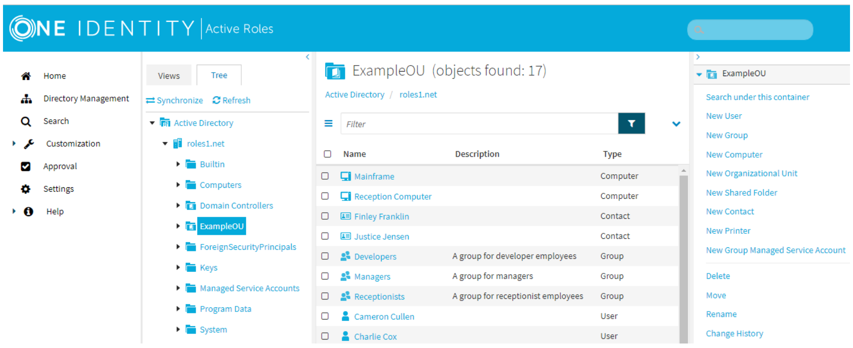
CAUTION: Hazard of data loss!

After you delete a user, it cannot be recovered. Therefore, One Identity recommends either deprovisioning or disabling user accounts before permanently deleting them. For more information on deprovisioning users with linked mailboxes, see Deprovisioning a user with a linked mailbox.

To delete a user with a linked mailbox

1. In the Active Roles Web Interface, under **Directory Management** > **Tree** > **Active Directory**, navigate to the OU for which ERFM is configured.

   Figure 158: Active Roles Web Interface – Navigating to the OU supporting linked mailboxes



ExampleOU (objects found: 17)

Active Directory / roles1.net

Filter

Name	Description	Type
Mainframe		Computer
Reception Computer		Computer
Finley Franklin		Contact
Justice Jensen		Contact
Developers	A group for developer employees	Group
Managers	A group for managers	Group
Receptionists	A group for receptionist employees	Group
Cameron Cullen		User
Charlie Cox		User

2. Select the master user account you want to delete.
3. In the list of actions available for the selected master account, click **Delete**.
4. To confirm deletion, in the pop-up dialog, click **OK**. To deprovision a user instead of permanently deleting them, click **Deprovision**.

Active Roles then deletes the master account in the account forest, then disables the linked mailbox of the associated shadow account in the resource forest.

Configuring remote mailboxes for on-premises users

Active Roles supports remote mailboxes, that is, managing cloud-only Exchange Online mailboxes assigned to on-premises users. Configuring cloud mailboxes for on-premises users allows your organization to store user mailboxes and mailbox data in the Exchange Online cloud, even if the user accounts in your organization are not hybrid or cloud-only user accounts.

By configuring remote mailboxes for your on-premises users, you can:

- Improve mailbox availability and accessibility.
- Improve data security by storing mailbox content in the Exchange Online cloud.
- Improve mailbox security via the integration of your on-premises Active Directory environment with Exchange Online.
- Use the flexibility and scalability of Exchange Online cloud mailboxes.
- Use the feature set of Microsoft 365 (such as real-time collaboration, document sharing, simultaneous editing, and so on).
- Use the administration automation features of Exchange Online.

To assign a remote mailbox for an on-premises user, you must set the user to a mail-enabled state, then assign a cloud email address to them in the Active Roles Console.

NOTE: Alternatively, Active Roles supports configuring remote mailboxes for existing on-premises users by converting them to hybrid users. After the conversion, you can configure and manage the remote mailbox settings of the new hybrid users either via the Active Roles Console or in the Active Roles Web Interface.

- For more information on converting an on-premises user to a hybrid user, see Sample Azure Hybrid Migration and *Converting an on-premises user with an Exchange mailbox to a hybrid Azure user* in the *Active Roles Web Interface User Guide*.
- For more information on managing the remote mailbox of a hybrid user, see *Viewing or modifying the Exchange Online properties of a hybrid Azure user* in the *Active Roles Web Interface User Guide*.

Assigning a remote mailbox to an on-premises user

You can assign a remote Exchange Online mailbox to an on-premises Active Directory (AD) user via the Active Roles Console.

Prerequisites

To assign a remote mailbox to an on-premises user, make sure that the following conditions are met.

- Your organization must have an on-premises Exchange server deployed in the same forest or domain where you want to configure remote mailboxes for on-premises users. The Exchange server will indicate later for Active Roles that the affected users have remote mailboxes.
- The on-premises user must already exist, and it cannot have a mailbox.
- The Exchange Online mailbox that you will assign to the on-premises user must already exist. To create a new cloud mailbox, use any of the following:
  - The Azure Portal.
  - The **Recipients** > **Mailboxes** menu of the Exchange Online Admin Center.
  - The `New-Mailbox` Windows PowerShell command.

  > ⚠ **CAUTION: After the cloud mailbox is created, it will enter into a 30-day grace period. To prevent deleting the remote mailbox after this period, you must assign an Exchange Online (Plan 2) license to it.**
  >
  > **To assign an Exchange Online license to the cloud mailbox, in the Microsoft 365 Admin Center, select the user, then navigate to Manage product licenses.**

- Note down the value of the Microsoft Online Services ID (that is, the `MicrosoftOnlineServicesID` attribute) of the remote mailbox. You will need to specify the value of this attribute to connect the on-premises user with the remote mailbox. You can check the value of the attribute either in the Microsoft 365 Admin Center, or via the `Get-User` PowerShell command.

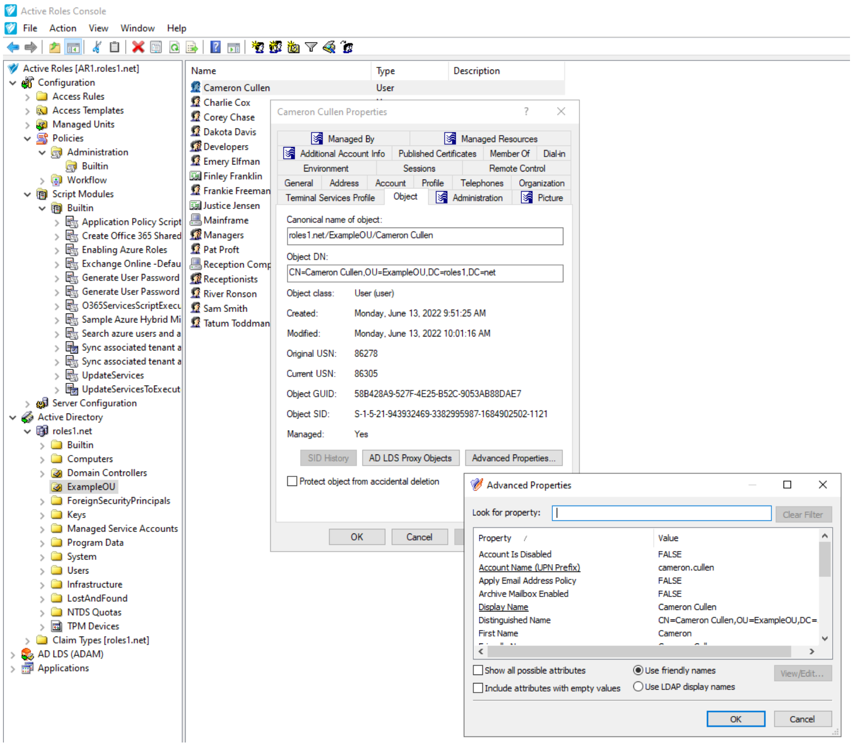
  > TIP: If the remote mailbox has multiple aliases configured, the `MicrosoftOnlineServicesID` attribute always takes the value of the primary email address and user name.

To assign a remote mailbox to an on-premises user

1. Open the **Advanced Properties** of the on-premises user for which you want to assign the remote mailbox. In the Active Roles Console, in the Active Directory (AD) tree, navigate to the Organizational Unit (OU) where the user is located, double-click

the user, then in the **Properties** window, click **Object** > **Advanced Properties**.

Figure 159: Active Roles Console – Opening the Advanced Properties of a user



Canonical name of object: roles1.net/ExampleOU/Cameron Cullen

Object DN: CN=Cameron Cullen,OU=ExampleOU,DC=roles1,DC=net

Object class: User (user)

Created: Monday, June 13, 2022 9:51:25 AM

Modified: Monday, June 13, 2022 10:01:16 AM

Original USN: 86278

Current USN: 86305

Object GUID: 58B428A9-527F-4E25-B52C-9053AB88DAE7

Object SID: S-1-5-21-943932469-3382995987-1684902502-1121

Managed: Yes

□ Protect object from accidental deletion

Look for property:

Property	Value
Account Is Disabled	FALSE
Account Name (UPN Prefix)	cameron.cullen
Apply Email Address Policy	FALSE
Archive Mailbox Enabled	FALSE
Display Name	Cameron Cullen
Distinguished Name	CN=Cameron Cullen,OU=ExampleOU,DC=
First Name	Cameron

□ Show all possible attributes

□ Include attributes with empty values

● Use friendly names

○ Use LDAP display names

2. Search for the **edsvaMsExchEnableRemoteMailRoutingAddress** property.

   TIP: To find the property faster, enter its name (or part of its name) in the **Look for property** field. If you cannot find the property, select **Show all possible attributes** and **Include attributes with empty values**, too.

   After you found the property, open its settings by double-clicking it.

3. In the **Edit Attribute** dialog, in **Value**, enter the value of the `MicrosoftOnlineServicesID` attribute (that is, the primary email address of the remote mailbox).

4. To apply your changes, click **OK** in each open window.

NOTE: Assigning a remote mailbox to an on-premises user may take up to 15 minutes to complete, with Active Roles attempting to establish connection up to 9 times. If the procedure fails (for example, because Active Roles cannot find the specified email address), Active Roles will log an error in the Windows Event Viewer under the **Applications and Services Logs** > **Active Roles Admin Service** category.

For more information on how to check if Active Roles could assign the remote mailbox to the user, see Verifying that a remote mailbox is assigned to an on-premises user.

TIP: If Active Roles could not assign the remote mailbox to the on-premises user within the expected time frame, perform the following troubleshooting steps:

- Check network connectivity.
- Check the status of the on-premises Exchange server and the Exchange Online service.
- Verify that the specified remote mailbox email address is correct.

Verifying that a remote mailbox is assigned to an on-premises user

Once you assigned an Exchange Online mailbox to an on-premises user, you can check if Active Roles completed the remote mailbox assignment by any of the following methods.

NOTE: Assigning a remote mailbox to an on-premises user may take up to 15 minutes to complete, with Active Roles attempting to establish connection up to 9 times. If the procedure fails (for example, because Active Roles cannot find the specified email address), Active Roles will log an error in the Windows Event Viewer under the **Applications and Services Logs** > **Active Roles Admin Service** category.

NOTE: If your environment has a large number of Microsoft Exchange mailboxes (or a complex Microsoft Exchange deployment), Active Roles may retrieve the properties of users with Exchange mailboxes slower than for users without Exchange mailboxes.

To solve this problem, enable a performance fix by creating a new registry key as described in Knowledge Base Article 4336544:

1. On the machine(s) running the Administration Service and the Web Interface, launch the Windows Registry Editor.
2. In the Registry Editor, navigate to the following registry path:

   `HKEY_LOCAL_ MACHINE\SOFTWARE\One Identity\Active Roles\Configuration`
3. Create a new **DWORD (32-bit) Value** named `PerformanceFlag`.
4. Double-click the new **PerformanceFlag** DWORD, and set its **Value data** to `1`.
5. To apply the fix, restart the Active Roles Administration Service and IIS. If the fix is enabled successfully, the following Active Roles event log with Event ID 2508 will appear in the Event Viewer:

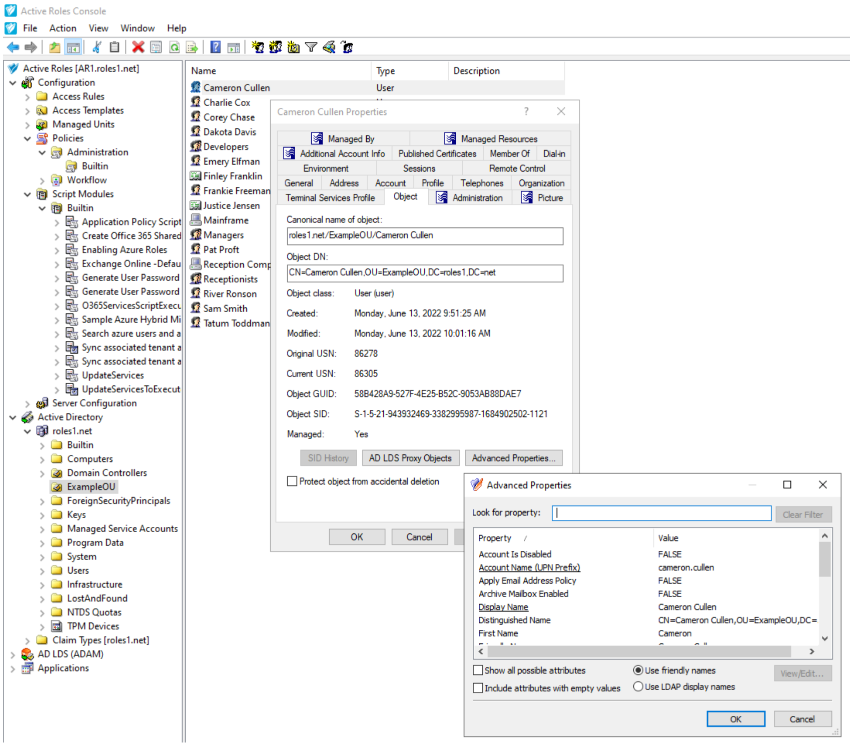
   ```
   Performance flag value set to 1.
   ```
6. (Optional) To deactivate the fix later, set the **Value data** of the **PerformanceFlag** DWORD to 0.

The **PerformanceFlag** registry key accepts only a value of `1` (to activate the fix) or `0` (to deactivate it).

To verify with the msExchRemoteRecipientType property whether Active Roles assigned the remote mailbox

1. Open the **Advanced Properties** of the on-premises user to which you assigned the remote mailbox. In the Active Roles Console, in the Active Directory (AD) tree, navigate to the Organizational Unit (OU) where the user is located, double-click the user, then in the **Properties** window, click **Object** > **Advanced Properties**.

   Figure 160: Active Roles Console – Opening the Advanced Properties of a user

   Canonical name of object: roles1.net/ExampleOU/Cameron Cullen

   Object DN: CN=Cameron Cullen,OU=ExampleOU,DC=roles1,DC=net

   Object class: User (user)

   Created: Monday, June 13, 2022 9:51:25 AM

   Modified: Monday, June 13, 2022 10:01:16 AM

   Original USN: 86278

   Current USN: 86305

   Object GUID: 58B428A9-527F-4E25-B52C-9053AB88DAE7

   Object SID: S-1-5-21-943932469-3382995987-1684902502-1121

   Managed: Yes

   □ Protect object from accidental deletion

   Look for property:

Property	Value
Account Is Disabled	FALSE
Account Name (UPN Prefix)	cameron.cullen
Apply Email Address Policy	FALSE
Archive Mailbox Enabled	FALSE
Display Name	Cameron Cullen
Distinguished Name	CN=Cameron Cullen,OU=ExampleOU,DC=
First Name	Cameron

   □ Show all possible attributes ● Use friendly names

   □ Include attributes with empty values ○ Use LDAP display names

2. Search for the **msExchRemoteRecipientType** property.

   TIP: To find the property faster, enter its name (or part of its name) in the **Look for property** field. If you cannot find the property, select **Show all possible attributes** and **Include attributes with empty values**, too.

3. Check the value of the **msExchRemoteRecipientType** property. For users with no mailboxes, the value of this property is empty. Once Active Roles finished assigning the remote Exchange Online mailbox to the user, the value of the property changes to 1.

To verify with the Exchange mailbox GUID whether Active Roles assigned the remote mailbox

1. Open Windows PowerShell, and connect to Exchange Online with the following command:

```
Connect-ExchangeOnline
```

2. In the Microsoft login popup that appears, log in with the Azure AD administrator account associated with the Azure tenant that stores the remote mailbox.
3. After logging in, in Windows PowerShell, fetch the identity information of the remote mailbox with the following command:

   ```
   Get-Mailbox -Identity '<email-address>' | Format-List ExchangeGUID
   ```

   `<email-address>` is the Microsoft Exchange alias of the mailbox.
4. Note down the value of the **ExchangeGUID** parameter.
5. In the Active Roles Console, open the **Advanced Properties** of the on-premises user to which you assigned the remote mailbox. In the Active Roles Console, in the Active Directory (AD) tree, navigate to the Organizational Unit (OU) where the user is located, double-click the user, then in the **Properties** window, click **Object** > **Advanced Properties**.
6. Search for the **msExchMailboxGuid** property.

   > TIP: To find the property faster, enter its name (or part of its name) in the **Look for property** field. If you cannot find the property, select **Show all possible attributes** and **Include attributes with empty values**, too.
7. Compare the value of the **msExchMailboxGuid** property with the Exchange GUID returned by the Get-Mailbox PowerShell command. If the two values match, Active Roles successfully assigned the remote mailbox to the on-premises user.

To verify with the RecipientType attribute of the user whether Active Roles assigned the remote mailbox

1. On the on-premises Microsoft Exchange server that stores the mailbox data of the user, open Windows PowerShell and run the following command:

   ```
   Get-User '<user-name>'
   ```

   `<user-name>` is the fully qualified user name of the on-premises user.
2. Check the value of the **RecipientType** property:
   - If the value is `MailUser`, Active Roles assigned the remote mailbox to the user.
   - If the value is `User`, the on-premises user does not have any mailboxes assigned to them.

> TIP: If Active Roles could not assign the remote mailbox to the on-premises user within the expected time frame, perform the following troubleshooting steps:
>
> - Check network connectivity.
> - Check the status of the on-premises Exchange server and the Exchange Online service.
> - Verify that the specified remote mailbox email address is correct.

20

Migrating Active Roles configuration with the Configuration Transfer Wizard

For large enterprises which implement a complex administrative structure using Active Roles, one of the greatest challenges becomes exporting Active Roles configuration from a test environment to a production environment.

With Active Roles Configuration Transfer Wizard, you can export Active Roles configuration objects (such as Access Templates, Managed Units, Policy Objects, Policy Type objects, and so on) to an XML file, then import them from that file to populate another instance of Active Roles. The export and import operations provide a way to move configuration objects from a test environment to a production environment.

Configuration Transfer Wizard components

Configuration Transfer Wizard includes the following components, all installed during the setup process of Configuration Transfer Wizard:

- Configuration Collection wizard
- Configuration Deployment wizard
- ARSconfig command-line tool

Configuration Collection wizard

Configuration Collection Wizard is intended to collect the Active Roles configuration data in a source environment. During the collection process, the selected Active Roles configuration objects are packed into an XML file, called "configuration package".

Configuration Deployment wizard

Configuration Deployment Wizard is designed to deploy a configuration package, earlier created with the Configuration Collection Wizard, in a destination Active Roles environment. When deploying the configuration data, the target Active Roles instance is populated with the configuration objects collected from the source Active Roles instance.

ARSconfig command-line tool

The ARSconfig command-line tool provides a script-based interface that enables automation of Active Roles configuration transfer. By using the command-line script, you can create or deploy an Active Roles configuration data package, or roll back changes made to a target Active Roles configuration during deployment of a configuration package.

For information on how to use the solution components, see Using the Configuration Transfer Wizard.

Installing Configuration Transfer Wizard

This section describes the installation requirements and the installation process of Configuration Transfer Wizard.

Configuration Transfer Wizard requirements

Configuration Transfer Wizard runs on top of Active Roles, and requires the Active Roles Administration Service deployed in your Active Directory environment before installing the Wizard. Configuration Transfer Wizard supports the following versions of Active Roles Administration Service:

- 7.0.2
- 7.1
- 7.2
- 7.3
- 7.4.x

Before you install Configuration Transfer Wizard, also make that you have any of following Active Roles components installed on the computer where you plan to install Configuration Transfer Wizard:

- Active Roles Administration Service
- Active Roles Console (MMC Interface)

Depending on whether you use the Wizard to collect Active Roles configuration data or to deploy a configuration package, Configuration Transfer Wizard must be installed on a computer from which you can connect to the Active Roles Administration Service in the source or destination environment. If the source and destination environments are physically separated, you must install the solution in each environment.

Assuming default security settings, the Domain Admins permissions are sufficient to install the Wizard.

Installing Configuration Transfer Wizard

You can install Configuration Transfer Wizard from the Active Roles `*.iso` file, if the installation requirements are met. For more information, see Configuration Transfer Wizard requirements.

To install Configuration Transfer Wizard

1. In the Active Roles `*.iso` file, navigate to the following folder:

   `\Solutions\Configuration Transfer Wizard`
2. To start installing the Wizard, double-click **ConfigurationTransferWizard_8.1.3.msi**.
3. Follow the instructions of the installer.

Using the Configuration Transfer Wizard

This section describes how to use Configuration Transfer Wizard to import and export Active Roles configuration data.

General considerations for using Configuration Transfer Wizard

To use Configuration Transfer Wizard, you must have the necessary security permissions. It is sufficient to be a member of the Active Roles Admin account, in both the source and destination environments. The Active Roles Admin account is specified during installation of the Administration Service and defaults to the Administrators group on the computer running the Administration Service.

> IMPORTANT: Before transferring the Active Roles configuration data, ensure that the Active Directory Organizational Unit (OU) structure in the destination environment is

identical to the OU structure in the source environment.

These are the general steps required to transfer Active Roles configuration data by using this solution:

1. **Collect configuration data from a source Active Roles environment** In this step, you select the Active Roles configuration objects you want the configuration package to include, and then create a configuration package XML file. This step is performed in the source environment.
2. **Deploy the collected configuration data to a destination Active Roles environment** In this step, the target Active Roles instance is populated with configuration objects from an earlier created package. This step is performed in the destination environment.

NOTE: If an object to deploy already exists in the target configuration, then the properties of the object are updated during the deployment process.

To perform these steps, you can use either the Configuration Collection Wizard and Configuration Deployment Wizard, or the ARSconfig command-line tool. Both methods have the same effect and can be used interchangeably, depending on your requirements.

You can use the Configuration Transfer Wizard to transfer the following Active Roles configuration objects:

- Access Templates and containers that hold Access Templates.
- Managed Units and containers that hold Managed Units.
- Policy Objects and containers that hold Policy Objects.
- Scheduled Task objects and containers that hold such objects.
- Application objects and containers that hold such objects.
- Script Modules and containers that hold Script Modules.
- Virtual attributes.
- Access Template links (`edsACE` object type).
- Policy Object links (`edsPolicyObjectLink` object type).
- Mail Configuration objects (`edsMailConfiguration` object type).
- Workflow definition objects (`edsWorkflowDefinition` object type).
- Automation Workflow definition objects (`edsAutomationWorkflowDefinition` object type).
- Policy Type objects (`edsPolicyType` object type).
- Entitlement Profile Specifier objects and containers (`edsOneViewSpecifier` or `edsOneViewSpecifiersContainer` object type).
- Display specifiers and containers that hold display specifiers (`displaySpecifier` or `edsDisplaySpecifierContainer` object type).

However, the Configuration Transfer Wizard cannot transfer the following configuration object categories:

- Built-in objects (that is objects that have "built-in" in their name).
- Web Interface configuration data (that is objects held in the **Configuration/Application Configuration/Web Interface** container)

If you need to roll back the changes made to the configuration of the target Active Roles instance, during the package deployment, you can do so by using the command-line tool included with Configuration Transfer Wizard. For more information, see Example: Rolling back the configuration changes.

Dangling links during configuration transfer

When collecting Access Templates and Policy Objects, Configuration Transfer Wizard analyzes their links and writes the links to the destination package. Every link record includes information about the directory object and, if applicable, the trustee to which the respective Access Template or Policy Object is applied. In the configuration package file, this information normally takes the form of the distinguished name (DN), while in the Active Roles environment the links refer to the objects by security identifier (SID) or globally unique identifier (GUID). The Wizard needs DN rather than SID or GUID to identify an object as in a different environment, the object SID or GUID differs from that in the original environment. By identifying the link reference objects by DN, the solution enables the delegation and policy settings to be properly transferred from the source environment to the destination environment.

To have the link records identify the link reference objects by DN, the Wizard has to look up object SID or GUID to object DN. If this process fails for a given link, the link record is created that identifies the link reference object by SID or GUID. Such a record is referred to as "dangling link".

If any dangling links have been recorded to the destination package, Configuration Transfer Wizard indicates this condition. Deploying a package that contains dangling links may create links in the destination environment that refer to non-existent objects. As a result, some delegation and policy settings configured by deploying the package may not match the settings found in the source environment from which the package was collected.

The ARSconfig tool provides the **danglingLinks** parameter that allows you to specify how you want the deployment process to handle dangling links. For more information, see Using the ARSconfig command-line tool.

Using the Configuration Collection Wizard and the Configuration Deployment Wizard

To transfer an Active Roles configuration, you can collect configuration objects from one Active Roles environment, then deploy them to another environment with the following steps:

1. Create a configuration package file with the Configuration Collection Wizard.
2. Deploy the package with the Configuration Deployment Wizard.

To create a configuration package with the Configuration Collection Wizard

1. Start the wizard by running the **Configuration Collection Wizard** application from the Start menu or the Apps page.
2. On the **Collect Active Roles Configuration Data** page, do the following:
   a. Click **Connect** and using the **Connect to Administration Service** dialog that opens, select the Administration Service to which you want the wizard to connect.
   b. Under **Select configuration objects to package**, select the objects you want to include in the configuration package, and specify whether you want to collect the child objects of the selected objects.
   c. When finished, click **Create Package**.
3. On the **Specify a location for the configuration package** page, do the following:
   a. Click **Browse** to specify a location and name for the configuration package file.
   b. (Optional) Enter a **Package description**.
   c. To collect Access Templates associated with the selected objects, leave the **Do not collect associated Access Templates** check box clear. Otherwise, select this check box.
   d. To cause the wizard to collect Policy Objects associated with the selected objects, leave the **Do not collect associated Policy Objects** check box clear. Otherwise, select this check box.
4. On the **Verify the information you specified** page, click **Start**.

To deploy a configuration package with the Configuration Deployment wizard

1. Start the wizard by running the **Configuration Deployment Wizard** application from the Start menu or the Apps page.
2. On the **Deploy Active Roles Configuration Data** page, do the following:
   a. Click **Browse** to select the configuration package file.
   b. (Optional) Select the **Ignore errors** check box for the wizard to ignore any errors during the configuration deployment.
   c. Click **Deploy Package**.
3. On the **Connect to Administration Service** page, select the Administration Service to which you want the wizard to connect, and then click **Next**.
4. On the **Add Domain Name Mapping** page, if names of the managed domains differ in the test and production environments, add domain name mapping entries, and then click **Next**.
5. On the **Verify the information you specified** page, click **Start.**

Using the ARSconfig command-line tool

As an alternative to using the graphical user interface tools, you can use the ARSconfig command-line tool. The ARSconfig tool is the arsconfig.wsf Windows Script File (WSF) that defines the command line parameters and the required object references.

Using the ARSconfig tool requires two files to be pre-configured, before running the script. These are a file that lists the configuration objects that the package must include, and, if necessary, a file containing domain mapping entries.

To run the ARSconfig command-line tool

1. Open the Windows Command Prompt.
2. From the command prompt, run the `arsconfig.wsf` script, specifying the required type of task and parameters. The script syntax is described in ARSconfig syntax.

ARSconfig syntax

The ARSconfig Windows Script File has the following syntax.

```
Cscript arsconfig.wsf [/?] /task:<'collect' | 'deploy' | 'rollback'>
[/selection:"<filename.xml>"] [/package:"<filename.xml>"] [/map:"<filename.csv>"]
[/verbose] [/log:"<filename>"] [/deletelog] [/server:<servername>]
[/login:<username>] [/password:<userpassword>] [/danglingLinks:<'Stop' | 'Skip' |
'Deploy'>] [/ignoreLinks:<'0' | '1' | '2' | '3'>] [/ignoreErrors] [/upgrade]
```

ARSconfig parameters

The ARSconfig Windows Script File (WSF) has the following parameters.

Table 86: Parameters

Parameter	Description
task	This is a required parameter which defines the type of task you want to perform by using this script. Specify one of these parameter values: • 'collect' - Collects configuration data from the source Active Roles environment, and creates a configuration package file. • 'deploy' - Populates the target Active Roles instance with objects from a configuration package created earlier by Configuration Transfer Wizard. • 'rollback' - Reverts the configuration of the target Active Roles instance to the state it was in before deployment of the

Parameter	Description
	configuration package.
selection	The path and name of the XML file containing a list of the source configuration objects to be included in the configuration package. This parameter is required when you use this script to create a configuration package. The XML file you specify in this parameter must be manually created before you run the script.
package	The full path to the configuration package XML file. Add this parameter is you want to specify a custom name and location for the configuration package file. If you do not specify this parameter, the script assumes that the installation path, and the default package file name are used.
map	The name of the domain mapping file. Add this parameter if you want the test domain names to be replaced with the production domain names, during configuration package deployment. You can add this parameter only when you use this script to deploy a configuration package. The CSV file you specify in this parameter must be manually created before you run the script.
verbose	Enables log trace output. If this parameter is not specified, then no information is displayed in the Command Prompt while the script is running.
log	Specifies the name of the trace output file. You can also specify a target location for the log file. Add this parameter to create a log file with diagnostic information.
deletelog	Deletes the trace output file upon successful completion. Add this parameter if you want the log file deleted if a task was completed with no errors.
server	The fully qualified domain name of the computer running the Administration Service to connect to. If this parameter is not specified, the script attempts a connection to any available Administration Service.
login	The user logon name of the account with which you want to connect, in the form Domain\UserName, or in the form of a user principal name.
password	Password for the user logon name you specify in the *login* parameter.
danglingLinks	This parameter takes effect if the task parameter value is set to 'deploy', and specifies whether to deploy Access Template or Policy

Parameter	Description
	Object links, if any found in the package, that refer to objects which may fail to be resolved in the destination environment (dangling links). The acceptable parameter values are: • 'Stop' - The deployment process is not started if any dangling links are detected (default setting) • 'Skip' - The dangling links are not deployed in the destination environment • 'Deploy' - Deployment of the dangling links is attempted based on the data found in the package
ignoreLinks	Specifies whether to collect Access Template links and Policy Object links. This parameter can take any of the following values: • '0' - Collect all links (default setting). • '1' - Do not collect Policy Object links. • '2' - Do not collect Access Template links. • '3' - Do not collect Policy Object and Access Template links.
ignoreErrors	If this parameter is specified, the solution ignores any errors that can be encountered during the configuration deployment.
upgrade	If supplied together with /task:'deploy', preserves the existing links, policy parameters and scheduled task parameters. Without this parameter, the deployment of a configuration package replaces the existing links with the links found in the configuration package, and resets the policy and schedule task parameters to the default values.

Example: Transferring an Active Roles configuration

This example scenario explains how to use the ARSconfig command-line tool to transfer a set of configuration objects from a test Active Roles instance to a production instance.

Suppose you need to transfer the following configuration objects from a test Active Roles instance to a production Active Roles instance:

- The **Configuration/Access Templates/Common** container, including all child objects stored in this container.
- The **Configuration/Managed Units/Development** container, excluding the child objects stored in this container.
- All child objects stored in the **Script Modules/Corporate Policy/Priority Access** container, but excluding the container itself.

Also, assume that the names of the domains managed by the test (source) Active Roles instance are **test1.company.com** and **test2.company.com**, and the two corresponding domains managed by the production (target) Active Roles instance are **prod1.company.com** and **prod2.company.com**.

To implement this scenario, complete the following steps:

1. Create a list of the configuration objects to collect
2. Create configuration data package
3. Add domain mapping
4. Deploy the configuration data package

Creating a list of the configuration objects to package

In this step, you create a list of the configuration objects that you want to collect into the configuration package, and define how you want to collect their child objects.

To do that, create the **selection.xml** file, and save that file to the solution installation folder: *<Active Roles installation folder>*\Configuration Transfer Wizard\Scripts.

To clarify the file format, consider the following sample file that illustrates how to collect Access Templates, Managed Units, and Script Modules residing within specified containers:

```
<?xml version="1.0" encoding="utf-8"?>

<Configuration>

<include DN="CN=Common,CN=Access Templates,CN=Configuration" collectSelf="True"
collectChildren="True"/>>

<include DN="CN=Development,CN=Managed Units,CN=Configuration" collectSelf="True"
collectChildren="False"/>

<include DN="CN=Priority Access,CN=Corporate Policy,CN=Script
Modules,CN=Configuration" collectSelf="False" collectChildren="True"/>

</Configuration>
```

Creating configuration data package file

In this step, you use the ARSconfig command-line tool to create a configuration data package file using the data from the `selection.xml` file created in Step 1.

To create the configuration data package file

1. Open the Windows Command Prompt.
2. In the command prompt, navigate to the Configuration Transfer Wizard installation folder, and enter the following syntax:

   ```
   Cscript.exe arsconfig.wsf /task:collect /selection:selection.xml
   ```

As the result, the `package.xml` configuration data package file will be created in the following default location:

```
\Active Roles\Configuration Transfer Wizard\Scripts
```

Configuring domain mapping

If the names of the managed domains are different in the test and production environments, you must add domain mapping that defines the correspondence between the domain names. When the configuration package is deployed in the target environment, the domain names specified as a part of the objects' attributes are replaced with the names of the production domains, according to the name mapping entries.

In this step, you create the CSV domain name mapping file (**`mapping.csv`**), then save that file to the installation folder of the Configuration Transfer Wizard:

```
\Active Roles\Configuration Transfer Wizard\Scripts
```

In this scenario, the **`mapping.csv`** file contains the following lines:

```
"DC=test1,DC=company,DC=com","DC=prod1,DC=company,DC=com"
"DC=test2,DC=company,DC=com","DC=prod2,DC=company,DC=com"
```

Deploying the configuration data package

In this step, you use the ARSconfig command-line tool to deploy the **package.xml** configuration package in the production Active Roles environment. When running the arsconfig.wsf script, specify the package file to deploy (**`package.xml`**), and the domain name mapping file (**`mapping.csv`**) you have created in the previous step.

To deploy the configuration data package

1. Open the Windows Command Prompt.
2. Navigate to the Configuration Transfer Wizard installation folder, and enter the following syntax:

   ```
   Cscript.exe arsconfig.wsf /task:deploy /package:package.xml /map:mapping.csv
   ```

Example: Rolling back the configuration changes

You may need to roll back the configuration changes if you encounter any errors when deploying a configuration package to the production environment. By rolling back changes in the target configuration, you bring it to the state it was in before the package was deployed.

To roll back configuration changes

1. Open the Windows Command Prompt.
2. Navigate to the Configuration Transfer Wizard installation folder, and enter the following syntax:

   ```
   Cscript.exe arsconfig.wsf /task:rollback /package:package.xml
   ```

21

Managing Skype for Business Server with Active Roles

The Skype for Business Server User Management feature allows you to administer Skype for Business Server user accounts via the Active Roles Web Interface by providing built-in policies to synchronize user account information between Active Roles and Skype for Business Server.

About Skype for Business Server User Management

With Skype for Business Server User Management, you can use Active Roles to perform the following tasks:

- Add and enable new Skype for Business Server users.
- View or change Skype for Business Server user properties and policy assignments.
- Move Skype for Business Server users from one Skype for Business Server pool to another.
- Disable or re-enable user accounts forSkype for Business Server.
- Remove users from Skype for Business Server.

Skype for Business Server User Management adds the following elements to Active Roles:

- Built-in Policy Object that enables Active Roles to perform user management tasks on Skype for Business Server.
- Built-in Policy Object that enables Active Roles to administer Skype for Business Server users in environments that involve multiple Active Directory forests.
- Commands and pages for managing Skype for Business Server users in the Active Roles Web Interface.
- Access Templates to delegate Skype for Business Server user management tasks.

The Skype for Business Server User Management policy allows you to control the following factors of creating and managing Skype for Business Server users:

- Rule for generating the SIP user name. When adding and enabling a new Skype for Business Server user, Active Roles can generate a SIP user name based on other properties of the user account.
- Rule for selecting a SIP domain. When configuring the SIP address for a Skype for Business Server user, Active Roles can restrict the list of selectable SIP domains and suggest which SIP domain to select by default.
- Rule for selecting a Telephony option. When configuring Telephony for a Skype for Business Server user, Active Roles can restrict the list of selectable Telephony options and suggest which option to select by default.
- Rule for selecting a Skype for Business Server pool. When adding and enabling a new Skype for Business Server user, Active Roles can restrict the list of selectable registrar pools and suggest which pool to select by default. This rule also applies to selection of the destination pool when moving a Skype for Business Server user from one pool to another.

The Skype for Business Server User Management feature provides a number of Access Templates allowing you to delegate the following tasks in Active Roles:

- Add and enable new Skype for BusinessSkype for Business Server users.
- View existing Skype for Business Server users.
- View or change the SIP address for Skype for Business Server users.
- View or change the Telephony option and related settings for Skype for Business users.
- View or change Skype for Business Server user policy assignments.
- Disable or re-enable user accounts for Skype for Business Server.
- Move users from one Skype for Business Server pool to another.
- Remove users from Skype for Business Server.

Active Directory topologies supported by Skype for Business Server User Management

Skype for Business Server User Management supports the following Active Directory Domain Services (AD DS) topologies:

- Single forest with a single tree or multiple trees. For more information, see Single forest topology for Skype for Business Server User Management.
- Multiple forests in a resource forest topology. For more information, see Resource forest topology for Skype for Business Server User Management.
- Multiple forests in a central forest topology. For more information, see Central forest topology for Skype for Business Server User Management.

Single forest topology for Skype for Business Server User Management

A single forest Active Directory topology assumes that the login-enabled user accounts managed by Active Roles are defined in the Active Directory forest in which Skype for Business Server is deployed.

To perform Skype for Business Server user management tasks on a given user account, Active Roles makes changes to the attributes of that use account. Then, based on the attribute changes, the Skype for Business Server User Management policy requests the Skype for Business Server remote shell to update the user account accordingly.

For example, when creating a new Skype for Business Server user, Active Roles sets a virtual attribute on that user account directing the policy to invoke the remote shell command for enabling the new user for Skype for Business Server. When making changes to an existing Skype for Business Server user, Active Roles populates the attributes of the user account with the desired changes, causing the policy to apply those changes via the remote shell.

Resource forest topology for Skype for Business Server User Management

The resource forest topology refers to a multi-forest environment where a separate forest (in this case, the Skype for Business Server forest) hosts servers running Skype for Business Server, but does not host any login-enabled user accounts.

Outside the Skype for Business Server forest, user forests host login-enabled user accounts but no servers running Skype for Business Server. When creating a Skype for Business Server account for a user from an external forest, Active Roles:

1. Creates an inactive user account in the Skype for Business Server forest.
2. Establishes a link between the user account in the user forest (master account) and the inactive user account in the Skype for Business Server forest (shadow account).
3. Enables the shadow account for Skype for Business Server.

The Master Account Management policy then ensures that the attributes of the shadow account are synchronized with the attributes of the master account, so that Skype for Business Server user properties can be administered on the master account via Active Roles. In the Skype for Business Server forest, the User Management policy detects the attribute changes replicated from the master account to the shadow account, and translates them to remote shell commands on Skype for Business Server, similarly to the case of single forests, as described in Single forest topology for Skype for Business Server User Management.

Central forest topology for Skype for Business Server User Management

The central forest topology refers to a multi-forest environment where a separate forest (in this case, a Skype for Business Server forest) hosts servers running Skype for Business Server and may also host login-enabled accounts. Outside the Skype for Business Server forest, user forests host login-enabled user accounts but no servers running Skype for Business Server.

With the Skype for Business Server User Management policy applied to login-enabled user accounts in the Skype for Business Server forest, Active Roles can enable and administer those user accounts for Skype for Business Server in the same way as in case of single forests, as described in as described in Single forest topology for Skype for Business Server User Management.

When creating a Skype for Business Server account for a user from an external forest, Active Roles:

1. Creates a contact in the Skype for Business Server forest.
2. Establishes a link between the user account in the user forest (master account) and the contact in the Skype for Business Server forest (shadow account).
3. Enables that contact for Skype for Business Server.

The Master Account Management policy then ensures that the attributes of the contact are synchronized with the attributes of the user account, so that Skype for Business Server user properties can be administered on the user account via Active Roles. In the Skype for Business Server forest, the User Management policy detects the attribute changes replicated from the user account to the contact, and translates them to remote shell commands on Skype for Business Server, similarly to the case of single forests, as described in Single forest topology for Skype for Business Server User Management.

