Microsoft Active Directory Connector

Version: 8.4

Copyright and Trademark Notices

SailPoint Technologies, Inc. makes no warranty of any kind regarding these materials, or the information included therein, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. SailPoint Technologies shall not be liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Patents Notice. https://www.sailpoint.com/patents

Restricted Rights Legend. All rights are reserved. No part of this document may be published, distributed, reproduced, publicly displayed, used to create derivative works, or translated to another language, without the prior written consent of SailPoint Technologies. The information contained in this document is subject to change without notice.

Use, duplication or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 for DOD agencies, and subparagraphs (c)(1) and (c)(2) of the Commercial Computer Software Restricted Rights clause at FAR 52.227-19 for other agencies.

Regulatory/Export Compliance. The export and re-export of this software is controlled for export purposes by the U.S. Government. By accepting this software and/or documentation, licensee agrees to comply with all U.S. and foreign export laws and regulations as they relate to software and related documentation. Licensee will not export or re-export outside the United States software or documentation, whether directly or indirectly, to any Prohibited Party and will not cause, approve or otherwise intentionally facilitate others in so doing. A Prohibited Party includes: a party in a U.S. embargoed country or country the United States has named as a supporter of international terrorism; a party involved in proliferation; a party identified by the U.S. Government as a Denied Party; a party named on the U.S. Department of Commerce’s Entity List in Supplement No. 4 to 15 C.F.R. § 744; a party prohibited from participation in export or re-export transactions by a U.S. Government General Order; a party listed by the U.S. Government’s Office of Foreign Assets Control as ineligible to participate in transactions subject to U.S. jurisdiction; or any party that licensee knows or has reason to know has violated or plans to violate U.S. or foreign export laws or regulations. Licensee shall ensure that each of its software users complies with U.S. and foreign export laws and regulations as they relate to software and related documentation.

Contents

Integrating SailPoint with Active Directory

Revised Date: 14 September 2023

> **Note**
>
> IdentityIQ Connector information is now available as online help and PDF. The online help describes the latest updates for the connector.
>
> To find documents related to a specific version of IdentityIQ, refer to the Supported Connectors for IdentityIQ page on Compass.
>
> Configuration details for connectors may vary not only by release version but also by patch version. Be sure to refer to the correct documentation for your specific release and patch level.

This document is designed to give specific information about the requirements and field definitions needed to get a working instance of an Active Directory connector in IdentityIQ.

The SailPoint Active Directory connector offers complete management of your Active Directory infrastructure, which can be distributed across multiple domains/multiple forests. You can manage users, contacts, groups, Exchange mailbox, mail users, mail contacts, and Skype users front a single source.

For more information on what you can do with the Active Directory connector, refer to Supported Features.

What's New in 8.4

- Supports aggregation of domain NetBIOSName as part of account and group aggregation. You need to add `NetBIOSName` as a schema attribute with the type as String in the Account and Group schema to leverage this feature.
- Supports Microsoft Windows Server 2022.

Supported Features

The connector supports the following features:

Account Management

Active Directory Users	• Manages Active Directory Users as Accounts • Aggregation, Delta Aggregation, Partitioning • Aggregation, Refresh Account, Pass Through

	• Authentication, Delta Partitioning Aggregation • Create, Update, or Delete • Enable, Disable, Unlock, or Change Password • Add/Remove Entitlements (includes Foreign Security Principals) • Terminal Services, Dial-in Attributes • Create, Update, or Delete Exchange User Mailbox • Create, Update, or Delete Exchange Mail User • Create, Update, or Delete Skype for Business user • Enable or Disable, setting policies for Skype for Business user • Reset Skype for Business user PIN • Password Interception
Active Directory Contacts	• Manages Active Directory Contacts as Accounts • Aggregation, Delta Aggregation, Partitioning Aggregation, or Refresh Account • Create, Update, or Delete • Add or Remove Entitlements • Create, Update, or Delete Exchange Mail Contact
Active Directory Service Accounts (Managed Service Accounts/Group Managed Service Accounts)	• Aggregation, Partitioning Aggregation, or Refresh Account • Create, Update, or Delete • Add or Remove Entitlements

Account - Group Management

- Manages Active Directory Groups as Account-Groups
- Aggregation, Delta Aggregation, Refresh Group
- Create, Update, or Delete
- Create or Delete Exchange Distribution List

Microsoft Exchange Shared Mailbox

Manage Shared Mailbox as Account Groups. For more information, see Microsoft Exchange Shared Mailbox.

Active Directory Resource Forest Exchange Management

For more information, see Active Directory Resource Forest Topology Exchange Management.

Permission Management

- Application can be configured for following unstructured target collectors to read permissions from the following end system:
  Windows File Share: Read Windows File Share permissions directly assigned to accounts and groups.
- Supports automated revocation of the aggregated permissions and creates work items for requests only when the default provisioning action is overridden, and **Manual Work** Item is selected as the provisioning action.

Other

- Restore deleted objects (Active Directory Accounts and Groups) using 'Active Directory Recycle Bin'
- Supports executing native before/after scripts for provisioning requests
- Provides support for Simple Authentication and Security Layer (SASL) when binding to Active Directory
- Active Directory Connector provides support for serverless configuration for better reliability and ease of configuration.
  For more information, see Prerequisites.
- IQService support TLS and client authentication to ensure the channel is secure and IQService is communicating with legit Client (IdentityIQ).
- Supports Auto Partitioning. For more information, see Account and Group Settings.

- Supports reusing of Ticket Granting Tickets (TGT) for Kerberos authentication during aggregation tasks. To revert to the earlier implementation (non-cached) an additional attribute named `adSystemConfUseUpdatedSASLCommunication` can be added to the system configuration.

Supported Managed Systems

Supported Active Directory Domain Services (AD DS) functional levels

- Microsoft Windows Server 2016
- Microsoft Windows Server 2012 R2
- Microsoft Windows Server 2012
- Microsoft Windows Server 2008 R2

> **Note**
>
> Microsoft has not made any changes in the functional level for Active Directory in Microsoft Windows Server 2019 and 2022.
>
> As a result, Microsoft Windows Server 2016 is the latest Active Directory functional level supported.

Supported Microsoft Exchange Servers

- Microsoft Exchange Server 2019
- Microsoft Exchange Server 2016
- Microsoft Exchange Server 2013

Supported Microsoft Lync\Skype for Business Servers

- Microsoft Skype for Business Server 2019
- Microsoft Skype for Business Server 2015
- Microsoft Lync Server 2013

Prerequisites

- Create an Active Directory service account with the required permissions. A service account is a special user account that is created for the sole purpose of running a particular service or application on the Windows operating system. Services use the service accounts to log on and interact with the operating system.
- Before you start using the connector, install and register IQService on any Windows system with any of the supported Operating Systems. For more information on installing and registering IQService, see IQService.
  - If the **Authentication Type** is set to **Strong**, then the IQService host must be in the same domain or in a trusted domain.
  - For managing Terminal Services (Remote Desktop Services profile) attributes, install the IQService on a server class Windows Operating System.
- Secure Active Directory connector.
- For an application managing multiple domain trees, either from same or different forests, there must be two-way trust relationship between them.
- For managing Managed Service Accounts (MSA) or Group Managed Service Accounts (GMSA), the following prerequisites are required:
  - For reading `msDS-GroupMSAMembership` and `msDS-AllowedToActOnBehalfOfOtherIdentity` GMSA object properties, IQService is required and Active Directory Module for Windows PowerShell must be enabled on the IQService Host.
  - For Provisioning operations of MSA and GMSA objects, IQService is required and Active Directory Module for Windows PowerShell must be enabled on the IQService Host.
- For access to the Active Directory domain controller, you must whitelist/allow list the following URL `<org>.api.identitynow.com` or you must implement some means of proxy server to allow the connection.