User Management policy for Skype for Business Server User Management

The **Skype for Business - User Management** built-in policy enables Active Roles to perform user management tasks on Skype for Business Server. The policy is intended for:

- Single-forest and multi-forest environments where login-enabled accounts of Skype for Business users are defined in the Active Directory forest in which Skype for Business Server is deployed.
- Multi-forest environments where login-enabled master accounts of Skype for Business Server users are defined in external forests with each master account being represented by a shadow account (inactive user account or contact) in the Active Directory forest in which Skype for Business Server is deployed.

The Policy Object that holds this policy is located in the following container in the Active Roles Console:

Configuration/Policies/Administration/Builtin/Built-in Policy - Skype for Business - User Management

Depending on your Active Directory topology, apply this Policy Object as follows to enable Skype for Business Server User Management in Active Roles.

Table 87: Applying the Built-in - Skype for Business - User Management Policy Object

Topology option	Where to apply the Policy Object
Single forest topology for Skype for Business Server User Management	Apply this Policy Object to Active Directory domains or containers that hold user accounts you want to administer by using Skype for Business Server User Management in Active Roles.
Resource forest topology for Skype for Business Server User Management	Apply this Policy Object to Active Directory domains or containers in the Skype for Business Server forest that hold shadow accounts (inactive user accounts) for users from external forests you want to administer by using Skype for Business Server User Management in Active Roles.
Central forest topology for Skype for Business Server User Management	Apply this Policy Object to: • Active Directory domains or containers in the Skype for Business Server forest that hold login-enabled user accounts you want to administer by using Skype for Business Server User Management in Active Roles • Active Directory domains or containers in the Skype for Business Server forest that hold shadow accounts (contacts) for users from external forests you want to administer by using Skype for Business Server User Management in Active Roles.

Skype for Business Server User Management policy settings

This section describes the policy settings available for the Skype for Business User Management built-in policy.

Connecting to Skype for Business Server

To administer Skype for Business Server users, Active Roles requires a connection to a computer running one of the following server roles in your Skype for Business Server deployment:

- Front End Server (if using Skype for Business Server Enterprise Edition).
- Standard Edition Server.

The computer must be from an Active Directory domain that is registered with Active Roles as a managed domain. By using the **Server** policy setting, you can specify how you want Active Roles to select a Skype for Business Server computer:

- **Connect to any available server**: With this option, Active Roles attempts to connect to any Front End Server or Standard Edition Server that runs the Central Management Server in your Skype for Business Server deployment. If no Central Management Server role holders are available in the managed domains, then Active Roles attempts to connect to the first Front End Server or Standard Edition Server found in the managed domains.
- **Connect to these servers only**: This option allows you to configure a list from which you want Active Roles to select a Skype for Business Server computer. You can:
    - Add or remove computers from the list. Active Roles searches the managed domains for computers running the appropriate Skype for Business Server role, allowing you to select the desired computers.
    - Set the default computer. Active Roles first attempts to connect to that computer.
    - Reorder the list. Active Roles first attempts to connect to computers that are higher in the list.

> NOTE: At least one of your Active Directory domains that hold computers running the Front End Server or Standard Edition Server must be registered with Active Roles as a managed domain. Otherwise, Active Roles cannot discover your Skype for Business Server deployment, and the Skype for Business Server User Management feature will not work.

SIP user name generation rule

The **SIP User Name** policy setting allows you to configure a rule for generating the SIP user name based on other properties of the user account. When adding a new Skype for Business Server user, Active Roles uses that rule to generate the SIP user name on the Web Interface page for enabling users for Skype for Business Server. The rule has an effect if you select the SIP address option that provides for entering a SIP user name. On the page where you edit Skype for Business Server users, the rule performs a validation function, preventing changes to the SIP user name that violate the rule.

To configure a rule, set up a value that acts as a template for the SIP user name. You can add one or more entries to the value, with each entry representing one of the following:

- **Text**: A text string. You can type the desired text when adding the entry.
- **User Property**: A particular property of the user account. You can choose the desired property and specify whether you want the entry to include the entire property or a part of the property.
- **Parent OU Property**: A particular property of the Organizational Unit that holds the user account. You can choose the desired property and specify whether you want the entry to include the entire property or a part of the property.
- **Parent Domain Property**: A particular property of the Active Directory domain that holds the user account. You can choose the desired property and specify whether you want the entry to include the entire property or a part of the property.

The rule sets the SIP user name to the string value obtained by calculating each entry and then concatenating the calculation results so that they form a single string value.

By default, the policy allows the generated name to be modified. The **SIP User Name** policy setting provides the option to prevent changing the generated name. If you select that option, the SIP user name is read-only on the Web Interface page for enabling users for Skype for Business Server.

SIP domain restriction rule

The **SIP Domain** policy setting allows you to configure a rule that restricts selection of a SIP domain for the user SIP address. When you add a new Skype for Business Server user or edit an existing Skype for Business Server user, this rule determines the list from which you can select a SIP domain for the user's SIP address. In case of adding a new Skype for Business Server user, the rule applies to any SIP address option that involves selecting a SIP domain from the list.

To configure a rule, you choose one of these policy options:

- **Allow selection of any SIP domain**: With this option, the policy does not restrict the list of SIP domains.
- **Restrict selection to these SIP domains**: This option allows you to configure a list of acceptable SIP domains. You can:
  - Add or remove SIP domains from the list. Active Roles identifies all SIP domains that exist in your Skype for Business Server deployment, allowing you to select the desired SIP domains.
  - Set the default SIP domain. When creating a SIP address, Active Roles selects the specified SIP domain by default.
  - Reorder the list. When prompting to select a SIP domain for a user's SIP address, Active Roles lists the SIP domain names in the order specified.

Pool restriction rule

The **Pool** policy setting allows you to configure a rule that restricts selection to an Enterprise Edition Front End pool or Standard Edition server to which Skype for Business Server users can be assigned. When you add a new Skype for Business user, this rule determines the list from which you can select a pool for the new user. When you move a Skype for Business Server user from one pool to another, this rule determines the list from which you can select the destination pool.

To configure a rule, choose one of the following policy options:

- **Allow selection of any pool**: With this option, the policy does not restrict the list of pools.
- **Restrict selection to these pool**: This option allows you to configure a list of acceptable pools. You can:
  - Add or remove pools from the list. Active Roles identifies all Front End pools and Standard Edition servers in your Skype for Business Server deployment, allowing you to select the desired pools or servers.
  - Set the default pool. When adding a new Skype for Business Server user or moving a user to another pool, Active Roles selects the specified pool by default.
  - Reorder the list. When prompting to select a pool, Active Roles lists the pools in the order specified.

Telephony restriction rule

The **Telephony** policy setting allows you to configure a rule that restricts selection of a Telephony option for Skype for Business Server users. When you add or edit a Skype for Business Server user, this rule determines the list from which you can select a Telephony option.

To configure a rule, you choose one of these policy options:

- **Allow selection of any option**: With this option, the policy does not restrict the list of Telephony options.
- **Restrict selection to these options**: This option allows you to configure a list of acceptable Telephony options. You can:
  - Add or remove Telephony options from the list.
  - Set the default Telephony option. When adding a new Skype for Business Server, Active Roles selects the specified Telephony option by default.
  - Reorder the list. When prompting to select a Telephony option, Active Roles lists the options in the order specified.

Master Account Management policy for Skype for Business Server User Management

The Master Account Management policy is intended for multi-forest environments where the login-enabled master accounts of Skype for Business Server users are defined in Active Directory forests where Skype for Business Server is not deployed.

In this setup, each master account is represented by a shadow account (inactive user account or contact) in the Active Directory forest in which Skype for Business Server is deployed. For more information on this forest topology, see Resource forest topology for Skype for Business Server User Management and Central forest topology for Skype for Business Server User Management.

The Master Account Management policy enables Active Roles to control master accounts of Skype for Business Server users, and operates together with the Skype for Business Server User Management policy that controls shadow accounts in the Skype for Business Server forest. For more information on the User Management policy, see User Management policy for Skype for Business Server User Management.

The Policy Object that holds the Master Account Management policy is located in the following container in the Active Roles Console:

Configuration/Policies/Administration/Builtin/Built-in Policy - Skype for Business - Master Account Management

Depending on your Active Directory topology, apply this Policy Object as follows to enable Skype for Business Server User Management in Active Roles.

Table 88: Applying the Built-in - Skype for Business - Master Account Management Policy Object

Topology option	How to apply the Policy Object
Single forest topology for Skype for Business Server User Management	Do not apply this Policy Object.
Resource forest topology for Skype for Business Server User Management	Configure the **Forest Mode** policy setting by selecting the **Resource forest** option, then apply this Policy Object to the Active Directory domains or containers that hold login-enabled user accounts in external forests (master accounts) you want to administer by using Skype for Business Server User Management in Active Roles.

Topology option	How to apply the Policy Object
Central forest topology for Skype for Business Server User Management	Configure the **Forest Mode** policy setting by selecting the **Central forest** option, then apply this Policy Object to Active Directory domains or containers that hold login-enabled user accounts in external forests (master accounts) you want to administer by using Skype for Business Server User Management in Active Roles.

Master Account Management policy settings for Skype for Business Server User Management

This section describes the policy settings of the Master Account Management policy for Skype for Business Server User Management.

Skype for Business Server forest mode

The Master Account Management policy is intended for multi-forest environments where the Skype for Business Server forest is used either as a resource forest or as a central forest.

- In the central forest mode, the Skype for Business Server forest may hold login-enabled Skype for Business Server user accounts in addition to shadow accounts (contacts) for Skype for Business Server users from external forests.
- In the resource forest mode, the Skype for Business Server forest holds only shadow accounts (logon-disabled user accounts) for Skype for Business Server users from external forests.

The **Forest Mode** policy setting allows you to choose the option that matches the Skype for Business Server forest mode in your Skype for Business Server deployment:

- **Resource forest**: The policy creates and administers login-disabled user accounts as shadow accounts for Skype for Business Server users from external forests. The user account from an external forest, referred to as a master account, is linked and synchronized with the shadow account that is enabled for Skype for Business Server in the Skype for Business Server forest.
- **Central forest**: The policy creates and administers contact objects as shadow accounts for Skype for Business Server users from external forests. The user account from an external forest, referred to as a master account, is linked and synchronizes with the contact that is enabled for Skype for Business Server in the Skype for Business Server forest.

Container for new Skype for Business Server shadow accounts

The Master Account Management policy allows you to specify the container in which you want Active Roles to create shadow accounts when enabling master accounts for Skype for Business Server. You can select the desired Organizational Unit (OU) in the Skype for Business Server forest or you can let Active Roles choose the default container.

If you select a specific OU, Active Roles creates shadow accounts in that OU. You can select an OU from any domain of the Skype for Business Server forest that is registered with Active Roles as a managed domain.

If you let Active Roles choose the default container for new shadow accounts, then Active Roles creates shadow accounts in the **Users** container in a particular domain of the Skype for Business Server forest. If the forest root domain of the Skype for Business Server forest is registered with Active Roles as a managed domain, then Active Roles creates shadow accounts in that domain. Otherwise, Active Roles creates shadow accounts in the domain that appears first in the ordered list of the managed domains from the Skype for Business Server forest.

> NOTE: Active Roles requires at least one domain of the Skype for Business Server forest to be registered with Active Roles as a managed domain.

Default description for new Skype for Business Server shadow accounts

The Master Account Management policy allows you to specify a text to use as the default description for new shadow accounts that Active Roles creates when enabling master accounts for Skype for Business Server. Active Roles writes that text to the **Description** property of every new shadow account.

Shadow account reference attribute

By default, the policy designates the **adminDescription** attribute of the master account to store the GUID of the shadow account. However, you can select a different attribute if needed.

The Skype for Business Server User Management feature uses this attribute to identify the shadow account (and, consequently, the linked mailbox) when managing a specific master account. The policy causes Active Roles to set this attribute on the master account when creating the linked mailbox.

Synchronized Skype for Business Server properties

The Master Account Management policy defines a list of properties to copy from the Skype for Business Server master account to the shadow account. These properties are referred to as "synchronized properties". When you use Active Roles to set or change a synchronized property of a master account, the policy causes Active Roles to set or change the value of that property on both the master account and the shadow account.

In addition, Skype for Business Server User Management provides a scheduled task that copies synchronized properties from every managed master account to the corresponding shadow account. The task runs on a scheduled basis to ensure that each of the synchronized properties of the shadow account has the same value as the corresponding property of the master account. If a synchronized property of the shadow account has changed for whatever reason, Active Roles changes that property back to the value found on the master account. For more details, see Scheduled Skype for Business Server synchronization.

The default list of synchronized properties are as follows:

- c (Country Abbreviation)
- co (Country)
- company (Company)
- countryCode (Country-Code)
- department (Department)
- displayName (Display Name)
- givenName (First Name)
- homePhone (Home Phone)
- initials (Initials)
- l (City)
- mobile (Mobile Number)
- otherTelephone (Phone Number (Others))
- physicalDeliveryOfficeName (Office Location)
- postalCode (ZIP/Postal Code)
- postOfficeBox (Post Office Box)
- sAMAccountName (Logon Name (pre-Windows 2000))
- sn (Last Name)
- st (State/Province)
- streetAddress (Street Address)
- telephoneNumber (Telephone Number)
- title (Job Title)

- url (Web Page Address (Others))
- wWWHomePage (Web Page Address)

> TIP: You can configure the policy to either synchronize additional properties or remove individual properties from synchronization.

Skype for Business Server

The Master Account Management policy defines a list of properties that appear on the Skype for Business Server master account but reflect the properties of the shadow account. These properties are referred to as substituted properties. When you use Active Roles to view properties of a master account, the policy causes Active Roles to retrieve the values of the substituted properties of the master account from the shadow account. When you use Active Roles to set or change a substituted property of a master account, the policy causes Active Roles to set or change the value of that property on the shadow account.

The Skype for Business Server substituted properties include the following:

- edsva-Skype for Business-AccountExists
- edsva-Skype for Business-ArchivingPolicy
- edsva-Skype for Business-ClientPolicy
- edsva-Skype for Business-ClientVersionPolicy
- edsva-Skype for Business-ConferencingPolicy
- edsva-Skype for Business-DialPlanPolicy
- edsva-Skype for Business-Disable
- edsva-Skype for Business-Enable
- edsva-Skype for Business-ExchangeArchivingPolicy
- edsva-Skype for Business-ExternalAccessPolicy
- edsva-Skype for Business-HostedVoiceMail
- edsva-Skype for Business-IsEnabled
- edsva-Skype for Business-LineServerURI
- edsva-Skype for Business-LineURI
- edsva-Skype for Business-LocationPolicy
- edsva-Skype for Business-MasterAccount
- edsva-Skype for Business-MobilityPolicy
- edsva-Skype for Business-Move
- edsva-Skype for Business-MoveTargetRegistrarPool
- edsva-Skype for Business-PersistentChatPolicy
- edsva-Skype for Business-PIN
- edsva-Skype for Business-PINPolicy

- edsva-Skype for Business-PrivateLine
- edsva-Skype for Business-ReEnable
- edsva-Skype for Business-RegistrarPool
- edsva-Skype for Business-SIPAddress
- edsva-Skype for Business-SIPAddressType
- edsva-Skype for Business-SIPDomain
- edsva-Skype for Business-SIPUserName
- edsva-Skype for Business-TasksAllowed
- edsva-Skype for Business-TelephonyOption
- edsva-Skype for Business-TemporarilyDisable
- edsva-Skype for Business-VoicePolicy

> NOTE: You cannot remove properties from the default list of substituted properties. However, you can create your custom list of substituted Skype for Business Server properties in addition to the default list. If you do so, the resulting list of substituted properties includes all properties from both the default list and your custom list.

Back-synchronized Skype for Business Server properties

The Master Account Management policy also defines a list of Skype for Business Server properties to copy from the shadow account to the master account. By default, the list is empty. If you add a property to that list, the policy ensures that any changes to that property on the shadow account will be replicated to the master account.

Master Account Management policy actions for Skype for Business User Management

The Master Account Management policy causes Active Roles to perform the following Skype for Business Server management actions depending on the change request submitted to the Active Roles Administration Service.

Table 89: Policy actions for Skype for Business Server User Management

Request	Actions
Enable an existing Active Directory user for Skype for Business Server	Active Roles retrieves the properties of the existing user (in the external forest), then performs the following actions: 1. Creates a shadow account in the Skype for Business forest, and populate its properties with the properties of

Request	Actions
	the user from the external forest. 2. Enables the shadow account for Skype for Business Server. 3. Sets the `msRTCSIP-OriginatorSID` attribute of the shadow account to the value of the `objectSID` attribute of the user from the external forest. 4. Creates a reference to the shadow account on the master account. If the user from the external forest already has a shadow account (for example, created by Exchange Resource Forest Management), then the policy reuses the existing shadow account instead of creating a new one. When creating the shadow account, Active Roles runs all policies that are applied to the container that holds the shadow account.
Modify Skype for Business Server user properties of a master account	If the change request includes any changes to substituted properties, Active Roles first makes the requested changes to the substituted properties of the shadow account. Next, Active Roles makes the requested changes to the properties of the master account, then updates the synchronized properties of the shadow account with the new property values found on the master account.
Deprovision a master account	Active Roles deprovisions the master account, then temporarily disables the shadow account for Skype for Business Server.
Undeprovision a deprovisioned master account	Active Roles undeprovisions the master account, and re-enables the shadow account for Skype for Business Server. For undeprovisioning master accounts to have an effect on shadow accounts, the container that holds deprovisioned master accounts must be in the scope of the **Built-in Policy - Skype for Business - Master Account Management** Policy Object (or a copy of that Policy Object).
Delete a master account	Active Roles deletes the master account, and then removes the shadow account from Skype for Business Server.

The Master Account Management policy requires that shadow accounts be in the scope of the User Management policy for Skype for Business Server User Management provided by Skype for Business Server User Management. This enables Active Roles to perform the Skype for Business Server-related actions on the shadow account.

Scheduled Skype for Business Server synchronization

Skype for Business Server User Management includes an Active Roles scheduled task that complements the Master Account Management policy to enforce synchronization of master and shadow account properties, and to capture existing Skype for Business Server users whose master account happens to fall under the control of that policy.

The scheduled task object is located in the following container in the Active Roles Console:

Configuration/Server Configuration/Scheduled Tasks/Builtin/Skype for Business - Master Account Management

The task is scheduled to run on a daily basis, and normally you do not need to modify its settings. The operation of the task affects only the user accounts that are in the scope of the **Built-in Policy - Skype for Business - Master Account Management** Policy Object (or a copy of that Policy Object). When run, the task performs the following actions on each of those user accounts:

- If the user account does not have a shadow account that is enabled for Skype for Business Server, it skips that user account.
- If the user account has a shadow account that is enabled for Skype for Business Server but does not store a reference to that shadow account, it creates the reference to the shadow account on that user account.

  This action enables the Skype for Business Server User Management feature to administer existing Skype for Business Server users.
- If the user account has a shadow account that is enabled for Skype for Business Server and stores a reference to the shadow account, then it copies the synchronized properties from the master account to the shadow account, and copies the back-synchronized properties from the shadow account to the master account.

  This action ensures that the shadow account properties are updated with the latest changes to the master account properties (and the opposite is also true).

Access Templates for Skype for Business Server

Skype for Business Server User Management provides a number of Access Templates (ATs) allowing you to delegate the tasks of managing Skype for Business Server users in Active Roles. You can find these ATs in the following container when using Active Roles Console:

Configuration/Access Templates/Skype for Business Server

Table 90: Skype for Business Server User Management Access Templates

Access Template	Description
Skype for Business Server - User Full Control	Gives permission to perform the following tasks by using Active Roles: • Add and enable new Skype for Business Server users. • View existing Skype for Business Server users. • View or change the SIP address. • View or change the telephony option and related settings. • View or change the user policy assignments in Skype for Business Server. • Temporarily disable or re-enable users for Skype for Business Server. • Move users to another server or pool in Skype for Business Server. • Remove users from Skype for Business Server.
Skype for Business Server - User Telephony	Gives permission to perform the following tasks by using Active Roles: • View existing Skype for Business Server users. • View the SIP address. • View or change the telephony option and related settings. • View the user policy assignments in Skype for Business Server.
Skype for Business Server - User Disable/Re-enable	Gives permission to perform the following tasks by using Active Roles: • View existing Skype for Business Server users. • View the SIP address. • View the telephony option and related settings. • View the user policy assignments in Skype for Business Server. • Temporarily disable or re-enable users for Skype for Business Server.
Skype for Business Server - User Policies	Gives permission to perform the following tasks by using Active Roles: • View existing Skype for Business Server users.

Access Template	Description
	- View the SIP address. - View the telephony option and related settings. - View or change the user policy assignments in Skype for Business Server.

When applying ATs for Skype for Business Server User Management, consider your Active Directory topology, and apply only the ATs applicable to your forest configuration.

Table 91: Applying Access Templates for Skype for Business Server User Management

Topology option	Where to apply Access Templates
Single forest topology for Skype for Business Server User Management	Apply ATs to Active Directory domains and containers to which the **Built-in Policy - Skype for Business - User Management** Policy Object (or a copy of that Policy Object) is applied, to allow access to user accounts of Skype for Business Server users managed by Active Roles.
Resource forest topology for Skype for Business Server User Management	Apply ATs to Active Directory domains and containers in external forests to which the **Built-in Policy - Skype for Business - Master Account Management** Policy Object (or a copy of that Policy Object) is applied, to allow access to master accounts of Skype for Business Server users managed by Active Roles. You do not need to apply these Access Templates in the Skype for Business Server forest.
Central forest topology for Skype for Business Server User Management	Apply ATs to: - Active Directory domains and containers in external forests to which the **Built-in Policy - Skype for Business - Master Account Management** Policy Object (or a copy of that Policy Object) is applied, to allow access to master accounts of Skype for Business Server users managed by Active Roles. - Active Directory domains and containers in the Skype for Business Server forest to which the **Built-in Policy - Skype for Business - User Management** Policy Object (or a copy of that Policy Object) is applied, to allow access to login-enabled user accounts of Skype for Business Server users managed by Active Roles in the Skype for Business Server forest.

Configuring the Skype for Business Server User Management feature

This section describes the prerequisites and procedure of deploying the Skype for Business User Management feature in your environment.

This section describes the prerequisites and procedure of deploying the Skype for Business User Management feature in your environment.

- Prerequisites of deploying the Skype for Business Server User Management feature
- Configuring Skype for Business Server User Management in a single-forest environment
- Configuring Skype for Business Server User Management in a multi-forest environment
- Upgrading the Skype for Business Server configuration from an earlier version

Prerequisites of deploying the Skype for Business Server User Management feature

This section lists the prerequisites that your environment must meet to deploy Skype for Business Server User Management.

Deployment conditions of Skype for Business Server

You can configure the Skype for Business Server User Management feature both for single-forest and multi-forest environments.

Deploying Skype for Business Server User Management in a single forest

In case of single forest, Skype for Business Server must be deployed in the forest that holds login-enabled accounts for Skype for Business Server users. For more details, see Single forest topology for Skype for Business Server User Management.

Deploying Skype for Business Server User Management in a multi-forest environment

In case of multiple forests, Skype for Business Server must be deployed in the Skype for Business Server forest only. Do not deploy Skype for Business Server in external user forests or extend the Active Directory schema with Skype for Business Server attributes in those forests. For more details about multi-forest topology options, see Resource forest topology for Skype for Business Server User Management and Central forest topology for Skype for Business Server User Management.

Active Directory forest trust

The multi-forest topology option requires a one-way trust relationship between the Skype for Business Server forest and each user forest so that users can authenticate to the user forest but access services in the Skype for Business Server forest.

> NOTE: Make sure to configure a forest trust instead of an external trust. An external trust relationship supports only NTLM, while a forest trust supports both NTLM and Kerberos, posing no limitations to Skype for Business client authentication options.

Trusts are configured as one-way to prevent unauthorized access to the user forest from the Skype for Business Server forest. For details, see How Domain and Forest Trusts Work in the *Windows Security Collection documentation*.

Skype for Business Server contact management rights

In case of central forest deployment, you must grant Skype for Business Server contact management rights on the container that will hold shadow accounts (contacts enabled for Skype for Business Server in the Skype for Business Server forest). Otherwise, Skype for Business Server security groups will not have sufficient rights to manage contact objects, resulting in a lack of access when Active Roles attempts to enable a shadow account for Skype for Business Server.

To grant Skype for Business Server contact management rights, run the following command in Skype for Business Management Shell.

```
Grant-CsOUPermission -OU "<DN-of-container>" -ObjectType "contact"
```

Replace `<DN-of-container>` with the Distinguished Name of the container where you want to store shadow account, for example:

```
OU=Shadow Accounts,DC=Skype for BusinessServer,DC=lab
```

If the domain has permission inheritance enabled (which is the default case), then you can supply the Distinguished Name of the domain as well, rather than container:

```
Grant-CsOUPermission -OU "DC=Skype for BusinessServer,DC=lab" -ObjectType
"contact"
```

> NOTE: You must be a domain administrator to run the `Grant-CsOUPermission` cmdlet locally.

Active Roles deployment prerequisites for Skype for Business Server User Management

To configure the Skype for Business Server User Management feature, you must install the following Active Roles components in your Active Directory environment:

- Administration Service
- Web Interface
- Active Roles Console

Install these components on the member servers of the account forest or in the Skype for Business Server forest. For installation instructions, see the *Active Roles Quick Start Guide*.

Logging in as an Active Roles Admin

To configure Skype for Business Server User Management, log in as an Active Roles Admin. This ensures that you have sufficient rights to make the necessary configuration changes.

If you use the default configuration of the Active Roles Administration Service, log in with a domain user account that is a member of the Administrators group on the computer running the Administration Service.

Registering domains with Active Roles

Skype for Business Server User Management requires the following domains to be registered with Active Roles:

- At least one domain that holds computers running the Front End Server or Standard Edition Server role in your Skype for Business Server deployment.
- Domains that hold login-enabled users you are going to administer with Skype for Business Server User Management.
- In case of multi-forest topology, the domain in the Skype for Business Server forest that holds shadow accounts for Skype for Business Server users.

When registering a domain, you are prompted to choose which account you want the Administration Service to use to access the domain. You can either specify a so-called override account or let the Administration Service use its service account. With either option, the account must have sufficient rights in the domain you are registering. At minimum, the account must have the following rights:

- In the domain that contains the Skype for Business Server computers, it must be a member of the **RTCUniversalUserAdmins** group.
- In the user domains, it must be a member of the **Account Operators** group.
- In the shadow accounts domain, it must also be a member of the **Account Operators** group.

- For a central forest deployment, the account must also have the rights to create, view, modify and delete contact objects in the shadow accounts domain. To ensure this, make the account a member of the **Domain Admins** group.

  For instructions on how to register domains with Active Roles, see Registering domains with Active Roles.

Configuring Skype for Business Server User Management in a single-forest environment

You can configure the Skype for Business Server User Management feature in a single-forest environment by linking the **Built-in Policy - Skype for Business - User Management** Policy Object to the Active Directory domains or containers that hold the Skype for Business user accounts you want to manage with Active Roles.

To link the Skype for Business User Management Policy Object to an Organizational Unit or domain

1. In the Active Roles Console, navigate to **Configuration** > **Policies** > **Administration** > **Builtin**.
2. In the details pane, right-click the **Built-in Policy - Skype for Business - User Management** Policy Object, then click **Policy Scope**.
3. In the dialog that appears, click **Add**, then select the Organizational Unit or domain.

Out of the box, the Policy Object has all policy settings configured. To change the default policy settings, use the Active Roles Console.

To view or change the settings of the Skype for Business User Management policy

1. In the Active Roles Console, navigate to **Configuration** > **Policies** > **Administration** > **Builtin**.
2. In the details pane, double-click the **Built-in Policy - Skype for Business - User Management** Policy Object.
3. In the **Properties** dialog that appears, go to the **Policies** tab, and double-click the entry in the list of policies.
4. In the **Properties** dialog that appears, modify the settings of the policy:
   - On the **Server** tab, specify how you want Active Roles to select a computer running Skype for Business Server.
   - On the **SIP User Name** tab, configure a rule for generating the SIP user name in the user SIP address.
   - On the **SIP Domain** tab, configure a rule to restrict selection of a SIP domain for the user SIP address.

- On the **Pool** tab, configure a rule to restrict selection of an Enterprise Edition Front End pool or Standard Edition server to which Skype for Business Server users can be assigned.
- On the **Telephony** tab, configure a rule to restrict selection of a Telephony option for Skype for Business Server users.

For more information on these policy settings, see Skype for Business Server User Management policy settings.

Configuring Skype for Business Server User Management in a multi-forest environment

You can configure the Skype for Business Server User Management feature in a multi-forest environment by performing the following main configuration steps:

1. Applying the Master Account Management policy: During this step, you must adjust the **Forest Mode** policy setting in the **Built-in Policy - Skype for Business - Master Account Management** Policy Object, then link that Policy Object to the Active Directory domains or containers in the user forest that contain the master accounts of the login-enabled user accounts you want to manage with Active Roles.
2. Applying the User Management policy: During this step, you must link the **Built-in Policy - Skype for Business - User Management** Policy Object to the Active Directory domains or containers in the Skype for Business Server forest that contains the shadow accounts.

   In case of a central forest, you must also link the **Built-in Policy - Skype for Business - User Management** Policy Object to Active Directory domains or containers in the Skype for Business Server forest that hold login-enabled user accounts you want to manage with Active Roles.

Applying the Master Account Management policy

To configure Skype for Business Server User Management in a multi-forest environment, apply the **Built-in Policy - Skype for Business - Master Account Management** Policy Object to user accounts in Active Directory forests that are external to the Skype for Business Server forest.

To enable the Skype for Business Server User Management feature:

1. Configure the Policy Object according to the Skype for Business Server forest mode in your organization (resource forest or central forest).
2. Link the Policy Object to the domains or containers in the external user forest(s) holding the user accounts you want to manage with Active Roles.

To configure the Master Account Management Policy Object

1. In the Active Roles Console, navigate to **Configuration** > **Policies** > **Administration** > **Builtin**.
2. In the details pane, double-click the **Built-in Policy - Skype for Business - Master Account Management** Policy Object.
3. In the **Properties** dialog that appears, go to the **Policies** tab, and double-click the entry in the list of policies.
4. In the **Properties** dialog that appears, go to the **Forest Mode** tab and select the option that matches the Skype for Business Server forest mode in your Skype for Business Server deployment (see Skype for Business Server forest mode).
5. (Optional) Review the rest of the policy settings if needed:
    - On the **Shadow Account** tab, view or change the container and default description for new shadow accounts.
    - On the **Master Account** tab, view or change the attribute to store a reference to shadow account.
    - On the **Synced** tab, view or change the list of synchronized properties.
    - On the **Substituted** tab, configure your custom list of substituted properties in addition to the default list.
    - On the **Back-synced** tab, view or change the list of back-synchronized properties.

For detailed description of the policy settings, see Master Account Management policy settings for Skype for Business Server User Management.

To link the Master Account Management Policy Object to an Organizational Unit or domain

1. In the Active Roles Console, navigate to **Configuration** > **Policies** > > **Builtin**.
2. In the details pane, right-click the **Built-in Policy - Skype for Business - Master Account Management** Policy Object, then click **Policy Scope**.
3. In the dialog that appears, click **Add**, then select the Organizational Unit or domain.

Applying the User Management policy

You can configure the Skype for Business Server User Management feature for user accounts in the Skype for Business Server forest with the **Built-in Policy - Skype for Business - User Management** Policy Object. To enable the feature, link the policy to domains or containers in the Skype for Business Server forest that contains the shadow accounts of the users.

If your organization uses a central forest topology, also link the policy to Active Directory domains or containers in the Skype for Business Server forest that contains the login-enabled Skype for Business user accounts you want to manage with Active Roles.

To link the User Management Policy Object to an Organizational Unit or domain

1. In the Active Roles Console, navigate to **Configuration** > **Policies** > **Administration** > **Builtin**.
2. In the details pane, right-click the **Built-in Policy - Skype for Business - User Management** Policy Object, then click **Policy Scope**.
3. In the dialog that appears, click **Add**, then select the Organizational Unit or domain.

By default, the Policy Object has all policy settings configured. To change the policy settings, use the Active Roles Console.

To view or change the settings of the User Management Policy Object

1. In the Active Roles Console navigate to **Configuration** > **Policies** > **Administration** > **Builtin**.
2. In the details pane, double-click the **Built-in Policy - Skype for Business - User Management** Policy Object.
3. In the **Properties** dialog that appears, go to the **Policies** tab, and double-click the entry in the list of policies.
4. In the **Properties** dialog box that appears, do any of the following:
   - On the **Server** tab, specify how you want Active Roles to select a computer running Skype for Business Server.
   - On the **SIP User Name** tab, configure a rule for generating the SIP user name in the user SIP address.
   - On the **SIP Domain** tab, configure a rule to restrict selection of a SIP domain for the user SIP address.
   - On the **Pool** tab, configure a rule to restrict selection of an Enterprise Edition Front End pool or Standard Edition server to which Skype for Business Server users can be assigned.
   - On the **Telephony** tab, configure a rule to restrict selection of a Telephony option for Skype for Business Server users.

For more information on the policy settings, see Skype for Business Server User Management policy settings.

Upgrading the Skype for Business Server configuration from an earlier version

If you already manage Skype for Business Server resources with Active Roles Add-on for Skype for Business Server, you can update your deployment to use the Skype for Business Server User Management feature. The procedure has the following main steps:

1. Identify the Active Directory topology option used by the add-on. For more information on how Skype for Business User Management works with the supported forest types, see the following sections:

   - Single forest topology for Skype for Business Server User Management
   - Resource forest topology for Skype for Business Server User Management
   - Central forest topology for Skype for Business Server User Management

   If your organization uses a multi-forest environment, take note of the Distinguished Name of the container in which the add-on creates the shadow accounts.

2. Uninstall Active Roles Add-on for Skype for Business Server from Active Roles Add-on Manager. Then, uninstall the add-on from the computer where it is installed.
3. Upgrade to the latest version of Active Roles. For more information, see the *Active Roles Quick Start Guide*.
4. Deploy the Skype for Business Server User Management feature. Depending on the Active Directory topology option used by the add-on, see the applicable section for more information:

   - In case of a single forest configuration, see Configuring Skype for Business Server User Management in a single-forest environment.
   - In case of a multi-forest configuration, see Configuring Skype for Business Server User Management in a multi-forest environment instructions. During the procedure, configure the **Built-in Policy - Skype for Business - Master Account Management** Policy Object to match the topology option and container for shadow accounts used by the add-on.

The following instructions provide more detailed information on the procedure.

NOTE: The instructions apply to Active Roles Add-on for Skype for Business Server 2.1.

NOTE: The instructions apply to Active Roles Add-on for Skype for Business Server 2.1.

To identify the Active Directory topology option used by the Skype for Business Server Add-on

1. In the Active Roles Console, select **Applications** > **Active Roles Add-on for Skype for Business Server**.
2. In the **Configure Add-on** area of the details pane, review the add-on settings:

   - The Active Directory topology option is selected in the **Active Directory topology** box.
   - If a multi-forest option is selected, the Distinguished Name of the container in which the add-on creates shadow accounts is specified in the **Container for shadow accounts/contacts** box.

If the add-on was configured with the resource forest or central forest option, you must configure and apply the **Built-in Policy - Skype for Business - Master Account Management** Policy Object.

To configure and apply the Master Account Management Policy Object

1. In the Active Roles Console, navigate to **Configuration** > **Policies** > **Administration** > **Builtin**.
2. In the details pane, double-click the **Built-in Policy - Skype for Business - Master Account Management** Policy Object.
3. In the **Properties** dialog that appears, go to the **Policies** tab, and double-click the entry in the list of policies.
4. In the **Properties** dialog that appears, go to the **Forest Mode** tab and select the option that matches the Active Directory topology option that was used by the add-on.
   - If the add-on was configured with the option **Multiple forests - Resource forest**, then select the **Resource forest** option on the **Forest Mode** tab.
   - If the add-on was configured with the option **Multiple forests - Central forest**, then select the **Central forest** option on the **Forest Mode** tab.
5. Go to the **Shadow Account** tab and configure the policy to use the container for shadow accounts that was used by the add-on. To do so, click **This container** > **Browse**, and select the container.
6. Close the **Properties** dialog for the policy entry by clicking **OK**.
7. In the **Properties** dialog box for the Policy Object, click **Apply**, go to the **Scope** tab, then click the **Scope** button on that tab.
8. In the dialog that appears, add the containers that hold the master accounts you managed using the add-on, then click **OK**.
9. Close the **Properties** dialog box for the Policy Object by clicking **OK**.

> TIP: The Skype for Business Server User Management feature will identify the existing master accounts, enabling Active Roles to manage their shadow accounts for Skype for Business Server in the same way as when using the add-on. To speed up the identification of the existing master accounts, you can run the Master Account Management scheduled task manually:
>
> 1. In the Active Roles Console, navigate to the following container:
>
>   **Configuration/Server Configuration/Scheduled Tasks/Builtin**
> 2. Right-click the **Skype for Business - Master Account Management** scheduled task.
> 3. Select **All Tasks**, then click **Execute**.

Managing Skype for Business Server users

The Skype for Business Server User Management feature lets you manage Skype for Business Server users with the Active Roles Web Interface. This includes:

- Enabling or disabling users for Skype for Business Server
- Managing Skype for Business Server user properties

Enabling or disabling users for Skype for Business Server

You can enable, temporarily disable, or remove Active Directory users from the configured Skype for Business Server via the Active Roles Web Interface.

Adding and enabling a new Skype for Business Server user

For an existing Active Directory user account, you can use the Active Roles Web Interface to create and enable a new Skype for Business Server user account by adding the Active Directory user to Skype for Business Server.

To add and enable a new Skype for Business Server user

1. Select the user account in the Active Roles Web Interface for administrators.
2. Click **Enable for Skype for Business Server**. The command is available if:
   - You have sufficient rights in Active Roles to enable users for Skype for Business Server.
   - The selected account is in the scope of the policy provided by Skype for Business Server User Management.
   - The selected account is not yet enabled for Skype for Business Server.

   If any of these conditions are not met, **Enable for Skype for Business Server** will not appear in the Web Interface.
3. On the page that appears:
   - Assign the user to a Skype for Business Server pool.
   - Specify any additional details.
   - Assign Skype for Business Server policies to the user as needed.
4. When ready, click **Finish**.

Disabling or re-enabling a user account for Skype for Business Server

You can use the Active Roles Web Interface to disable a user account for logging in to Skype for Business Server. This allows you to disable a previously enabled user account in Skype for Business Server while retaining all the Skype for Business Server settings that were configured for the user account.

As you do not lose the Skype for Business Server user account settings, you can re-enable a disabled user account again without having to reconfigure the user account.

To disable or re-enable a previously enabled user account for Skype for Business Server

1. In the Active Roles Web Interface, select the user account that you want to disable or re-enable.
2. (Optional) To disable the user account, click **Temporarily Disable for Skype for Business Server**.

   > NOTE: You can disable a Skype for Business Server user account only if:
   >
   > - You have sufficient rights in Active Roles to perform the action.
   > - The selected user account is in the scope of the policy configured for the Skype for Business Server User Management feature.
   > - The selected user account is currently enabled.

3. (Optional) To re-enable the user account, click **Re-enable for Skype for Business Server**.

   > NOTE: You can enable a Skype for Business Server user account only if:
   >
   > - You have sufficient rights in Active Roles to perform the action.
   > - The selected user account is in the scope of the policy configured for the Skype for Business Server User Management feature.
   > - The selected user account is currently disabled.

Removing a user account from Skype for Business Server

You can use the Active Roles Web Interface to remove a user account from Skype for Business Server. This removes all Skype for Business Server-related attributes from the user account, including the identities of any per-user policies that have been assigned to that user account.

You can later re-add the account to Skype for Business Server as described in Adding and enabling a new Skype for Business Server user). However, you will need to reconfigure all Skype for Business Server-related information (including policy assignments) previously associated with that account.

> TIP: If you want to prevent a user from logging on to Skype for Business Server, but do not want to lose all of their account information, you can temporarily disable the user account for Skype for Business Server, as described in Disabling or re-enabling a user account for Skype for Business Server.

To remove a user account from Skype for Business Server

1. In the Active Roles Web Interface, select the user account that you want to remove from Skype for Business Server.
2. Click **Remove from Skype for Business Server**.

   NOTE: This option appears only if:

   - You have sufficient rights in Active Roles.
   - The user account you selected is in the scope of the policy provided by Skype for Business Server User Management.
   - The user account is either enabled or temporarily disabled for Skype for Business Server.

Managing Skype for Business Server user properties

By using the Active Roles Web Interface, you can:

- View or change Skype for Business Server user properties such as the user's SIP address, telephony options and Skype for Business Server policy assignments.
- Move Skype for Business Server users to a different Enterprise Edition Front End pool or Standard Edition server.

Viewing or changing Skype for Business Server user properties

You can view or modify the Skype for Business user settings and user policies of a user account that is enabled or temporarily disabled for Skype for Business Server with the Active Roles Web Interface.

To view or change Skype for Business Server user properties

1. In the Active Roles Web Interface, select the user account whose properties you want to view or change.
2. Click **Skype for Business Server User Properties**.

   NOTE: This option appears only if:

   - You have sufficient rights in Active Roles.
   - The user account you selected is in the scope of the policy provided by Skype for Business Server User Management.
   - The account is enabled or temporarily disabled for Skype for Business Server.
3. On the page that appears, view or change the following settings:

- **Enabled for Skype for Business Server**: Specifies whether the user can log in to Skype for Business Server.

  If you clear this check box, the user will no longer be able to log in to Skype for Business Server.

  Selecting this check box re-enables the user to log in to Skype for Business Server. The function of this check box is equivalent to the **Temporarily Disable for Skype for Business Server** and **Re-enable for Skype for Business Server** options. For more information, see Disabling or re-enabling a user account for Skype for Business Server.

- **SIP address**: Specifies the user’s SIP address (SIP URI), a unique identifier allowing the user to communicate using SIP devices, such as Microsoft Skype for Business. The SIP address consists of the SIP user name on the left side of the @ symbol, and the SIP domain name on the right side. It must be prefaced by “sip:”, such as:

  ```
  sip:Sam.Smith@company.com
  ```

- **Registrar pool**: Identifies the Enterprise Edition Front End pool or Standard Edition server where the Skype for Business Server user is stored. If you need to move the user to a different server or pool, see Moving a user to another server or pool in Skype for Business Server.

- **Telephony**: Specifies whether the Skype for Business Server user can make PC-to-PC calls with audio and video, route incoming and outgoing calls, and control the desktop phone. The possible telephony options are as follows:

  - **PC-to-PC only**: The user can make only PC-to-PC audio or video calls.
  - **Audio/video disabled**: The user cannot make calls with audio and video.
  - **Remote call control**: The user can use Skype for Business Server to control the desktop phone, and can also make PC-to-PC calls.
  - **Enterprise Voice**: The user can use Skype for Business Server to route all incoming and outgoing calls, and can also make PC-to-PC calls.
  - **Remote call control only**: The user can use Skype for Business Server to control the desktop phone, but cannot make PC-to-PC audio calls.

- **Line URI**: Specifies the primary phone number assigned to the Skype for Business Server user.

  > NOTE: Consider the following when using this setting:
  >
  > - This setting applies to all telephony options but **Audio/video disabled**.
  > - The line URI must use the E.164 format and have the `TEL:` prefix. For example:
  >
  >   ```
  >   TEL:+12345678997
  >   ```

> The extension number, if any, must be added at the end of the line URI. For example:
>
> `TEL:+12345678997;ext=65431`

- **Line server URI**: Specifies the URI of the remote call control telephone gateway assigned to the Skype for Business Server user.

  The line server URI is the gateway URI, prefaced by `sip:`. For example:

  `sip:rccgateway@company.com`

  > NOTE: This setting applies to the **Remote call control** and **Remote call control only** options.

4. (Optional) If you need to apply any user policies to the user, configure the following policy settings:

   - **Dial plan policy**: Specifies the dial plan currently assigned to the Skype for Business user, and allows you to assign a different dial plan.

     > NOTE: This setting applies to the **Enterprise Voice** option.

   - **Voice policy**: Specifies the voice policy currently assigned to the Skype for Business Server user, and allows you to assign a different voice policy.
   - **Conferencing policy**: Specifies the conferencing policy currently assigned to the Skype for Business Server user, and allows you to assign a different conferencing policy to the user.
   - **Client version policy** Specifies the client version policy currently assigned to the Skype for Business Server user, and allows you to assign a different client version policy.
   - **PIN policy**: Specifies the personal identification number (PIN) policy currently assigned to the Skype for Business Server user, and allows you to assign a different PIN policy.
   - **External access policy**: Specifies the external access policy currently assigned to the Skype for Business Server user, and allows you to assign a different external access policy.
   - **Archiving policy**: Specifies the archiving policy currently assigned to the Skype for Business Server user, and allows you to assign a different archiving policy.

   > NOTE: Consider the following when configuring any of the policy settings:
   >
   > - The Skype for Business Server user account policy settings allow you to assign specific policies to users that differ from the policy settings assigned to other users (such as global policies). These policies are referred to as "user policies".
   > - In Skype for Business Server, configuring user policies is optional; you can deploy only global policies or site policies. If you configure user policies, you must assign them explicitly to users. When managing Skype for Business Server user settings, you can select the appropriate user policy from a list.

> The list also includes the `<Automatic>` entry: selecting it will apply the global policy (or, if defined, the site policy) to the user.

Moving a user to another server or pool in Skype for Business Server

You can move a user account that is enabled or temporarily disabled for Skype for Business Server to a specific Enterprise Edition Front End pool or Standard Edition server with the Active Roles Web Interface.

To move a Skype for Business Server user account to a different server or pool

1. In the Active Roles Web Interface, select the user account you want to move.
2. Click **Move to Skype for Business Server Pool**.

   > NOTE: This option appears only if:
   >
   > - You have sufficient rights in Active Roles.
   > - The user account you selected is in the scope of the policy provided by Skype for Business Server User Management.
   > - The selected user is enabled or temporarily disabled for Skype for Business Server.
3. On the page that appears, select the server or pool to which you want to move the Skype for Business Server user.
4. Click **Finish**.

22

Exchanging provisioning information with Active Roles SPML Provider

Active Roles SPML Provider is designed to exchange the user, resource, and service provisioning information between SPML-enabled enterprise applications and Active Directory.

Active Roles SPML Provider supports the Service Provisioning Markup Language Version 2 (SPML v2), an open standard approved by the Organization for the Advancement of Structured Information Standards (OASIS). SPML is an XML-based provisioning request-and-response protocol that provides a means of representing provisioning requests and responses as SPML documents. The use of open standards provides the enterprise architects and administrators with the flexibility they need when performing user management and user provisioning in heterogeneous environments.

Key SPML Provider features

The key features of Active Roles SPML Provider are as follows:

- **Support for two operation modes**: SPML Provider can be configured to operate in *proxy mode* or in *direct access mode*. In proxy mode, SPML Provider accesses Active Directory or Active Directory Lightweight Directory Services (AD LDS, formerly known as ADAM) through Active Roles used as a proxy service, while in direct access mode, SPML Provider directly accesses Active Directory or AD LDS.
- **Support for equivalent LDAP operations:** SPML Provider can perform equivalent LDAP operations such as addRequest, modifyRequest, deleteRequest, and lookupRequest.
- **Support for Azure AD, AD, and AD LDS data management:** SPML Provider enables SPML-conformant applications to read from and write to Azure AD, Active Directory (AD), and AD LDS.

- **Search Capability support:** SPML Provider allows SPML-enabled applications to search for relevant directory objects based on various search criteria.
- **Password Capability support:** SPML Provider allows SPML-enabled applications to perform basic password management tasks such as setting and expiring user passwords.
- **Suspend Capability support**: SPML Provider allows SPML-enabled applications to effectively enable, disable and deprovision user accounts in Active Directory.
- **Flexible Configuration options:** There is support for many different configuration options that enable the administrator to adjust the behavior and optimize the SPML Provider performance.
- **IIS Security Support:** SPML Provider supports all IIS security configurations, including integrated Windows authentication, basic authentication, and basic authentication over Secure Sockets Layer (SSL).
- **Support for using Active Roles controls**: In proxy mode, you can send Active Roles controls to the Active Roles Administration Service with an SPML request to perform an administrative operation. In your request, you can also define the Active Roles controls that the Administration Service must return in the SPML response.

SPML Provider usage scenarios

SPML Provider can be used for a variety of purposes. Some common scenarios for using SPML Provider are as follows:

- **Non-Windows applications:** The systems running non-Windows applications that need to communicate with Active Directory can do this through SPML Provider. For example, with SPML Provider, Unix applications can manage Unix-enabled user accounts in Active Directory. In proxy mode, SPML Provider allows existing SPML-compatible provisioning systems, such as SUN Java System Identity Manager and IBM Tivoli Directory Integrator to take advantage of the functionality of Active Roles.
- **Web services:** The use of directories in Web services is growing rapidly. Additionally, XML is becoming the default language for use with Web services. SPML Provider fills the gap between XML documents and Active Directory services, enabling applications that must provide or use Web services to communicate with Active Directory.
- **Handheld and portable devices:** Data-enabled cell phones or PDAs that need an access to directory data may not contain a client for the ADSI LDAP Provider but might be able to use the SPML communication protocol to access Active Directory over the Internet.
- **Firewall access:** Certain firewalls cannot pass LDAP traffic because they cannot audit it, but these firewalls can pass XML. In such cases, applications can use SPML Provider to communicate with Active Directory across a firewall.

Basic SPML Provider concepts and definitions

Active Roles SPML Provider operates based on the concepts defined in SPML v2. This section introduces and describes these key concepts and definitions as applied to SPML Provider.

A **Client** (Requesting Authority or Requestor) is any SPML-compliant application that sends well-formed SPML requests to the Active Roles SPML Provider and receives responses from it. Clients can include various business applications, such as human resources (HR) databases or Identity Management systems. There is no direct contact between a client and the target (Active Roles or an Active Directory server).

Active Roles SPML Provider (Provisioning Service Provider or PSP) is a Web service that uses the Simple Object Access Protocol (SOAP) over HTTP for communications. SPML Provider can directly access Active Directory data or communicate with Active Directory using the Active Roles proxy service. SPML Provider acts as an intermediary between a client and the target (Active Directory domain controller or Active Roles).

In proxy mode, **Active Roles** represents the Provisioning Service Target (or Target) that is available for provisioning actions through SPML Provider. The target has a unique identifier (targetID) that is maintained by SPML Provider and is used in a request or a response.

AD Objects (Provisioning Service Objects or PSO) represent directory objects that SPML Provider manages. A client can add, delete, modify, or look up a directory object. Each object has a unique identifier (PSO ID). In SPML Provider, an object DN is used as a PSO ID.

> NOTE: The Requestor, Provisioning Service Provider, Provisioning Service Target, and Provisioning Service Objects are key notions described in the official SPML v2 specification.

For detailed information on the concepts defined in SPML v2, see Section 2 “Concepts” of the OASIS SPML v2 specification, available for download at http://www.oasis-open.org/specs/index.php#spmlv2.0.

How SPML Provider works

With SPML Provider, applications can use SPML documents to look up, retrieve and update directory data in Active Directory, Azure AD, and AD LDS. SPML Provider converts XML elements and attributes into commands used to make changes to Active Directory and retrieve data from Active Directory. SPML Provider can also convert the response received from Active Roles or Active Directory to XML format. These conversions are based on and are in compliance with the OASIS SPML v2 - DSML v2 Profile specification.

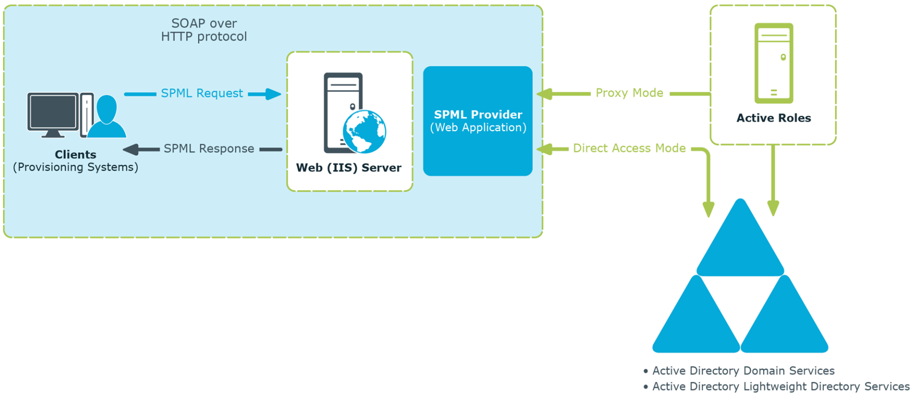
SPML Provider runs as a Web application on a Web server running Microsoft Internet Information Services (IIS), and uses SOAP over HTTP to transmit and receive directory requests from client computers.

The SPML Provider environment includes the following components:

- **Clients that use SPML v2**: These clients are applications that manage directory objects (for example, user accounts). A client issues SPML requests that describe operations to be performed on the directory object and send these requests to SPML Provider.
- **SPML Provider**: Receives and processes client requests, and returns a response to the client.
- **Active Roles:** In proxy mode, this is the endpoint for provisioning requests and the actual software that manages directory objects.
- **Active Directory, Azure AD, or AD LDS**: In proxy mode, SPML Provider can access Active Directory or Azure AD domains and AD LDS instances that are registered with Active Roles as managed domains, Azure AD tenants, and managed AD LDS instances, respectively. In direct access mode, SPML Provider can access the domain controller or the AD LDS instance defined in the `SPML.Config` file. For more information, see "Configuring SPML Provider" later in this document.

The following diagram illustrates the flow of requests and responses through the SPML Provider environment components:

Figure 161: Flow of requests and responses through the SPML Provider environment components



As shown in the diagram, the client/SPML Provider communications are based on the simple request/response protocol.

In proxy mode, SPML Provider works in the following way:

1. A client issues a well-formed SPML request using the SOAP over HTTP protocol. This request goes to a server running IIS, where it is routed to SPML Provider.
2. SPML Provider examines the request for conformance to the SPML format.
3. If the request complies with the SPML format, the SPML Provider submits the request to Active Roles. Based on the client request, Active Roles retrieves or modifies data in Active Directory, Azure AD, or in AD LDS.
4. After performing the requested operation, Active Roles sends the result of the operation back to SPML Provider.
5. SPML Provider then processes this result data and sends the result of the performed operation back to the client in the form of an SPML response.

In direct access mode, SPML Provider works in the following way:

1. A client issues a well-formed SPML request using the SOAP over HTTP protocol. This request goes to a server running IIS, where it is routed to SPML Provider.
2. SPML Provider examines the request for conformance to the SPML format.
3. If the request conforms to the SPML format, SPML Provider retrieves or modifies the relevant data in Active Directory or in AD LDS (ADAM).
4. SPML Provider sends the result of the performed operation back to the client in the form of an SPML response.

If the client request does not conform to the SPML format, the client receives an SPML response that describes the encountered error.

Configuring Active Roles SPML Provider

Configuration settings allow the administrator to configure SPML Provider and its schema in order to adjust the SPML Provider behavior. Administrators can, for example, specify the required managed objects and attributes in the schema, or choose the type of execution (disabling or deprovisioning objects) for the Suspend operation.

Configuring SPML Provider settings in the SPML.Config file

The SPML Provider configuration settings can be found in the `SPML.Config` file located in the **Web** subfolder of the SPML Provider installation folder. The `SPML.Config` file contains data in the XML format. You can open and edit the configuration file with a common text editor, such as Notepad.

> NOTE: After you modify configuration settings,restart the IIS application pool for the SPML Provider website for the changes to take effect.

The following table describes the XML elements used in the SPML Provider configuration file.

Table 92: XML elements used in the SPML Provider configuration file

Element	Parent element	Description
service	configuration	In proxy mode, specifies the name of the computer running the Active Roles Administration Service. In direct access mode, specifies the name of the AD domain controller or AD LDS server. The name of the AD LDS server must be in the form

Element	Parent element	Description
		<servername:portnumber>.
adsiProvider	configuration	Specifies the progID of the ADSI Provider. In proxy mode, the progID is EDMS. In direct access mode, the progID is LDAP.
schemaFile	configuration	Contains the name of the file that defines the DSML Profile schema for SPML Provider. By default, the file name is SPMLSchema.Config. The schema file must be located in the same folder as the SPML.Config file.
defaultMaxSelect	search	Specifies the maximum number of search results that SPML Provider can return without page splitting. The default value is 1000.
pageSize	search	Specifies the maximum number of search results per page. The default value is 25. NOTE: If **pageSize** is set to **`0`**, SPML Provider returns search results without page splitting.
class	password	Contains the LDAP display name of the schema class of objects on which SPML Provider is expected to perform the Password Capability-related operations such as **`setPassword`** and **`expirePassword`**.
class	suspend	Contains the LDAP display name of the schema class of objects on which SPML Provider is expected to perform the Suspend Capability-related operations such as **suspend**, **resume**, and **active**.
suspendAction	suspend	Possible values: disable or deprovision. The default value is disable. If **suspendAction** is set to **`disable`**, SPML Provider disables the specified user account on the target. If **suspendAction** is set to **`deprovision`**, SPML Provider deprovisions the specified user account in accordance with the deprovisioning policies defined by Active Roles.
checkOutput	configuration	Possible values: true or false. The default value is false. **`true`** causes SPML Provider to check the string attribute values retrieved from the underlying directory before adding them to a response. If an

Element	Parent element	Description
		attribute value contains illegal characters that could break the XML parser on the client side, SPML Provider converts the attribute value to the base64binary format and then adds the result of the conversion to the response. Note that this option may result in performance degradation of SPML Provider as checking every attribute value is a resource-intensive operation. `false` causes SPML Provider not to check the string attribute values retrieved from the underlying directory. An attribute value is added to the response without any conversion even if the value contains illegal characters. NOTE: In accordance with the XML specification, the legal character range is as follows: `#x9 | #xA | #xD | [#x20-#xD7FF] | [#xE000-#xFFFD] | [#x10000-#x10FFFF]`. With **checkOutput** set to `true`, SPML Provider ensures that attribute values in a response contain only characters from the legal character range.

Sample SPML Provider configuration file

The following is an example of the configuration file for SPML Provider configured to operate in proxy mode. If SPML Provider and the Active Roles Administration service are installed on the same computer, the default configuration settings look as follows:

```
<?xml version="1.0"?>
  <configuration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="urn:quest:names:SPMLProvider">
    <service>localhost</service>
    <adsiProvider>EDMS</adsiProvider>
    <schemaFile>SPMLSchema.Config</schemaFile>
    <capabilities>
      <search>
        <defaultMaxSelect>1000</defaultMaxSelect>
        <pageSize>25</pageSize>
      </search>
      <password>
        <appliesTo>
          <class>user</class>
        </appliesTo>
      </password>
      <suspend>
        <appliesTo>
          <class>user</class>
        </appliesTo>
        <suspendAction>disable</suspendAction>
      </suspend>
    </capabilities>
    <checkOutput>false</checkOutput>
  </configuration>
```

Extending the SPML Provider schema

The SPML Provider schema defines the XML structure of the objects and attributes that SPML Provider manages. You can modify the schema to manage new types of objects or object properties. Thus, you can add the class and attribute definitions to the schema in order to meet the needs of your organization.

> NOTE: In proxy mode, you can add only those object classes and attributes that are valid according to the Active Roles schema.

The SPML Provider schema is stored in the `SPMLSchema.Config` file. The `SPMLSchema.Config` file is located in the **Web** subfolder of the SPML Provider installation folder.

The schema format corresponds to the DSML Version 2 profile (DSMLv2). For detailed information on the DSML v2 profile, refer to the OASIS SPML v2 - DSML v2 Profile specification. The specification describes the use of the DSML protocol as a data model for SPML- based provisioning and can be accessed from the OASIS Web site at http://www.oasis-open.org/specs/index.php#spmlv2.0.

Using Active Roles SPML Provider

To access SPML Provider, enter the following URL in your web browser:

`http://<hostname>/ARServerSPML/SPMLProvider.asmx`

In this URL, `<hostname>` is the name of the computer where SPML Provider is installed.

> NOTE: The SPML Provider web service is described by a Web Services Description Language (WSDL) file. To obtain a WSDL description of SPML Provider, open the following URL:
>
> **`http://<hostname>/ARServerSPML/SPMLProvider.asmx?WSDL`**

SPML Provider operation modes

You can configure SPML Provider to operate in two modes.

- **Proxy mode**: In this mode, SPML Provider accesses Active Directory, Azure AD, or AD LDS using the Active Roles proxy service. While in proxy mode, SPML Provider can manage objects in all Active Directory domains and/or AD LDS instances that are registered with Active Roles as managed domains and managed AD LDS instances, respectively.

  With proxy mode, SPML Provider not only extends the functionality of Active Roles, but also offers better interoperability than Active Roles ADSI Provider, due to SPML Provider using open standards, such as HTTP, XML, and SOAP.

  > TIP: To take full advantage of the management capabilities of Active Roles, One Identity recommends using proxy mode when configuring SPML Provider.

- **Direct access mode**: In this mode, SPML Provider directly accesses Active Directory, Azure AD, or AD LDS.

  When working in direct access mode, SPML Provider can only manage objects that are located in the Active Directory / Azure AD domain or AD LDS instance to which SPML Provider is connected via the Active Directory domain controller (DC) or the AD LDS server.

Active Roles controls supported by SPML Provider

Active Roles implements special parameters called "Active Roles controls" (hereafter "controls"). Controls allow you to customize request processing.

In proxy mode, SPML Provider clients can send controls to the Active Roles Administration Service with an SPML request to perform an administrative operation. The Administration Service can process the controls. On the other hand, the Administration Service can return its own control to the SPML Provider client, then the client can process that control. The controls a client sends to the Administration Service are referred to as `InControls`, while the controls the Administration Service returns to the client are referred to as `OutControls`.

For more information, see the following sections:

- For details on sending the `InControl`-type controls to the Active Roles Administration Service with an SPML request, see Sending controls to the Active Roles Administration Service.
- For details on specifying a set of the `OutControl`-type controls that the Active Roles Administration Service will return with an SPML response, see Specifying controls to return to the SPML Provider client.

For more information about Active Roles controls and for the list of available built-in controls, see the *Active Roles SDK documentation*.

IMPORTANT: All elements described in this section must be defined at the beginning of your SPML request. For a sample of use, see Sample SPML requests.

Sending controls to the Active Roles Administration Service

This section covers the `controls` and `control` XML elements that your SPML request must include to send controls to the Active Roles Administration Service.

Element name: `controls`

Element description: Specifies a collection of InControl-type controls to send to Administration Service.

Child elements: `control`

Attributes:

Table 93: Controls attributes

attribute name	attribute description
xmlns	Declares the namespase for all child elements of the `controls` element. This attribute must be set to `quest:ars:SPML:2:0`

Element name: `control`

Element description: Describes a control to send to the Administration Service.

Parent elements: `controls`

Child elements: None

Attributes:

Table 94: Control attributes

attribute name	attribute description
name	Specifies the name of the control.

The control value in the `control` element body must be specified as follows:

```
<control name=%control name%>%control value%</control>
```

To send an empty control, use the following syntax:

```
<control name=%control name% />
```

Specifying controls to return to the SPML Provider client

This section covers the `controlsForOutput` and `control` XML elements that your SPML request must include to specify a set of controls to return to the SPML Provider client.

Element name: `controlsForOutput`

Element description: Specifies a collection of OutControl-type controls to return to SPML client.

Child clements: `control`

Attributes:

Table 95: Attributes for controlsForOutput

attribute name	attribute description
xmlns	Declares the namespase for all child elements of the `controls` element. This attribute must be set to `quest:ars:SPML:2:0`

Element name: `control`

Element description: Describes a control to return to SPML Provider client with an SPML response.

Parent elements: `controlsForOutput`

Child elements: None

Attributes:

Table 96: Attributes for control

attribute name	attribute description
name	Specifies the name of the control.

The `control` elements used to specify controls to return with SPML response must be defined as follows:

```
<control name=%control name% />
```

Sample SPML requests

This section provides sample SPML requests and SPML responses to show how to use the feature with Active Roles.

> NOTE: You must modify the sample SPML requests to adjust them to your environment. For example, before using the first sample, set the `ID` attribute of the `psoID` element to the distinguished name of the user account you want to modify.

SPML request to modify a user object

This sample shows how an SPML Provider client can send a request to modify the specified user object. With this request, the client sends the `AllowApproval` built-in control set to `Confirm`, and the `CustomControl` control set to `MyCustomValue`. The request also contains the `controlsForOutput` element, which specifies that Active Roles Administration Service will return values of the `OperationStatus` and `CustomControl` controls in the SPML response.

> TIP: For more information about the use of the `AllowApproval` and `OperationStatus` controls, refer to the *Active Roles SDK documentation*.

```
<?xml version="1.0"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
<spml:modifyRequest xmlns:spml="urn:oasis:names:tc:SPML:2:0">
<controls xmlns="quest:ars:SPML:2:0">
<control name="AllowApproval">Confirm </control>
<control name="CustomControl">MyCustomValue </control>
</controls>
<controlsForOutput xmlns="quest:ars:SPML:2:0">
<control name="OperationStatus"/>
<control name="CustomControl"/>
</controlsForOutput>
<spml:psoID ID="CN=JDOE,OU=Users,DC=mycompany,DC=com"/>
<spml:modification>
<modification name="description" operation="replace"
xmlns="urn:oasis:names:tc:DSML:2:0:core">
<value>New description</value>
</modification>
</spml:modification>
</spml:modifyRequest>
</soap:Body>
</soap:Envelope>
```

SPML response of modifying a user object

The following example provides a sample response to the previous request of modifying a user object.

```
<?xml version="1.0" encoding="UTF-16"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<soap:Body>
<modifyResponse status="success" xmlns="urn:oasis:names:tc:SPML:2:0">
<controls xmlns="quest:ars:SPML:2:0">
<control name="OperationStatus">Completed</control>
```

```
<control name="CustomControl">ReturnedValue</control>
</controls>
<pso>
<psoID ID="CN=JDOE,OU=Users,DC=mycompany,DC=com"/>
<data>
<attr name="cn" xmlns="urn:oasis:names:tc:DSML:2:0:core">
<value xsi:type="xsd:string">Admin1</value>
</attr>
<attr name="objectClass" xmlns="urn:oasis:names:tc:DSML:2:0:core">
<value xsi:type="xsd:string">top</value>
<value xsi:type="xsd:string">person</value>
<value xsi:type="xsd:string">organizationalPerson</value>
<value xsi:type="xsd:string">user</value>
</attr>
<attr name="objectCategory" xmlns="urn:oasis:names:tc:DSML:2:0:core">
<value
xsi:type="xsd:string">CN=Person,CN=Schema,CN=Configuration,DC=dom,DC=lab,DC=loca
l</value>
</attr>
<attr name="objectGUID" xmlns="urn:oasis:names:tc:DSML:2:0:core">
<value xsi:type="xsd:base64Binary">Aodvua6TAE+Ja9O3vnRntg==</value>
</attr>
<attr name="description" xmlns="urn:oasis:names:tc:DSML:2:0:core">
<value xsi:type="xsd:string">New description</value>
</attr>
</data>
</pso>
</modifyResponse>
</soap:Body>
</soap:Envelope>
```

Supported Azure features

Active Roles SPML Provider supports creating Azure users, Azure groups, and Azure contacts.

> NOTE: To create Azure users, groups or contacts in an Azure AD deployment with SPML Provider, you must configure an Azure tenant in the Active Roles Configuration Center, and consent Active Roles as an Azure application.
>
> For more information, see Configuring Active Roles to manage Azure AD using the GUI.

Sample SPML requests for creating Azure users, groups or contacts in Azure AD

The following sample SPML requests show how to create Azure objects in an Azure AD deployment configured for Active Roles.

Sample SPML request for creating an Azure user

```
<?xml version="1.0"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
<addRequest xmlns="urn:oasis:names:tc:SPML:2:0" returnData="everything">
<containerID ID="OU=AzureOU, DC=Sample,DC=local,DC=com"/>
<data>
<attr name="cn" xmlns="urn:oasis:names:tc:DSML:2:0:core">
<value>AzureUser</value>
</attr>
<attr name="description" xmlns="urn:oasis:names:tc:DSML:2:0:core">
<value>Azure test user</value>
</attr>
<attr name="sAMAccountName" xmlns="urn:oasis:names:tc:DSML:2:0:core">
<value>AzureUser</value>
</attr>
<attr name="objectClass" xmlns="urn:oasis:names:tc:DSML:2:0:core">
<value>user</value>
</attr>
<attr name="mail" xmlns="urn:oasis:names:tc:DSML:2:0:core">
<value>AzureUser@ARStestdev.onmicrosoft.com</value>
</attr>
<attr name="otherHomePhone" xmlns="urn:oasis:names:tc:DSML:2:0:core">
<value>12135555555</value>
<value>12134444444</value>
</attr>
<attr name="edsaPassword" xmlns="urn:oasis:names:tc:DSML:2:0:core">
<value>P@ssw0rd123</value>
</attr>
<attr name="edsaAccountIsDisabled" xmlns="urn:oasis:names:tc:DSML:2:0:core">
<value>FALSE</value>
</attr>
<attr name="userPrincipalName" xmlns="urn:oasis:names:tc:DSML:2:0:core">
<value>AzureUser@ARStestdev.onmicrosoft.com</value>
</attr>
<attr name="edsvaAzureOffice365Enabled"
xmlns="urn:oasis:names:tc:DSML:2:0:core">
<value>TRUE</value>
</attr>
<attr name="edsaAzureUserPrincipalName"
xmlns="urn:oasis:names:tc:DSML:2:0:core">
<value>AzureUser@ARStestdev.onmicrosoft.com</value>
</attr>
<attr name="edsaAzureUserAccountEnabled"
xmlns="urn:oasis:names:tc:DSML:2:0:core">
<value>TRUE</value>
</attr>
```

```
<attr name="edsaAzureUserDisplayName" xmlns="urn:oasis:names:tc:DSML:2:0:core">
<value>AzureUser</value>
</attr>
</data>
</addRequest>
</soap:Body>
</soap:Envelope>
```

Sample SPML request for creating an Azure group

```
<?xml version="1.0"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
<addRequest xmlns="urn:oasis:names:tc:SPML:2:0" returnData="everything">
<psoID ID="CN=GroupName,OU=AzureOU,DC=Sample,DC=local,DC=com"/>
<data>
<attr name="objectClass" xmlns="urn:oasis:names:tc:DSML:2:0:core">
<value>group</value>
</attr>
<attr name="description" xmlns="urn:oasis:names:tc:DSML:2:0:core">
<value>My test group</value>
</attr>
<attr name="mailEnabled" xmlns="urn:oasis:names:tc:DSML:2:0:core">
<value>false</value>
</attr>
<attr name="mail" xmlns="urn:oasis:names:tc:DSML:2:0:core">
<value> GroupName@company.com</value>
</attr>
<attr name="mailNickName" xmlns="urn:oasis:names:tc:DSML:2:0:core">
<value> GroupName</value>
</attr>
<attr name="edsvaAzureOffice365Enabled"
xmlns="urn:oasis:names:tc:DSML:2:0:core">
<value>TRUE</value>
</attr>
<attr name="edsaAzureGroupDisplayName" xmlns="urn:oasis:names:tc:DSML:2:0:core">
<value> GroupName</value>
</attr>
<attr name="edsaEstablishGroupEmail" xmlns="urn:oasis:names:tc:DSML:2:0:core">
<value>false</value>
</attr>
<attr name="edsaAzureGroupType" xmlns="urn:oasis:names:tc:DSML:2:0:core">
<value>-2147483646</value>
```

```
</attr>
</data>
</addRequest>
</soap:Body>
</soap:Envelope>
```

Sample SPML request for creating an Azure contact

```
<?xml version="1.0"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
<addRequest xmlns="urn:oasis:names:tc:SPML:2:0" returnData="everything">
<containerID ID="OU=AzureOU,DC=Sample,DC=local,DC=com"/>
<data>
<attr name="cn" xmlns="urn:oasis:names:tc:DSML:2:0:core">
<value>AzureContact</value>
</attr>
<attr name="description" xmlns="urn:oasis:names:tc:DSML:2:0:core">
<value>AzureContact</value>
</attr>
<attr name="objectClass" xmlns="urn:oasis:names:tc:DSML:2:0:core">
<value>Contact</value>
</attr>
<attr name="edsvaAzureOffice365Enabled"
xmlns="urn:oasis:names:tc:DSML:2:0:core">
<value>TRUE</value>
</attr>
<attr name="edsaAzureContactEmail" xmlns="urn:oasis:names:tc:DSML:2:0:core">
<value>AzureContact@test.com</value>
</attr>
</data>
</addRequest>
</soap:Body>
</soap:Envelope>
```

Supported SPML Provider operations

SPML Provider implements the SPML v2 core protocol and supports core operations that are required for compliance with the official SPML v2 specification. The following table lists the core operations supported by SPML Provider.

Table 97: Core operations supported by SPML Provider

Operation	Description
listTargets	Lists targets available for provisioning through SPML Provider and the SPML Provider's supported set of capabilities for targets.
add	Creates a new object on the target.
modify	Changes the specified object on the target.
lookup	Obtains the XML that represents the specified object on the target.
delete	Removes the specified object from the target.

In addition to core operations required for conformance to the SPML v2 specification, SPML Provider supports a set of optional operations (Capabilities) that are functionally related. The following tables list the Capabilities supported by SPML Provider.

Search capability

Table 98: Search capabilities supported by SPML Provider

Operation	Description
search	Obtains every object that matches the specified query.
iterate	Obtains the next set of objects from the result set selected for a search operation.
closeIterator	Informs SPML Provider that the client no longer intends to iterate the search result.

Suspend capability

Table 99: Suspend capabilities supported by SPML Provider

Operation	Description
suspend	Disables/deprovisions the specified object on the target.
resume	Re-enables the specified object on the target.
active	Checks whether the specified object on the target has been suspended.

Password Capability

Table 100: Password capabilities supported by SPML Provider

Operation	Description
setPassword	Specifies a new password for a user account.
expirePassword	Marks as invalid the current password for a user account.

For detailed information on the SPML v2 operations, refer to the “Operations” section in the official SPML v2 specification, available for download at http://www.oasis-open.org/specs/index.php#spmlv2.0.

SPML Provider samples of use

SPML Provider implements the SPML v2 core protocol and supports the DSML v2 Profile for SPML operations. SPML Provider comes with a sample client that includes examples illustrating how to construct SOAP messages that contain SPML payloads to perform common directory operations.

To work with the examples in the SPML Provider sample client

1. From the **Start** menu on the computer on which SPML Provider is installed, select **Active Roles SPML Provider** to open the home page of the sample client in your web browser.
2. On the **Samples of Use** home page, under **How do I**, click the example you want to examine.

   For instance, you might click **Create new user** to view, modify, and perform the SPML v2 request that creates a user object.
3. On the page that opens, in the **SPMLv2 request** box, view the SOAP message that will be sent to SPML Provider.

   You may need to modify the SOAP message in order to adjust it to your environment. Thus, with the **Create new user** example, you have to set the ID attribute of the `<ContainerID>` element to the distinguished name (DN) of the container where you want to create a new user.
4. To send the SOAP message to SPML Provider, click **Send Request**.
5. In the **SPMLv2 response** box, view the SOAP message returned by SPML Provider in response to your request.
6. To examine another example, return to the home page, then click the desired example.

SPML Provider configuration settings in the sample.config file

You can set SPML Provider configuration options in a sample client configuration file. This is useful to test the SPML Provider functionality before live deployment. Administrators can, for example, specify the desired settings for the sample container object (OU) that will be used in sample SPML v2 operations.

The configuration settings of the SPML Provider sample client can be found in the `sample.config` file located in the **Samples** sub-folder of the SPML Provider installation folder.

The `sample.config` file contains data in XML format. You can open and edit the configuration file with any common text editor, such as Notepad. The default configuration settings in the `sample.config` file look as follows:

```
<samples>
<server>localhost</server>
<url>ARServerSPML/spmlprovider.asmx</url>
<sampleContainerName>OU=MyOU,DC=Company,DC=com</sampleContainerName>
</samples>
```

The following table provides reference information for the XML elements used in the `sample.config` file.

Table 101: XML elements used in the sample.config. file

Element	Parent element	Description
server	samples	Specifies the name of the computer running SPML Provider.
url	samples	Specifies Web address of SPML Provider. The default address is `ARServerSPML/spmlprovider.asmx`.
sampleContainerName	samples	Specifies the distinguished name of the container (OU) used in the sample SPML v.2 requests.

Core SPML Provider operation samples

The following table lists all examples included in the SPML Provider core operation samples.

Table 102: Core operation samples

Operation	Description
List targets available for provisioning with SPML Provider	This example illustrates how to retrieve the targets available for provisioning with SPML Provider. To do this, SPML Provider performs the **listTargets** operation. The request message includes the following XML elements: • The `<soap:Envelope>` and `<soap:Body>` SOAP elements enclose the SPML payload. • The `<listTargetsRequest>` element asks SPML Provider to declare the set of targets that SPML Provider exposes for provisioning operations. The response lists the supported targets, including the schema definitions for each target and the set of capabilities that SPML Provider supports for each target. The contents of the `<listTargetsResponse>` element conform to the OASIS SPML v2 specification.
Create new user Create new user (using direct access mode)	These examples illustrate how to create a user account object in two operation modes. To create a new object, SPML Provider performs the **add** operation. The request message includes the following XML elements: • The `<soap:Envelope>` and `<soap:Body>` SOAP elements enclose the SPML payload. • The `<addRequest>` element asks SPML Provider to create a new object. • The `<containerID>` element specifies the distinguished name of the container in which to create the new object. • The `<data>` element encloses the elements that specify attribute values on the new object. Thus, in accordance with the `objectClass` attribute value, SPML Provider is requested to create a user account. The operation response indicates whether the user account is successfully created. NOTE: To provision a user account in direct access mode, perform the following steps: 1. Create a request to create a new user account, as described above. 2. Create a request to set the user password (see *Set user password* in *Password capability samples*. 3. Create a request to enable the user account (see *Resume user*

Operation	Description
	account in *Suspend capability samples*).
Create new user (approval aware)	This example illustrates how to create a user account if this operation is subject to approval by designated approvers. For more information about approval activities and workflows, see Workflows. If the creation of user is subject to approval, to perform the operation, your SPML request must contain the `AllowApproval` built-in control. For information about how to use controls in SPML requests, see Active Roles controls supported by SPML Provider. To create a new object, SPML Provider performs the **add** operation. The request message includes the following XML elements: - The `<soap:Envelope>` and `<soap:Body>` SOAP elements enclose the SPML payload. - The `<addRequest>` element asks SPML Provider to create a new object. - The `<controls>` element includes the child element `<control>` that sets the `AllowApproval` control to the `Confirm` value. - The `<controlsForOutput>` element includes the child element `<control>`, which specifies that the `OperationStatus` control will be returned with the SPML response. - The `<containerID>` element specifies the distinguished name of the container in which to create the new object. - The `<data>` element encloses the elements that specify attribute values on the new object. Thus, in accordance with the `objectClass` attribute value, SPML Provider is requested to create a user account. The operation response contains the `OperationStatus` control value that indicates the creation operation status. For example, if the user creation operation is subject to approval, the `OperationStatus` control returns the `Pending` value. In this case, the operation is waiting for approval by designated approvers. For more information about possible values of the `OperationStatus` control, see the *Active Roles SDK documentation*.
Create a user whose logon name is not in compliance with Active Roles policies	This example illustrates an attempt to create a new user account whose logon name does not conform to the Active Roles policies. Because the user logon name does not conform to the Active Roles policies, the creation operation fails and the operation response includes an error message returned by Active Roles. For example, an attempt to set the `sAMAccountName` attribute to a string of more than 20 characters causes the user creation operation to fail, with the response containing a message that provides some details on the error condition.