Required Permissions

Service Account Permissions

A service account is a special user account that is created for the sole purpose of running a particular service or application on the Windows operating system. Services use the service accounts to log on and interact with the operating system. The service account must have appropriate permissions on Active Directory. The Domain Controller must be accessible from the IQService host computer.

> **Note**
> The permissions discussed in the following section grant limited account creation privileges to a user. This user can create and modify most accounts. It cannot manage the administrator user account, the user accounts of administrators, the server operators, account operators, backup operators, and print operators. To manage these user types, you must assign the appropriate security permissions or add the user to groups having higher permissions. For example, domain administrators.

The service account specified in the application must be the member of the Account Operators group.

More granular permissions can be assigned to users for specific portions of the directory, but this is discouraged by Microsoft best practices for Active Directory access control. The required permissions depend on the use cases that are implemented, but could include:

Operations	Service Account Permissions
Load Accounts	• Read All Properties • Read Members
Provision Accounts	• Write All Properties • Write Members • Create User Objects
Password Management	• Change Password • Reset Password

Permissions for Special Operations

Some special operations need additional permissions.

Operations	Service Account Permissions
Delta Aggregation	Additional permissions are required for "Replicating directory changes" and "Read permissions on the Deleted Object Container". For more information, refer to Required Permissions for Delta Aggregation.
Provision Exchange Mailbox	Must be a member of Exchange Recipient Management group.

Microsoft Skype for Business Server

For the Active Directory connector, there are updated service account permissions to load and provision Microsoft Lync/Skype for Business. One of the following permissions is required, depending on the service account type:

- For Microsoft Skype for Business Server user management, the service account must be a member of the CSUserAdministrator and one of the following domain groups:
  - RTCUniversalServerAdmins
  - Custom group with SQL permission

The required permissions for the Custom group and CSUserAdministrator domain group in SQL are:

Database Instance	Security login	Database Role Membership	Databases
RTCLOCAL	Group required to be added in SQL server: Custom Group and CSUserAdministrator	DB_Owner	RTC, XDS, RTCDYN
RTC	Group required to be added in SQL server: Custom Group and CSUserAdministrator	DB_Owner	RTCXDS, XDS

Permissions for Managing Group Managed Service Accounts (gMSA)

For managing Managed Service Accounts and Group Managed Service Accounts, the following permissions are required:

- Aggregation and Refresh Account: Member of Account Operators group.
- Create, Update, and Delete: In addition to Account Operators, service accounts must have full permission on the Active Directory container from which service account is to be managed.

Strong Authentication (SASL) Permissions

For Strong authentication (SASL), a single service account can be used for multiple domains/forests.

Prerequisites

- The domains must have two-way trust.
- The service account must have delegated permissions across other domains for user, contact, and group objects.

  Permissions must be delegated to the service account. Use the **Delegation Control Wizard** to delegate permissions to the contact.

To delegate permissions using the Delegation Control Wizard, complete the following:

1. Open **Active Directory Users and Computers**.
2. Right-click on the **Domain** and select **Delegate Control to open Delegation of Control Wizard** and then select **Next**.
3. Select the **Add** button to add a service account user and then select **Next**.
4. Select **Create a custom task to delegate** and then select **Next**.
5. Only select the following objects in the folder option: **User Objects**, **Contact Objects**, **Group Objects**, and **Create/Delete the selected objects in the folder**.
6. On the next screen, under **Permissions** select **Full Control**, then select **Next**.
7. Select **Finish**.

Foreign Security Principals Permissions

For Foreign Security Principals (FSP) to be aggregated, created, or modified a single service account must have full delegated permissions on the FSP container. Use the **Delegation Control Wizard** to delegate permissions to the service Account.

To delegate permissions using the Delegation Control Wizard, complete the following:

1. Open **Active Directory Users and Computers**.
2. Right-click on the **ForeignSecurityPrincipals** container, select **Delegate Control**, and then select **Next.**
3. Select the **Add** button to add a service account user and then select **Next**.
4. Select **Create a custom task to delegate** and then select **Next**.
5. Select **This folder, existing objects in this folder and creation of objects in this folder**.
6. On the next screen, select **Full Control under Permissions**, and then select **Next**.
7. Select **Finish**.

Permissions for Managing Contact Objects

For managing contacts, the contacts must be delegated to **Account Operators** group. Use the **Delegation Control Wizard** to delegate permissions to the contact.

To delegate permissions using the Delegation Control Wizard, complete the following:

1. Open **Active Directory Users and Computers**.
2. Right-click on the domain and select **Delegate Control**.
3. Select **Next**.
4. Select **Account Operators** group.
5. Select **Next**.
6. Select **Create a custom task to delegate**.
7. Select **Next**.
8. Select **Only the following objects in the folder** option, then select **Contact Objects** and **Create and Delete selected objects in the folder**.
9. On the next screen, select **Full Control** under **Permissions**, and then select **Next.**
10. Select **Finish**

Required Permissions for Delta Aggregation

You must give the service account additional both of the additional permissions to support delta aggregation.

To provide the additional permissions, complete the following:

1. To provide Replicating directory changes permissions to the service account.
   a. In the **Active Directory Users and Computers** browser menu, select **View**, and then right-click and select the **Advanced features** checkbox.
   b. Right-click the domain node and select **property**, and then open the **Security** tab.
   c. Add the user to the list of **Security Principals**.
   d. Select the user and select the **Allow** checkbox for **Replicating Directory Changes** permission.
2. To provide Read permissions on the Deleted Object Container to the service account.
   a. Log on to any domain controller in the target domain with a user account that is a member of the Domain Administrators group.

b. Open a command prompt. Enter the following command: dsacls "*Deleted objects container DN*" /*takeownership*In the above command, the Deleted objects container DN is the distinguished name of the deleted objects container. For example, `dsacls "CN=Deleted Objects,DC=SailPoint,DC=Com" /takeownership`

c. Press the **Enter** key.

d. To grant Read permission to the objects in the Deleted Objects container to a user type, enter the following command: dsacls "Deleted *objects container DN*" /G domainName\userName:LCRP

In the above command, LCRP stands for the list object and read properties permission. For example, `dsacls "CN=Deleted Objects,DC=SailPoint,DC=Com" /G SailPoint\John:LCRP`

e. Press the **Enter** key.