Operation	Description
Create new group	This example illustrates how to create the group object **SPMLGroup** in the **mycompany.com** domain. To create a new object, SPML Provider performs the **add** operation. The request message includes the following XML elements: • The `<soap:Envelope>` and `<soap:Body>` SOAP elements enclose the SPML payload. • The `<addRequest>` element asks SPML Provider to create a new object. • The `<psoID>` element specifies the distinguished name of the object to be created. • The `<data>` element encloses the elements that specify attribute values on the new object. Thus, in accordance with the `objectClass` attribute value, SPML Provider is requested to create a group object.
Modify user attributes	This example illustrates how to modify the `description` attribute of the **John Smith** user object in the **mycompany.com** domain. To modify the object attribute, SPML Provider performs the **modify** operation. The request message includes the following XML elements: • The `<soap:Envelope>` and `<soap:Body>` SOAP elements enclose the SPML payload. • The `<modifyRequest>` element asks SPML Provider to make changes to a specified object. • The `<psoID>` element specifies the distinguished name of the user account to be modified. • The `<modification>` element specifies the type of change as `replace`, causing the new values to replace the existing attribute values. • The `<data>` element encloses the elements that specify the new attribute values.
Modify Shared mailbox user permissions	Modify or replace the **edsaUserMailboxSecurityDescriptorSddl** attribute of the Shared mailbox object. To modify the object attribute, SPML Provider performs the **modify** operation. The request message includes the following XML elements: • The `<soap:Envelope>` and `<soap:Body>` SOAP elements enclose the SPML payload.

Operation	Description
	• The `<modifyRequest>` element asks SPML Provider to make changes to a specified object. • The `<psoID>` element specifies the distinguished name of the user account to be modified. • The `<modification>` element specifies the type of change as `replace`, causing the new values to replace the existing attribute values. • The `<data>` element encloses the elements that specify the new attribute values, in SDDL format along with the SID of the user specified. For an example, see Sample SPML Provider request to modify shared mailbox user permissions.
Add user to group	This example illustrates how to add the **John Smith** user account to the **SPMLGroup** group object in the **mycompany.com** domain. To do this, SPML Provider performs the **modify** operation. • The request message includes the following XML elements: • The `<soap:Envelope>` and `<soap:Body>` SOAP elements enclose the SPML payload. • The `<modifyRequest>` element asks SPML Provider to make changes to a specified object. • The `<psoID>` element specifies the distinguished name of the group object to be modified. • The `<modification>` element specifies the type of change as `add`, causing the new values to be appended to the existing attribute values. • The `<data>` element encloses the elements that specify the distinguished name of the user account to be appended to the existing values of the member attribute.
Look up user attributes	This example illustrates how to get the XML representation of the **John Smith** user account in the **mycompany.com** domain. To get the XML representation of an object, SPML Provider performs the **lookup** operation. The request message includes the following XML elements: • The `<soap:Envelope>` and `<soap:Body>` SOAP elements enclose the SPML payload. • The `<lookupRequest>` element asks SPML Provider to return the XML document that represents a specified object.

Operation	Description
	• The `<psoID>` element specifies the distinguished name of the object. The response contains the object identifier, the XML representation of the object and its attributes, and information about SPML Provider capabilities that are supported on the object (the capability-specific data that is associated with the object).
Delete user	This example illustrates how to delete the **John Smith** user account. To do this, SPML Provider performs the **delete** operation. The request message includes the following XML elements: • The `<soap:Envelope>` and `<soap:Body>` SOAP elements enclose the SPML payload. • The `<deleteRequest>` element asks SPML Provider to delete a specified object. • The `<psoID>` element specifies the distinguished name of the user account to delete.
Delete group	This example illustrates how to delete the **SPMLGroup** group object in the **mycompany.com** domain. To do this, SPML Provider performs the **delete** operation. The request message includes the following XML elements: • The `<soap:Envelope>` and `<soap:Body>` SOAP elements enclose the SPML payload. • The `<deleteRequest>` element asks SPML Provider to delete a specified object. • The `<psoID>` element specifies the distinguished name of the group object to delete.

Sample SPML Provider request to modify shared mailbox user permissions

This section provides a sample request that shows how to use Active Roles controls in your SPML requests to modify shared mailbox user permissions.

```
<?xml version="1.0"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
<spml:modifyRequest xmlns:spml="urn:oasis:names:tc:SPML:2:0">
<spml:psoID ID="CN=shmb1,OU=NOV_OU,DC=ars,DC=cork,DC=lab,DC=local"/>
<spml:modification>
```

```
<modification name="edsaUserMailboxSecurityDescriptorSddl" operation="replace"
xmlns="urn:oasis:names:tc:DSML:2:0:core">
<value>O:PSG:PSD:AI(A;CI;RC;;;S-1-5-21-2064067869-2662360268-1970296196-3772)
(A;CI;RC;;;S-1-5-21-2064067869-2662360268-1970296196-3773)
</value>
</modification>
</spml:modification>
</spml:modifyRequest>
</soap:Body>
</soap:Envelope>
```

SPML Provider capability samples

The following tables list all search, password and suspend capability examples included in the Capability samples.

Search Capability samples

Table 103: Search Capability samples

Operation	Description
Perform one-level search	This example illustrates how to obtain a list of the child objects (direct descendants) of the **Active Directory container object**. In proxy mode, you can use this example to list the domains that are registered with Active Roles (managed domains). To do this, SPML Provider performs the **`search`** operation. The request message includes the following XML elements: • The `<soap:Envelope>` and `<soap:Body>` SOAP elements enclose the SPML payload. • The `<searchRequest>` element asks SPML Provider to perform a search and return the identifiers of the objects found. • The `<query>` element determines that SPML Provider is to perform a one-level search (that is, to search only direct descendants of the object specified by `<basePsoID>`). • The `<basePsoID>` element specifies the distinguished name of the container object to search. The response contains the identifiers (distinguished names) of the objects residing in the container object specified by the `<basePsoID>` element.

Operation	Description
Perform subtree search	This example illustrates how to obtain a list of objects that reside below the **Active Directory object** in the directory tree. You can use this example to list the objects that reside in a given domain. To do this, SPML Provider performs the **`search`** operation. The request message includes the following XML elements: • The `<soap:Envelope>` and `<soap:Body>` SOAP elements enclose the SPML payload. • The `<searchRequest>` element asks SPML Provider to perform a search and return the identifiers of the objects found. • The `<query>` element determines that SPML Provider is to perform a subtree search (that is, to search any direct or indirect descendant of the object specified by `<basePsoID>`). • The `<basePsoID>` element specifies the distinguished name of the container object to search. For instance, this could be the distinguished name of a domain that is registered with Active Roles (managed domain). The response contains the identifiers (distinguished names) of the objects that reside in the directory tree below the container object specified by the `<basePsoID>` element.
Perform base search	This example illustrates how to obtain an XML representation of the specific object. To do this, SPML Provider performs the **`search`** operation. The request message includes the following XML elements: • The `<soap:Envelope>` and `<soap:Body>` SOAP elements enclose the SPML payload. • The `<searchRequest>` element asks SPML Provider to perform a search and return the XML representation of the object found. • The `<query>` element determines that SPML Provider is to perform a base search (that is, to search only the object identified by `<basePsoID>`). • The `<basePsoID>` element specifies the distinguished name of the object to search. For instance, this could be the distinguished name of a user account. The response contains the identifier of the object and the XML representation of the object (as defined in the schema of the

Operation	Description
	target).
Iterate search results	This example illustrates how to obtain the next set of objects from the result set that SPML Provider selected for a search operation. In this case, SPML Provider performs the **`iterate`** operation. The request message includes the following XML elements: - The `<soap:Envelope>` and `<soap:Body>` SOAP elements enclose the SPML payload. - The `<iterateRequest>` element asks SPML Provider to return additional objects that matched a previous search request but that the Provider has not yet returned to the client. - The `<iterator>` element supplies the iterator ID found either in the original search response or in a subsequent iterate response.
Stop iterating search results	This example illustrates how to tell SPML Provider that the client has no further need for the search results that a specific iterator represents. In this case, SPML Provider performs the **`closeIterator`** operation. The request message includes the following XML elements: - The `<soap:Envelope>` and `<soap:Body>` SOAP elements enclose the SPML payload. - The `<closeIteratorRequest>` element tells SPML Provider that the client no longer intends to iterate search results. - The `<iterator>` element specifies the ID of the iterator to close. This could be the iterator ID found in the original search response or in a subsequent iterate response.
Find inactive users	This example illustrates how to get a list of inactive (disabled or deprovisioned) user accounts found within a specified container. To do this, SPML Provider performs the **`search`** operation. The request message includes the following XML elements: - The `<soap:Envelope>` and `<soap:Body>` SOAP elements enclose the SPML payload. - The `<searchRequest>` element asks SPML Provider to

Operation	Description
	perform a search and return the identifiers of the objects found. - The `<query>` element determines SPML Provider is to perform a subtree search. - The `<basePsoID>` element specifies the distinguished name of the container object to search. For instance, this could be the distinguished name of a certain organizational unit. - The `<filter>` element encloses the elements that direct SPML Provider to search for inactive user accounts. Thus, the `<equalityMatch>` elements are configured so as to limit the search to user accounts; the `<isActive>` element combined with the `<not>` element causes SPML Provider to select the user accounts that are inactive. - The response contains the identifiers (distinguished names) of the inactive user accounts that exist in the directory tree below the container object specified by the `<basePsoID>` element.
Perform complex search	This example illustrates how to have SPML Provider find all objects that meet certain search criteria and return the values of certain attributes of the objects found. In this case, SPML Provider performs the **`search`** operation. The request message includes the following XML elements: - The `<soap:Envelope>` and `<soap:Body>` SOAP elements enclose the SPML payload. - The `<searchRequest>` element asks SPML Provider to perform a search and return the identifiers and attribute values of the objects found. - The `<query>` element determines the scope of the search. - The `<basePsoID>` element specifies the distinguished name of the container object to search. For instance, this could be the distinguished name of a certain Organizational Unit. - The `<filter>` element encloses the elements that specify the search criteria. - The `<attributes>` element specifies the object attributes to be included in the response. The response contains the identifiers (distinguished names)

Operation	Description
	of the objects found and, for each object, the values of the attributes specified by the `<attributes>` element in the search request.
Find only security groups	This example illustrates how to obtain a list of security groups found in a specified container. In this case, SPML Provider performs the **search** operation. The request message includes the following XML elements: • The `<soap:Envelope>` and `<soap:Body>` SOAP elements enclose the SPML payload. • The `<searchRequest>` element asks SPML Provider to perform a search and return the identifiers of the objects found. • The `<query>` element determines that SPML Provider is to perform a subtree search. • The `<basePsoID>` element specifies the distinguished name of the container object to search. For instance, this could be the distinguished name of a certain organizational unit. • The `<filter>` element encloses the elements that direct SPML Provider to search for security groups. Thus, the `<equalityMatch>` elements are configured so as to limit the search to group objects; the `<extensibleMatch>` element specifies a matching rule that is equivalent to the LDAP filter `(groupType:1.2.840.113556.1.4.803:=2147483648)` where `2147483648` is the decimal equivalent of the `ADS_GROUP_TYPE_SECURITY_ENABLED` flag (`0x80000000`). The response contains the identifiers (distinguished names) of the security groups that exist in the directory tree below the container object specified by the `<basePsoID>` element.

Password Capability samples

Table 104: Password capability samples

Operation	Description
Set user password	This example illustrates how to set a new password for the specific user account. To set a new password, SPML Provider performs the **setPassword** operation. The request message includes the following XML elements:

Operation	Description
	- The <soap:Envelope> and <soap:Body> SOAP elements enclose the SPML payload. - The <setPasswordRequest> element asks SPML Provider to change to a specified value the password that is associated with a certain user account. - The <psoID> element specifies the distinguished name of the user account. - The <password> element specifies the new password to assign to the user account.
Expire user password	This example illustrates how to force a given user to change the password at next logon. To do this, SPML Provider performs the **expirePassword** operation. The request message includes the following XML elements: - The <soap:Envelope> and <soap:Body> SOAP elements enclose the SPML payload. - The <expirePasswordRequest> element asks SPML Provider to mark expired the current password that is associated with a certain user account. The remainingLogins attribute is set to 1 so as to disallow grace logons once the expirePassword operation is completed, forcing the user to change the password at next logon. - The <psoID> element specifies the distinguished name of the user account.

Suspend Capability samples

Table 105: Suspend capability samples

Operation	Description
Suspend user account	This example illustrates how to either disable or deprovision a specified user account, depending on the SPML Provider configuration (see the description of the <suspendAction> element in the “Configuring SPML Provider” section earlier in this document). To do this, SPML Provider performs the **suspend** operation. The request message includes the following XML elements: - The <soap:Envelope> and <soap:Body> SOAP elements enclose the SPML payload.

Operation	Description
	- The <suspendRequest> element asks SPML Provider to perform the suspend action on a certain user account (either disable or deprovision, depending on the configuration of SPML Provider). - The <psoID> element specifies the distinguished name of the user account to suspend.
Resume user account	This example illustrates how to enable a disabled user account. This operation requires that the suspend action be set to disable in the SPML Provider configuration file (see the description of the <suspendAction> element in the "Configuring SPML Provider" section earlier in this document). In this case, SPML Provider performs the **resume** operation in order to enable a disabled user account. The request message includes the following XML elements: - The <soap:Envelope> and <soap:Body> SOAP elements enclose the SPML payload. - The <resumeRequest> element asks SPML Provider to re-enable a user account that has been disabled. - The <psoID> element specifies the distinguished name of the user account to re-enable.
Check whether user is active	This example illustrates how to determine whether a specified user account is active, that is, has not been suspended. A user account is considered to be suspended if the suspend action was performed on that account. The suspend action can be either disable or deprovision, depending on the SPML Provider configuration (see the description of the <suspendAction> element in the "Configuring SPML Provider" section earlier in this document). The request message includes the following XML elements: - The <soap:Envelope> and <soap:Body> SOAP elements enclose the SPML payload. - The <activeRequest> element asks SPML Provider to check whether the suspend action has been performed on a given user account (either disable or deprovision, depending on the SPML Provider configuration). - The <psoID> element specifies the distinguished name of the user account to check. The <activeResponse> element in the response message has the active attribute that indicates whether the specified user

Operation	Description
	account is suspended. If the user account is suspended, the active attribute is set to `false`. Otherwise, the active attribute is set to `true`.

Active Roles SPML Provider terminology

Direct Access Mode

In this mode, SPML Provider directly connects to the specified domain or AD LDS instance.

Capabilities

A set of optional, functionally related operations defined in SPML v2.

Core Operations

The minimum set of operations that a provider must implement to conform to the official SPML v2 specification.

Extensible Markup Language (XML)

A meta-markup language that provides a format for describing structured data. This facilitates more precise declarations of content and more meaningful search results across multiple platforms. In addition, XML enables a new generation of Web-based data viewing and manipulation applications.

Organization for the Advancement of Structured Information Standards (OASIS)

An international consortium that drives the development, convergence, and adoption of e-business and Web service standards.

Provider

See Provisioning Service Provider.

Provisioning Service Object (PSO)

Represents a data entity or an information object on a target.

Provisioning Service Provider (PSP)

A software component that listens for, processes, and returns the results for well-formed SPML requests from a known requestor.

Provisioning Service Target (PST)

Represents a destination or endpoint that a provider makes available for provisioning actions.

Proxy Mode

In proxy mode, SPML Provider accesses directory data using the Active Roles proxy service.

Requesting Authority (RA)

A software component that issues well-formed SPML requests to a Provisioning Service Provider.

Requestor

See Requesting Authority.

Simple Object Access Protocol (SOAP)

An XML/HTTP-based protocol for platform-independent access to objects and services on the Web. SOAP defines a message format in XML that travels over the Internet using HyperText Transfer Protocol (HTTP). By using existing Web protocols (HTTP) and languages (XML), SOAP runs over the existing Internet infrastructure without being tied to any operating system, language, or object model.

SPML

An XML-based framework for exchanging user, resource, and service provisioning information between cooperating organizations.

SPML v2

An OASIS standard that provides a means of representing provisioning requests and responses as SPML documents.

Target

See Provisioning Service Target.

Target Schema

Defines the XML structure of the objects (PSO) that the target may contain.

Troubleshooting SPML Provider

This section briefly discusses some error statements that you may encounter when using SPML Provider.

Cannot remove the specified item because it was not found in the specified Collection

When sending an SPML request to remove a user from a group, the requested operation fails with the following error:

```
Cannot remove the specified item because it was not found in the specified
Collection.
```

Solution

This error has one of the following causes:

- The `<value>` element of the `<attr>` element specifies a user account that is not a member of the group.
- The Distinguished Name fields, such as CN or OU, used in the distinguished name of the user account to be removed, have invalid spelling or case. The Distinguished Name fields must be in upper case. For example, using `cn=Robert Smith` instead of `CN=Robert Smith` can result in this error.

Verify that the `<value>` element specifies the distinguished name of the user that is the group member. Make sure that the **Distinguished Name** fields are in upper case.

The following example illustrates how to create a request to remove user **Robert Smith** from the **Sales** group.

```
<?xml version="1.0"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
<modifyRequest xmlns="urn:oasis:names:tc:SPML:2:0" returnData="everything">
<psoID ID="CN=Sales,OU=SPML2,DC=Mycompany,DC=com"/>
<modification modificationMode="delete">
<data>
<attr name="member" xmlns="urn:oasis:names:tc:DSML:2:0:core">
<value>CN=Robert Smith,OU=Staff,DC=MyCompany,DC=com</value>
</attr>
```

```
</data>
</modification>
</modifyRequest>
</soap:Body>
</soap:Envelope>
```

Some of the specified attributes for the object class are not defined in the schema

When sending a request to change the values of the virtual attributes of an object, the requested operation fails with the following error:

```
Some of the specified attributes for the <object-class-name> object class are
not defined in the schema.
```

Solution

This error has one of the following causes:

- The `spmlschema.config` configuration file has changed since you started SPML Provider.
- The Default Application Pool idle timeout period has ended.

To resolve this issue, recycle the Default Application Pool or change its settings using Internet Information Services (IIS) Manager.

23

Monitoring Active Roles with Management Pack for SCOM

The Active Roles Management Pack for Microsoft System Center Operations Manager (SCOM) provides a basic solution for monitoring the availability and health of:

- The Active Roles Administration Service and its information store.
- The Active Roles replication status.
- The availability of the Active Roles Web Interface.

By detecting, alerting on, and automatically responding to critical events, Management Pack for SCOM helps indicate, correct, and in many cases, prevent outages of the Administration Service and the Web Interface.

Management Pack for SCOM features

Management Pack for SCOM provides several features that you can use to monitor your Active Roles environment, including:

- Automated discovery.
- Availability and performance monitoring.
- Replication monitoring.

Using these features, Management Pack for SCOM can alert you to the following error conditions:

- Administration Service is not responding.
- Active Roles replication failure has occurred.
- Connection to the Active Roles database has been lost.
- Administration Service failed to update a Dynamic Group.
- Administration Service failed to update a Group Family.
- Active Roles Web Interface is unavailable.

Management Pack for SCOM monitoring views

Management Pack for SCOM has monitoring views used to centrally monitor the availability and health of the various Active Roles components. The monitoring views are available in the **Monitoring** pane of the **Operation Manager** console, and are as follows:

- **Alerts**: Displays the alerts on the computers running the Active Roles Administration Service or Web Interface.
- **Computers**: Shows information about the state of the computers running the Active Roles Administration Service or Web Interface.
- **Events**: Displays the events on the computers running the Active Roles Administration Service or Web Interface.
- **Performance**: Shows performance information collected from the computers running the Active Roles Administration Service or Web Interface.
- **Service Level Exceptions**: Displays the unresolved alerts that have exceeded service levels on the computers running the Active Roles Administration Service or Web Interface.
- **Task Status**: Shows information indicating task results on the computers running the Active Roles Administration Service or Web Interface.
- **Discovery**: Contains separate views allowing you to examine the state of the computers running the Web Interface, the Administration Service, or both.
- **Services Monitoring**: Contains the views allowing you to monitor availability, health and performance of the Administration Service instances in your environment.
- **Web Interfaces Monitoring**: Contains the views allowing you to monitor availability and health of the Web Interface instances in your environment.

Getting started with Management Pack for SCOM

Install Management Pack for SCOM by importing the `ActiveRoles.SCOM.MP.xml` file into System Center Operations Manager (SCOM). You can install this Management Pack on the following SCOM versions:

- System Center Operations Manager 2016
- System Center Operations Manager 2019

Monitoring Active Roles Administration Service with Management Pack for SCOM

This section describes the processing rules that Management Pack for SCOM uses to monitor the availability and health of the Active Roles Administration Service. These include the following:

- General response
- Replication monitoring
- Monitoring of connection to configuration database
- Monitoring of Dynamic Group-related operations
- Monitoring of Group Family-related operations
- Internal error
- Critical error on startup
- License system failure

Monitoring of general response and replication is performed via custom, script-based processing rules. Those rules run on a scheduled basis, analyzing information returned by the scripts and raising an appropriate event if an error is detected. The schedule is stored as part of rule configuration data, and can be adjusted by managing rule properties in the Operations Manager console.

General response - Script

This rule uses a script to check the responsiveness of the Administration Service by periodically issuing a simple request to the Service. By default, this rule is scheduled to run every 10 minutes. The schedule can be adjusted by managing rule properties in the Operations Manager console.

General response - Alert

This rule generates an alert when the **General response** script detects that the Administration Service is unavailable.

Possible causes of the alert include:

- The Administration Service is not running.
- The Administration Service is not configured properly.

- The Administration Service has encountered a critical error.
- The administration database is unavailable.

Replication monitoring - Script

This rule uses a script to check the status of Active Roles replication. The script is intended to run on the Publisher Administration Service so as to verify the replication status of the Publisher and Subscribers. By default, this rule is scheduled to run every 30 minutes. The schedule can be adjusted by managing rule properties in the Operations Manager console.

Replication monitoring - Alert

This rule generates an alert when the **Replication monitoring** script detects that the Active Roles replication status indicates a replication failure.

Possible causes of the alert include:

- The SQL Server Agent service is not started on the computer running the Publisher SQL Server.
- The Snapshot Agent or a Merge Agent is not started at the Publisher SQL Server.
- The Merge Agent uses incorrect credentials when connecting to the Publisher or a Subscriber.
- The Snapshot Agent uses incorrect credentials when connecting to the Publisher.

For more information, and details on how resolve replication-related problems, see Identifying replication-related problems.

Monitoring connection to configuration database

This category includes the event-based processing rules to monitor health of the connection to the configuration database:

- **Connection to database has been lost**: Administration Service has lost connection to the configuration database, and is attempting to re-establish the connection.
- **Connection to database has been restored**: Administration Service restored connection to the configuration database.

The following sub-sections elaborate on each of these processing rules.

Connection to database has been lost - Alert

This rule generates an alert indicating that the Administration Service has lost a connection to the configuration database, and is making attempts to restore the connection. For details, refer to the alert description generated by this rule. Losing the connection to the database does not affect the directory management functions of the Administration Service. All operations related to Active Directory management continue to work as expected.

As long as there is no connection to the database, the following Administration Service functions will not be available:

- Collecting data related to change history and user activity.
- Retrieving and updating configuration data.
- Retrieving changes to configuration data made by other Administration Service instances (both directly and via replication).
- Retrieving and updating virtual attributes stored in the configuration database.

Connection to database has been restored - Alert

This rule generates an alert indicating that the Administration Service has restored the connection to the configuration database. For details, refer to the alert description generated by this rule. Once the connection has been restored, all Administration Service functions that require access to the database will be restored.

Monitoring of Dynamic Group-related operations

This category includes the event-based processing rules to monitor the background activities of Active Roles related to Dynamic Groups:

- **Rebuilding has been started**: Administration Service has been forced to re-calculate (rebuild) the membership list of a Dynamic Group.
- **Failed to add object to Dynamic Group**: Administration Service failed to add an object to a Dynamic Group.
- **Failed to remove object from Dynamic Group**: Administration Service failed to remove an object from a Dynamic Group.
- **Failed to process membership rule**: Administration Service failed to apply a query-based membership rule when updating the membership list of a Dynamic Group.
- **Failed to update membership list**: Administration Service failed to update the membership list of a Dynamic Group in accordance with the membership rules.
- **Failed to update membership list of nested group**: Administration Service failed to update the membership list of an additional (nested) group generated to

accommodate extra members of a Dynamic Group.

- **Failed to update membership rule upon deletion of object**: When updating a Dynamic Group, Administration Service failed to delete or update a membership rule of a Dynamic Group upon deletion of an object.
- **Failed to look up object when updating**: When updating a Dynamic Group, Administration Service failed to locate an object that is referred to by a certain membership rule. The object may have been deleted.
- **Failed to retrieve information from domain**: Administration Service failed to retrieve information about Dynamic Groups from a certain domain.
- **Membership rule domain unavailable**: When updating a Dynamic Group, Administration Service failed to apply a membership rule because the rule applies to a domain unavailable on the network.
- **Membership rule failed**: When updating a Dynamic Group, Administration Service failed to apply one of the membership rules, which prevented all rules from being applied and stopped changes to the members list of the Dynamic Group.

The following sub-sections provide more details about these processing rules.

Dynamic Group - Rebuilding has been started - Alert

This rule generates an alert indicating that an administrator has forced Active Roles to re-calculate (rebuild) the membership list of a Dynamic Group. For details, refer to the alert description generated by this rule.

You can start rebuilding the Dynamic Group from the **Properties** > **Members** tab of the Dynamic Group, in the Active Roles Console.

Failed to add object to Dynamic Group - Alert

This rule generates an alert indicating that the Administration Service failed to add an object to a Dynamic Group due to a certain problem. The object is missing from the Dynamic Group until after the problem has been resolved. For details, refer to the alert description generated by this rule.

To solve the problem, try to force rebuilding the Dynamic Group from the **Properties** > **Members** tab of the Dynamic Group, in the Active Roles Console.

Failed to remove object from Dynamic Group - Alert

This rule generates an alert indicating that the Administration Service failed to remove an object from a Dynamic Group due to a certain problem. The object remains in the Dynamic Group until after the problem has been resolved. For details, refer to the alert description generated by this rule.

To solve the problem, try to force rebuilding the Dynamic Group from the **Properties** > **Members** tab of the Dynamic Group, in the Active Roles Console.

Dynamic Group - Failed to process membership rule - Alert

This rule generates an alert indicating that the Administration Service failed to apply a query-based membership rule when updating the membership list of a Dynamic Group. The failed rule is not taken into account, so the membership list may not comply with the membership rules. For details, refer to the alert description generated by this rule.

To solve the problem, try to force rebuilding the Dynamic Group from the **Properties** > **Members** tab of the Dynamic Group, in the Active Roles Console. Check membership rules by using the **Membership Rules** tab in that dialog.

Dynamic Group - Failed to update membership list - Alert

This rule generates an alert indicating that the Administration Service failed to update the membership list of a Dynamic Group in accordance with the membership rules. The membership list may not be compliant with the membership rules. For details, refer to the alert description generated by this rule.

To solve the problem, try to force rebuilding the Dynamic Group from the **Properties** > **Members** tab of the Dynamic Group, in the Active Roles Console.

Dynamic Group - Failed to update membership list of nested group - Alert

This rule generates an alert indicating that the Administration Service failed to update the membership list of an additional (nested) group generated to accommodate extra members of a Dynamic Group. The membership list of the nested group may not be compliant with the membership rules. For details, refer to the alert description generated by this rule.

To solve the problem, try to force rebuilding the Dynamic Group from the **Properties** > **Members** tab of the Dynamic Group, in the Active Roles Console.

Dynamic Group - Failed to update membership rule upon deletion of object - Alert

This rule generates an alert indicating that the Administration Service failed to delete or update a membership rule of a Dynamic Group when deleting a certain object. The membership rule could be one of the following:

- Implicit inclusion or exclusion of that object from the Dynamic Group.
- Query with a filter referring to that object.
- Inclusion or exclusion of the members of the group represented by that object.

For details, refer to the alert description generated by this rule.

To resolve the issue, delete or update membership rules with the **Properties** > **Membership Rules** tab of the Dynamic Group in the Active Roles Console. Then, force rebuilding of the Dynamic Group from the **Members** tab in that dialog.

Dynamic Group - Failed to look up object when updating - Alert

This rule generates an alert indicating that the Administration Service failed to locate an object when updating the membership list of a Dynamic Group in accordance with the membership rules. The object may have been deleted. The object could be referred to by:

- A membership rule to explicitly include or exclude that object from the Dynamic Group.
- A query-based membership rule (the object may represent the base of a search or be a member of the search result set).
- A membership rule to include or exclude the members of a certain group (the object may represent the domain of that group).
- A directory synchronization (DirSync) query (this may be one of the objects returned by that query).

For details, refer to the alert description generated by this rule.

The membership rules referring to that object are inoperative and are not taken into account when updating the Dynamic Group, so the membership list may not be compliant with the membership rules.

To prevent issues with the membership list of the Dynamic Group, check membership rules by using the **Properties** > **Membership Rules** tab of the Dynamic Group in the Active Roles Console. Then, force rebuilding of the Dynamic Group from the **Members** tab in that dialog.

Dynamic Group - Failed to retrieve information from domain - Alert

This rule generates an alert indicating that the Administration Service failed to retrieve information about Dynamic Groups from a certain domain. The Dynamic Groups contained in that domain are inoperative until after the problem has been resolved. For details, refer to the alert description generated by this rule.

Dynamic Group - Membership rule domain unavailable - Alert

This rule generates an alert indicating that Active Roles failed to update the members list of the Dynamic Group in accordance with one of the membership rules. The failed membership rule applies to a domain that is currently unavailable. The membership rule is disregarded, so the members list of the Dynamic Group may not be compliant with the membership rules. For details, refer to the alert description generated by this rule.

To solve the problem, ensure that the domain is available on the network, then update the Dynamic Group by clicking **Properties** > **Members** > **Rebuild** in the dialog of the group in the Active Roles Console. Alternatively, wait for Active Roles to update the Dynamic Group on a schedule.

Dynamic Group - Membership rule failed - Alert

This rule generates an alert indicating that Active Roles failed to update the members list of the Dynamic Group in accordance with one of the membership rules. As one of the membership rules failed, no membership rules are applied until the issue is resolved, so the members list of this Dynamic Group remains unchanged. For details, refer to the alert description generated by this rule.

To solve the problem, try to force update the Dynamic Group by clicking **Properties** > **Members** > **Rebuild** in the dialog of the group in the Active Roles Console. Check the membership rules on the **Membership Rules** tab in that dialog.

Monitoring of Group Family-related operations

This category includes the event-based processing rules to monitor the background activities of Active Roles related to Group Families:

- **Cannot find configuration storage group**: Administration Service failed to run a Group Family due to the following problem:

  The Group Family configuration storage group cannot be found.
- **Failed to retrieve configuration data**: Administration Service failed to run a Group Family due to the following problem:

  Group Family configuration data cannot be retrieved from the Group Family configuration storage group.
- **Incorrect configuration data**: Administration Service failed to run a Group Family due to the following problem:

  Incorrect configuration data was encountered in the Group Family configuration storage group.
- **Failed to retrieve configuration data for controlled group**: Administration Service encountered an error when running a Group Family, failed to retrieve configuration data for a controlled group. Changes to the controlled group may not be applied until a subsequent run of the Group Family.
- **Failed to retrieve data from container**: Administration Service encountered an error when running a Group Family, failed to search a certain container within the Group Family scope. Until a subsequent run, Group Family does not consider information about objects held in that container.
- **Failed to update configuration data**: Administration Service encountered an error when running a Group Family, failed to update data in the Group Family configuration storage group. Information about controlled groups may be incorrect until a subsequent run of the Group Family.
- **Failed to update configuration data for controlled group**: Administration Service encountered an error when running a Group Family, failed to update configuration data for a controlled group. The controlled group is not linked with the Group Family until a subsequent run of the Group Family.
- **Cannot find controlled group**: Administration Service encountered an error when running a Group Family, failed to find a controlled group. Changes to the controlled group, if any, are not applied until a subsequent run of the Group Family.

- **Failed to create controlled group**: Administration Service encountered an error when running a Group Family, failed to create a controlled group. Administration Service will attempt to create that controlled group during a subsequent run of the Group Family.
- **Failed to update membership list of controlled group**: Administration Service encountered an error when running a Group Family, failed to update membership data for a controlled group. The membership list of the controlled group may be incorrect until a subsequent run of the Group Family.
- **Failed to create run task**: Administration Service failed to create a task to run a Group Family. The Group Family is inoperative until the task is created.
- **Failed to modify run task**: Administration Service failed to update a task to run a Group Family. The Group Family runs in accordance with the earlier schedule settings of that task.
- **Failed to delete run task**: Administration Service failed to delete a task to run a Group Family. The Group Family continues to run in accordance with the schedule settings of that task.
- **Run task has been started manually**: A task to run a Group Family was started manually.
- **Group Family run has been completed**: Administration Service has completed a run of a Group Family.

The following sub-sections provide more information about these alerts.

Group Family - Cannot find configuration storage group - Alert

This rule generates an alert indicating that the Administration Service failed to run a Group Family due to the following problem:

The Group Family configuration storage group cannot be found. The Administration Service cannot run the Group Family until the problem is resolved.

The configuration storage group may have been either inaccessible or deleted. For details, refer to the alert description generated by this rule.

Group Family - Failed to retrieve configuration data - Alert

This rule generates an alert indicating that the Administration Service encountered the following problem when running a Group Family:

Group Family configuration data cannot be retrieved from the configuration storage group. The Administration Service cannot run the Group Family until the problem is resolved.

For details, refer to the alert description generated by this rule.

Group Family - Incorrect configuration data - Alert

This rule generates an alert indicating that the Administration Service failed to run a Group Family due to the following problem:

Incorrect configuration data was encountered in the Group Family configuration storage group. The configuration storage group may have been corrupted. The run of the Group Family has been canceled.

For details, refer to the alert description generated by this rule.

Group Family - Failed to retrieve configuration data for controlled group - Alert

This rule generates an alert indicating that the Administration Service encountered the following problem when running a Group Family:

Failed to retrieve configuration data for a certain group that is under the control of the Group Family (controlled group). Changes to the controlled group may not be applied until a subsequent run of the Group Family.

For details, refer to the alert description generated by this rule.

Group Family - Failed to retrieve data from container - Alert

This rule generates an alert indicating that the Administration Service encountered the following problem when running a Group Family:

Failed to search a certain container within the Group Family scope. The groupings that were calculated during this run of the Group Family may not take into account information about some objects held in that container.

For details, refer to the alert description generated by this rule.

During a subsequent run of the Group Family, the Administration Service will attempt to search the entire Group Family scope, including the failed container to recalculate the Group Family groupings.

Group Family - Failed to update configuration data - Alert

This rule generates an alert indicating that the Administration Service encountered the following problem when running a Group Family:

Failed to update configuration data in the Group Family configuration storage group. The Active Roles Console may display incorrect information about results of the Group Family run and about groups that are under the control of the Group Family (controlled groups).

For details, refer to the alert description generated by this rule.

During a subsequent run of the Group Family, the Administration Service will attempt to update configuration data in the configuration storage group.

Group Family - Failed to update configuration data for controlled group - Alert

This rule generates an alert indicating that the Administration Service encountered the following problem when running a Group Family:

Failed to update configuration data for a certain group that is under the control of the Group Family (controlled group). The group is removed from the control of the Group Family.

For details, refer to the alert description generated by this rule.

During a subsequent run of the Group Family, the Administration Service will attempt to locate the failed group and put it under the control of the Group Family.

Group Family - Cannot find controlled group - Alert

This rule generates an alert indicating that the Administration Service encountered the following problem when running a Group Family:

Cannot find a certain group that is under the control of the Group Family (controlled group). Some changes to the controlled group may not be applied.

For details, refer to the alert description generated by this rule.

During a subsequent run of the Group Family, the Administration Service will attempt to locate the controlled group and apply the changes, if any, to that group.

Group Family - Failed to create controlled group - Alert

This rule generates an alert indicating that the Administration Service encountered the following problem when running a Group Family:

Failed to create a certain group to be put under the control of the Group Family (controlled group).

For details, refer to the alert description generated by this rule.

During a subsequent run of the Group Family, the Administration Service will attempt to create the controlled group and apply the changes, if any, to that group.

Group Family - Failed to update membership list of controlled group - Alert

This rule generates an alert indicating that the Administration Service encountered the following problem when running a Group Family:

Failed to update the membership list of a certain group that is under the control of the Group Family (controlled group). Some changes to the membership list of the controlled group may not be applied.

For details, refer to the alert description generated by this rule.

During a subsequent run of the Group Family, the Administration Service will attempt to locate the controlled group and apply the changes, if any, to the membership list of that group.

Group Family - Failed to create run task - Alert

This rule generates an alert indicating that the Administration Service failed to create a task to run a Group Family. The Group Family is inoperative until the task is created.

For details, refer to the alert description generated by this rule.

Group Family - Failed to modify run task - Alert

This rule generates an alert indicating that the Administration Service failed to update a task to run a Group Family. The Group Family continues to run in accordance with the earlier schedule settings of that task.

For details, refer to the alert description generated by this rule.

To solve the problem, try to adjust the schedule settings via the **Properties** > **Schedule** tab of the Group Family configuration storage group in the Active Roles Console.

Group Family - Failed to delete run task - Alert

This rule generates an alert indicating that the Administration Service failed to delete a task to run a Group Family while the configuration storage group of that Group Family was successfully deleted. The Group Family continues to run in accordance with the schedule settings of that task, which may cause an error situation. For details, refer to the alert description generated by this rule.

To solve the problem, delete the run task manually by switching the Active Roles Console into Raw view mode, then deleting the appropriate task from the following container:

Configuration/Server Configuration/Scheduled Tasks/Group Family

Group Family - Run task has been started manually - Alert

This rule generates an alert indicating that an administrator has forced Active Roles to run a Group Family. For details, refer to the alert description generated by this rule.

To solve the problem, start the run task for a Group Family by using the **Force Run** command on the configuration storage group of that Group Family in the Active Roles Console.

Group Family run has been completed - Alert

This rule generates an alert indicating that the Administration Service has completed the run task for a Group Family. For task results, refer to the alert description generated by this rule.

The alert description also includes the name of the Group Family configuration storage group, so you can use the **Properties** dialog box for that group to examine task results in more detail.

Internal error - Alert

This rule generates an alert when a fatal error occurs at Administration Service run time. Normally, the alert indicates that Administration Service stopped.

Critical error on startup - Alert

This rule generates an alert when a fatal error occurs at Administration Service startup time. The alert includes information detailing the error.

License system failure - Alert

This rule generates an alert when a failure occurs in the Administration Service licensing system. The alert includes information detailing the problem.

Monitoring Active Roles Web Interface

This section describes the processing rules that you can use to monitor the availability of the Active Roles Web Interface.

Availability - Script

This rule uses a script to check the availability of the Active Roles Web Interface. The script invokes a self-diagnostic script built into the Web Interface to verify the Web Interface configuration, including the customization settings, and to check whether the Administration Service is available.

The rule ensures that both the default and customized Web Interface sites are monitored properly if customization is performed by using the point-and-click tools included in the Web Interface.

> NOTE: This rule cannot check the availability of custom Web Interface functions based on custom ASP files. To monitor such functions, implement custom rules to Operations Manager.

By default, this rule is scheduled to run every 30 minutes. You can adjust the default schedule in the Operations Manager console.

Availability - Alert

This rule generates an alert when the **Availability** script detects that the Active Roles Web Interface is not available. The possible causes of the alert can include:

- The Web Interface is not running.
- The Web Interface is not configured properly.

- The Administration Service is unavailable.

Monitoring performance with Management Pack for SCOM

This section describes the processing rules of Management Pack for SCOM based on performance counters that allow you to evaluate the performance of the Administration Service.

AD changes processed/sec

This rule collects **AD changes processed/sec** counter samples for the **AR Server:External Changes** performance object. A sample of the counter is the number of changes received from Active Directory and processed by the Active Roles Administration Service per second.

Changes queue length (AD + Database)

This rule collects **Changes queue length (AD + Database)** counter samples for the **AR Server:External Changes** performance object. A sample of the counter is the number of unprocessed changes that the Active Roles Administration Service received from Active Directory and from the Active Roles database.

Connected clients

This rule collects **Connected clients** counter samples for the **AR Server:Miscellaneous** performance object. A sample of the counter is the current number of the clients connected to the Active Roles Administration Service.

Database changes processed/sec

This rule collects **Database changes processed/sec** counter samples for the **AR Server:External Changes** performance object. A sample of the counter is the number of changes received from the Active Roles database and processed by the Active Roles Administration Service per second.

LDAP operations in progress

This rule collects **LDAP operations in progress** counter samples for the **AR Server:LDAP Operations** performance object. A sample of the counter is the current number of the LDAP operation requests that are in progress on the Active Roles Administration Service.

LDAP operations/sec

This rule collects **LDAP operations/sec** counter samples for the **AR Server:LDAP Operations** performance object. A sample of the counter is the number of LDAP operations initiated by the Active Roles Administration Service per second.

Private bytes

This rule collects **Private Bytes** counter samples for the **Process** performance object specific to the Active Roles Service (arssvc) process. A sample of the counter is the amount of virtual memory (in bytes) that the Active Roles Administration Service process allocates (process private bytes).

Queued post-processing policies

This rule collects **Queued post-processing policies** counter samples for the **AR Server:Miscellaneous** performance object. A sample of the counter is the number of the post-processing policy operations queued by the Active Roles Administration Service.

Requests in progress

This rule collects **Requests in progress** counter samples for the **AR Server:Requests** performance object. A sample of the counter is the current number of the client requests being processed by the Active Roles Administration Service.

Requests/sec

This rule collects **Requests/sec** counter samples for the **AR Server:Requests** performance object. A sample of the counter is the number of requests received by the Active Roles Administration Service per second.

Script module average execution time

This rule collects **Script module average execution time** counter samples for the **AR Server:Script Modules** performance object. A sample of the counter is the average running time of all script module instances run by the Active Roles Administration Service.

Script modules executing

This rule collects **Script modules executing** counter samples for the **AR Server:Script Modules** performance object. A sample of the counter is the current number of the script module instances being run by the Active Roles Administration Service.

Configuring Active Roles for AWS Managed Microsoft AD

> NOTE: This feature is officially supported starting from Active Roles 8.1.3 SP1 (build 8.1.3.10). It is not supported on Active Roles 8.1.3 (build 8.1.3.2) and earlier versions.

Active Roles supports deployment and configuration in the Amazon cloud to manage AWS Managed Microsoft AD instances hosted via AWS Directory Service.

This allows you to:

- Perform Active Directory management tasks in your AWS Managed Microsoft AD environment.
- Synchronize directory data from an on-premises AD environment to AWS Managed Microsoft AD.
- Synchronize passwords from an on-premises Active Directory to AWS Managed Microsoft AD (with certain limitations).

For more information about the Active Roles features supported with AWS Managed Microsoft AD, see *Support for AWS Managed Microsoft AD* in the *Active Roles Feature Guide*.

Supported AWS Managed Microsoft AD deployment configuration

To manage AWS Managed Microsoft AD environments, you must deploy Active Roles in Amazon Web Services (AWS) in the following configuration:

- Active Roles must be deployed on an Amazon Elastic Compute Cloud (EC2) instance or instances. For more information, see the *Amazon Elastic Compute Cloud documentation*.
- The SQL Server required by Active Roles Administration Service must run on a separate Amazon Relational Database Service for Microsoft SQL Server (RDS for SQL Server) instance. For more information, see the *Amazon RDS documentation*.
- The Active Directory environment must be hosted in AWS via AWS Directory Service. For more information, see the *AWS Directory Service documentation*.

> NOTE: Support for AWS Managed Microsoft AD by Active Roles was tested only in this configuration. Active Roles does not officially support managing AWS Managed Microsoft AD environments in a hybrid deployment, that is, using an on-premises Active Roles and/or SQL Server installation and hosting AD via AWS Directory Service.

Deployment requirements for AWS Managed Microsoft AD support

Before starting the deployment and configuration of Active Roles to manage AWS Managed Microsoft AD via AWS Directory Service, make sure that the following requirements are met.

> NOTE: When setting up a virtual environment, carefully consider the configuration aspects such as CPU, memory availability, I/O subsystem, and network infrastructure to ensure the virtual layer has the necessary resources available. Please consult One Identity's Product Support Policies for more information on environment virtualization.

Connectivity requirements

You must have:

- Stable network connectivity to Amazon Web Services (AWS).
- Port `1433` open and available for the Amazon Relational Database Service (RDS) service.
- Access to the AWS service with the **AWSAdministratorAccess** permission.

  > NOTE: Make sure that you have **AWSAdministratorAccess** permission, as it is required for certain configuration steps. The **AWSPowerUserAccess** permission is not sufficient for completing the entire configuration procedure.

Infrastructure requirements

To deploy and configure Active Roles for AWS Managed Microsoft AD, you must have access to the following AWS services and resources:

- AWS Managed Microsoft AD deployed via AWS Directory Service.
- One or more Amazon Elastic Compute Cloud (EC2) instance(s) hosting the Active Roles services and components.

  The EC2 instance(s) must have, at minimum:

  - 2 vCPUs running at 2.0 GHz.
  - 4 GB of RAM.

  > TIP: One Identity recommends hosting the main Active Roles services and components (the Active Roles Service and Console, and the Active Roles Web Interface) on separate EC2 instances. If you deploy all Active Roles services and components in a single EC2 instance, use a more powerful instance to ensure a better user experience for the product.

  > NOTE: AWS Managed Microsoft AD support was tested with a single **t2.large** EC2 instance.

- An Amazon Relational Database Service for SQL Server (RDS for SQL Server).

> NOTE: AWS Managed Microsoft AD support was tested with an RDS instance running the latest version of Microsoft SQL Server.

Make sure that all these components are discoverable or visible to each other.

Main steps of configuring Active Roles for AWS Managed Microsoft AD

If your organization and environment meet the Deployment requirements for AWS Managed Microsoft AD support, configuring Active Roles for managing AWS Managed Microsoft AD via AWS Directory Service has the following main steps:

1. Creating your AWS Managed Microsoft AD environment.
2. Creating an Amazon Elastic Compute Cloud (EC2) instance for Active Roles.
3. Joining the EC2 instance to AWS Managed Microsoft AD.
4. Creating an Amazon Relational Database Service for SQL Server (RDS for SQL Server) instance to host the Active Roles Management History and Configuration databases.
5. Verifying the connectivity between the EC2 and RDS instances.
6. Installing and configuring Active Roles on the EC2 instance.
7. (Optional) Installing and configuring Active Roles Synchronization Service on the EC2 instance. For more information, see *Installing and configuring Synchronization Service to manage AWS Managed Microsoft AD resources* in the *Active Roles Synchronization Service Administration Guide*.

Creating the AWS Managed Microsoft AD instance

To deploy and configure Active Roles in Amazon Web Services (AWS) for managing AWS Managed Microsoft AD, first you must create an AWS Directory Service instance hosting your AWS Managed Microsoft AD instance in the AWS console. For more information on configuring the service in the AWS console, see Setting up AWS Directory Service in the *AWS Directory Service documentation*.

> NOTE: Consider the following when creating the AWS Managed Microsoft AD instance:
>
> - Make sure that the connectivity requirements listed in Deployment requirements for AWS Managed Microsoft AD support are met.
> - During the procedure, take note of the following values, as they will be required in later procedures:

    - **Directory DNS name**: The fully qualified domain name (FQDN) of your AD service (for example, `activeroles.demo`).
    - **Directory NetBIOS name**: The NetBIOS name (or shortname) of your AD service (for example, `ARDEMO`).
    - **Admin password**: The password of the default admin account (named `admin`).
- After specifying all required settings, it takes approximately 30-40 minutes to create the AWS Managed Microsoft AD instance. If you run into any issues when creating the environment, see Troubleshooting AWS Managed Microsoft AD in the *AWS Managed Microsoft AD documentation*.

Creating the EC2 instance for Active Roles

To deploy and configure Active Roles in Amazon Web Services (AWS) for managing AWS Managed Microsoft AD, you must create an Amazon Elastic Compute Cloud (EC2) instance hosting your Active Roles installation.

Complete the procedure in AWS as described in Set up to use Amazon EC2 in the *Amazon EC2 documentation*. If you run into any problems when configuring or connecting to the EC2 instance, see Troubleshoot EC2 Windows instances in the *Amazon EC2 documentation*.

NOTE: Consider the following when creating the EC2 instance:

- Make sure that the connectivity requirements listed in Deployment requirements for AWS Managed Microsoft AD support are met.
- For the operating system on the EC2 instance, select a **Microsoft Windows Server** AMI supported by Active Roles. For the list of supported Windows Server operating systems, see *System requirements* in the *Active Roles Release Notes*.
- Select an EC2 instance type that has, at minimum:
    - 2 vCPUs running at 2.0 GHz.
    - 4 GB of RAM.
- One Identity recommends setting the storage to a minimum of 60 GiB of gp2 root volume.

TIP: For consistency, after you logged in to the EC2 instance, rename the virtual machine to the same name that you originally defined for the EC2 instance in the AWS console.

Joining the EC2 instance to AWS Managed Microsoft AD

After you created your AWS Managed Microsoft AD service and your EC2 instance(s), you must join the configured Amazon Elastic Compute Cloud (EC2) instance(s) to AWS Managed Microsoft AD.

Complete the procedure in Amazon Web Services (AWS) as described in Join an EC2 instance to your AWS Managed Microsoft AD directory in the *AWS Directory Service documentation*.

> NOTE: Consider the following when joining the EC2 instance(s) to AWS Managed Microsoft AD:
>
> - Make sure that the connectivity requirements listed in Deployment requirements for AWS Managed Microsoft AD support are met.
> - You need to use the fully qualified domain name that your configured during Creating the AWS Managed Microsoft AD instance.

> TIP: If the domain join process ends with an error, check the specified DNS addresses and Domain Admin credentials in the AWS console.

Creating the RDS instance for the Active Roles SQL Server

If you manage AWS Managed Microsoft AD with Active Roles in Amazon Web Services (AWS), you must store the Active Roles Management History and Configuration databases in an Amazon Relational Database Service (RDS) instance.

Configure the RDS instance in AWS as described in Setting up for Amazon RDS in the *Amazon RDS documentation*.

> NOTE: Consider the following when creating the EC2 instance:
>
> - Make sure that the connectivity requirements listed in Deployment requirements for AWS Managed Microsoft AD support are met.
> - Select the SQL Server edition that suits your needs the most. For most Active Roles use cases, **SQL Server Standard Edition** is an optimal choice.
> - Take note of the **Master username** and **Master password**, as these credentials will be required later.
> - For **Storage type**, select **General Purpose SSD (gp2)**, and allocate a minimum storage of 60 GiB.
> - Consider selecting **Enable storage autoscaling**. Selecting this setting is useful if the SQL Server is utilized with a heavy load most of the time, but it may incur additional operational costs.

Verifying connectivity between the EC2 and RDS instances

After you created the RDS instance, you can test in the EC2 instance with the telnet client or Microsoft SQL Server Management Studio (SSMS) if the RDS connectivity was successfully configured.

To verify RDS connectivity in the EC2 instance

1. Log in to the EC2 instance created for Active Roles.
2. To test connectivity to RDS, install the telnet client. To do so:
    a. Open Windows Server Manager.
    b. On the **Dashboard**, click **Add roles and features**.
    c. In **Installation Type**, select **Role-based or feature-based installation**, then click **Next**.
    d. In **Server Selection**, choose **Select a server from the server pool**, and make sure that the local server (the EC2 instance) is selected.
    e. In **Server Roles**, just click **Next**.
    f. In **Features**, select **Telnet Client**.
    g. In **Confirmation**, click **Install**, then **Close** the application.
3. To verify connectivity to the RDS instance, open the Windows Command Prompt, and run the following command:

    ```
    telnet <rds-server-endpoint> <port-number>
    ```

    To find the RDS server endpoint and port to specify, open the entry of the RDS instance in the AWS console, and check the values under **Connectivity & Security** > **Endpoint & port**.

    > NOTE: If the command returns an empty prompt, that indicates connectivity between the EC2 instance and the RDS instance.

4. Download and install Microsoft SQL Server Management Studio (SSMS) on the EC2 instance.
5. To test the connection with SSMS, start the application, then in the **Connect to Server** dialog, specify the following attributes:
    - **Server type**: Select **Database Engine**.
    - **Server name**: The same RDS instance endpoint used in the telnet command.
    - **Authentication**: Select **SQL Server Authentication**, then specify the admin user name and password created when configuring the RDS instance.
6. After you specified all connection properties, click **Connect**.

Installing and configuring Active Roles on the EC2 instance

After you checked the connectivity between the EC2 and RDS instances, you can deploy and configure Active Roles on the EC2 instance.

Prerequisites

Before starting the procedure, make sure that the following requirements are met:

- The EC2 and RDS instances are connected.
- Microsoft SQL Server Management Studio (SSMS) is installed on the EC2 instance. If you followed the steps of Verifying connectivity between the EC2 and RDS instances, SSMS must already be installed on the EC2 instance.

To install Active Roles on the EC2 instance

1. Download the Active Roles installation media to the EC2 instance.
2. Run the setup and install Active Roles with all required prerequisites as described in *Active Roles installation* in the *Active Roles Quick Start Guide*.

After installing Active Roles, configure the Active Roles Administration Service.

To configure Active Roles Administration Service for managing AWS Managed Microsoft AD in SQL Server Management Studio

1. Start Microsoft SQL Server Management Studio (SSMS) and connect to the RDS for SQL Server instance as described in Verifying connectivity between the EC2 and RDS instances.
2. Under the **Databases** node of the **Object Explorer**, create two new empty databases to be used later for configuring Active Roles:
   - A database for the Management History database. Name it, for example, `ARMH`.
   - A database for the Active Roles Configuration database. Name it, for example, `ARConfig`.
3. Create a new user that Active Roles will use to connect to the SQL database in the RDS instance. To do so, right-click the **Security** > **Logins** node of the **Object Explorer**, then select **New login** and specify the following details:
   a. Under **General** > **Login name**, enter the name of the user (for example, `sql-activeroles`). Then, select **SQL Server authentication**.
   b. Under **User Mapping**, select the databases that you created (in this example, `ARMH` and `ARConfig`), and assign the **db_owner** role to both of them.

To configure Active Roles Administration Service for managing AWS Managed Microsoft AD in Active Roles Configuration Center

1. Start the Active Roles Configuration Center.
2. On the **Dashboard**, under **Administration Service**, click **Configure**.
3. In **Service Account**, enter the user name and password of the Active Roles Service account. This can be, for example, the domain admin account supplied by Amazon Web Services (AWS).
4. In **Active Roles Admin**, specify the security group or administrator user in the EC2 instance who will hold Active Roles Admin permissions.
5. In **Configuration Database Options**, select **New Active Roles database** and **Use a pre-created blank database**.
6. In **Connection to Configuration Database**, configure the following settings:

- **Database type**: Select **On Premise.** In the context of Active Roles, the Amazon RDS for SQL Server instance functions like an on-premises SQL Server.
- **Database Server name**: Specify the endpoint URL of the RDS instance. This is the same endpoint you specified during Verifying connectivity between the EC2 and RDS instances.
- **Database name**: Specify the name of the blank database that you created as the Active Roles Configuration database (in this example, `ARConfig`).
- **Connect using**: Select **SQL Server authentication**, and enter the user name and password of the user created as the owner of the database.

7. In **Management History Database Options**, select **New Active Roles database** and **Use a pre-created blank database**.
8. In **Connection to Management History Database**, specify the same **Database type**, **Database Server name** and connection settings that you set for the Configuration database. However, for **Database name**, enter the name of the blank database that you created for use as the Active Roles Management History database (in this example, `ARMH`).
9. In **Encryption Key Backup**, specify the file name and save location of the Active Roles database encryption key.
10. (Optional) Still in **Encryption Key Backup**, specify a password for additional protection. To continue, click **Next**.
11. Review your settings. Then, to apply your changes, click **Configure**.

After you configured the Active Roles Administration Service, you can also configure the Active Roles Console to manage your AWS Managed Microsoft AD instance.

To configure Active Roles Console for managing AWS Managed Microsoft AD

1. Start the Active Roles Console.
2. Due to limitations with Service Connection Points (SCPs) in the Amazon cloud, Active Roles Console is likely unable to automatically discover the Administration Service instance you configured previously.

   To manually connect to the Administration Service, in the **Connect to Administration Service** dialog, under **Service**, specify `localhost`. Under **Connect as**, select **Current user**, then click **Connect**.

   > NOTE: If you cannot connect to the Administration Service by specifying `localhost`, then specify the full **Device name** as indicated in the **Settings** > **About** page of the operating system.
3. After you connected, in the Active Roles Console landing page, click **Add Domain**.
4. In the Add Managed Domain Wizard, in **Domain Selection**, click **Browse** and select the domain configured by AWS for the EC2 instance.
5. In **Active Roles Credentials**, select **The service account information the Administration Service uses to log on**.
6. To finish adding the domain, click **Next**, then **Finish**.

7. To make sure that the contents of the AWS Managed Microsoft AD domain appear in the Active Roles Console, click **Refresh** or right-click the Active Roles node, then click **Reconnect**.

   > NOTE: The connected AWS Managed Microsoft AD environment will contain several built-in and AWS-specific containers with read-only access. You can create and manage AD objects only in the Organizational Unit whose name matches the shortname of the connected domain's name (specified during Creating the AWS Managed Microsoft AD instance).

Azure AD, Microsoft 365, and Exchange Online Management

Active Roles facilitates the administration and provisioning of Active Directory (AD), Exchange, and 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 AD 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 AD 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 AD objects (users, guest users, contacts, and so on) Active Roles synchronizes the on-premises AD objects and their properties to the AD cloud. This synchronization is performed by the Active Roles Synchronization Service between Active Roles and Microsoft Microsoft 365, whenever you configure an AD object with the Active Roles Web Interface.

> NOTE: Active Roles Web Interface supports AD-related operations only on sites based on the Administrators template. While some of the configuration procedures described in this document are also supported through the Active Roles Management Shell, they are all described with using the Active Roles Web Interface.

Fore more information about the management of Azure AD, Microsoft 365, and Exchange Online objects, see *Managing Azure AD, Microsoft 365, and Exchange Online objects* in the *Active Roles Web Interface User Guide*.

Configuring Active Roles to manage Hybrid AD objects

When a user signs up for a Microsoft cloud service, for example, 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. You can add or modify existing tenants to the management scope through the Web Interface and Management Shell. Active Roles also supports the 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

Use the Active Roles Web Interface and the Active Roles Configuration Center to configure and manage Azure AD deployments with the following actions:

- 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 tenant type
- Removing an Azure tenant
- Viewing the Azure Health status for Azure tenants and applications
- Viewing the Azure Licenses Report of an Azure tenant
- Viewing the Office 365 Roles Report of an Azure tenant

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 Active Directory (Azure AD) objects (Azure users, guest users, contacts, M365 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.

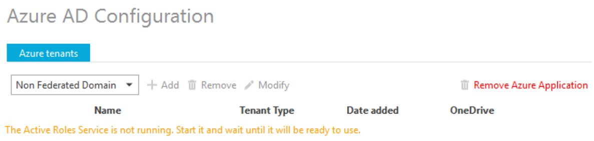
> NOTE: If you have already used an Azure tenant (or tenants) in a previous version of Active Roles, you can import and reconfigure them in two ways:
>
> - If you perform an in-place upgrade of Active Roles (that is, you install the latest version without uninstalling the previous version of Active Roles 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 8.1.3 Quick Start Guide*. For more information on the supported upgrade paths, see *Upgrade and installation instructions* in the *Active Roles 8.1.3 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**.

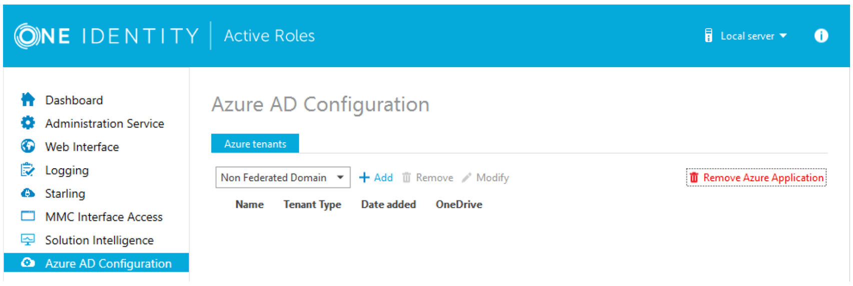
> TIP: If the Active Roles Administration Service is not running, the **Azure AD Configuration** page indicates it with an on-screen warning.



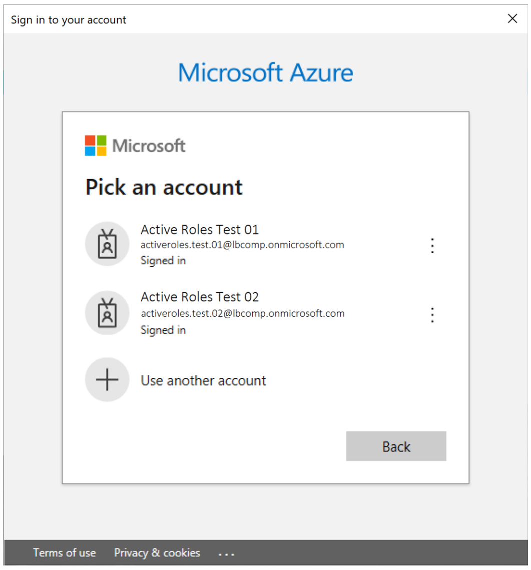
The Active Roles Service is not running. Start it and wait until it will be ready to use.

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 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. Authenticate your Azure AD administrator account.
   - If you already used one or more Azure AD administrator accounts on your PC, select your account from the **Pick an account** list, then provide the account password. If you do not find your account in the list, specify your account by clicking **Use another account**.



Sign in to your account

Microsoft Azure

Microsoft

Pick an account

Active Roles Test 01
activeroles.test.01@lbcomp.onmicrosoft.com
Signed in

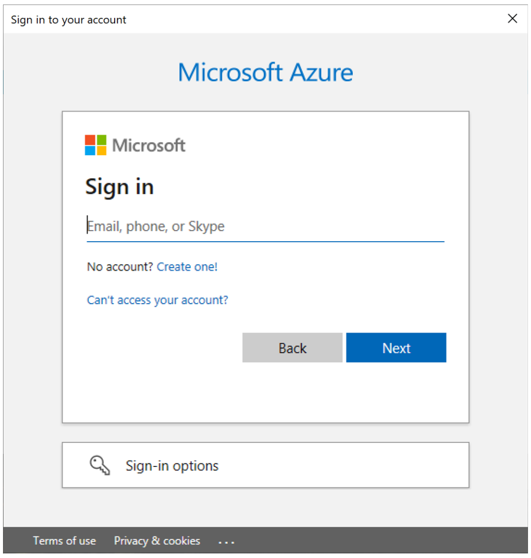
Active Roles Test 02
activeroles.test.02@lbcomp.onmicrosoft.com
Signed in

Use another account

Back

Terms of use   Privacy & cookies   ...

- If you have not used any Azure AD administrator accounts yet on the PC (for example, because you are configuring a fresh Active Roles installation), specify your account user name in the **Sign in** field, then provide your password.



Sign in to your account

Microsoft Azure

Microsoft

Sign in

Email, phone, or Skype

No account? Create one!

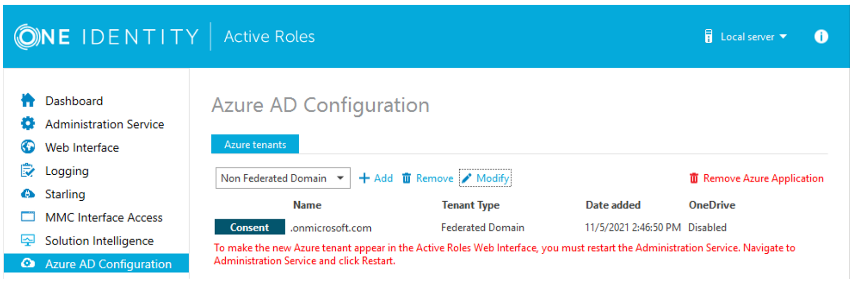
Can't access your account?

Back   Next

Sign-in options

Terms of use   Privacy & cookies   ...

Upon successful authentication, the new Azure tenant appears in the list.



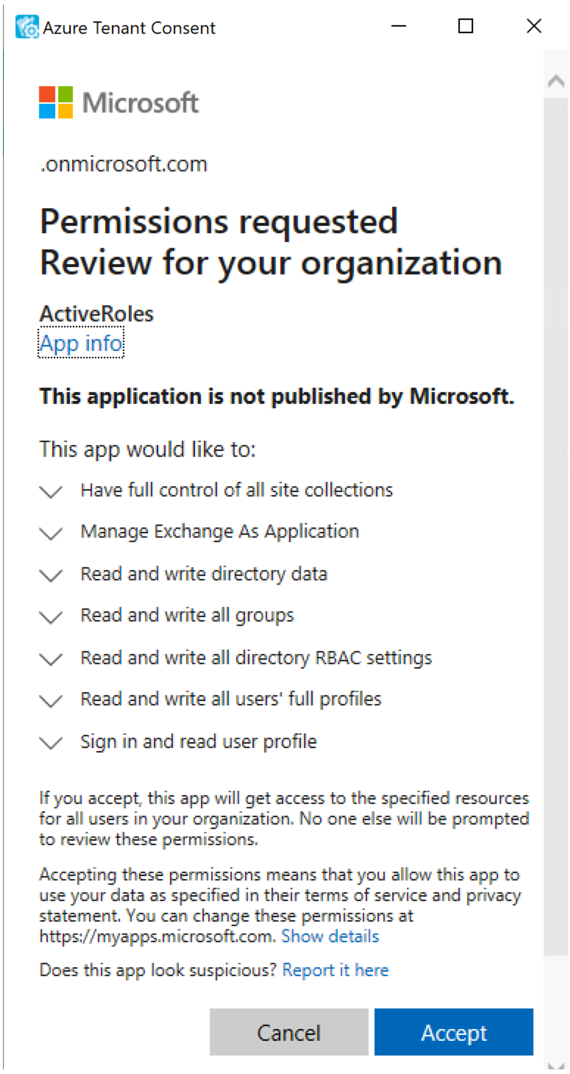
	Name	Tenant Type	Date added	OneDrive
Consent	.onmicrosoft.com	Federated Domain	11/5/2021 2:46:50 PM	Disabled

To make the new Azure tenant appear in the Active Roles Web Interface, you must restart the Administration Service. Navigate to Administration Service and click Restart.

5. 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. Authenticate your Azure AD administration account again. Depending on the type of Microsoft pop-up that appears (**Pick an account** or **Sign in**), either select the Azure AD account you used for adding the Azure tenant, or specify its user name and password again.

   > NOTE: Make sure to specify the account used for adding the Azure tenant (that is, the account name listed under the **Name** column of the Azure tenant). Authenticating with another account will result in an error.

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**.



Azure Tenant Consent

Microsoft

.onmicrosoft.com

Permissions requested
Review for your organization

ActiveRoles
App info

This application is not published by Microsoft.

This app would like to:

- Have full control of all site collections
- Manage Exchange As Application
- Read and write directory data
- Read and write all groups
- Read and write all directory RBAC settings
- Read and write all users' full profiles
- Sign in and read user profile

If you accept, this app will get access to the specified resources for all users in your organization. No one else will be prompted to review these permissions.

Accepting these permissions means that you allow this app to use your data as specified in their terms of service and privacy statement. You can change these permissions at https://myapps.microsoft.com. Show details

Does this app look suspicious? Report it here

Active Roles then authenticates every Azure AD administrative operation performed in the Azure tenant with a set of generated client ID and client secret.

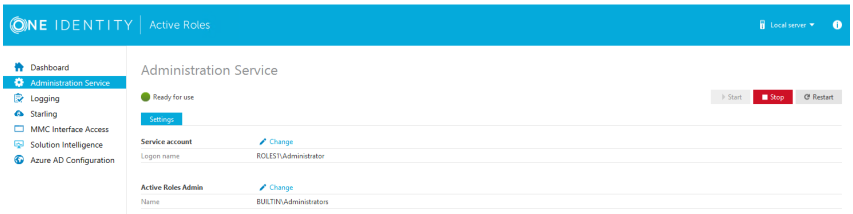
NOTE: Once you click **Accept**, Windows may show a **Security Warning** pop-up with the following message:

```
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. To restart the Administration Service, open the Configuration Center, click **Administration Service** on the left pane, then either click **Restart**, or first click **Stop** and then **Start**.



Administration Service

Ready for use

Service account	
Logon name	ROLES1\Administrator
Active Roles Admin	
Name	BUILTIN\Administrators

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 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 Viewing the Azure Health status for Azure 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 Viewing the Azure Licenses Report of an Azure tenant.
- 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 Viewing the Office 365 Roles Report of an Azure tenant.

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**.

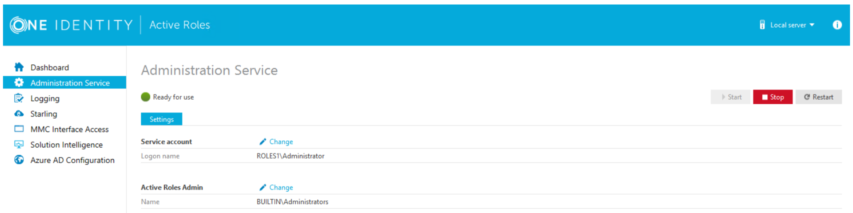
Importing an Azure tenant and consenting Active Roles as an Azure application

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 fActive 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, M365 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 8.1.3 Quick Start Guide*. For more information on the supported upgrade paths, see *Upgrade and installation instructions* in the *Active Roles 8.1.3 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**.

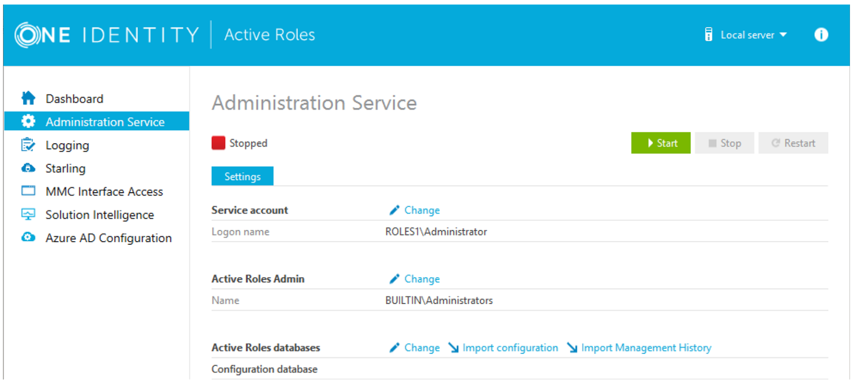
   Administration Service

   Ready for use

   Service account
   Logon name ROLES1\Administrator

   Active Roles Admin
   Name BUILTIN\Administrators

2. After the Active Roles Administration Service stopped, open the **Import configuration** wizard by clicking **Active Roles databases** > **Import configuration**.



Administration Service

Stopped

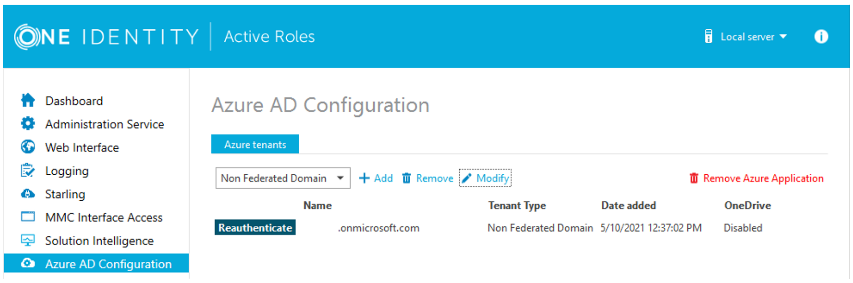
Service account	
Logon name	ROLES1\Administrator
Active Roles Admin	
Name	BUILTIN\Administrators
Active Roles databases	
Configuration database	

3. Perform the steps of the wizard. For more information, see Importing 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. After 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**.

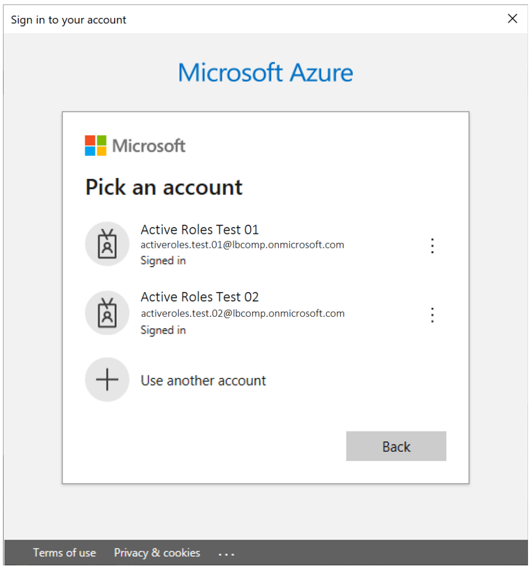
   The list of imported Azure tenants appears.



Azure AD Configuration

	Name	Tenant Type	Date added	OneDrive
Reauthenticate	.onmicrosoft.com	Non Federated Domain	5/10/2021 12:37:02 PM	Disabled

6. To configure an imported Azure tenant, click **Reauthenticate**.
7. Authenticate your Azure AD administrator account.
   - If you already used one or more Azure AD administrator accounts on your PC, select your account from the **Pick an account** list, then provide the account password. If you do not find your account in the list, specify your account by clicking **Use another account**.


Sign in to your account

Microsoft Azure

Microsoft

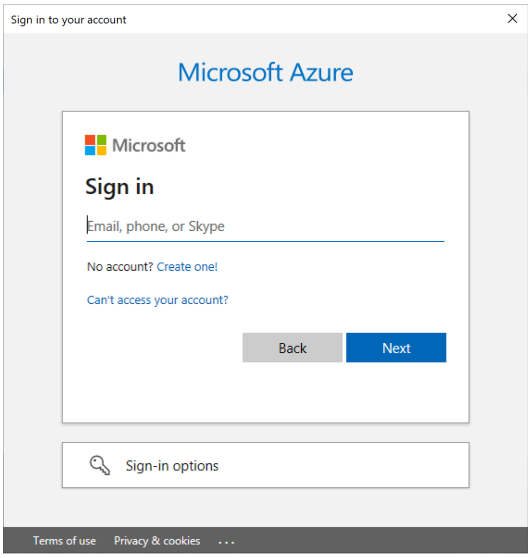
Pick an account

Active Roles Test 01
activeroles.test.01@lbcomp.onmicrosoft.com
Signed in

Active Roles Test 02
activeroles.test.02@lbcomp.onmicrosoft.com
Signed in

Use another account

- If you have not used any Azure AD administrator accounts yet on the PC (for example, because you are configuring a fresh Active Roles installation), specify your account user name in the **Sign in** field, then provide your password.


Sign in to your account

Microsoft Azure

Microsoft

Sign in

Email, phone, or Skype

No account? Create one!

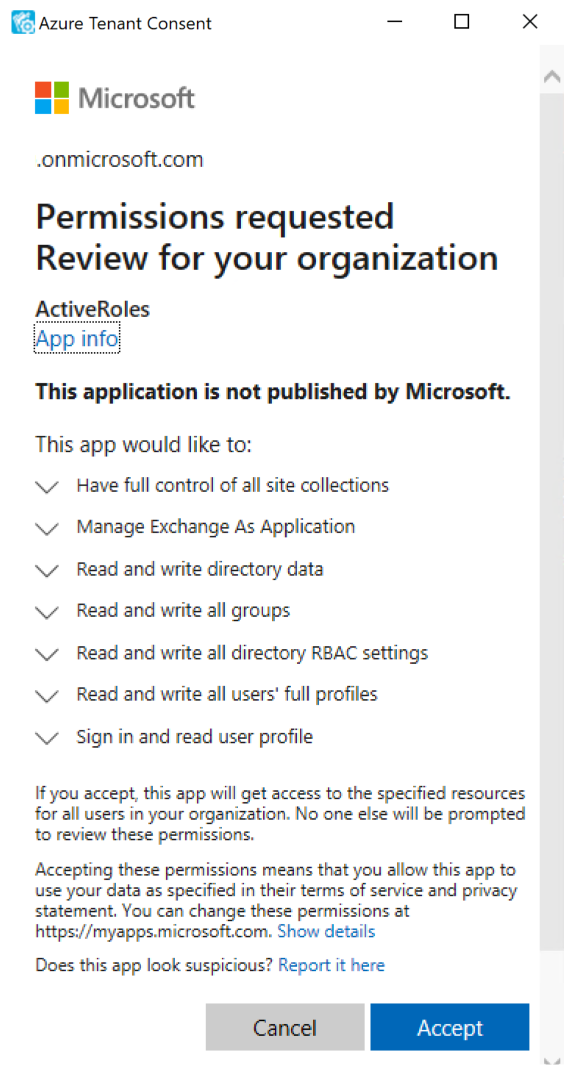
Can't access your account?

8. 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. Authenticate your Azure AD administration account again. Depending on the type of Microsoft pop-up that appears (**Pick an account** or **Sign in**), either select the Azure AD account you used for adding the Azure tenant, or specify its user name and password again.

   NOTE: Make sure to specify the account used for adding the Azure tenant (that is, the account name listed under the **Name** column of the Azure tenant).

Authenticating with another account will result in an error.

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**.



Azure Tenant Consent

Microsoft

.onmicrosoft.com

Permissions requested
Review for your organization

ActiveRoles
App info

This application is not published by Microsoft.

This app would like to:

- Have full control of all site collections
- Manage Exchange As Application
- Read and write directory data
- Read and write all groups
- Read and write all directory RBAC settings
- Read and write all users' full profiles
- Sign in and read user profile

If you accept, this app will get access to the specified resources for all users in your organization. No one else will be prompted to review these permissions.

Accepting these permissions means that you allow this app to use your data as specified in their terms of service and privacy statement. You can change these permissions at https://myapps.microsoft.com. Show details

Does this app look suspicious? Report it here

Active Roles then authenticates every Azure AD administrative operation performed in the Azure tenant with a set of generated client ID and client secret.

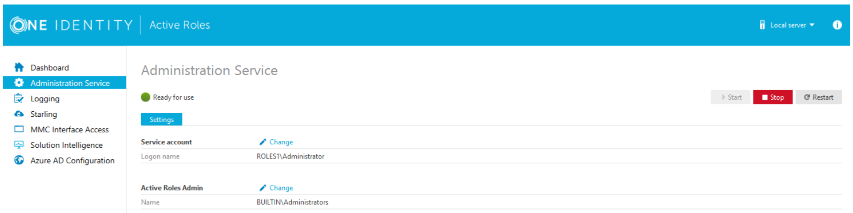
NOTE: Once you click **Accept**, Windows may show a **Security Warning** pop-up with the following message:

```
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. To restart the Administration Service, open the Configuration Center, click **Administration Service** on the left pane, then either click **Restart**, or first click **Stop** and then **Start**.



Service account	
Logon name	ROLES1\Administrator

Active Roles Admin	
Name	BUILTIN\Administrators

> 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 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 Viewing the Azure Health status for Azure 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 Viewing the Azure Licenses Report of an Azure tenant.
> - 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 Viewing the Office 365 Roles Report of an Azure tenant.

> 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 tenant type

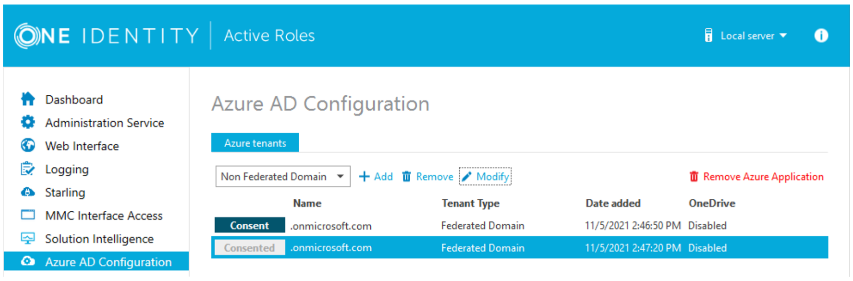
Use the Active Roles Administration Center to view or modify the tenant type of an existing Azure tenant. This is useful if you need to change the default domain settings of an Azure tenant due to an IT or organizational change.

> NOTE: Consider the following limitations when modifying the properties of the selected Azure 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 tenant.

To view or modify the Azure tenant properties

1. In the Active Roles Configuration Center, on the left pane, click **Azure AD Configuration**.

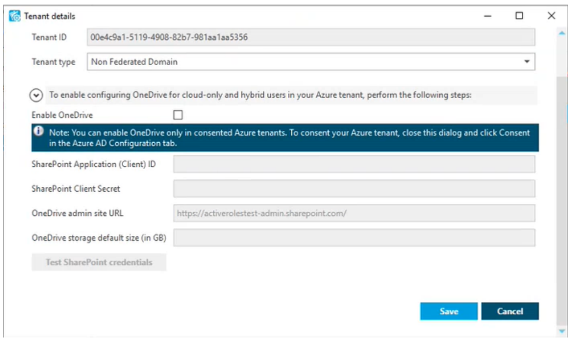
   The list of existing Azure tenants appears.