Managing Shared Mailbox Permissions

The Active Directory connector supports managing **Full Access**, **Send As**, and **Send On Behalf** permissions on the Shared Mailbox.

Permissions can be assigned in the following ways:

Using the `memberOfSharedMailbox`attribute in the provisioning plan for the User Object.

1. Assign **Full Access** and **Send As** permissions to user.

   For example:

```
<ProvisioningPlan>
    <AccountRequest op="Modify">
        <AttributeRequest name="memberOfSharedMailbox" op="Add" >
             <Value>
                <List>
                   <String>DN of the shared Mailbox </String>
                </List>
             </Value>
        </AttributeRequest>
    </AccountRequest>
</ProvisioningPlan>
```

   The above request assigns **Full Access** and **Send As** permissions to the user. These are the default permissions that are assigned if permission names are not provided explicitly in the request. The default permission to be assigned can be changed by using the `defaultSharedMBPermissions` application configuration attribute.

For example:

```
<entry key="defaultSharedMBPermissions" value="fullAccess,sendAs,sendonbehalf"/>
```

This configuration sets default permissions to `fullAccess`, `sendAs`, and `sendOnBehalf`.

2. Assign specific permission to user.

   Add specific permissions (other than the default) by passing additional information in the `AttributeRequest` by using the `sharedMailboxPermission` attribute in the request.

   The following example only assigns the `sendOnBehalf` permission to the user:

```
    <ProvisioningPlan>
      <AccountRequest op="Modify">
           <AttributeRequest name="memberOfSharedMailbox" op="Add" >
           <Attributes>
              <Map>
                  <entry key="sharedMailboxPermission" value="sendOnBehalf" />
              </Map>
            </Attributes>
             <Value>
               <List>
                 <String>DN of the Shared Mailbox</String>
               </List>
            </Value>
       </AttributeRequest>
      </AccountRequest>
    </ProvisioningPlan>
```

> **Note**
> The connector supports only assigning of specific permissions. To remove a specific permission on the Shared Mailbox, update those permissions properties on Shared Mailbox entitlement object.

Assigning Permission by Updating Shared Mailbox Permission Attributes

1. Assign Shared Mailbox permission to Active Directory User or Group by updating the Shared Mailbox `fullAccess`, `sendAs`, and `sendOnbehalf` properties.

   For example, the following plan executes the modify operation on the shared mailbox and assigns `fullAccess` and `sendAs` permission to the user or group:

```
<ObjectRequest application=<Application Name> nativeIdentity=<DN of the Shared
mailbox> op="Modify" type="sharedMailbox">
    <AttributeRequest name="fullAccess" op="Add ">
        <Value>
            <List>
                <String> <DN of the User or Group ></String>
            </List>
        </Value>
    </AttributeRequest>
    <AttributeRequest name="sendAs" op="Add ">
        <Value>
            <List>
                <String> <DN of the User or Group ></String>
            </List>
        </Value>
    </AttributeRequest>
</ObjectRequest>
```

Securing the Active Directory Application

Secure the Active Directory application by using the following communication paths based on the operations performed.

- **IdentityIQ and Active Directory Domain Controller/ Target system**: For read operations *
- **IdentityIQ and IQService**: For provisioning operations **
- **IQService and Active Directory Domain Controller/ Target system**: For provisioning operations **

The asterisk (*) symbols represent:

* IQService is used for read operation for Skype and terminal attributes if defined in schema.

**Out-of-the-box IQService uses a fixed, known default encryption key when IQService is installed. This enables IdentityIQ to communicate with IQService with no specific configuration for encryption being put in place ahead of time, while still providing encryption for the data payload. No data persists on the disk with these keys so observers would have to trace the data in-flight to be able to decrypt any communications. Because of this extremely temporary and transitory nature of the communication stream the risk associated with using default keys here is considered extremely low. The risk can be further reduced by deployment specific keys which can be easily configuring using the IQService public key exchange task.

From this point forward, IdentityIQ and IQService use TLS for encrypting the XML data payload.

> **Note**
> SailPoint recommends securing every communication path for the Active Directory application by following the configurations outlined in the following sections:

Securing Communication Path Between IQService and Active Directory Domain Controller / Target system

For IQService to connect using TLS and self-signed certificates, you must install the certificate in **Trusted Root Certification Authorities** on the IQService host.

To install a certificate, complete the following:

1. Export the server certificate and copy the exported .cer file to the IQService host.
2. Double-click the .cer file, select **Install Certificate**, and select **Next**.
3. Choose **Place all certificates in the following store** and select **Browse**.
4. Select **Show physical stores**.
5. Expand **Trusted Root Certification Authorities** and select **Local Computer**.
6. Select **OK**.
7. Select **Next**, and then **Finish**.

Securing Communication Path Between IdentityIQ and Active Directory Domain Controller/ Target System

To secure a TLS connection for Active Directory, TLS communication must be enabled between the Active Directory connector and the Active Directory Server. For a Java client to connect using TLS and self-signed certificates, install the certificate into the JVM key-store.

The Common Name (CN) in the **Subject** and DNS entry in the **Subject Alternative Name** fields in SSL certificate must match the fully qualified domain name (FQDN) of the server.

> **Note**
> The FQDN of the Active Directory host must be specified (instead of IP address) in the **Servers** field when the **Use TLS** option is selected under **Domain Configuration**.

To enable TLS communication, complete the following:

1. Export server certificate and copy the exported `.cer` file to the IdentityIQ host.

2. Execute the following command from the bin directory of JDK:

   ```
   keytool -importcert -trustcacerts -alias aliasName -file <absolute path of
   certificate> -keystore <JAVA_HOME>/jre/lib/security/cacerts
   ```

   In the preceding command line, `aliasName` is the name of the alias.

3. Create the Active Directory application and provide all the required values after selecting the **Use TLS for IQService** checkbox.

4. Select **Test Connection** and then **Save**.

Securing Communication Path Between IdentityIQ and IQService

TLS Communication

The Active Directory connector supports TLS communication for IQService.

> **Note**
> Client Authentication is mandatory for operations that use IQService with TLS Communication. Before configuring TLS Communication, install the IQService on the TLS port with the following command as client authentication is mandatory: `IQService.exe -i -o <TLS Port Number>`

To enable TLS communication, complete the following:

1. On the application configuration page, select the **Use TLS for IQService** checkbox.

   For more information on the TLS communication between IQService and an IdentityIQ, see IQService.

2. Select **Save**.

Client Authentication

The Active Directory connector supports client authentication for IQService. This ensures that IQService is communicating with an authorized version of IdentityIQ.

To configure the client authentication, you need to have a domain user whose credentials can be used for connection between IdentityIQ and IQService. This user must be able to self-authenticate on the IQService host machine. To ensure that the authentication works correctly, IQService expects the client to send the credentials of a user that is already registered with every request.

Note

- For client authentication, configure the IQService with the following command: `IQService.exe -a <Domain User/s>`
- For client authentication, if the IQService host machine is not added to any domain, instead of the domain user the Windows local user from the IQService host machine can be configured as the IQService User.
- In cases where a load balancer configured for IQServices (which are under same domain), you can register a single domain user for client authentication on each IQService. This avoids creating separate local users on each IQService host.

To enable client authentication, complete the following:

1. On the application configuration page, enter the credentials in the **IQService User** and **IQService Password** fields.

   For more information on the client authentication, refer to IQService.

2. Select **Save**.

Connecting SailPoint and Active Directory

To connect SailPoint and Active Directory, perform the following tasks:

Configuring the Connector in SailPoint

An application is an instance of third-party software connected to IdentityIQ. The connector is configured to seamlessly provide governing and provisioning access to the application. The connector configuration includes all of the configuration and connection details required to connect IdentityIQ to the application.

> **Note**
>
> This procedure provides the basic information necessary to connect your connector. For additional information, refer to the Application Configuration Guide PDF for your deployed version of IdentityIQ found in the IdentityIQ Product Guides page on Compass.
>
> To view the latest online IdentityIQ guides, refer to the Documentation Portal.

> **Caution**
> Do not open the application configuration in multiple tabs or browsers. Doing so may overwrite changes made in the other.

1. Go to **Applications > Application Definition**.

2. Select **Add New Application**.

3. The **Edit Application** page opens to the **Details** page. Enter the following information:

- **Name** – The name of the application. This is the named used to identify the application throughout IdentityIQ.

  > **Note**
  > IdentityIQ does not support application names that start with a numeric value or that are longer than 31 characters.

- **Owner** – The owner of the application. The owner specified here is responsible for certifications and account group certifications requested on this application if no revoker is specified.

  Application ownership can be assigned to an individual identity or to a workgroup. If the application ownership is assigned to a workgroup, all members share certification responsibilities, are assigned certification requests associated with the application, and all can take action on those requests.

- **Application Type** – The dropdown list contains the applications to which IdentityIQ can connect. This list will grow and change to meet the needs of IdentityIQ users.

- **Description** – The brief description of the application. Use the language selector to enter the description in multiple languages. The dropdown list displays languages supported by your instance of IdentityIQ.

- **Revoker** – The default IdentityIQ user or workgroup to be assigned revocation requests associated with entitlements on this application. If no user is specified in this field, all revocation requests are assigned to the to application owner by default.

- **Proxy Application** – Specify an application to manage accounts and provide connector and schema settings for this application. The proxy application is an application that handles the processing (aggregation and provisioning) on behalf of your application.

- **Profile Class** – A class used to associate this application with a larger set of applications for role modeling purposes.

- **Authoritative Application** – Select this option if this application in an authoritative application. An authoritative source is a repository for employee information for your enterprise that represents the primary and most trusted information about identities, such as a human resources application.

- **Case Insensitive** – Select this option to remove case sensitivity and ignore capitalization differences within values.

- **Native Change Detection** – Select this option if this application should be included when IdentityIQ performs native change detection during aggregation.

      - **Native Change Operations** – Select which operations are included when detecting native change. If no operations are selected, native change detection is disabled.

      - **Attributes to Detect** – Indicates which attributes are compared when accounts are modified. If the Entitlement option is selected, all entitlement attributes are included. If you select User Defined, enter the name of the attributes to compare in the Attribute Names box.

   - **Maintenance Enabled** – Select this option to exclude this application from provisioning and aggregation during the defined maintenance period.

     For more information, refer to Application Maintenance Windows.

      - **Maintenance Expiration** – The date at which the maintenance will end. If no date is defined, this application will be in maintenance indefinitely.

   - **Extended Attributes** – This section displays any extended attributes that were configured for your deployment of IdentityIQ.

   - For more information on the fields displayed on the **Details** page, refer to the IdentityIQ Application Configuration Guide for your release.

4. Select **Configuration** and enter the information required for IdentityIQ to connect and interact with the application. The information required varies by application.

5. Select **Save**.

Forest Settings

An Active Directory forest is a set of all the directory partitions in a particular Active Directory instance that includes all domain, configuration, schema, and optional application information. Multiple forests can share the Active Directory responsibilities across an enterprise. To support a multi-forest configuration for the Active Directory source, configure multiple forests.

You can find the details of your existing configuration by using PowerShell commands. For more information, refer to Active Directory PowerShell Commands.

To configure forest settings, complete the following:

1. Enter the **Forest Name** you want to set for a new forest to use in an organization.

2. (Optional) Enter the **Global Catalog Server** information using the following format: *`IP address`*/*`FQDN`*:*`Port Number`*

Configuring the Global Catalog details also helps improve the pass-through authentication performance. The Active Directory connector provides preference to connect to the Global Catalog if details are provided, if not details are provided it uses the server configured for respective domains to authenticate the users.

The Global Catalog configuration also facilitates domain discovery within that forest.

3. (Optional) Enter the **User** with the required permissions using the following format: *`Domain Name/User Name`*

   For Strong Authentication (SASL) to work, the user must use the following format: `User-Name@DNSDomainName.com`. For more information, refer to Required Permissions.

4. (Optional) Enter the **Password** for the service account.

5. (Optional) Select the **Authentication and Security** from the drop-down menu.

   - **Simple** - The account to authenticate is identified by the DN of the entry for that account, and the proof identity comes in the form of a password. SailPoint recommends that you **Use TLS** with simple authentication as this encrypts data during transit.

   - **Strong (SASL)** - Strong authentication bind is performed, which uses Kerberos or NTLM depending upon whether the IdentityIQ system is in a network (of service account domain) or outside network. SASL has implicit security layer for data encryption.

6. (Optional) Select the **Use TLS** checkbox if the connection is over TLS. If you select this option, you must specify the TLS port in the **Global Catalog Server** field. For more information on enabling TLS communication, refer to Securing the Active Directory Application.

7. (Optional) Select the **Resource Forest** checkbox if this is a dedicated resource forest to manage Microsoft Exchange resources. For more information, refer to Active Directory Resource Forest Topology Exchange Management.

8. (Optional) Select **Manage All Domains** to manage all domains under that forest using the forest credential. If selected, you do not have to configure the domain configuration section. For domains that the application manages, you can preview them by selecting the **Preview** button. If you do not select this option, domains in this forest can be enumerated in the domain configuration by selecting the **Discover** button.

   > **Note**
   > If you change the **Authentication and Security** type, ensure that the **Manage All Domains** attribute is reloaded to view the updated configuration.

9. (Optional) To create another forest, select **Add** and repeat the previous steps.

10. Select **Save**.

Domain Settings

This page displays the list of forests that you have configured and enables you to configure domains.

To create and set up a new domain, complete the following:

1. Enter the **Forest Name** you want to configure for this domain.

2. Enter the **Domain**.

3. Enter the **Service Account** with the required permissions using the following format: `Domain Name\User Name`.

4. Enter the Service Account **Password**.

5. (Optional) Enter the **Servers** information for the domain controller servers that you want to configure using the following format: `IP Address` or `FQDN`. To configure multiple servers, enter a server and then press the **Enter** key. If you have configured two or more servers and the connection to the first servers fails, the source attempts to bind to the next domain controller server in the list.