	Name	Tenant Type	Date added	OneDrive
Consent	.onmicrosoft.com	Federated Domain	11/5/2021 2:46:50 PM	Disabled
Consented	.onmicrosoft.com	Federated Domain	11/5/2021 2:47:20 PM	Disabled

2. Select the Azure tenant you want to view or modify, then click **Modify**.

   The **Tenant details** window appears.

   Tenant ID: 00e4c9a1-5119-4908-82b7-981aa1aa5356

   Tenant type: Non Federated Domain

   To enable configuring OneDrive for cloud-only and hybrid users in your Azure tenant, perform the following steps:

   Enable OneDrive □

   Note: You can enable OneDrive only in consented Azure tenants. To consent your Azure tenant, close this dialog and click Consent in the Azure AD Configuration tab.

   SharePoint Application (Client) ID

   SharePoint Client Secret

   OneDrive admin site URL: https://activerolestest-admin.sharepoint.com/

   OneDrive storage default size (in GB)

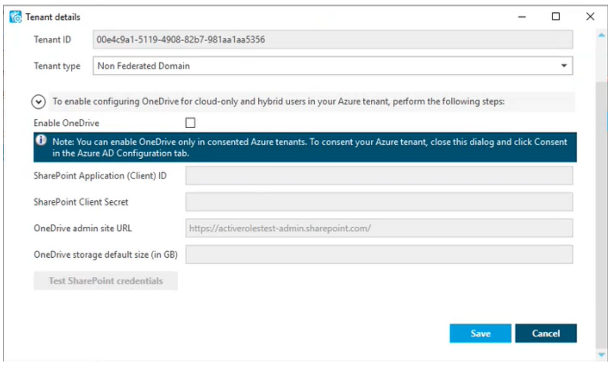
3. (Optional) To change the domain type of the Azure tenant, select the applicable type from the **Tenant type** drop-down list.

- **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.



Tenant ID: 00e4c9a1-5119-4908-82b7-981aa1aa5356

Tenant type: Non Federated Domain

To enable configuring OneDrive for cloud-only and hybrid users in your Azure tenant, perform the following steps:

Enable OneDrive □

Note: You can enable OneDrive only in consented Azure tenants. To consent your Azure tenant, close this dialog and click Consent in the Azure AD Configuration tab.

SharePoint Application (Client) ID

SharePoint Client Secret

OneDrive admin site URL: https://activerolestest-admin.sharepoint.com/

OneDrive storage default size (in GB)

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* in the *Active Roles Web Interface User Guide*.
- 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.

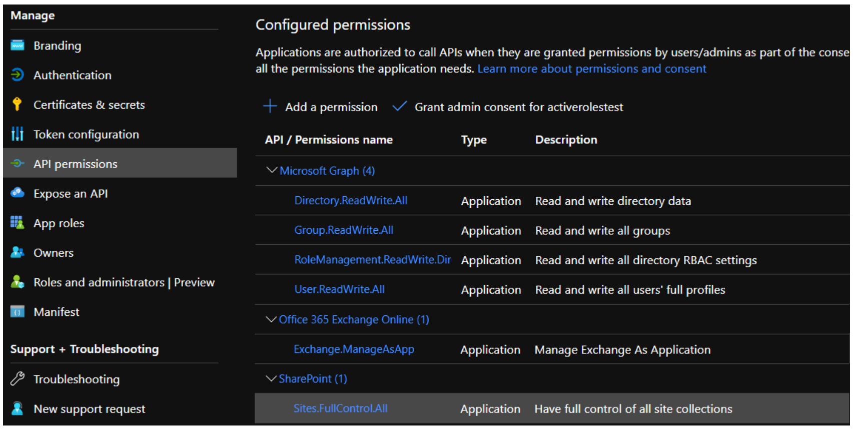
Checking and adding the Sites.FullControl.All permission for Active Roles

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 AD** 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 162: List of configured permissions under Azure AD > Manage > API Permissions of Azure Portal



Configured permissions

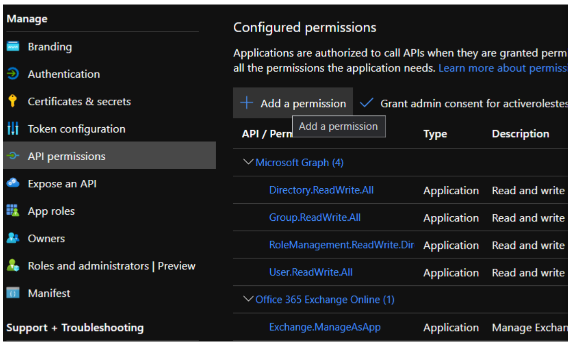
Applications are authorized to call APIs when they are granted permissions by users/admins as part of the conse all the permissions the application needs. Learn more about permissions and consent

API / Permissions name	Type	Description
Microsoft Graph (4)		
Directory.ReadWrite.All	Application	Read and write directory data
Group.ReadWrite.All	Application	Read and write all groups
RoleManagement.ReadWrite.Dir	Application	Read and write all directory RBAC settings
User.ReadWrite.All	Application	Read and write all users' full profiles
Office 365 Exchange Online (1)		
Exchange.ManageAsApp	Application	Manage Exchange As Application
SharePoint (1)		
Sites.FullControl.All	Application	Have full control of all site collections

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**.



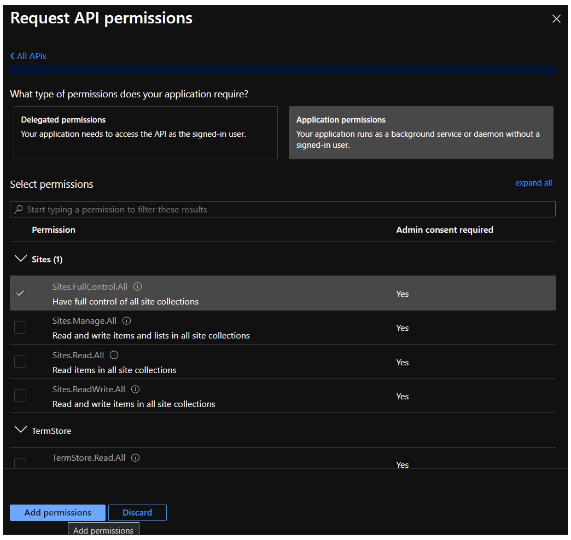
Configured permissions

Applications are authorized to call APIs when they are granted perm
all the permissions the application needs. Learn more about permiss

API / Pern	Type	Description
Microsoft Graph (4)		
Directory.ReadWrite.All	Application	Read and write
Group.ReadWrite.All	Application	Read and write
RoleManagement.ReadWrite.Dir	Application	Read and write
User.ReadWrite.All	Application	Read and write
Office 365 Exchange Online (1)		
Exchange.ManageAsApp	Application	Manage Exchan

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**.



What type of permissions does your application require?

Delegated permissions
Your application needs to access the API as the signed-in user.

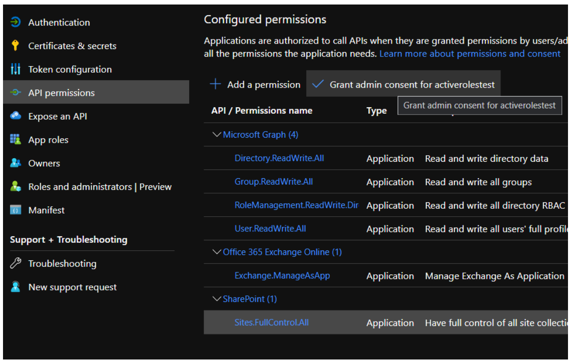
Application permissions
Your application runs as a background service or daemon without a signed-in user.

Select permissions

Start typing a permission to filter these results

	Permission	Admin consent required
	Sites (1)	
✓	Sites.FullControl.All Have full control of all site collections	Yes
□	Sites.Manage.All Read and write items and lists in all site collections	Yes
□	Sites.Read.All Read items in all site collections	Yes
□	Sites.ReadWrite.All Read and write items in all site collections	Yes
	TermStore	
□	TermStore.Read.All	Yes

5. To apply your changes, select **Sites.FullControl.All** under **Configured permissions** and click **Grant admin consent for <azure-tenant-name>**.



Configured permissions

Applications are authorized to call APIs when they are granted permissions by users/ad
all the permissions the application needs. Learn more about permissions and consent

API / Permissions name	Type	
Microsoft Graph (4)		
Directory.ReadWrite.All	Application	Read and write directory data
Group.ReadWrite.All	Application	Read and write all groups
RoleManagement.ReadWrite.Dir	Application	Read and write all directory RBAC
User.ReadWrite.All	Application	Read and write all users' full profil
Office 365 Exchange Online (1)		
Exchange.ManageAsApp	Application	Manage Exchange As Application
SharePoint (1)		
Sites.FullControl.All	Application	Have full control of all site collecti

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.

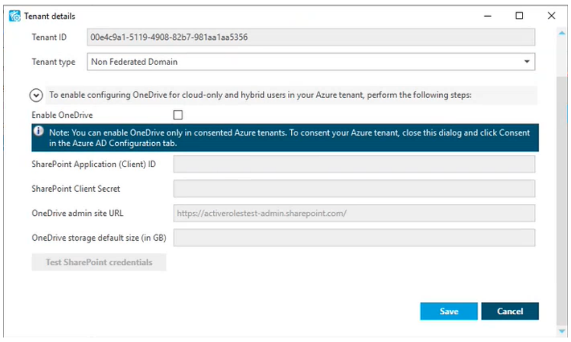
Prerequisites

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 163: Active Roles Configuration Center > Azure AD Configuration > Modify

   Tenant details

   Tenant ID: 00e4c9a1-5119-4908-82b7-981aa1aa5356

   Tenant type: Non Federated Domain

   To enable configuring OneDrive for cloud-only and hybrid users in your Azure tenant, perform the following steps:

   Enable OneDrive □

   Note: You can enable OneDrive only in consented Azure tenants. To consent your Azure tenant, close this dialog and click Consent in the Azure AD Configuration tab.

   SharePoint Application (Client) ID:

   SharePoint Client Secret:

   OneDrive admin site URL: https://activerolestest-admin.sharepoint.com/

   OneDrive storage default size (in GB):

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* in the *Active Roles Web Interface User Guide*.

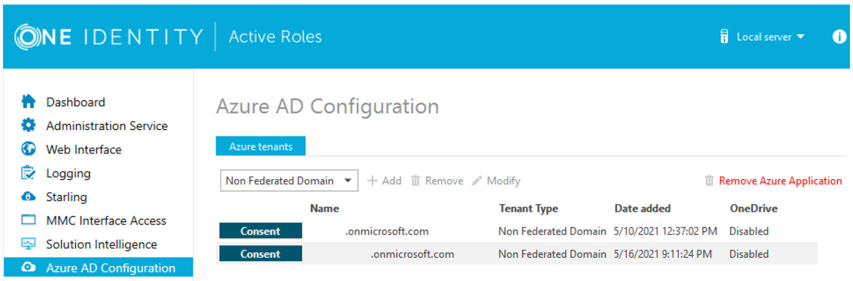
Removing an Azure tenant

You can use the Active Roles Configuration Center to delete an Azure tenant. This is typically required when an Azure tenant and its directory objects become obsolete because of organizational reasons.

To remove an Azure tenant

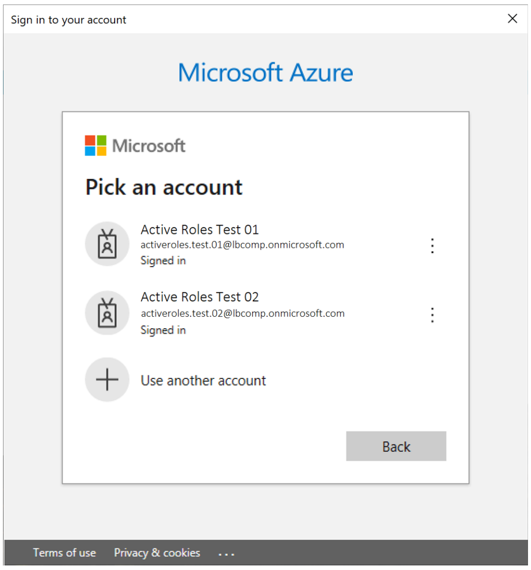
1. In the Active Roles Configuration Center, on the left pane, click **Azure AD Configuration**.

   The list of existing Azure tenants appears.



	Name	Tenant Type	Date added	OneDrive
Consent	.onmicrosoft.com	Non Federated Domain	5/10/2021 12:37:02 PM	Disabled
Consent	.onmicrosoft.com	Non Federated Domain	5/16/2021 9:11:24 PM	Disabled

2. On the **Azure AD Configuration** page, from the list of Azure tenants, select the tenant that you want to remove.
3. Click **Remove**.
4. Authenticate your Azure AD administrator account.
   - If you already used one or more Azure AD administrator accounts on your PC, select your account from the **Pick an account** list, then provide the account password. If you do not find your account in the list, specify your account by clicking **Use another account**.



Sign in to your account

Microsoft Azure

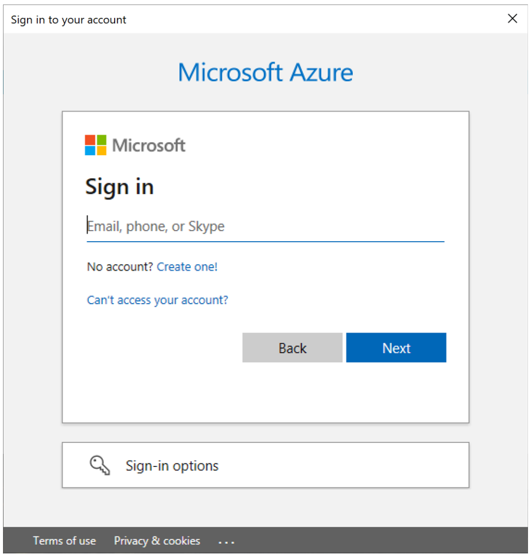
Microsoft

Pick an account

Active Roles Test 01
activeroles.test.01@lbcomp.onmicrosoft.com
Signed in

Active Roles Test 02
activeroles.test.02@lbcomp.onmicrosoft.com
Signed in

   - If you have not used any Azure AD administrator accounts yet on the PC (for example, because you are configuring a fresh Active Roles installation), specify your account user name in the **Sign in** field, then provide your password.



Sign in to your account

Microsoft Azure

Microsoft

Sign in

Email, phone, or Skype

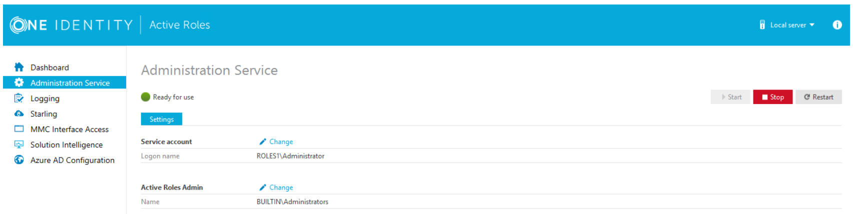
No account? Create one!

Can't access your account?

> NOTE: Make sure to specify the account used for adding the Azure tenant (that is, the account name listed under the **Name** column of the Azure tenant). Authenticating with another account will result in an error.

5. The Azure tenant and all the related domains and applications are then deleted upon successful login.

6. To apply the changes, you must restart the Administration Service. To restart the Administration Service, open the Configuration Center, click **Administration Service** on the left pane, then either click **Restart**, or first click **Stop** and then **Start**.



Administration Service

Ready for use

Service account

Logon name ROLES1\Administrator

Active Roles Admin

Name BUILTIN\Administrators

7. (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 the Active Roles Configuration Center.**
   > 2. **Add the Azure tenant again, as described inConfiguring a new Azure tenant and consenting Active Roles as an Azure application.**

8. 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.

Viewing the Azure Health status for Azure tenants and applications

You can view the Azure Health Check status of your configured Azure tenants and Azure applications in the Active Roles Web Interface. This informs you about the connection status of Active Roles to your Azure AD deployment, and the health status of your Azure AD deployment itself.

To view the Azure AD health status in Active Roles

1. On the Active Roles Web Interface, navigate to **Directory Management** > **Views** > **Azure** > **Azure Configuration** > **Azure Health Check**.
2. In the **Tenant** drop-down list, select the Azure tenant for which you want to view the Azure health status.

Active Roles then shows the following health status information:

- **Graph Connectivity**: Indicates if Active Roles is connected to the Microsoft Graph API. Successful connection is indicated with a green status.
- **Tenant Connectivity** Indicates if the Azure tenant user name and password credentials are validated. Successful connection is indicated with a green status.

  > NOTE: Active Roles can establish **Tenant Connectivity** only if **Graph Connectivity** is established successfully.

- **Azure Application Connectivity** Indicates if Active Roles is consented, validated and verified as an Azure AD application. Successful connection is indicated with a green status.

  > NOTE: Active Roles can establish **Azure Application Connectivity** only if **Tenant Connectivity** and **Graph Connectivity** are established successfully.

Viewing the Azure Licenses Report of an Azure tenant

You can view the Azure Licenses Report of an Azure tenant in the Active Roles Web Interface. Use this feature to check the Office 365 (O365) licenses available in an Azure tenant and assigned to the (guest) users of the Azure tenant.

To view the Azure AD licenses report of an Azure tenant

1. On the Active Roles Web Interface, navigate to **Directory Management** > **Views** > **Azure** > **Azure Configuration** > **Azure Licenses Report**.
2. In the **Tenant** drop-down list, select the Azure tenant for which you want to view the Azure licenses report.

Active Roles then shows the list of O365 licenses available in the Azure AD domain with the following information:

- **Valid**: The total number of a specific O365 license available for the Azure AD domain.
- **Expired**: The number of licenses for a specific O365 license that are in renewal period or have expired.
- **Assigned**: The number of licenses for a specific O365 license that have been assigned to any users in the domain.

Viewing the Office 365 Roles Report of an Azure tenant

You can view the Office 365 Roles Report of an Azure tenant in the Active Roles Web Interface. Use this feature to check the Office 365 (O365) roles that are available and assigned to the users within your Azure tenant.

To view the Office 365 roles report

1. On the Active Roles Web Interface, navigate to **Directory Management** > **Views** > **Azure** > **Azure Configuration** > **Office 365 Rules Report**.
2. In the **Tenant** drop-down list, select the Azure tenant for which you want to view the O365 roles report.

The O365 Roles Report wizard then appears, showing the list of available O365 roles and the users assigned with those roles in the Azure AD domain.

Adding an Azure AD tenant using Management Shell

To add an Azure AD tenant, use the Active Roles Management Shell. To do so, run the **New-QADAzureConfigObject** cmdlet on the Management Shell interface.

Description

New-QADAzureConfigObject allows you add an Azure AD tenant to Active Directory.

Usage Recommendations

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), use **New-QADAzureConfigObject**.

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 106: 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 107: 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 108: 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 109: AzureTenantDescription

Required	false
Position	named
Accepts pipeline input	false
Accepts wildcard characters	false

- **AzureAdminUserID**: Specifies the administrative user name for Microsoft Azure AD.

  NOTE: To perform license management or Azure user, guest user, and group management, the administrative user must have the required privileges (for example, License Administrator, User Administrator or Groups Administrator roles).

  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 110: 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 111: 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 112: 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.

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'
```

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
```

Adding an Azure AD application using Management Shell

To add an Azure AD application to the Azure AD tenant, you can use the Active Roles Management Shell.

To add an Azure AD application

On the Management Shell interface, run the **New-QADConfigObject** cmdlet.

Synopsis

This cmdlet allows 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

To add an Azure AD application, use this cmdlet.

Parameters

- **type (string)**: To specify the object class of the directory object to be created, use this parameter. 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 113: Parameters: type (string)

Required	true
Position	named
Accepts pipeline input	false
Accepts wildcard characters	false

- **name (string)**: To set the **name** attribute to this parameter value on the new object created by this cmdlet in the directory, use this parameter.

  Table 114: Parameters: name (string)

Required	true
Position	named
Accepts pipeline input	false
Accepts wildcard characters	false

- **AzureTenantId (string)**: To enter the Azure AD tenant ID obtained from the default tenant created after subscribing for Microsoft Azure, use this parameter.

  > ⚠ **CAUTION: The values that you enter when you configure the Azure AD tenant must exactly match the values configured for Azure AD. Otherwise, the Azure AD application creation and the management of the Azure AD objects will fail.**

Table 115: Parameters: AzureTenantId (string)

Required	true
Position	named
Accepts pipeline input	false
Accepts wildcard characters	false

- **Displayname (string)**: To specify the **displayName** attribute to this parameter value, use this parameter.

Table 116: Parameters: Displayname (string)

Required	false
Position	named
Accepts pipeline input	false
Accepts wildcard characters	false

- **AzureAppPermissions**: To specify the permission scope for applications for Azure AD, use this parameter.

Table 117: Parameters: AzureAppPermissions

Required	true
Position	named
Accepts pipeline input	false
Accepts wildcard characters	false

- **AzureApplicationDescription**: To specify the description of the Azure AD application, use this parameter.

Table 118: Parameters: AzureApplicationDescription

Required	false
Position	named
Accepts pipeline input	false
Accepts wildcard characters	false

Example 1

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 2

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

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: Consider the following when configuring Active Roles to manage Hybrid AD objects

- After an upgrade the **edsvaAzureOffice365Enabled** is not available for viewing or editing from **Organizational Unit** > **Advanced Properties** or through the management shell cmdlet. However, the organizational unit container continues to be an Azure-enabled container because the Azure policy is already applied.

Configuring the Azure - Default Rules to Generate Properties policy

If you want to manage hybrid Azure objects (such as hybrid Azure users) in your Organizational 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* in the *Active Roles Web Interface User Guide*.

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, to continue using the functionality, you must synchronize back the existing users and groups' information, such as Id from Azure AD to on-premises AD. To synchronize existing AD users and groups from Azure AD to Active Roles use back synchronization.

When creating objects such as users, groups, or contacts in Federated or synchronized Identity environment, they are first created on-premise and then they are synchronized to Azure using AAD Connect. To allow further management, the **BackSync** is performed to obtain the ObjectID of these objects and update the **edsvaAzureObjectID** in Active Roles.

Back synchronization can be performed automatically or manually using the 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 *Active Roles Synchronization Service Administration Guide.*

- **Manual Back Synchronization** is performed by using the existing functionality of Synchronization Service component of Active Roles. Sync 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 sync workflows for Azure AD, see *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

Prerequisites

- You must install and configure Azure AD Connect for the hybrid environment.
- The user account that is used for performing back synchronization configuration must have the following privileges:
  - User Administrator
  - Exchange Administrator
  - Application Administrator

- For the back synchronization to work as expected, install the Windows Azure Active Directory (Azure AD) module version 2.0.0.131 or later.
- You must enable the **Directory Writers Role** in Azure Active Directory. To enable the role, run 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 Active Roles must have write permissions for `edsvaAzureOffice365Enabled`, `edsaAzureContactObjectId` and `edsvaAzureObjectID`. The user must also have a local administrator privilege where the Active Roles 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 is displayed.
2. In the dialog 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**: Prompts 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 whether you want to override the existing back synchronization settings with the new settings.

- To override the existing back synchronization settings with the new settings, click **Override BackSync Settings**.
- To retain the existing back synchronization settings, click **Cancel**.

Configuring Sync Workflow to back synchronize Azure AD objects to Active Roles manually

Prerequisites

- You must install and configure Azure AD Connect for the hybrid environment.
- You must install and configure the Synchronization Service Component for Active Roles.
- You must complete the Azure AD configuration and the Administrator Consent for Azure AD application through the web interface.
- You must enforce the Azure AD built-in policy for the container where Active Roles performs the back synchronization.
- For the back synchronization to work as expected, the user in Active Roles must have write permissions for `edsvaAzureOffice365Enabled`, `edsaAzureContactObjectId`, `edsvaAzureObjectID`, and `edsvaAzureAssociatedTenantId`. The user must also have a local administrator privileges where the Active Roles Synchronization Service is running.

To configure sync workflow to back synchronize users and groups

1. **Create a 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:

   a. 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** set 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**.

b. Open the application properties and copy the following:

   - Client ID
   - Valid key of the application

c. 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 Synchronization 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 synchronization operations.

2. Create a 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. To select the container that the objects for synchronization must be selected from, define the scope.

3. Create a Sync Workflow

Create a Sync Workflow using the Microsoft 365 and Active Roles connections. Add a **Synchronization** step to update Microsoft 365 Contacts to Active Roles Contacts. To synchronize the following, configure the **Forward Sync Rule**:

- Set the 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.

4. Create a Mapping rule

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: Consider the following when creating a Mapping rule:
>
> - Based on the environment, make sure to create the correct Mapping rule to identify the contacts uniquely. An incorrect mapping rule might create duplicate objects and the back-sync operation might not work as expected.

- The initial configuration and running of the back synchronization 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 it 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.
- You must configure the Sync engine to synchronize the data back to AD based on the frequency of groups creation.

Configuring Sync Workflow to back synchronize AD contacts

To configure sync workflow to back synchronize contacts

1. **Create Connection to Microsoft 365 in the hybrid environment**

   Create a connection to Microsoft 365 using the Microsoft 365 Connector. The configuration requires Microsoft Online Services ID, Password, Proxy server (if required) and Exchange Online services.

   NOTE: The back-synchronization of contacts uses Microsoft 365 Connector to establish connection to Microsoft 365. The back synchronization of users and groups uses the Azure AD Connector to establish connection to Azure AD.

2. **Create a 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. To select the container that the objects for synchronization must be selected from, define the scope.

3. **Create a Sync Workflow**

   Create a Sync Workflow using the Microsoft 365 and Active Roles connections. Add a **Synchronization** step to update Microsoft 365 Contacts to Active Roles Contacts. To synchronize the following, configure the **Forward Sync Rule**:

   - Set the 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.

4. **Create a Mapping rule**

Create a **Mapping rule**, which identifies the contact in Microsoft 365 and on-premises AD uniquely and map the specified properties from Microsoft 365 to Active Roles appropriately.

> NOTE: Consider the following when creating a Mapping rule:
>
> - Based on the environment, make sure to create the correct Mapping rule to identify the contacts uniquely. An incorrect mapping rule might create duplicate objects and the back-sync operation might not work as expected.
> - In Federated or Synchronized environments, Azure AD group creation is not supported. The group is created in Active Roles and it 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.

Changes to Azure M365 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. You can manage the Azure objects from multiple tenants 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/Microsoft 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/Microsoft 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 Microsoft 365 and Azure Tenant Selection.

Unified provisioning policy for Azure M365 Tenant Selection, Microsoft 365 License Selection, Microsoft 365 Roles Selection, and OneDrive provisioning

The **O365 and Azure Tenant Selection** provisioning policy is a unified policy for all M365 user license and user role management as well as OneDrive provisioning for Azure AD users. This M365 management for users is controlled or restricted by creating a new provisioning policy and applying the policy to the Organizational Unit.

How the M365 and Azure Tenant Selection policy works

The **O365 and Azure Tenant Selection** provisioning policy is a unified policy for Azure Microsoft 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, Microsoft 365 license selection, and Microsoft 365 roles selection, and OneDrive provisioning for Azure AD users.

This policy is also used for tenant selection for groups and contacts.

Configuring an M365 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, Azure security 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

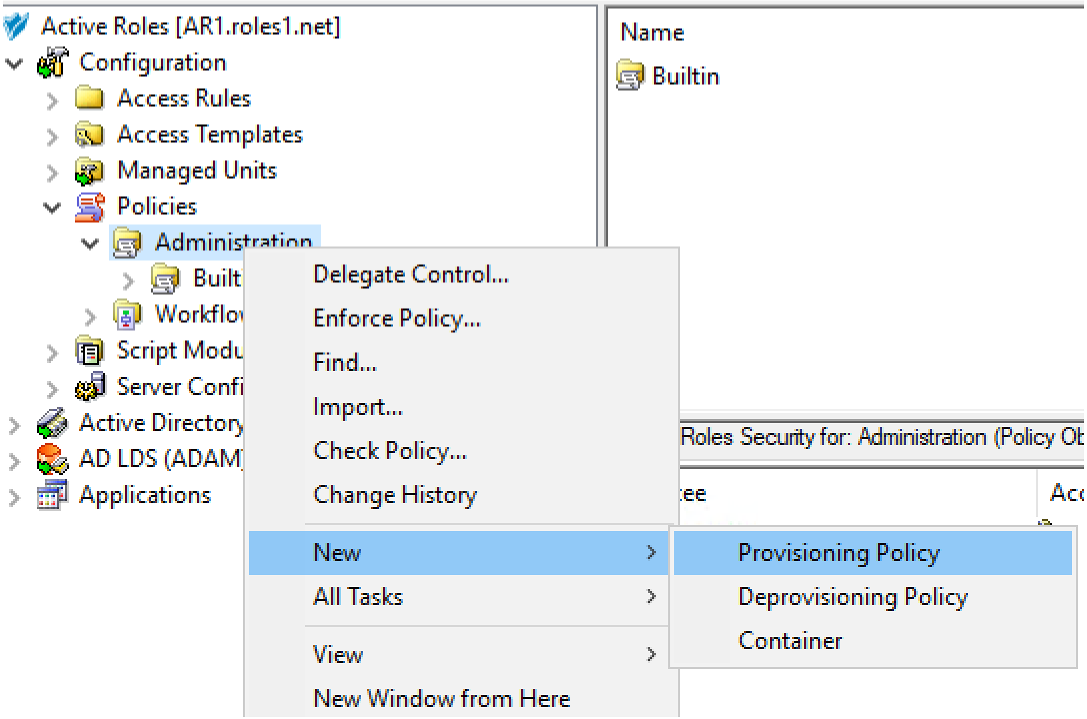
Consider the following before configuring an **O365 and Azure Tenant Selection** policy:

- 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

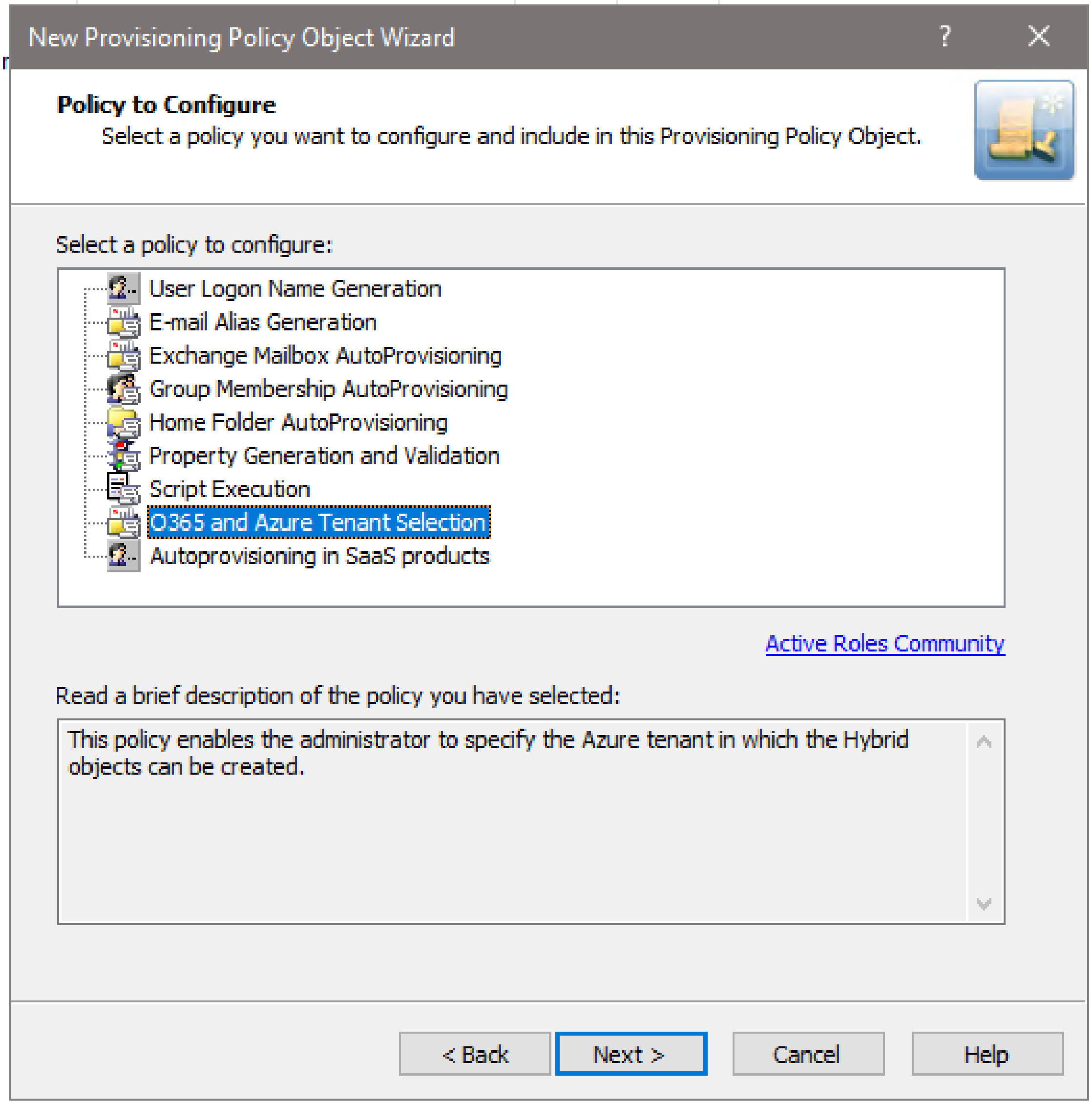
1. Navigate to **Configuration** > **Policies** > **Administration**.
2. To open the **New Provisioning Policy Object Wizard** dialog, right-click in the middle pane to open the context menu, and then select **New** > **Provisioning**

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**.



New Provisioning Policy Object Wizard

Policy to Configure

Select a policy you want to configure and include in this Provisioning Policy Object.

Select a policy to configure:

- User Logon Name Generation
- E-mail Alias Generation
- Exchange Mailbox AutoProvisioning
- Group Membership AutoProvisioning
- Home Folder AutoProvisioning
- Property Generation and Validation
- Script Execution
- O365 and Azure Tenant Selection
- Autoprovisioning in SaaS products

Active Roles Community

Read a brief description of the policy you have selected:

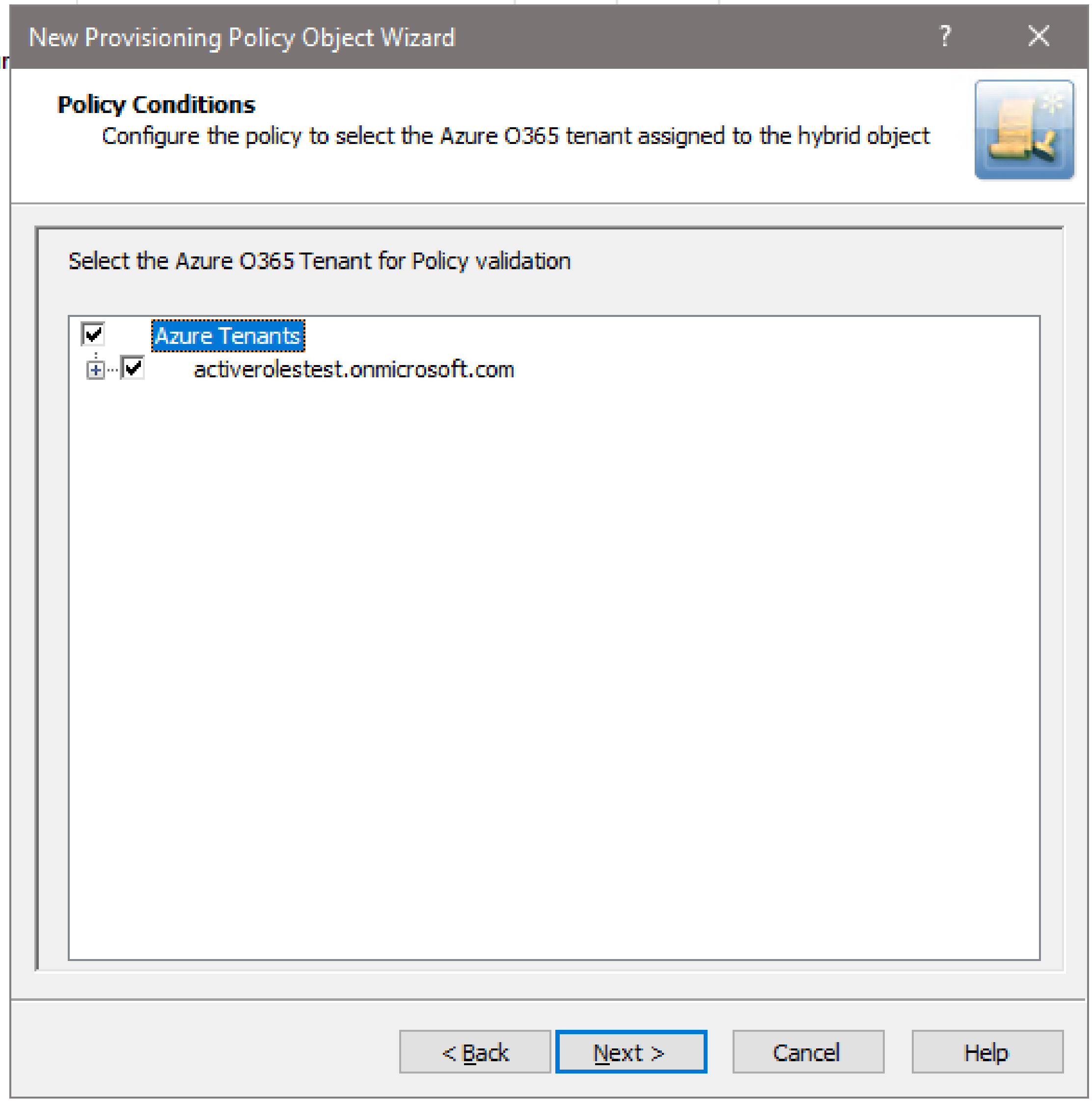
This policy enables the administrator to specify the Azure tenant in which the Hybrid objects can be created.

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**.



New Provisioning Policy Object Wizard

Policy Conditions

Configure the policy to select the Azure O365 tenant assigned to the hybrid object

Select the Azure O365 Tenant for Policy validation

☑ Azure Tenants

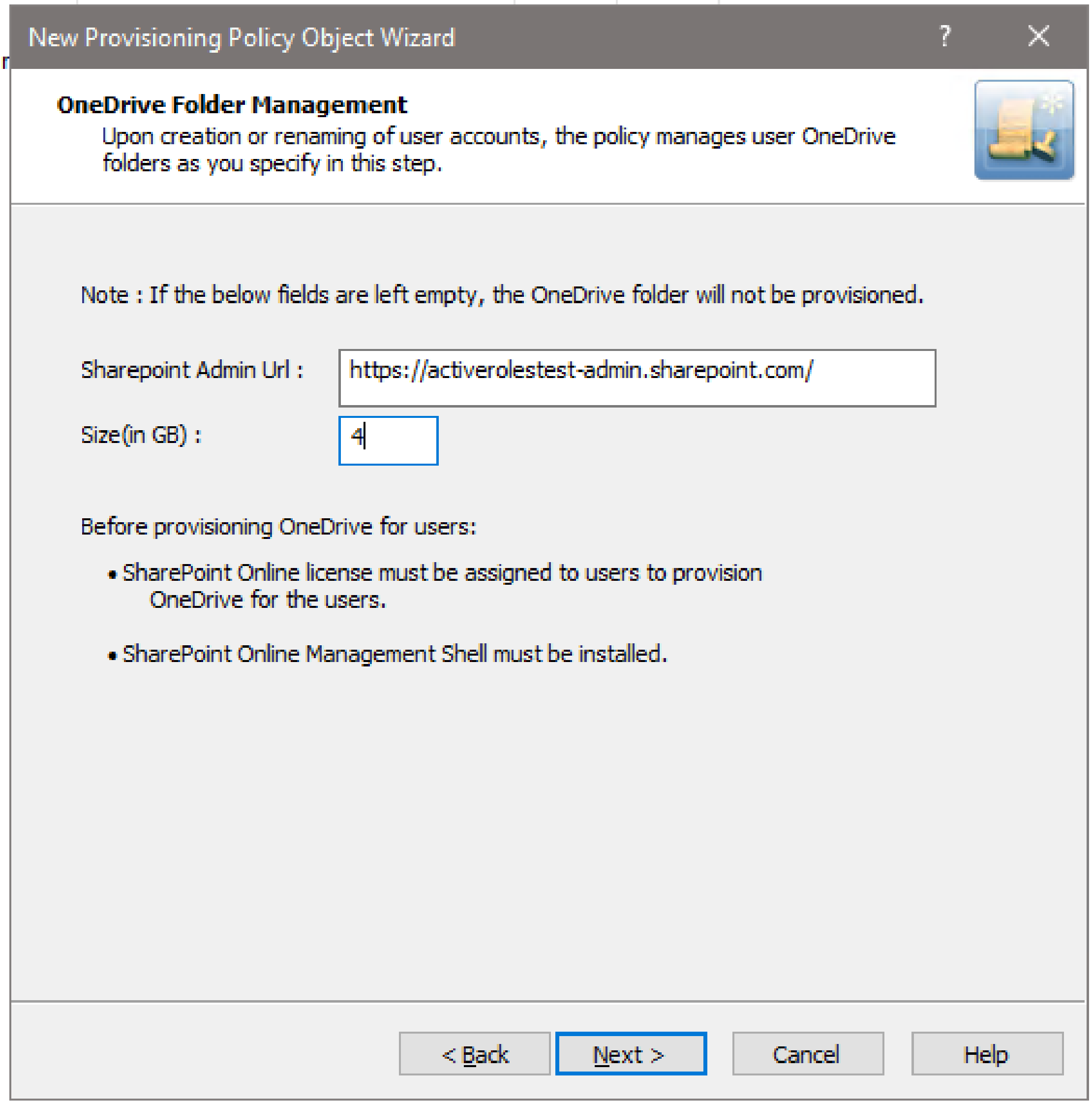
☑ activerolestest.onmicrosoft.com

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* in the *Active Roles Web Interface User Guide*.