   > **Note**
   > If you do not provide the IP or FQDN information, it is a server-less bind that requires the correct DNS configuration.

6. (Optional) Select the **Authentication and Security** from the drop-down menu.

   - **Simple** - The account to authenticate is identified by the DN of the entry for that account, and the proof identity comes in the form of a password. SailPoint recommends that you **Use TLS** with simple authentication as this encrypts data during transit.

   - **Strong** - Strong authentication bind is performed, which uses Kerberos or NTLM depending upon whether the IdentityIQ system is in a network (of service account domain) or outside network. SASL has implicit security layer for data encryption.

7. (Optional) Select the **Use TLS** checkbox if the connection is over TLS.

8. (Optional) To configure another domain, select **Add** and repeat the previous steps.

9. Select **Save**.

Account and Group Settings

This page displays enables you to configure account, and group search options. The search DNs define the list of distinguished names of the containers along with other relevant attributes the define the scope for this connector. The search scope defines how far down in the tree to search from the base DN.

If auto partitioning is not enabled, each of these search DNs is also considered as a partition for partitioned full aggregation. Users, Contacts, Managed Service Accounts, and Groups can have different set of searchDNs to define different scope for each of them. The search scopes are stored in the following search DNs respectively:

- Contact Search Scope: `contact.searchDNs`
- Managed Service Account Search Scope: `gmsa.searchDNs`

In cases where the scope is not defined for Groups, it follows **Account's Search Scope**. Defining one search DN to the minimum is required to successfully configure the connector.

Auto Partitioning

SailPoint recommends that you enable the Auto Partitioning feature to enable faster retrieval of Active Directory data. For more information on configuring partitions manually, refer to Partitioning Aggregation.

> **Note**
> The Allow Partitioning feature is only available for account aggregation.

To configure auto partitioning, complete the following:

1. Verify the **Enable Partitioning** checkbox is selected in your Account Aggregation task. For more information on configuring your account aggregation, refer to Account Aggregation.
2. In your application configuration, go to **Configuration > Settings > Account**.
3. Select the **Allow Auto Partitioning** checkbox.
4. In the **Number of Partitions** dropdown, select the number of partitions. These help improve the performance of auto partitioning and can be tuned by selecting the appropriate partitioning count. For higher user populations, a higher partitioning count is preferred.
5. Once you have finished configuration on this page, select **Save**.

Account Search Scope

If the **Group Membership Search DN** attribute is not defined then connector brings all the group memberships associated with the respective account, which are returned by APIs instead of falling back to the scope defined by **Search DN**.

User Search Scope

To configure the user's search, complete the following:

1. In the **Search DN** field, enter the distinguished name of the domain or OU that defines the scope for users.
2. (Optional) Specify an **Iterate Search Filter** string to limit the results returned by the search DN. For more information on the search syntax, refer to the Microsoft Active Directory: LDAP Syntax Filters wiki.
3. (Optional) Specify **Group Membership Search DN** to determine the group membership of the users that you are loading. Separate multiple entries with a semicolon.
4. (Optional) Specify a **Group Member Filter String** as an LDAP search filter string that applies while fetching the user's group membership.
5. (Optional) Select **Add** to create another search filter or select **Delete** to remove a search filter.
6. Once you have configured all the search scopes, select **Save**.

Contact Search Scope

To configure the contact's search, complete the following:

1. (Optional) In the **Search DN** field, enter the distinguished name of the domain or OU that defines the scope for contacts.
2. (Optional) Specify an **Iterate Search Filter** string to limit the results returned by the search DN. For more information on the search syntax, refer to the Microsoft Active Directory: LDAP Syntax Filters wiki.
3. (Optional) Specify **Group Membership Search DN** to determine the group membership of the contacts that you are loading. Separate multiple entries with a semicolon.
4. (Optional) Select **Add** to create another search filter or select **Delete** to remove a search filter.
5. Once you have configured all the search scopes, select **Save**.

Managed Service Account Search Scope

To configure the managed service account's search, complete the following:

1. (Optional) In the **Search DN** field, enter the distinguished name of the domain or OU that defines the scope for the managed service account.
2. (Optional) Specify an **Iterate Search Filter** string to limit the results returned by the search DN. For more information on the search syntax, refer to the Microsoft Active Directory: LDAP Syntax Filters wiki.
3. (Optional) Specify **Group Membership Search DN** to determine the group membership of the managed service accounts that you are loading. Separate multiple entries with a semicolon.

4. (Optional) Select **Add** to create another search filter or select **Delete** to remove a search filter.

5. Once you have configured all the search scopes, select **Save**.

Group Search Scope

To configure the group's search, complete the following:

1. Optional) In the **Search DN** field, enter the distinguished name of a container for a group to define the search scope of groups.

2. (Optional) Specify an **Iterate Search Filter** string to limit the results returned by the search DN. For more information on the search syntax, refer to the Microsoft Active Directory: LDAP Syntax Filters wiki.

3. (Optional) Select **Add** to create another search filter or select **Delete** to remove a search filter.

4. Once you have configured all the search scopes, select **Save**.

IQService Settings

Install IQService to support execution of provisioning operations and native rules. IQService is required by the Active Directory connector for all provisioning, and for reading certain fields from a user's entry, including Terminal Services and Lync attributes.

Configure IQService Settings

To configure the IQService settings, complete the following:

1. (*Optional*) In the **IQService Host** field, enter the FQDN/IP of the system where IQService is installed.

2. (*Optional*) Enter the **IQService Port** number used to connect. If you enable TLS, configure the corresponding IQService TLS port.

3. (*Optional*) Enter the **IQService User** for client authentication.

4. (*Optional*) Enter the **IQService Password** for the IQService user.

5. (*Optional*) Select the **Use TLS** checkbox if the connection is over TLS. If you select this checkbox, IQService User and IQService Password attributes are mandatory. For more information, refer to IQService: TLS and Client Authentication Configuration.

6. Select **Save**.

Exchange Settings

Use this page to configure the exchange server.

To configure the exchange settings, complete the following:

1. (*Optional*) Enter the **Exchange Forest** where the exchange servers are installed.

   > **Important**
   > If you select an exchange forest, the **Exchange Host**, **Service Account**, **Password**, and **Account Forest** fields are mandatory.

2. (*Optional*) In the **Exchange Host** field, enter the exchange FQDN or IP of the Exchange server host.

3. (*Optional*) Enter the **User** with the required permissions using the following format: *`Domain Name\User Name`*

4. (*Optional*) Enter the Service Account **Password**.

5. (*Optional*) In the **Account Forest** field, enter the name of the account's or user's forests served by this exchange server.

6. (*Optional*) Select the **Use TLS** checkbox if the connection is over TLS.

7. (*Optional*) To configure another exchange server, select **Add** and repeat the previous steps.

8. Select **Save**.

Review and Test

Perform a test to confirm the connection to SailPoint.

1. Confirm that the entries in each field are correct.

   If you note any mistakes, return to the section and make corrections.

2. Select **Test Connection** to run the connection test.

Additional Configuration Parameters

This section contains the information on additional parameters you can configure to modify the connector behavior that aren't available on the UI. Configure these parameters using the application debug page. This page contains the following sections:

- Additional Configuration Parameters
- Additional binary/Sid/Guid Attributes
- Caching Ports

Additional Configuration Parameters

The following attributes can be added into the application debug page:

rollbackCreatedAccountOnError

To rollback a created account in case one or more requested attribute /s for that account fails during provisioning operations, set this attribute to true as follows:

```
<entry key="rollbackCreatedAccountOnError" value="true"/>
```

reportPostScriptFailuresAsWarnings

When set to true, the native postscript errors are returned as warnings instead of errors for all update operations.

This ensures that the attributes are successfully provisioned to Active Directory which reflect in IdentityIQ also.

unlockOnChangePassword

The default behavior of unlocking the account on change password can be turned off by setting the `unlockOnChangePassword` attribute to **false**. Default: **true**

setAttributeLevelResult

Set it to true to enable attribute request level results. Default: **False**

Enabling this parameter would marginally increase the time taken to process the request.

aggregationMaxRetries

Count of maximum retry attempts for Active Directory aggregation in case of failures with any of the retry-able errors. Default: **5**

aggregationRetryThreshold

Delay in seconds between each retry attempt of aggregation Default: **10 seconds**

manageLync

Microsoft Lync\Skype for Business Server to be managed by the application.

Add the **manageLync** attribute as follows in the application debug page:

```
<entry key="manageLync">
  <value>
     <Boolean>true</Boolean>
  </value>
</entry>
```

authSearchAttributes

List of attributes which would be used to search user during Pass Through Authentication.

The **authSearchAttributes** attribute can be changed as follows in the application debug page:

```
<entry key="authSearchAttributes">
  <value>
    <List>
     <String>sAMAccountName</String>
     <String>msDS-PrincipalName</String>
     <String>mail</String>
    </List>
  </value>
</entry>
```

memoryStoreSizeInElements

Defines the number of cache elements to be stored in memory (RAM). If all elements must be stored in-memory and nothing on the disk, specify the value as **-2** as follows:

```
<entry key="memoryStoreSizeInElements" value="-2"/>
```

disableComputePreloading

Default: false

To disable auto detection of group membership pre-loading for forests, set the value to true as follows:

```
<entry key="disableComputePreloading">
   <value>
      <Boolean>true</Boolean>
   </value>
</entry>
```

useSingleThreadedCookieSearch

During full aggregation, **dirsync** cookies are fetched as per domain basis using concurrent threads. To fetch cookies sequentially on a single thread, set the value to true as follows:

```
<entry key="useSingleThreadedCookieSearch" value="true"/>
```

displayAttributeForContacts

CN is used as default for display name of contact objects in IdentityIQ. To use any other schema attribute, define the name of the attribute as the as value of this attribute:

```
<entry key="displayAttributeForContacts" value="firstName"/>
```

disableFspAggregation

Default: false

To disable aggregating foreign memberships of any user, set the value to true as follows:

```
<entry key="disableFspAggregation">
   <value>
      <Boolean>true</Boolean>
   </value>
</entry>
```

ldapExtendedControls

For Active Directory Services managed system not to generate any further references (crossRef objects) in response to the search query add the following entry key in the application debug page:

```
<entry key="ldapExtendedControls">
 <value>
  <List>
   <String>1.2.840.113556.1.4.1339</String>
  </List>
 </value>
</entry>
```

Active Directory Connector search does not **rely** on referrals to fetch information from the managed system. To have the comprehensive data aggregated, Domain Setting configuration must be up to date with required information.

skipDeletedObjScopeCheckInDelta

Default: false

If set to true as follows during account delta aggregation, connector does not make a call to Active Directory to check whether deleted object was in scope of the application.

```
<entry key="skipDeletedObjScopeCheckInDelta" value="true"/>
```

If the deleted object is present in the IdentityIQ database, it gets deleted from the database. If the deleted object is not in the IdentityIQ database, then no further action is performed.

skipObjTypeCheckForMembersInDelta

Default: false

If set to true as follows during account delta aggregation, connector does not make a call to Active Directory to check if objectType of member is added/removed to a group:

```
<entry key="skipObjTypeCheckForMembersInDelta" value="true"/>
```

If object is present in the IdentityIQ database, then membership would get updated.

skipBindUsingDNS

Default: false

If set to true as follows, DNS server would not be used to find out Domain Controller for any given domain in serverless configuration:

```
<entry key="skipBindUsingDNS" value="true"/>
```

Connector would always call IQService to find domain controller.

skipGetObjInMembershipDelta

Default: false

If set to true as follows, Connector would not make a call to Active Directory to get additional attributes of the changed object intercepted during delta aggregation.

```
<entry key="skipGetObjInMembershipDelta" value="true"/>
```

These additional attributes are fetched by connector if the user has entitlement changes along with attribute changes, or if the user has add, remove, or both entitlement changes.

Hence, when **skipGetObjInMembershipDelta** is set to **true**, the Resource Object is sent to IdentityIQ containing only the attributes intercepted during delta aggregation.

searchInContainers

Default: false

By default, the pass-through authentication (PTA) searches for the users in the entire domain defined (in case of multiple searchDNs configured) which can delay PTA.

To enable PTA check for the users in configured search DNs only, set the following entry key to true (only applies to pass-through authentication) in the application debug page:

```
<entry key="searchInContainers" value="true"/>
```

disableLDAPHostnameVerification

To disable hostname verification during LDAP Communication over TLS, configure the following attribute in the application debug page:

```
<entry key="disableLDAPHostnameVerification" value="true"/>
```

skipIterateSearchFilterInPTA

Default: false

If set to true as follows, Connector would not consider iterate search filter configured for single search DN to authenticate the user in Pass through authentication (PTA):

```
<entry key="skipIterateSearchFilterInPTA" value="true"/>
```

If searchInContainers flag is set to true, it would take precedence over skipIterateSearchFilterInPTA.

buildPartialROOnAuthentication

Default: false

By default, when buildPartialROOnAuthentication is set to false, Connector would build full RO but sometimes that may take time and would cause delay in login.