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:



OneDrive Folder Management

Upon creation or renaming of user accounts, the policy manages user OneDrive folders as you specify in this step.

Note : If the below fields are left empty, the OneDrive folder will not be provisioned.

Sharepoint Admin Url : https://activerolestest-admin.sharepoint.com/

Size(in GB) : 4

Before provisioning OneDrive for users:

- SharePoint Online license must be assigned to users to provision OneDrive for the users.
- SharePoint Online Management Shell must be installed.

- **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 displays 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 M365 and Azure Tenant Selection policy

Microsoft 365 user license management

1. From the Web Interface, assign, or modify the Microsoft 365 license for an Azure AD User.

   The Policy is triggered for any Azure AD user in the Organization Unit for which the M365 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. To check whether there are any policy violations, right-click and select **Check Policy**

   For a container object, this displays the **Check Policy** dialog.

3. Review the options in the **Check Policy** dialog 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.

Microsoft 365 user roles management through provisioning policy

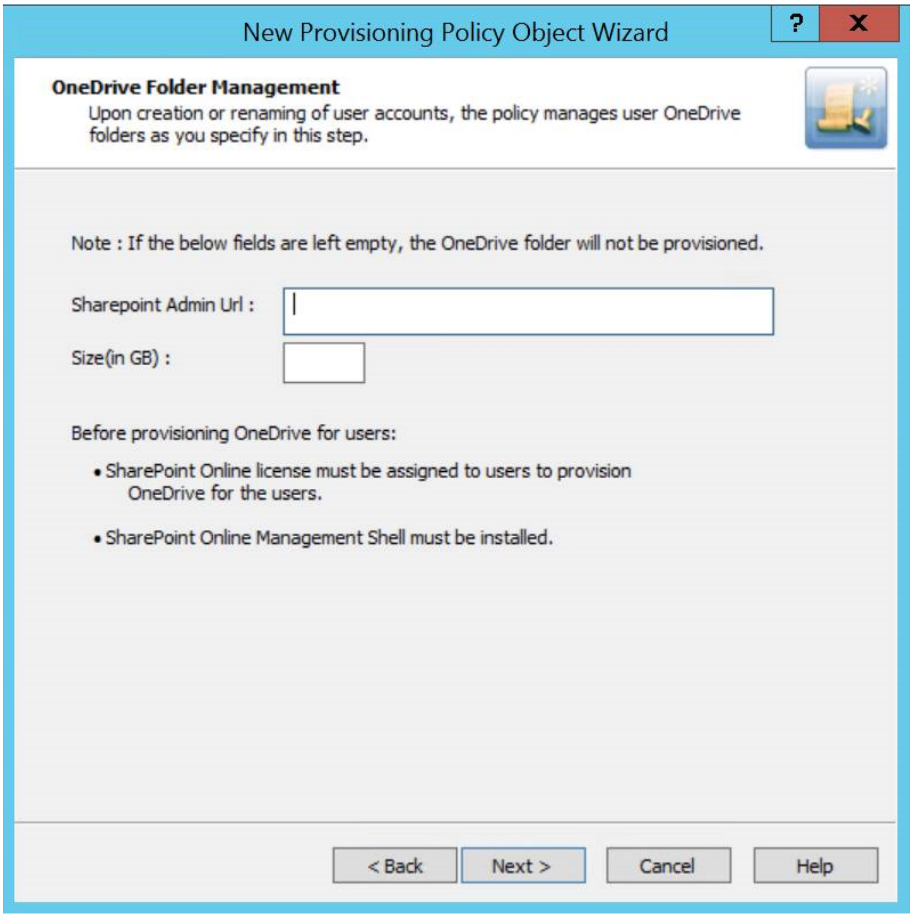
From the Web Interface, assign or modify the Microsoft 365 roles for an Azure AD User.

If the policy conditions are not satisfied while assigning Azure AD User roles while creating an Azure AD user from the Active Roles Web Interface, 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 164: OneDrive folder management wizard



OneDrive Folder Management

Upon creation or renaming of user accounts, the policy manages user OneDrive folders as you specify in this step.

Note : If the below fields are left empty, the OneDrive folder will not be provisioned.

Sharepoint Admin Url :

Size(in GB) :

Before provisioning OneDrive for users:

- SharePoint Online license must be assigned to users to provision OneDrive for the users.
- SharePoint Online Management Shell must be installed.

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: Consider the following when provisioning OneDrive for Azure AD users
   >
   > - 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 the **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.

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**

25

Managing the configuration of Active Roles

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 group. 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.

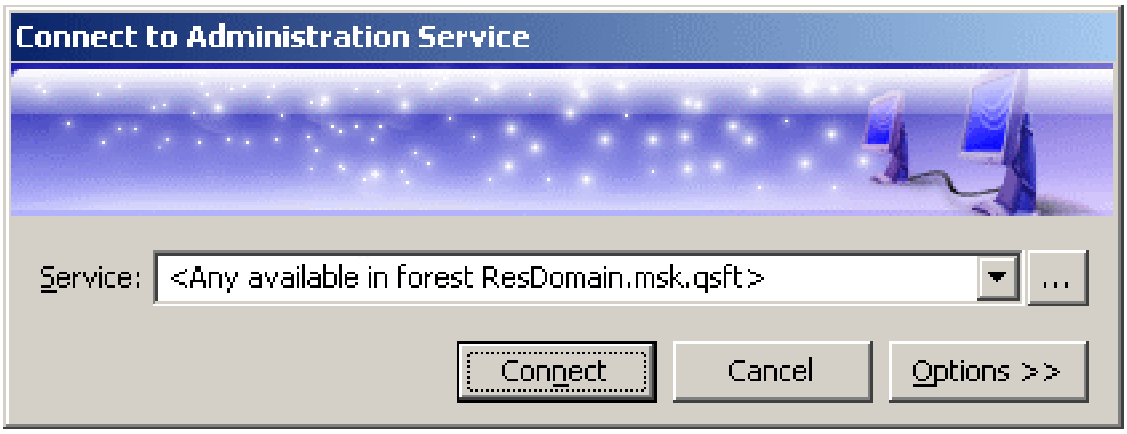
The authority to modify the Active Roles configuration can be delegated by applying the **Manage Configuration** Access Template to the **Server Configuration** container.

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 to select the appropriate Administration Service. To display this dialog, right-click **Active Roles** in the Console tree and click **Connect**. The dialog looks as shown in the following figure.

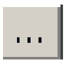
Figure 165: Connect to Administration Service



Service: <Any available in forest ResDomain.msk.qsft>

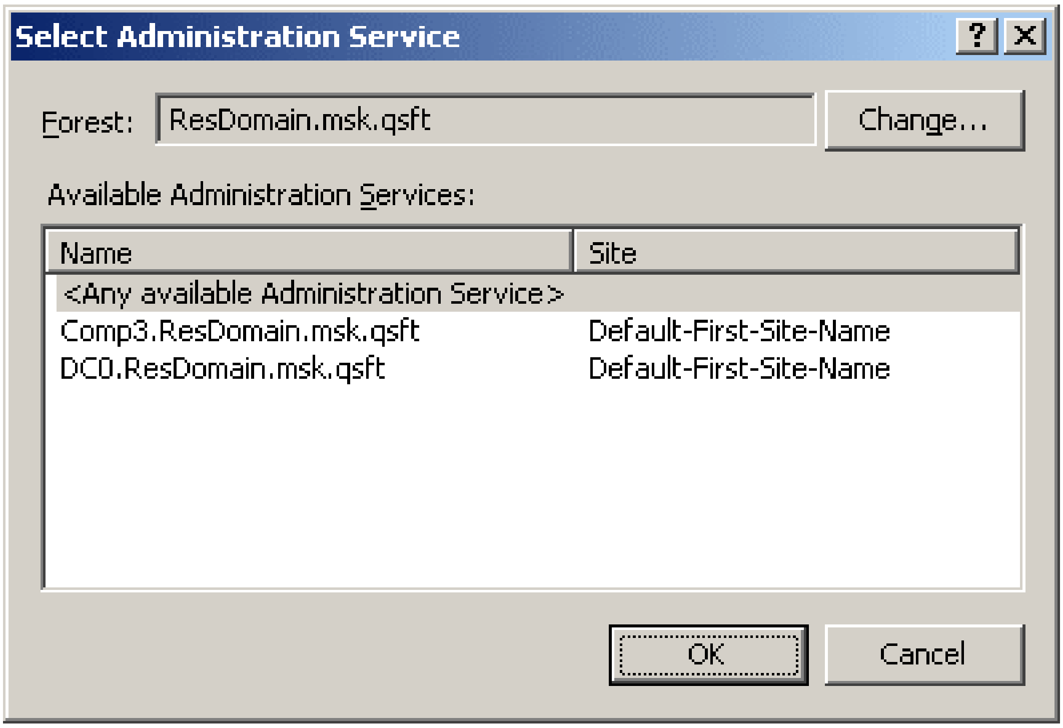
In the **Service** box, type or select the name of the computer running the Administration Service to connect to, then click **Connect**. The **Service** box provides a list of names that were specified for previous connection sessions. The last selected name is displayed by default.

To select the Administration Service that is not in the list, click **Select** next to the **Service** box:



This displays the **Select Administration Service** dialog, shown in the following figure.

Figure 166: Select Administration Service



Forest: ResDomain.msk.qsft

Available Administration Services:

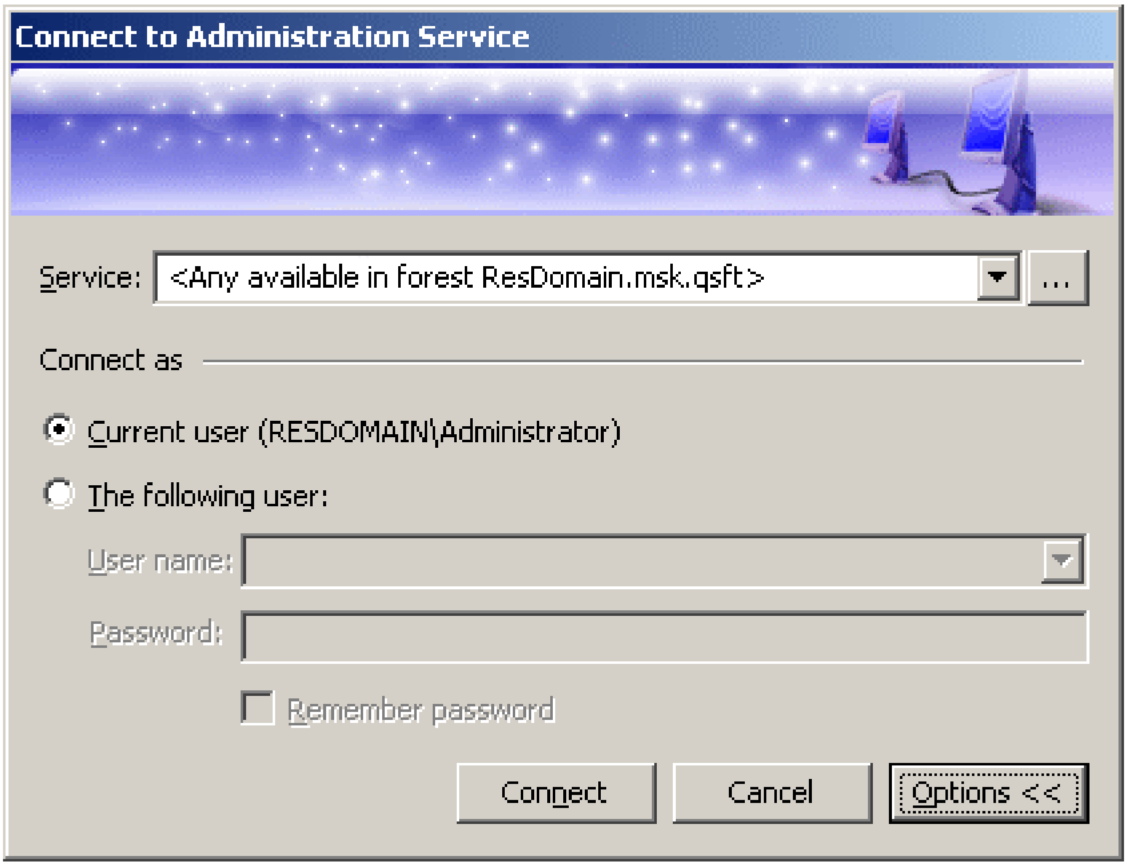
Name	Site
<Any available Administration Service>	
Comp3.ResDomain.msk.qsft	Default-First-Site-Name
DC0.ResDomain.msk.qsft	Default-First-Site-Name

The **Select Administration Service** dialog 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, 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, as shown in the following figure.

Figure 167: Connect to Administration Service



Connect to Administration Service

Service: <Any available in forest ResDomain.msk.qsft>

Connect as

● Current user (RESDOMAIN\Administrator)

○ The following user:

User name:

Password:

□ Remember password

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 Active Roles Console

By default, on installing Active Roles, all users are allowed to log in to the Active Roles Console. To manage the Console 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 Active Roles Console, 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 Active Roles Console.
7. Click **OK** to close the **Active Roles Security** dialog.

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:

1. Click the Console tree root.
2. 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*.

> NOTE: This option applies to all Administration Service 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**.

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 in.**
    - **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**, 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, then click **Delete**.

NOTE: Consider the following when managing managed domains:

- 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:
    1. On the **General** tab in the **Properties** dialog, choose the appropriate option.
    2. 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 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. For more information, 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 that appears, and then select the **Built-in Policy - Exclude from Managed Scope** Policy Object.
3. Click **OK** to close the dialogs.

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’s 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 email 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 `(&(objectCategory=person)(objectClass=user))` LDAP query.

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 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.
- 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:

1. Choose the **Enforce Policy** command on the domain object under the **Active Directory** node in the Active Roles Console.
2. Click **Add**.
3. 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:

1. In the Active Roles Console, choose the **Enforce Policy** command on the OU.
2. 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):

1. Create a Managed Unit with the appropriate membership rules.
2. 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 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’s 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 email.

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 group are authorized to configure thresholds and notification for the managed object count.

2. In the Console tree, select the **Active Roles** root node.

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 that appears, specify the desired threshold value for active 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 you of the threshold violation alert over email, then, in the **Threshold Value** dialog, 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 email addresses.

   c. Click the button next to the **E-mail server settings** field. Then, on the **Mail Setup** tab in the dialog 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.

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 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 **Active Roles** root node.

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 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.

   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 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 that opens, view or modify the value of the attribute.

Scenario: Implementing a Birthday attribute

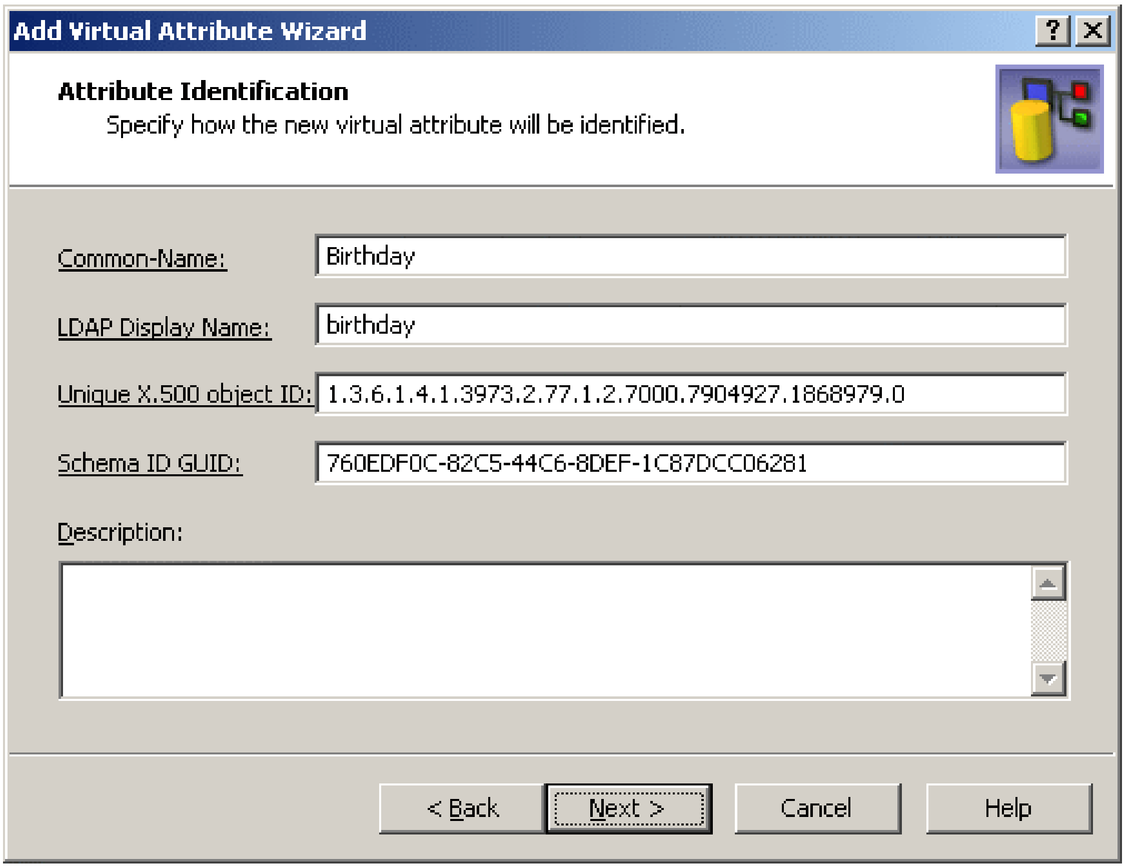
This scenario illustrates how to create and use a virtual attribute to store information on the birthdays of users.

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 168: Attribute identification



Add Virtual Attribute Wizard

Attribute Identification

Specify how the new virtual attribute will be identified.

Common-Name: Birthday

LDAP Display Name: birthday

Unique X.500 object ID: 1.3.6.1.4.1.3973.2.77.1.2.7000.7904927.1868979.0

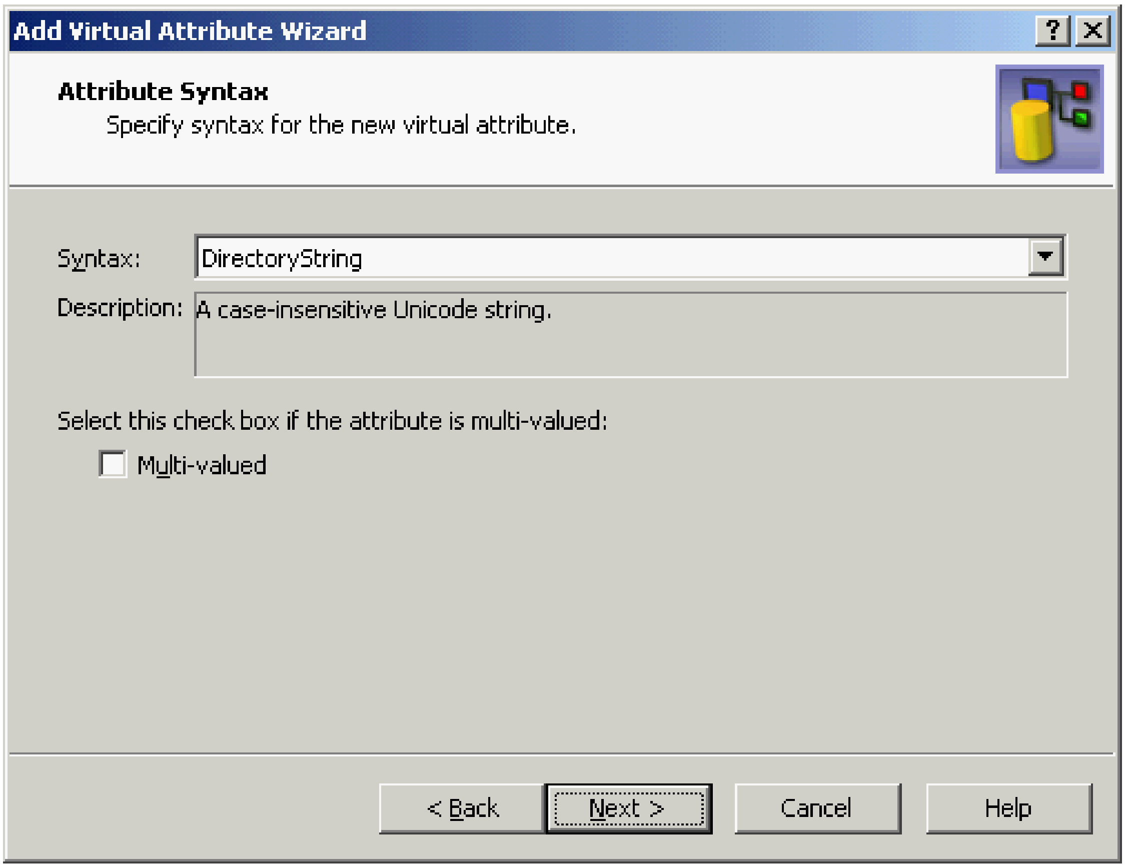
Schema ID GUID: 760EDF0C-82C5-44C6-8DEF-1C87DCC06281

Description:

5. Click **Next.**

   The **Attribute Syntax** page should look as shown in the following figure.

Figure 169: Attribute syntax



Attribute Syntax

Specify syntax for the new virtual attribute.

Syntax: DirectoryString

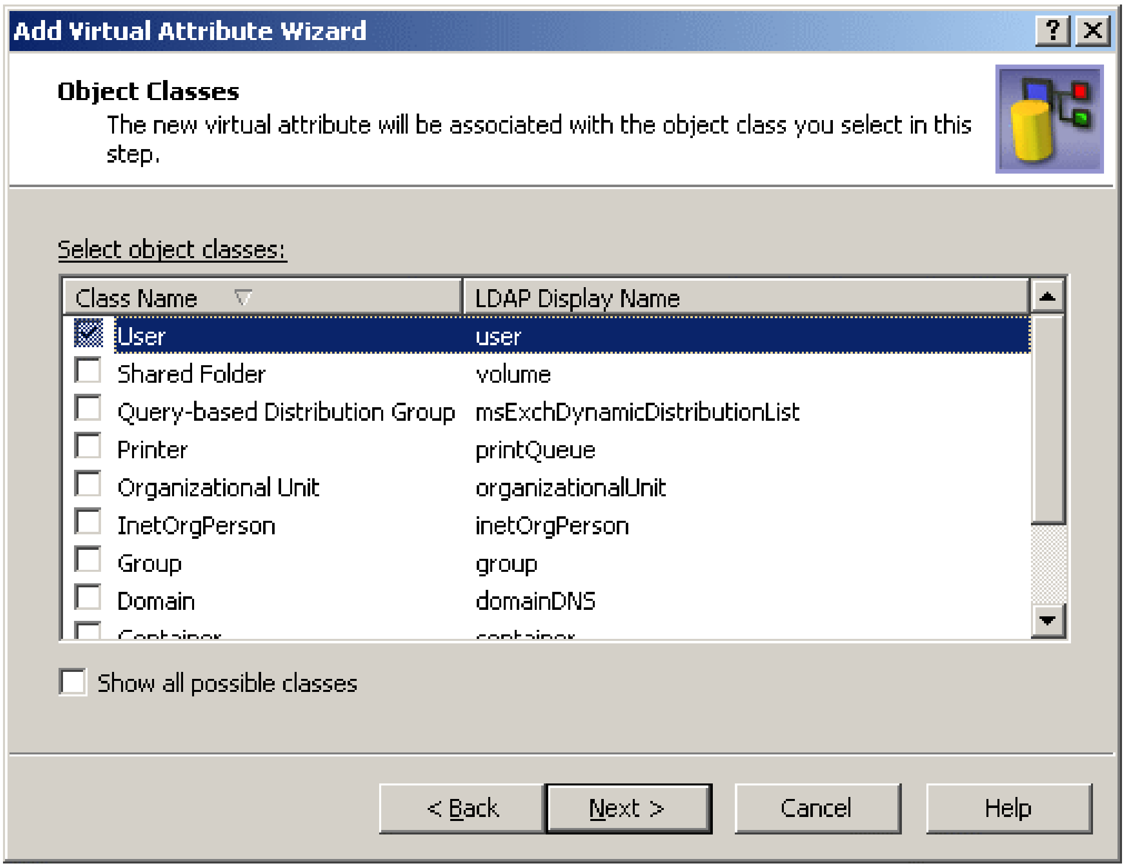
Description: A case-insensitive Unicode string.

Select this check box if the attribute is multi-valued:

☐ Multi-valued

6. Click **Next**.
7. On the **Object Classes** page, select the check box next to **User**, as shown in the following figure.

Figure 170: Object classes



Add Virtual Attribute Wizard

Object Classes

The new virtual attribute will be associated with the object class you select in this step.

Select object classes:

Class Name	LDAP Display Name
☑ User	user
□ Shared Folder	volume
□ Query-based Distribution Group	msExchDynamicDistributionList
□ Printer	printQueue
□ Organizational Unit	organizationalUnit
□ InetOrgPerson	inetOrgPerson
□ Group	group
□ Domain	domainDNS
□ Container	container

□ Show all possible 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**, 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, select both the **Show all possible attributes** and **Include attributes with empty values** check boxes.
3. Click **Birthday** in the list of properties, 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. For more information, see *Scenario 2: Implementing Self-Administration* in the *Active Roles Feature Guide*.

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 group, and thus has administrator rights on the Administration Service.
- **Client Version**: Client application, such as Active Roles Console 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 email 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 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, then press **CTRL+I** to display the **Add Counters** dialog.
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
- Customizing display names

Other Properties tab in the Properties dialog

The Active Roles Console also makes it possible to extend the **Properties** dialog 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 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 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 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 for Group, Computer, or Organizational Unit, create and configure a custom display specifier named **group-Display**, **computer-Display**, or **organizationalUnit-Display**, respectively.

> NOTE: 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 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 **EDS-Display-Specifier-Container** class.
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 **Display-Specifier** class. 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. Navigate 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 reconnect the Console to the Service, for your changes to take effect.

As a result of these steps, the **Properties** dialog 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 clicking **Properties**.

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 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: 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 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 **EDS-Display-Specifier-Container** class.
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 **Display-Specifier** class.

   > NOTE: 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. Navigate 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 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 selecting **New** > **User**. Follow the wizard steps to reach the page containing the list of “other” 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 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 Active Roles Console, 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 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.

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 all the required configuration settings.
- **Hub pages and management wizards**: Once the 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 website 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 Active Roles Console.
- **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, if no component 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. For more information, see Enable-PSRemoting. On Windows Server 2016 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 location: On Windows Server 2016 or later, click the **Active Roles 8.1.3 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 8.1.3** > **Active Roles 8.1.3 Configuration Center**.
- On Windows 8 or later, click the **Active Roles 8.1.3 Configuration Center** tile on the **Apps** page.

Prerequisites for running 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 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.

Before launching Configuration Center, it is recommended to perform the following steps:

1. On the system where Active Roles is installed, navigate to `C:\Program Files\One Identity\Active Roles\8.1.3\Shell`.
2. Right click on the **ActiveRolesServiceConfiguration.psm1** file and select **Properties**.
3. On the **ActiveRolesServiceConfiguration Properties** dialog, click **Digital Signatures** > **Details**.
4. On the **Digital Signatures Details** dialog, click **View Certificate**.
5. On the **Certificate** dialog, click **Install Certificate...**.

6. On the **Certificate Import Wizard** dialog, 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, 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 Active Roles Console 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.

Configuring 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. You can choose from the following options:
  - **Windows authentication**: The Administration Service will use the credentials of the service account.
  - **SQL Server authentication**: The Administration Service will use the SQL login name and password you supply in the wizard.
  - **Azure AD authentication**: 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 more information and step-by-step instructions, see *Steps to deploy the Administration Service* in the *Active Roles Quick Start Guide*.

Configuring 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 more 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, 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.

Viewing 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).

Changing 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.

Importing configuration data

When deploying the Administration Service, you may need to import configuration data from an existing database to ensure that the new Administration Service instance has the same configuration as the existing one. Importing configuration data to a newly created database instead of attaching the Administration Service to an existing database is necessary if the version of the Administration Service you are deploying is greater than the version of the database you want to use. Some examples of such a situation are the following:

- Upgrading the Administration Service while preserving its configuration.
- Restoring configuration data from a backup copy of the database whose version does not match the version of the Administration Service.

For more information, see *Importing configuration data* in the *Active Roles Upgrade Guide*.

Importing Management History data

A part of the Active Roles database, the Management History data storage is empty after you have configured the Administration Service with the option to create a new database. During import of configuration data, Configuration Center transfers only the administrative right assignments, policy definitions, administrative view settings, workflow definitions and other parameters that determine the Active Roles work environment. Management history data is excluded from the import operation to reduce the time it takes to upgrade the configuration of the Administration Service.

The Management History data describes the changes that were made to directory data via Active Roles. This includes information about who did what and when it was done as applied to the directory data management tasks. The Management History data is used as a source of information for the change history and user activity reports. In addition, the Management History data storage holds information about various tasks related to approval workflow and temporal group membership.

After configuring the Administration Service with the option to create a new database, and importing the configuration data from an existing database, you must take additional steps to transfer the Management History data from that database to the new database. Configuration Center provides the **Import Management History** wizard to perform this task.

The **Import Management History** wizard populates a new storage of Management History data with the data found in an existing Active Roles database, to make the data available to the Active Roles user interfaces after your configure a new Administration Service instance. The wizard merges the Management History data from the source database with the data stored in the destination database.

> NOTE: The **Import Management History** wizard only adds new data, keeping intact any data that already exists in the destination database. You may import your legacy Management History data at any time after you have configured the Administration Service, without the risk of losing any data.

Although importing Management History data 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), you must import configuration data first. You can import Management History data after that if 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 the following:

- 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 **Import Management History** wizard, click **Import Management History** on the **Administration Service** page in the Configuration Center main window. 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 more information, see *Importing Management History data* in the *Active Roles Upgrade Guide*.

Viewing 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.

Starting, stopping or restarting the Administration Service

You can start, stop or restart the Administration Service by clicking **Start**, **Stop** or **Restart** 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

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 website 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 with 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 more information and step-by-step instructions, see the “Additional configuration” topic 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 website 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), 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 more information and step-by-step instructions, see the “Additional configuration” topic 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 every 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 website 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: Consider the following when configuring the Web Interface for secure communication for the first time:
   >
   > - 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 website.

5. In the **Available HTTPS Bindings** field, click the drop-down list and select the required binding for the website.

6. Click **Modify**.

   After successful completion of configuration changes, in the Web Interface window, the Force SSL Redirection configuration state for the selected website 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 website, 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 every Web Interface site 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 website 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 Configuring federated authentication.

Configuring federated authentication

You can access an application or websites 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 Configuring federated authentication.

Starling Join configuration task

Active Roles version 8.1.3 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 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.

Active Roles Console access management

On installing Active Roles on a computer, the Active Roles Console user access setting is not enabled by default, and any user is enabled to log in to the Active Roles Console. You can use Configuration Center, to set the Active Roles Console user access.

To manage the Active Roles Console 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 **Component**, then click **Modify** or double-click **Component**.
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 Active Roles Console.
   - **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 Active Roles Console Access settings get configured successfully. A message is displayed prompting you to restart the Administrative Service to disconnect the current Active Roles Console user sessions and for the updated settings to be reflected on the Active Roles Console.

> NOTE: Consider the following when managing Active Roles Console:
>
> - The user must be delegated with the **User Interfaces** access rights on the **User Interfaces** container under **Server Configuration** to obtain access to the Active Roles Console. 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 Active Roles Console.

Logging management tasks

Active Roles writes most events to its own Event log in Windows Event Viewer, under **Applications and Services**, called **Active Roles Admin Service**. You can use this Event log to help determine root causes for issues and typically provide more detailed error information if any issues are encountered within the Console or the Web Interface.

In addition to the Event log, there is a debug option for the Active Roles Administration Service that is disabled by default. Enabling logging can be accessed either in the Active Roles Console or in the Configuration Center.

Use the Configuration Center to enable, disable or view diagnostic logs for the Active Roles components that are installed on the computer running Configuration Center. On the **Logging** page, the Configuration Center lists the following information:

- **Component**: Name of the component, such as Administration Service, Web Interface or Console.
- **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, then click **Open in Log Viewer** on the toolbar. For more information, see Active Roles Log Viewer.

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 Roles 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

After 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 settings

1. On the **Dashboard** page in the **Configuration Settings** main window, click **Solution Intelligence**.
2. On the **Solution Intelligence** page, select **Enable Solution Intelligence**.
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 as a service account. Before you configure a gMSA as an Active Roles Service account, the following prerequisites 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.
- The gMSA 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 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. 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.

For more information on managing gMSA accounts see *Managing Group Managed Service Accounts* in the *Active Roles User Guide*.

Configuring the Active Roles Service account to use a gMSA

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 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 more information, 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, click **Object Types**.

5. In the **Object Types** dialog, select the **Service Accounts** object type along with the Users object type and click **OK**.

6. In the **Service User** or **Service Account** dialog, click **Check Names** to select the required gMSA, and click **OK**.

   The **Configure Administration Service** dialog displays the new login name for the gMSA. 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, then an error is displayed prompting you to check if the system is permitted to use the provided gMSA.

   If the gMSA is not part of the **Local Administrators** group, then an error is displayed prompting you to check if the gMSA is a member of the **Local Administrators** group on the system.

8. If all the prerequisites 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 that use the same database.

   > NOTE: When you create a new database, you can add the `DB_owner` permission to the gMSA 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 Administration 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 save 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

Active Roles provides support to change an Active Roles account to use a gMSA.

To change the Administration Service account to use a gMSA 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 more information, 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, under **Service Account**, click **Browse**.
4. In the **Select User** or **Service Account** dialog, click **Object Types**.
5. In the **Object Types** dialog, select the **Service Accounts** object type along with the **Users** object type and click **OK**.
6. In the **Service User** or **Service Account** dialog, click **Check Names** to select the required gMSA, and click **OK**.

The **Change Service Account** dialog displays the new login name for the gMSA. The **Password** field is disabled.

7. Click **Change** to save the changes for the service account.

   If the system that is running the Active Roles Service is not linked to the gMSA, then an error is displayed prompting you to check if the system is permitted to use the provided gMSA.

   If the gMSA is not part of the **Local Administrators** group, then an error is displayed prompting you to check if the gMSA is a member of the **Local Administrators** group on the system.

   If all the prerequisites are met, the service account is changed to gMSA 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 **Change** on the **Active Roles Admin** page.

> NOTE: 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, this 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 diagnosing the issue.

- 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). 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 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 from the Administration Service object in the **Configuration/Server Configuration/Administration Service** container.
5. Use the **Diagnostics** page to perform the following tasks:
   - To turn the Administration Service's log on or off, click the appropriate option. 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.
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 diagnosing the issue.
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 by clicking **Open** on the Log Viewer toolbar, and supplying 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.

SQL Server replication

SQL Server database replication allows copying and distributing data between different nodes to maintain 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.

> NOTE: For more information about SQL Server replication, see *SQL Server Replication* in the *Microsoft SQL documentation* or in the *SQL Server Books Online*.

SQL Server replication terminology

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 the replication group, with each server in the replication group being referred to as the replication partner.

The replication group is comprised of replication partners that include a single Publisher and can 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 it is 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.

SQL Server replication model overview

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 more information, see Viewing replication settings.

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, the 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. For more information, see Create and Apply the Initial Snapshot.

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, the 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, the 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 might 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*.

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). One Identity recommends 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. For more information, 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.

About replication groups

The Publisher and its Subscribers constitute a replication group. Every replication group must include a single Publisher and it can include any number of Subscribers. The members of a replication group are referred to as replication partners.

Each member of a replication group (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 it is 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 can 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, you must designate a standalone database server as the Publisher. The new replication group will then have a single member—the Publisher. Later, you can 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

To create a replication group

1. Connect to a standalone Administration Service.
2. Promote the Administration Service database server to Publisher.

For more information on how to connect to the Administration Service, see Connecting to the Administration Service.

Once connected to the Administration Service, perform the following steps to promote the Administration Service database server to Publisher.

To promote the Administration Service databse 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. For more information, 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.

⚠ **CAUTION: Hazard of data loss!**

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.

To add a replication partner to a replication group

1. Connect to the Administration Service whose database server holds the Publisher role.

   For more information on how to connect to the Administration Service, see Connecting to the 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 click **Add Replication Partner**.
4. Follow the instructions in the **New Replication Partner** wizard.
5. On the **Database Selection** page, click **Browse**.
6. To configure the SQL Server of an Administration Service as a Subscriber to a Publisher, specify the corresponding Administration Service in the **Connect to Administration Service** dialog.

   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. For more information, 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.

   Choose one of these options:

   - **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 you select this option, 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.
7. 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 more information, see *Replication Agent permissions* in the *Active Roles Quick Start Guide*.

8. Click **Next**, and the click **Finish**.

> NOTE: Consider the following when adding a replication partner to a replication group:
>
> - 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.
> - 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

Consider the following when removing members from a replication group:

- 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.
- 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. For more information, see “Replication configuration permissions” in the Active Roles Quick Start Guide.

Removing Subscribers from a replication gorup

To remove Subscribers from a replication group

1. Connect to the Publisher Administration Service.

   For more information on how to connect to the Administration Service, see Connecting to the 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**.

Removing the Publisher from a replication group

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: The **Demote** command is not displayed unless the Publisher is the only member of the replication group.

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.

To 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 for the replication partner and navigate to the **Replication Status** tab.

For more information on how to connect to the Administration Service, see Connecting to the Administration Service.

Once connected to the Administration Service, perform the following steps to open the **Properties** dialog for a replication partner.

To open the Properties dialog for a replication partner

1. In the Console tree, expand **Configuration** > **Server Configuration**, and click **Configuration Databases**.
2. In the Details pane, right-click the replication partner, and click **Properties**.

The **Replication Status** tab in the **Properties** dialog 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 displays ⊗ next to **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 *Active Roles Replication: Best Practices and Troubleshooting*.

Always On Availability Groups

To improve the availability of the Active Roles Administration Service, you can use Always On Availability Groups introduced in Microsoft SQL Server 2012. With Always On Availability Groups, 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: a primary 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 ensures the 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 more information about Always On Availability Groups, see AlwaysOn Availability Groups (SQL Server) in the *Microsoft SQL documentation*.

Configuring AlwaysOn Availability Groups in Active Roles

If you have the Active Roles Administration Service installed, you can configure it to use a database belonging to an Always On availability group (also called an availability database). When configuring Active Roles, you must store the Management History data and Configuration data in separate databases. Each of the two databases (or both) can belong to an availability group.

> NOTE: Active Roles does not support the replication of availability databases. Therefore, if the Administration Service is configured to use an availability database (either for the Management History Database or for the Configuration Database), then the data of that database cannot be replicated.

For more information on how to install and configure the Administration Service, see the *Active Roles Quick Start Guide*.