When buildPartialROOnAuthentication is set to true as follows in the application debug page, Connector would build partial RO and would set identity, display and some other attributes which would be used in correlation like samAccountName, hence improving the login performance:

```
<entry key="buildPartialROOnAuthentication" value="true"/>
```

domainIterateSearchFilter

(*Applicable only for User Delta Aggregation*) Define this attribute in domain settings to override the Iterate filter defined in Search Scope for Users.

disableContainerFilterForDelta

(*Applicable only for Delta Aggregation*) This attribute is used to skip the iterate search filter when set to **true** while performing **DirSync** delta aggregation.

adSystemConfUseUpdatedSASLCommunication

This attribute is used to resort back to previous implementation (non-cached). Add the following attribute to the **IdentityIQ > Debug Page > Configuration > System Configuration**

Or

IdentityIQ > Debug Page > Configuration Object (drop-down) > System Configuration

as follows:

```
<entry key="adSystemConfUseUpdatedSASLCommunication" value="false"/>
```

Additional binary/Sid/Guid Attributes

To display any additional binary/Sid/Guid attributes, use the following entries :

attrsDisplayInBinaryFormat

To display attributes values in binary format which is also the default display format:

```
<entry key="attrsDisplayInBinaryFormat">
  <value>
    <List>
      <String>Attribute name1</String>
      <String>Attribute name2</String>
    </List>
  </value>
</entry>
```

attrsDisplayInSIDFormat

To display attributes values in Sid format:

```
<entry key="attrsDisplayInSIDFormat">
  <value>
    <List>
      <String>Attribute name1</String>
      <String>Attribute name2</String>
    </List>
  </value>
</entry>
```

attrsDisplayInGUIDFormat

To display attributes values in Guid format:

```
<entry key="attrsDisplayInGUIDFormat">
  <value>
    <List>
```

```
      <String>Attribute name1</String>
      <String>Attribute name2</String>
    </List>
  </value>
</entry>
```

Caching Ports

Port numbers for caching mechanism to replicate the cached data across different task servers.

> **Note**
> SailPoint recommends that the ports are open and not in use by any other application.

enableCache

To enable cache, set the value to true as follows:

```
<entry key="enableCache">
  <value>
     <Boolean>true</Boolean>
   </value>
</entry>
```

cacheRmiPort

The default value is 40001

cacheRemoteObjectPort

The default value is 40002

cacheReplicationTimeout

Maximum time in minutes to wait for membership cache replication on task server. Default: 10 minutes

```
<entry key="cacheReplicationTimeout" value="20"/>
```

cacheSocketTimeoutMillis

Maximum time in milliseconds to wait for the client sockets to send messages to a remote listener. Default: 2000 milliseconds.

```
<entry key="cacheSocketTimeoutMillis" value="5000"/>
```

JNDI Configuration

This section covers information on JNDI systems.

JNDI System Properties

The Active Directory connector supports all JNDI system properties. For more information, refer to https://docs.oracle.com/javase/jndi/tutorial/ldap/connect/config.html.

Following are the available system properties:

```
com.sun.jndi.ldap.connect.pool.maxsize
```

```
com.sun.jndi.ldap.connect.pool.protocol
```

```
com.sun.jndi.ldap.connect.pool.timeout
```

```
com.sun.jndi.ldap.connect.pool.initsize
```

```
com.sun.jndi.ldap.connect.pool.authentication
```

```
com.sun.jndi.ldap.connect.pool.debug
```

For example:

```
<entry key="com.sun.jndi.ldap.connect.pool.maxsize" value="10"/>
<entry key="com.sun.jndi.ldap.connect.pool.protocol" value="plain ssl"/>
<entry key="com.sun.jndi.ldap.connect.pool.timeout" value="20000"/>
<entry key="com.sun.jndi.ldap.connect.pool.initsize" value="5"/>
<entry key="com.sun.jndi.ldap.connect.pool.authentication" value="plain ssl"/>
<entry key="com.sun.jndi.ldap.connect.pool.debug" value="fine"/>
<entry key="com.sun.jndi.ldap.connect.pool" value="true"/>
<entry key="com.sun.jndi.ldap.read.timeout" value="120000"/>
```

Provisioning Policy

When SailPoint provisions new accounts to the Active Directory source, it uses the attributes on the **Provisioning Policy** page as instructions or a template for what to include in the account. Each source can have its own configuration that specifies which attributes to include in account creation and how to set their values. SailPoint predefines this for most source types, but you can edit the way the attributes are mapped.

When new access is granted on a source where a user does not already have an account, IdentityIQ automatically includes account creation in the provisioning. This applies whether provisioning started from an access request or from automated role or lifecycle state assignment.

For direct-connect sources, IdentityIQ automatically creates the account from this configuration. If the source is not configured as a direct-connect source, IdentityIQ creates and assigns a provisioning task to the source owner and includes the values for the source owner to use in manually creating the account.

> **Warning**
> This section describes the configuration of the default Provisioning Policy. However, SailPoint recommends that you work with Services to define a Create Profile specific to your company's needs. Be sure to verify large changes to the provisioning policy before implementation. Failure to do so may result in your provisioning to fail.

Note the following when provisioning the Active Directory source:

- The Active Directory source has a `skipDeletedObjScopeCheckInDelta` attribute that you can set to configure the binding of deleted and recycled objects in Active Directory and process them in IdentityIQ accordingly.
- Active Directory does not show the `InvalidCastException` attribute in logs when provisioning a Lync account.
- Active Directory updates the `msExchHideFromAddressLists` attribute value in 'modify' provisioning operation.
- Active Directory returns the attribute level results by setting the `setAttributeLevelResult` attribute to `"true"`. Any attribute provisioning failures do not result in the failure of subsequent attributes with same error, when `setAttributeLevelResult` is set to `"true"`.

  > **Note**
  > The value must be inside quotation marks as it is being passed as a string and not a boolean value.

- Active Directory clears the description attribute when provisioned with an empty string.
- To pass additional information (metadata) in a provisioning plan to be used for any customization (for example, IQService Before/After script), see the example of **AccountRequest metadata** xml provided in the IQService Before/After Scripts.
- For more information on provisioning attributes, refer to the Default Provisioning Attributes Reference.

Configuring Account Schema

The application schema is used to configure the objects returned from a connector. When a connector is called, the schema is supplied to the methods on the connector interface. This connector currently supports two types of objects, accounts (users and contacts) and group. Account objects are used when building identities Link objects. The group schema is used when building Account_Group objects that are used to hold entitlements shared across identities.

> **Note**
>
> - The Schema tab is used to define the attributes for each object type in the application being configured. The schema attributes can be defined as Indexed, Entitlement, or Multi-Valued.
> - The schema attributes that are not present in the out-of-the-box must be defined as string if not specified.
> - For more information on the account schema attributes, see Schema Attributes Reference.

Adding a New Schema Attribute

To add a new attribute on the Schema tab, complete the following:

1. Go to **Configuration > Schema**.
2. Select **Add New Schema Attribute**.

   A new line is added to the Attributes table.
3. Enter the attribute **Name**.
4. Enter the attribute **Description**.
5. Select the attribute **Type**.
6. Enter the attribute properties. For example, Entitlement or Multi-Valued.
7. Select **Save**.

Notes for creating custom schema attributes:

- Attribute names cannot contain a space.
- When creating custom names, be aware that the name that displays in Active Directory may be different than the LDAP name. For more information, refer to List of LDAP Attribute Names and Associated Name in Active Directory.