By using the availability group 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 the SQL Server that hosts the primary replica. The listener also enables support for failover redirection. This means that in case of a failover, the listener automatically redirects the Administration Service connection to the new primary replica.

Prerequisites

- The Active Roles database is added to an Always On availability group on the SQL Server.

  For instructions on how to configure an availability group, and how to add a database to an availability group, see Getting Started with Always On Availability Groups (SQL Server) in the *Microsoft SQL documentation*.
- Active Roles replication is not configured for the Configuration data and the Management History data.

To configure the Active Roles Administration Service to connect to the database via the availability group listener

1. Start the Active Roles Configuration Center on the computer running the Administration Service, or connect the Active Roles Configuration Center to that computer.
2. On the Active Roles Configuration Center **Dashboard**, in **Administration Service**, click **Manage Settings**.

   The **Connection to Database** page opens.
3. To modify the database connection of the Administration Service, in **Connection to Database** > **Active Roles databases**, click **Change**.
4. If either or both of the databases belong to an availability group in your Active Roles environment, specify the availability group listener. Otherwise, do not change the value of **SQL Server**.

   a. If the Configuration database belongs to an availability group, enter the DNS host name and, optionally, the TCP port of the listener of that availability group in **Connection to Database** > **SQL Server**.

   b. If the Management History database belongs to an availability group, enter the DNS host name and, optionally, the TCP port of the listener of that availability group in **Connection to Management History Database** > **SQL Server**.

   The value of **SQL Server** must be identical to the DNS host name and, optionally, the TCP port of the listener of the availability group to which the database belongs.

Example: Specifying the availability group listener in the SQL Server

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 can omit the port number in case of the default port, `1433`.

5. Click **Next**.
6. To complete the configuration, follow the instructions of the wizard.

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 in the SQL Server product documentation.

> 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

This section is based on the assumption 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\8.1.3\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\8.1.3\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, for example, `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 **Restart** at the top of the **Administration Service** page in the Configuration Center main window. For more information on how to run Configuration Center, see Running Configuration Center.

To view the mirroring status of the Configuration or Management History database that is used by a particular instance of the Administration Service in the Active Roles Console

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, click the **Configuration Database** or **Management History Database** tab, and view 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 Mirroring States.

   If no information is displayed in the **Database Mirroring** area, it means that the 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 for the object representing that database in the **Configuration/Server Configuration/Configuration Databases** or **Configuration/Server Configuration/Management History Databases** container, respectively.

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 in Troubleshooting replication failures.

The following table lists the values that Active Roles assigns to certain replication parameters.

Table 119: 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 00:00 at the Publisher.

To view the replication parameters using SQL Server Management Studio,

> ⚠ **CAUTION: Do not change these settings. Replication might not work correctly if you manually modify replication settings with the use of SQL Server tools.**

1. **Start Management Studio and connect to the Publisher SQL Server:**
   a. In Object Explorer, click **Connect**, and then click **Database Engine**.
   b. Complete the **Connect to Server** dialog to connect to the instance of the SQL Server Database Engine that holds the Publisher role.
2. **Open the Publication Properties dialog:**
   a. In Object Explorer, under the Publisher SQL Server, expand **Replication** > **Local Publications**.
   b. In Object Explorer, under **Local Publications**, right-click **AelitaReplica**, and click **Properties**.

   In the **Publication Properties** dialog, you can review the Active Roles publication settings.
3. **Open the Subscription Properties dialog:**
   a. In Object Explorer, under **Local Publications**, expand **AelitaReplica**.
   b. In Object Explorer, under **AelitaReplica**, right-click a Subscription, and click **Properties**.

   In the **Subscription Properties** dialog, you can review the Active Roles subscription settings.

Replication Agent schedule

By default, Active Roles schedules the Replication Agents to run at the Publisher as follows:

- The Snapshot Agent starts every day at 00:00.
- The Merge Agents start automatically when SQL Server Agent starts, and runs continuously.

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. To view the Replication Agent schedule settings in detail, click **Edit**.

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. To view the Replication Agent schedule settings in detail, click **Edit**.

Identifying replication-related problems

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 that is running the Publisher SQL Server, and then use SQL Server Management Studio to get more information on that failure.

To ensure that the SQL Server Agent service is started on the computer that is running the Publisher SQL Server

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, under **Agents and jobs related to this publication**, look for the ⊗ icon. 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, read 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, in the list of subscriptions, look for the ⊗ icon. 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**.

For more information on typical errors and how to resolve them, see Troubleshooting replication failures.

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. The security credentials of the Replication Agents depend on the authentication mode of the Administration Service. To determine which authentication mode is actually used:

Windows authentication or SQL Server authentication, view the Administration Service database connection settings.

To 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 view, and click **Properties**.
3. In the **Properties** dialog, go to the **Configuration Database** tab.

The **Configuration Database** tab displays the following information:

- **SQL Server**: Identifies the SQL Server instance that is used by the Administration Service.
- **Database**: The name of the Administration Service database.
- **Use Windows authentication**: When this is selected, it indicates that the Administration Service uses Windows authentication mode when connecting to SQL Server.
- **Use SQL Server authentication**: When this is selected, it 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. This only applies to the **Use SQL Server authentication** option.

Modifying database connection settings

You might have 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 Modifying Replication Agent credentials.

You can modify connection settings by using Active Roles Configuration Center.

To modify the connection settings

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 8.1.3 Configuration Center** on the **Apps** page or **Start** menu, depending upon the version of your Windows operating system. For more information, see Running Configuration Center.
2. In the Configuration Center main window on the **Dashboard** page, in the **Administration Service** area, click **Manage Settings**.

3. On the **Administration Service** page that opens, in the **Active Roles database** area, click **Change**.
4. To view or change the login or password of the Administration Service for SQL Server authentication, use the **Change Active Roles Database** wizard that appears: 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*.

Because of the requirements of the service account, you might have to specify a different service account with sufficient SQL Server permissions. Also, you might have to change the service account's password.

To view or change the service account by using Active Roles Configuration Center

1. Start the Configuration Center on the computer running the Administration Service, or connect the Configuration Center to that computer.

   You can start Configuration Center by selecting **Active Roles 8.1.3 Configuration Center** on the **Apps** page or **Start** menu, depending upon the version of your Windows operating system. For more information, see Running Configuration Center.
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, the SQL Server Agent logon account on the Publisher SQL Server must have appropriate access permissions on the Subscriber SQL Server. For details, see the "SQL Server permissions" section in the *Active Roles Quick Start Guide*.

Because of these requirements of the SQL Server Agent logon account, there might be a case where you have to specify a different logon account with sufficient access permissions. You might also have to change the password of the logon account. This section provides instructions on how to change the SQL Server Agent logon account.

To 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 that are 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 that is running on the PublisherSQL Server computer.
- **SQL Server authentication**: In this mode, Replication Agents use the credentials of the SQL Server login that is specified for the Administration Service connection to SQL Server.

The following sections describe 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. For more information, 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.

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, 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.

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 ... in that entry to display the **Publisher Properties** dialog.
5. In the **Publisher Properties** dialog, 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:

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 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 toSQL Server in the security context of the SQL Server login that is specified for the Administration Service connection to SQL Server.

If the login does not have sufficient rights, it has been deleted, or had the password changed, the Active Roles replication fails. To resolve this problem, do the following:

To avoid replication fail

1. Choose an SQL Server login with sufficient rights. For more information, see the "SQL Server permissions" section in the *Active Roles Quick Start Guide*.
2. Configure the Administration Service to use that login. For more information, see Viewing database connection settings.
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:

To configure the Replication Agent to connect to the Publisher

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 ... in that entry to display the **Publisher Properties** dialog.
5. In the **Agent Connection to the Publisher** area, click **Login**, and type the login name.
6. Click **Password**, and then click ... in the **Password** entry.
7. In the **Enter Password** dialog, type and confirm by retyping the password of that login.
8. To close the **Enter Password** dialog, click **OK**.
9. To close the **Publisher Properties** dialog, click **OK**.

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:

To configure the Replication Agent to connect to the Subscriber

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 ... in the **Subscriber Connection** entry.

   This displays the **Enter Connection Information** dialog.
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. To close the **Enter Connection Information** dialog, click **OK**.
9. To close the **Subscription Properties** dialog, click **OK**.

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 might 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. Because 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. The increased volume of requests 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 which has a database server currently holding the Publisher role (Publisher Administration Service).

NOTE: The Publisher Administration Service must be up and running. If the Publisher is unavailable, you can assign the Publisher role to a different SQL Server. For more information, see Promoting an SQL Server to Publisher.

To connect to the Publisher Administration Service via the Active Roles Console

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.

Demoting the Publisher

Use the Active Roles Console to remove all Subscribers and to demote the Publisher as follows.

To remove all Subscribers and demote the Publisher

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 the deletion steps 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.

Promoting an SQL Server to Publisher

After demoting the previous Publisher, 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 using the Active Roles Console

1. Right-click the Console tree root, click **Connect**, and then select the Administration Service the SQL Server of which you want to hold the Publisher role.
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. To configure the SQL Server of an Administration Service as a Subscriber to this Publisher, specify the corresponding Administration Service in the **Connect to Administration Service** dialog. Click **OK**.
10. In the **New Replication Partner** wizard, click **Next**, click **Next**, and then click **Finish**.
11. Repeat the steps for adding a replication partner for each SQL Server you want to make a Subscriber.

Recovering replication if the 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.

Removing Subscribers if the Publisher is not available

Given that the Publisher is unavailable, you can remove a Subscriber from the replication group by using the Active Roles Console.

To remove a Subscriber from a replication group using the Active Roles Console

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**.

Promoting an SQL Server to Publisher

After demoting the previous Publisher, 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 using the Active Roles Console

1. Right-click the Console tree root, click **Connect**, and then select the Administration Service the SQL Server of which you want to hold the Publisher role.
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. To configure the SQL Server of an Administration Service as a Subscriber to this Publisher, specify the corresponding Administration Service in the **Connect to**

Administration Service dialog. Click **OK**.

10. In the **New Replication Partner** wizard, click **Next**, click **Next**, and then click **Finish**.
11. Repeat the steps for adding a replication partner 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. For more information, see Monitoring replication.

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

To 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** button is disabled, it means that the SQL Server Agent service is already started.

To 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 is scheduled correctly at the Publisher. The Merge Agents must be configured to run continuously. The Snapshot Agent must be configured to start daily at 00:00. For more information, see Replication Agent schedule.

Replication Agent authentication problems

The following section describes the symptoms and solutions for Replication Agent authentication problems.

Symptoms

Replication fails with one of the following errors on the Snapshot Agent or Merge Agent (for more information, see Identifying replication-related problems):

- `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 120: 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). For more information on impersonating the SQL Server Agent account, see .
	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

Server role	Authentication mode	Replication Agent credentials
		computer running the Publisher SQL Server (trusted connection). For more information on impersonating the SQL Server Agent account, see Adding members to a replication group.
	SQL Server Authentication	SQL Server login and password that the Subscriber Administration Service uses to connect to its SQL Server

For more 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.

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 is usually caused by one of the following reasons:

- Incorrect server name. The computer that is running SQL Server has been 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, run 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`**, run 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

  In this syntax, `computername` is the (short) NetBIOS name of the computer running SQL Server.

- `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.

Using regular expressions

When configuring search filter conditions or property validation criteria, you may need to use regular expressions. This section helps you learn 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 121: 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.
{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

Character	Definition
	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 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].

Character	Definition
`\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 122: 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
`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 123: Order of precedence

Character	Description
`\`	Escape
`()`, `[]`	Parentheses and Brackets
`*`, `+`, `?`, `{n}`, `{n,}`, `{n,m}`	Quantifiers
`^`, `$`, `\anymetacharacter`	Anchors and Sequences
|	Alteration

Administrative Template

The Active Roles Administrative Template allows you to control the behavior and appearance of the Active Roles Console by using Group Policy. For more information, 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's Administration Service instances for auto-connect. For more information, see Administration Service auto-connect settings.

Active Roles snap-in settings

With the Active Roles Snap-in policy settings you can:

- Hide some areas of the user interface with the Console.
- 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 124: Policy settings to control the behavior and appearance of the Active Roles Console

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 mode	Specifies view mode in which the Active Roles Console will start. If you enable this policy, you can select view mode from a list. When started,

Policy Setting	Explanation
	the Active Roles Console will switch to the view mode you have selected. By default, users are allowed to change view mode by using the **Mode** command on the **View** menu. If you want to enforce a 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 that 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. If you enable this policy, the **Connect as: The following user** option in the Active Roles Console requires the user to enter their password every time when using that option, rather than encrypting and storing the password once it has been entered. NOTE: Saving passwords may introduce a potential security risk.
Disable **Connect as** options	Disables the **Connect as** options in the **Connect to Administration Service** dialog, 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 125: Policies allowing to register extension snap-ins with Active Roles Console

Policy Setting	Explanation
Namespace	Allows you to register extension snap-ins to extend the namespace of the

Policy Setting	Explanation
extensions	Active Roles Console.
Context menu extensions	Allows you to register extension snap-ins to extend a context menu in the Active Roles Console.
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}`.

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’s 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 predefined 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 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’s 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 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 `%systemroot%\sysvol\domain\policies\PolicyDefinitions` folder.
2. Copy the ADML file to the `%systemroot%\sysvol\domain\policies\PolicyDefinitions\en-US` folder.

Create those folders if they do not exist. For more information about ADMX files, see Managing Group Policy ADMX Files Step-by-Step Guide.

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** > **Templates** > **Active Roles Snap-in Settings** or **Computer Configuration** > **Templates** > **Active Roles** > **Administration Service Auto-connect Settings**, then apply the Group Policy object as appropriate.

Configuring federated authentication

Federated authentication (also known as claim-based authentication) allows users to access applications or websites by authenticating them against a certain set of rules, known as "claims". When federated authentication is configured, users are validated across multiple applications, websites or IT systems via authentication tickets or their token.

During claim-based authentication, authorization is performed by acquiring the identity-related information of users both for on-premises and cloud-based products. Based on the predefined claims to identify the users trying to access the applications or websites, a single token is created for each user. This security token is used to identify the user type once the user is successfully identified.

Active Roles supports federated authentication using the WS-Federation protocol, allowing users to access websites or sign in to an application once with the single sign-on option.

Configuring federated authentication settings

To configure the federated authentication settings, configure the **Identity provider configuration**, and set claims in the **Claim editor**.

> 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.

To set identity provider configuration

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

   The **Web Interface** page displays all the Active Roles Web Interface sites that are deployed on the web server running the Active Roles 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 setting is configured.

3. To configure the federated authentication settings, click **Federated**.

4. In **Identity provider configuration**, from the **Identity provider** drop-down, select the security identity provider. The available options are **Azure**, **ADFS**, and **Custom**.

   > NOTE: For the **Custom** identity provider option, Active Roles supports the WS-Federation standard. However, One Identity Support cannot assist with custom

> WS-Federation-related configurations of third-party identity providers. For assistance in configuring Active Roles with a custom WS-Federation-related configuration of a third-party identity provider, contact One Identity Professional Services.

5. From **Options**, select the required additional options.
6. In **Federated metadata URL**, enter a valid URL.

   > 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. From **Options**, if you select the **Token encryption**, you must enter the **Certificate thumbprint** manually. If the **Token encryption** option is not selected, this field is not available.

   > NOTE: You must enter the **Certificate thumbprint** manually. Copying the key and pasting in the field is not supported.

10. In the **Realm** field, enter the realm URL of the requesting realm.
11. In the **Reply URL** field, enter the URL to send a response. 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 **Claim editor**, to add claims, click **Add**.

   The **Add claim** window is displayed.

2. From the **Claim type** drop-down, select the type of claim.

   > IMPORTANT: Active Roles supports **UPN**, **EMAIL**, **SID**, and **GUID** claims.

3. Select the **Claim value**.
4. In the **Display name** field, enter a name for the claim.
5. In the **Claim description** field, enter a description.
6. Click **Save**.

   The claim is added successfully.

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

7. Click **Modify**.

   If the operation is completed successfully, a message is displayed.

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

Examples of configuring identity providers

See the following examples of configuring the identity providers when using federated authentication.

> NOTE: For the **Custom** identity provider option, Active Roles supports the WS-Federation standard. However, One Identity Support cannot assist with custom WS-Federation-related configurations of third-party identity providers. For assistance in configuring Active Roles with a custom WS-Federation-related configuration of a third-party identity provider, contact One Identity Professional Services.

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/`

29

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 How to restrict Active Directory RPC traffic to a specific port.
- 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 email 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 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 Active Roles Web Interface

To access the Active Roles Web Interface through a firewall, 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).

30

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 Microsoft 365 using Graph API calls or cmdlet calls. Azure users or guest users are typically created 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.

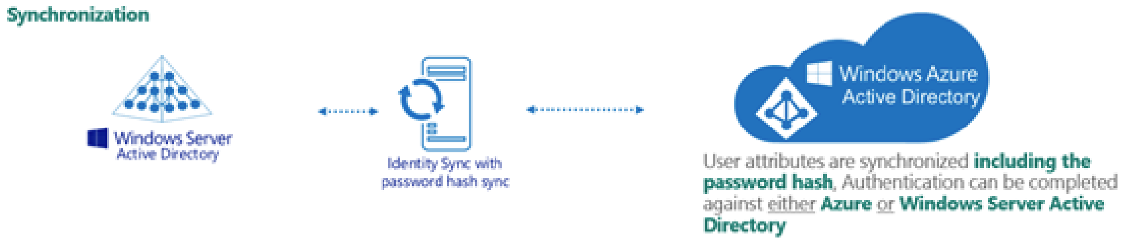
The on-premises domain is not registered in Azure. The Azure user is created in Active Roles with the ID of **user001@test.local** and in Azure as **user001@ARSAzure.onmicrosoft.com**. The user is created in Azure simultaneously when it is created in Active Roles using a Graph API call.

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 171: Synchronized identity configuration


Synchronization
Windows Server Active Directory
Identity Sync with password hash sync
Windows Azure Active Directory
User attributes are synchronized including the password hash, Authentication can be completed against either Azure or Windows Server Active Directory

Example: Synchronized identity configuration

A synchronized identity environment may have the following settings:

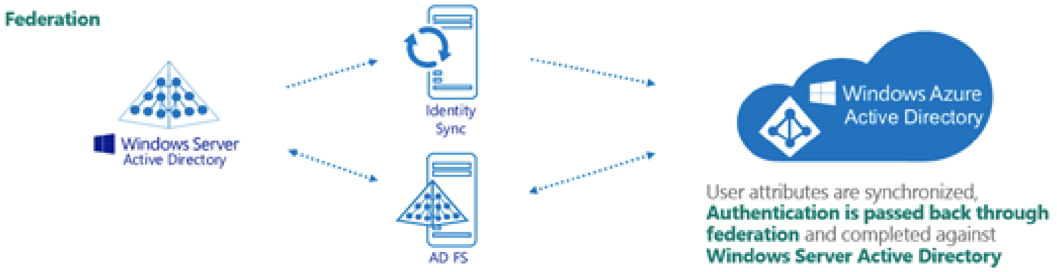
- On-premises domain: `test.local`
- Azure AD Domain: `rd4.qsftdemo.com`
- Azure AD Connect is configured for synchronization.

The on-premises domain is optionally registered in Azure. The Azure 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. Azure users and guest users are typically created with the **onmicrosoft.com** UPN suffix of the selected on-premises domain.

Figure 172: Federated configuration


Federation
Windows Server Active Directory
Identity Sync
AD FS
Windows Azure Active Directory
User attributes are synchronized, Authentication is passed back through federation and completed against Windows Server Active Directory

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 Azure 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 Azure object operations and methods in various Azure environments using the Active Roles Web Interface. Active Roles supports Non-federated, Federated and Synchronized Identity environments.

You can select the Azure environment configuration type in the Active Roles Configuration Center when creating the Azure tenant, as described in Configuring a new Azure tenant and consenting Active Roles as an Azure application. You can modify the configuration type later by changing the Azure properties of the tenant.

Active Roles identifies the environment based on the Azure tenant type and applies the changes accordingly.

Azure object management in a Non-Federated environment

A Non-federated environment is typically used for testing purposes. In a Non-federated environment, most of the Azure properties can be modified, with the exception of attributes that uniquely identify the object (such as `UserPrincipalName` and `ObjectId`).

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 126: Supported Azure configurations comparison chart

Object	Operation	Non-Federated : Method
User	Create	Using Graph API
	Read	Using Graph API and Exchange Online cmdlets
	Update	Using Graph API and Exchange Online cmdlets
	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 cmdlets
	Read	Using Graph API
	Update	Using Graph API
	Delete	Using Graph API
Distribution Group	Create	Using Exchange Online cmdlets
	Read	Using Graph API

Object	Operation	Non-Federated : Method
	Update	Using Graph API
	Delete	Using Graph API
Native Microsoft 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 cmdlets
	Read	Using Graph API
	Update	Using Exchange Online cmdlets
	Delete	Using Graph API

NOTE: Active Roles provides cloud-only support only for Native Microsoft 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 Microsoft 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 127: 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 cmdlets
	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 synchronized 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 synchronized to Active Roles. • Synced using AAD Connect.

Object	Operation	Commands	Tabs	Federated/Synchronized : Method
	Read			Using Graph API
	Update			Synced using AAD Connect
	Delete			Using Graph API
Distribution Group	Create			• Created in Azure. • Back synchronized to Active Roles. • Synced using AAD Connect.
	Read			Using Graph API
	Update			Synced using AAD Connect
	Delete			Using Graph API
Native Microsoft 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 Microsoft 365 Group management.
- "Synced using AAD Connect" in the table means that the object operation is initially performed on the on-premises object. Once the Microsoft Azure AD Connect synchronization cycle is completed, the object is updated in Azure AD or Microsoft 365.
- For more information on how to perform back synchronization, see Active Roles configuration to synchronize existing Azure AD objects to Active Roles.

31

Integrating Active Roles with other products and services

You can integrate Active Roles with several One Identity, Quest and third-party products or services to complement and extend identity and access management in your organization.

Supported One Identity and Quest products

The supported One Identity and Quest products include the following:

- Change Auditor
- Defender
- Enterprise Reporter
- Identity Manager
- Recovery Manager for Active Directory
- Safeguard
- Safeguard Authentication Services
- Starling

For more information on these products, see Active Roles integration with other One Identity and Quest products.

Supported third-party services

The supported third-party services include Duo and Okta.

- For more information on Duo integration, see Active Roles integration with Duo.
- For more information on Okta integration, see Active Roles integration with Okta.

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.

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 Connect service.

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.

Active Roles integration with Duo

Active Roles can be integrated with Duo to complement and extend identity and access management. For more information about Duo, see https://duo.com.

Starting from Active Roles 7.5.2, the rSTS API Admin Tool is no longer available and supported, so you will need assistance from One Identity Professional Services in configuring Active Roles with Duo. To use Active Roles with Duo, contact One Identity Professional Services. For more information, see https://support.oneidentity.com/active-roles/professional-services.

Active Roles integration with Okta

Active Roles can be integrated with Okta 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 functions such as Single Sign-on (SSO) and Multi-Factor Authentication (MFA), Active Roles integrates with the Okta identity management service through Federated Authentication. 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. Okta supports a broad range of authentication methods, including software, hardware, and mobile-based solutions.

By enabling this integration with Okta, Active Roles can use your users' Okta accounts to authenticate them when accessing the Active Roles

Web Interface. To enable this functionality with Active Roles, you need to configure it using the Federated Authentication login method in the Active Roles Configuration Center. The MFA functionality is an additional configuration that you need to perform in the Okta Admin Console.

Configuring the Active Roles application in Okta

From version 7.5.2, Active Roles can be integrated with Okta, a cloud-based identity service offering identity, authentication, and access control functions as a service to complement and extend identity and access management.

To configure the Active Roles application in Okta, follow these steps.

To configure the Active Roles application in Okta

1. Log into the Okta Admin Console.
2. Navigate to **Applications** > **Applications**.
3. Click **Browse App Catalog**.
4. Find and select `Template WS-Fed`.
5. Click **Add**.
6. Enter and set the following:
    a. **Application label**: Enter a label for the Okta application.
    b. **Web Application URL**: Enter the URL for the Active Roles Web Interface, for example, `https://localhost/arwebadmin`.
    c. **ReplyTo URL**: Enter the same URL that you entered as the **Web Application URL** value.
    d. **Name ID Format**: Enter `Persistent`.

e. **Audience Restriction**: Temporarily enter the same value that you entered as the **Web Application URL** value. This will be updated.

f. **Custom Attribute Statements**: Enter `http://schemas.xmlsoap.org/ws/2005/05/identity/claims/email|${user.email}|`.

7. Click **Done**.
8. Click **General**.
9. Copy the value from **Realm**.
10. Click **Edit**.
11. Paste the **Realm** value as the **Audience Restriction** value.
12. Click **Save**.
13. Click **Sign On**.
14. To open a new tab with information needed to configure WS-Federation in Configuring Okta in the Active Roles Configuration Center, click **View Setup Instructions**.

Configuring Okta in the Active Roles Configuration Center

From version 7.5.2, Active Roles can be integrated with Okta, a cloud-based identity service offering identity, authentication, and access control functions as a service to complement and extend identity and access management.

To configure Okta in the Active Roles Configuration Center, follow these steps.

Prerequisites

Before you can configure Okta in the Active Roles Configuration Center, you must configure the Active Roles application in Okta. For more information, see Configuring the Active Roles application in Okta.

To configure Okta in the Active Roles Configuration Center

1. In the Active Roles Configuration Center, navigate to **Web Interface** > **Authentication**.
2. In the **Site authentication settings** window, select the **Federated** tab.
3. In the **Identity provider configuration** tab that you opened in Step 14 of Configuring the Active Roles application in Okta, configure the settings of the identity provider.

   a. From **Identity provider**, select `Custom`.

   b. In **Okta Setup Instructions**, copy the **Public Link** URL.

   c. In the Active Roles Configuration Center, paste it into the **Federated metadata URL**.

   d. To validate the metadata, click **Test metadata**.

   e. To close the prompt about opening the XML file in a web browser, click **No**.

4. In the **Okta Setup Instructions** tab that you opened in Step 14 of Configuring the Active Roles application in Okta, copy the **Realm (APP ID URL)** value.

5. In the Active Roles Configuration Center, paste the **Realm (APP ID URL)** value as the **Realm** value.

6. In **Reply URL**, enter the same value that you entered as the **Web Application URL** value in Step 6 of Configuring the Active Roles application in Okta.

7. In **Claim editor**, click **Add** to open the **Add claim** window, and enter or select the following.

   a. **Claim Type**: Based on the values of the local AD objects, select `UPN` or `EMAIL`.

   > NOTE: The UPN or the email address of the local AD objects must match the email value of the Okta objects.

   b. **Claim value**: Select `http://schemas.xmlsoap.org/ws/2005/05/identity/claims/email`.

   c. **Display name**: Enter the display name in `user.email` format.

   d. **Description**: Enter any description (this is typically the value the user logged in with).

   e. Click **Save**.

8. Click **Domain user login credentials**.

9. To access the local domain, enter the **Username** in `domain/username` format, and the **Password**.

10. Click **Modify**.

Active Roles Language Pack

Available for download from the One Identity Support Portal, the Active Roles Language Pack provides product localization for Active Roles. To enable localization, install the Language Pack on the machine(s) running the Active Roles Administration Service, Configuration Center, Console, Synchronization Service or Web Interface components.

> NOTE: You can install the Active Roles Language Pack on 64-bit operating systems only.

To install the Active Roles Language Pack

1. From the One Identity Support Portal, download the Language Pack applicable to your Active Roles release. For more information on the system requirements, see the *Active Roles Release Notes*.
2. Open the ISO or extract the ZIP archive, and run `x64\ActiveRolesLanguagePack.msi`.
3. Follow the instructions of the installer.
4. After the Language Pack is installed, restart the Active Roles Administration Service. To restart the Administration Service, open the Configuration Center, click **Administration Service** on the left pane, then either click **Restart**, or first click **Stop** and then **Start**.
5. Reset Internet Information Services (IIS) for the Active Roles Web Interface. To do so, open the Windows Command Prompt, and run the **iisreset** command.

Supported languages

The Active Roles Language Pack supports the following languages:

- For the Active Roles Administration Service, Configuration Center, Console and Synchronization Service components, the Language Pack provides support for German language.
- For the Active Roles Web Interface component, the Language Pack provides support for the following languages:
  - Chinese (Simplified and Traditional)
  - French

- German
- Portuguese (Brazilian and European)
- Spanish

Localization limitations

For Active Roles 8.1.3, the Active Roles Language Pack has the following limitations:

- The Active Roles documentation and the built-in help files are not translated, and are available only in English.
- If you use a Windows operating system set to German language on the machine running the Active Roles Administration Service, Configuration Center, Console and Synchronization Service components, installing the Language Pack on that machine will switch the language of these components to German. If Windows is set to any other language, these Active Roles components will default to English.

  > TIP: To change the language of these components manually to English or German, update their **Language** value in the Windows registry. For more information, see Modifying the language of Active Roles components in the Windows registry.
- As the Administration Service will use the German localization on an operating system set to German language after installing the Language Pack, the Active Roles Web Interface will also show errors, messages and reports originating from the Administration Service component in German, regardless of the language selected for the Web Interface.

Modifying the language of Active Roles components in the Windows registry

When the Active Roles Language Pack is installed on the machine(s) running the various Active Roles components, the configured language for the Active Roles Administration Service, Configuration Center, Console and Synchronization Service is set within the Windows registry.

If you want to change the configured language of any of these components (for example, because of localization testing), modify the language value in the applicable registry entry of the respective Active Roles component.

To change the language of an Active Roles component in the Windows registry

1. Open the Windows Registry Editor.
2. Navigate to the language setting of the Active Roles component that you want to modify:
   - Administration Service: `HKEY_LOCAL_MACHINE\SOFTWARE\One Identity\Active Roles\Configuration\Service\Language`
   - Configuration Center: `HKEY_LOCAL_MACHINE\SOFTWARE\One Identity\Active Roles\Configuration\ConfigCenter\Language`
   - Console: `HKEY_LOCAL_MACHINE\SOFTWARE\One Identity\Active Roles\Configuration\Console\Language`
   - Synchronization Service: `HKEY_LOCAL_MACHINE\SOFTWARE\One Identity\Active Roles\Configuration\SyncService\Language`
3. Double-click the **Language** key, then under **Value data**, modify the string value.

   > TIP: The configured language is indicated with an ISO 639 language code and ISO 3166 subculture code pair. For example, German (Germany) is specified as `de-DE`, while English (United States) is specified as `en-US`.
4. (Optional) If you changed the language of the Administration Service, restart it to apply your changes. To restart the Administration Service, open the Configuration Center, click **Administration Service** on the left pane, then either click **Restart**, or first click **Stop** and then **Start**.
5. (Optional) If you changed the language of the Active Roles Console, the Configuration Center, or the Synchronization Service, close and reopen them to apply your changes.

Modifying the language of the Web Interface

When the Active Roles Language Pack is installed on the machine running the Active Roles Web Interface, you can change its language via the user interface settings.

To modify the language of the Active Roles Web Interface

1. In the Active Roles Web Interface header, click **Active Roles 8.1.3** > **Settings**.
2. Under **User interface language**, select the language you want to use.
3. To apply your change, click **Save**.

33

Active Roles Diagnostic Tools

The Active Roles Diagnostic Tools package provides optional tools for checking system requirements, logs and changes in your Active Directory domain. The package contains the following tools:

- **System Checker**: Checks your computer, SQL Server, and Active Directory domains to see if you are ready to deploy Active Roles. For more information on using the tool, see *Using System Checker* in the *Active Roles Feature Guide*.
- **Log Viewer**: Examines Active Roles diagnostic logs and event logs, and helps finding Knowledge Base Articles that may help you resolve errors.
- **Directory Changes Monitor**: Gets the statistics of directory change operations that occurred in a particular Active Directory domain.

For more information on installing the Active Roles Diagnostic Tools package, see *Steps to install Diagnostic Tools* in the *Active Roles Quick Start Guide*.

Using the System Checker

To check if a computer and your organization environment supports installing Active Roles, use the Active Roles System Checker tool.

To check system readiness with Active Roles System Checker

1. From your operating system, launch Active Roles System Checker.
2. Select **Computer** > **System Readiness Checks**.
3. To check the computer specifications, in the **Confirm System Readiness Checks** window, select the appropriate components to check for and click **Check**.
4. In the **System Readiness Checks** window, review the summary and confirm that the computer has passed the required checks. Take appropriate action before installing Active Roles. For example, if there is a warning about insufficient memory (RAM), then upgrade the computer's memory to the recommended amount.

5. To check the SQL Server requirements of Active Roles, in the Active Roles System Checker, select **Environment** > **SQL Server Checks**.
6. Enter the SQL Server name and appropriate credentials for the Active Roles service account, then click **Check**.
7. Review the summary to confirm that the SQL Server passed the checks.
8. To check the Active Directory requirements, in the Active Roles System Checker, select **Environment** > **Active Directory Checks**. Enter the domain controller (DC) name and the appropriate credentials for the Active Roles service account, then click **Check**.

   A progress window appears. Once the check completes, System Checker shows the summary.
9. Review the summary to confirm the account has adequate permissions in Active Directory.
10. To learn more about Active Roles. click **Additional Resources**. To finish the check, click **Finish**.

Using the Log Viewer tool

You can run the **Active Roles Log Viewer** application from the **Start** menu or from the **Apps** page, depending on the version of your Windows operating system. Alternatively, you can also start the application by navigating to its `.exe` file in the installation folder:

```
\Active Roles Diagnostic Tools\Log Viewer\arlogviewer.exe
```

To open a log in Log Viewer

1. In the Active Roles Log Viewer application, click **Open**.
2. Browse to the diagnostic log file or saved event log file you want to open.
3. Select the file, and click **Open**.

By default, Log Viewer displays a list of errors encountered by the Administration Service and recorded in the log file. To analyze these errors further, and look for information about them, right-click an error in the list, then click **Look for solution in Knowledge Base**. Log Viewer then searches the One Identity Knowledge Base to list the Knowledge Base Articles related to the error you selected.

Additional tasks in Log Viewer

Besides opening and troubleshooting logs, you can also perform the following tasks in Log Viewer:

- 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, then 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**.

  > NOTE: Stack tracing is not available for event log files.

- 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.

  > NOTE: Stack tracing is not available for event log files.

- 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.

Log file size

The logs grow in size quickly. Therefore, One Identity recommends to enable logging only when attempting to reproduce an issue, and disable it immediately once reproduction is successful.

The log file captures every activity performed by the service, including the tasks performed by connected users while debug logging is enabled.

> TIP: Sometimes, you may need to keep logging enabled for an extended period of time. As the log files are stored on the computer running Active Roles, the logging service requires a substantial amount of free space, which may not be available on the system.
>
> In such cases, to save disk space, set logging to a specific interval and move the logs to another drive or network share. For more information, see Knowledge Base Article How to automate Active Roles debug logging in the One Identity support portal.

Web Interface logs

The Active Roles Web Interface component uses separate log files, named after the configured Web Interface sites. The logs are stored in the following location:

```
C:\Program Files\One Identity\Active Roles\8.1.3\Web\Public\Log
```

> TIP: Similarly to the `ds.log` file, the Web Interface log can grow quickly in size. Therefore, One Identity recommends enabling Web Interface logging only when reproducing an issue.

Using the Directory Changes Monitor command-line interface

Active Roles Directory Changes Monitor is a command-line tool.

To run Directory Changes Monitor

1. Open the Windows command prompt.
2. Navigate to the folder where Directory Changes Monitor is installed.
3. Run the tool with the appropriate parameters.

   > TIP: To list the available parameters, run the following command:
   >
   > **`dirchangesmon.exe /?`**

> NOTE: Directory Changes Monitor has a single required parameter, `/TargetDC`, specifying the Domain Controller (DC) of the Active Directory domain from which to get directory change statistics. To retrieve information from a domain, make sure to run Directory Changes Monitor with a domain user account of that domain, or from a trusted domain.

Active Roles Add-on Manager

Active Roles Add-on Manager is an application for installing and managing add-ons for Active Roles. You can also create new addons with the solution's Add-on Editor.

For more information on installing Add-on Manager, see *Steps to install Add-on Manager* in the *Active Roles Quick Start Guide*.

Using the Add-on Manager command-line interface

You can use the Active Roles Add-on Manager from the command-line.

To run Add-on Manager

1. Open the Windows command prompt.
2. Navigate to the folder where the Add-on Manager is installed.
3. Run the `ActiveRolesAddOnManager_8.1.3.exe` file with the appropriate parameters.

   > TIP: To list the available parameters, run the following command:
   >
   > **`ActiveRolesAddOnManager_8.1.3.exe /?`**

Creating an add-on

You can create new add-ons with the Add-on Editor component of the Add-on Manager.

To create a new add-on

1. To open the Add-on Editor, perform one of the following steps:
   - In the Windows **Start** menu, click **Add-on Editor**.
   - In the Active Roles Console, navigate to the Add-on Manager application page, and click **Create New**.
2. In the **Connect to Administration Service** dialog, select the Active Roles Administration Service you want the Add-on Editor to connect to, then specify a user name and password.
3. On the **Create or Edit Add-on** page, select **Create a new add-on** and click **Next**.
4. On the **General Add-on Settings** page, configure all settings and click **Next**.
5. On the **Add-on Objects** page, select the Active Roles objects and/or Web Interface customization items you want to include in your add-on.
   - To add Active Roles objects, click **Add Active Roles Objects**. Then, on the **Add Active Roles Objects** page, select the objects to include in your add-on. To apply your selection, click **OK**.
   - To add Web Interface customization items, click **Add WI Customization items**. On the **Add Web Interface Customization Items** page, specify the configuration from which you want to export customization items, then select the items to include in the add-on. To apply your selection, click **OK**.
6. Once you selected the objects to include in the add-on, click **Next**.
7. On the **Save Add-on** page, specify the file name of your add-on and click **Next**.
8. On the **Ready to Create Add-on** page, review the settings for your add-on. If you want to specify advanced settings, click the **Advanced** button.
9. (Optional) On the **Advanced Settings** page, configure the following settings for your add-on. To apply your advanced settings, click **OK**.
   - **Show in Raw mode only**: If selected, Add-on Manager will display your add-on only if the Active Roles Console is in Raw view mode.
   - **Show Uninstall link**: If selected, Add-on Manager will show the **Uninstall** option for your add-on in the Active Roles Console.

     > NOTE: If this option is not selected, you can only uninstall your add-on later via the Add-on Manager command line.
     >
     > For the list of command-line options, use the `AddOnManager.exe /?` command in the Windows command prompt.

   - **Show Add-on Configuration page**: If selected, the add-on title link in the Add-on Manager page will open the **Add-on Details** dialog.

     > TIP: One Identity recommends selecting this option if your add-on has a configuration page, then configuring the add-on title link to open that configuration page.

> To do so, select **Show Add-on Configuration page**, then supply the Distinguished Name (DN) of the application object included in your add-on.

- **Web Interface customization label**: Contains the Web Interface customization label. All Web Interface customization items added by the configured add-on will be marked in the add-on XML with this label. By default, the label consists of the add-on name and version.
- **Show Web Interface customization link**: If selected, Add-on Manager will provide the **Web Interface Customization** link for your add-on in the Active Roles Console. When clicking this customization link, you can select the Web Interface configurations and sites you want your add-on to customize, then also apply the customization items to the selected configurations and sites.
- **Apply customization to Site for Administrators**: If selected, the customization items of your add-on will be applied to the Active Roles Web Interface site for Administrators.
- **Apply customization to Site for Help Desk**: If selected, the customization items of your add-on will be applied to the Active Roles Web Interface site for Helpdesk.
- **Apply customization to Site for Self-Service**: If selected, the customization items of your add-on will be applied to the Active Roles Web Interface site for Self-Service.
- **System Requirements**: Use this setting to specify the minimum and maximum versions of Active Roles supported by your add-on.
- **Required add-ons**: Use this setting to specify add-ons that must be installed before installing your add-on. To specify a new add-on, click **Add**.
- **Pre-install script**: If selected, you can specify a script to run before installing your add-on. Enter the script in the text box of the setting.

  > NOTE: Add-on Manager supports only PowerShell scripts.

- **Post-install script**: If selected, you can specify a script to run after installing your add-on. Enter the script in the text box of the setting.
- **Pre-uninstall script**: If selected, you can specify a script to run before uninstalling your add-on. Enter the script in the text box of the setting.
- **Post-uninstall script**: If selected, you can specify a script to run after uninstalling your add-on. Enter the script in the text box of the setting.

10. To create the add-on, click **Finish**.

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