List of LDAP Attribute Names and Associated Name in Active Directory

Active Directory Display Names and LDAP Names to be used while importing as .csv file.

Name in Active Directory	LDAP Name (Header in CSV File)
First Name	givenName
Middle Name/Initials	initials
Last Name	sn
Logon Name	rPrincipalName
Logon Name (Pre Windows 2000)	sAMAccountName
Display Name	displayName
Full Name	name/cn
Description	description
Office	physicalDeliveryOfficeName
Telephone Number	telephoneNumber
Email	mail
Web Page	wWWHomePage
Password	password
Street	streetAddress
PO Box	postOfficeBox
City	l
State/Province	st
Zip/Postal Code	postalCode
Country	co
Country 2 Digit Code - e.g., US	c
Country Code - e.g., 840 for the US	countryCode
Add to Groups	memberOf
Remove from Groups	removememberOf
Account Expires (use the same date format as	accountExpires

Name in Active Directory	LDAP Name (Header in CSV File)
the server)	
User Account Control	userAccountControl
User Photo	thumbnailPhoto / exchangePhoto (supports high resolution photos) / jpegPhoto / photo / thumbnailLogo
Profile Path	profilePath
Login Script	scriptPath
Home Folder	homeDirectory
Home Drive	homeDrive
Log on to	userWorkstations
Home	homePhone
Pager	pager
Mobile	mobile
Fax	facisimileTelephoneNumber
IP Phone	ipPhone
Notes	info
Title	title
Department	department
Company	company
Manager	manager
Mail Alias	mailNickName
Simple Display Name	displayNamePrintable
Hide from Exchange Address Lists	msExchHideFromAddressLists
Sending Message Size (KB)	submissionContLength
Receiving Message Size (KB)	delivContLength
Accept Messages from Authenticated Users Only	msExchRequireAuthToSendTo
Reject Messages From	unauthOrig
Accept Messages From	authOrig
Send on Behalf	publicDelegates
Forward To	altRecipient
Deliver and Redirect	deliverAndRedirect
Recipient Limits	msExchRecipLimit
Use Mailbox Store Defaults	mDBuseDefaults

Name in Active Directory	LDAP Name (Header in CSV File)
Issue Warning at (KB)	mDBStorageQuota
Prohibit Send at (KB)	mDBOverQuotaLimit
Prohibit Send and Receive at (KB)	mDBOverHardQuotaLimit
Do not Permanently Delete Messages Until the Store Has Been Backed Up	deletedItemFlags
Keep Deleted Items for (days)	garbageCollPeriod
Outlook Mobile Access	msExchOmaAdminWirelessEnable
Outlook Web Access	protocolSettings
Allow Terminal Server Logon	tsAllowLogon
Terminal Services Profile Path	tsProfilePath
Terminal Services Home Directory	tsHomeDir
Start the Following Program at Logon	tsInheritInitialProgram
Starting Program File Name	tsInitialProgram
Start In	tsWorkingDir
Connect Client Drive at Logon	tsDeviceClientDrives
Connect Client Printer at Logon	tsDeviceClientPrinters
Default to Main Client Printer	tsDeviceClientDefaultPrinter
End Disconnected Session	tsTimeOutSettingsDisConnections
Active Session Limit	tsTimeOutSettingsConnections
Idle Session Limit	tsTimeOutSettingsIdle
When Session Limit Reached or Connection Broken	tsBrokenTimeOutSettings
Allow Reconnection	tsReConnectSettings
Remote Control	tsShadowSettings
Protect Accidental Deletion	preventDeletion
Manager Can Update Members	managerCanUpdateMembers
Primary Group ID	primaryGroupID
Administrative Group	msExchAdminGroup
Exchange Server Name	msExchHomeServerName
Managed By	managedBy
Target Address	targetAddress
Add Proxy Addresses	proxyAddresses
Automatically Update Email-Address Based on Recipient Policy	msExchPoliciesExcluded

Active Directory Resource Forest Topology Exchange Management

In Active Directory Account Forest - Resource Forest Topology, all user accounts exist in one or more Forests called Account Forests, while resources have a dedicated Active Directory Forest called a Resource Forest. The Resource Forest may have deployments like Microsoft Exchange or Skype Server.

The Active Directory connector supports managing Exchange Linked Mailbox, Mail user, and Mail contact from the Resource Forest. Whenever a user from the Account Forest requests a mailbox, a Linked Mailbox is created on the Resource Forest Exchange server with an associated disabled user. The connector uses the following terms:

- Shadow Account for disabled user
- Master Account for the user of Account Forest

The connector aggregates all Exchange properties of the Shadow Account and maps these to the corresponding Master Account.

The connector relies on the connection details provided under the Exchange Settings, Forest Settings, and Domain Settings to carry out all the supported operations.

Supported Operations

Operations	Features
Aggregation	• Aggregate Linked Mailbox properties for the Account Forest User • Aggregate Mail user, Mail contact from the Resource Forest Exchange
Delta Aggregation	Supports aggregating for the following delta changes: • Create Linked Mailbox, Update Linked Mailbox properties • Mail enabled Distribution List membership changes for the shadow account • Create, Update, Delete Mail User object from the Resource Forest Exchange
Create, Update, Delete	• Linked Mailbox for the Account Forest User • Mail enabled Distribution List from the Resource Forest

Prerequisite

Minimum one-way trust from Exchange Resource Forest to Account Forest.

Administrator Permissions

- For read operations of the Linked mailbox properties, service account from the Resource Forest Domain must be a member of Account Operator group.
- For all provisioning operations of Linked mailbox, service account from the Resource Forest Domain must be a member of Recipient Management group.

Resource Forest Specific Domain Configuration Attributes

This section lists additional Resource Forest specific configuration parameters:

disableShadowAccountMembership

By default, the connector considers the memberships of Shadow Account as Master Accounts memberships.

To discard membership of shadow account, set this Boolean (or as a String) attribute to `true` under `domainSettings` of respective Resource Forest domain, as follows:

```
<entry key=" disableShadowAccountMembership" value=""true/>
```

shadowAccountMembershipFilter

By default, the connector retrieves all memberships of Shadow Account, but these memberships can be filtered based on a LDAP filter.

For example, the following entry key only considers distribution group of shadow account:

```
<entry key="shadowAccountMembershipFilter" value=" (!
(groupType:1.2.840.113556.1.4.803:=2147483648))">
```

provisionGroupToShadowAccount

By default, the connector supports assigning of only Universal and Global Distribution List from Resource Forest Domain to the Shadow Account. To override this and to support all other types of group provisioning to the Shadow Account, pass this attribute in the metadata of the `AttributeRequest` for `memberOf` attribute as given in the following example:

```
<AttributeRequest op="Add" name="memberOf" value=<group-nativeIdentity>>
    <Attributes>
      <Map>
```

```
        <entry key="provisionGroupToShadowAccount" value="true" />
      </Map>
    </Attributes>
</AttributeRequest>
```

Resource Forest Specific Application Configuration Attributes

This section lists additional Resource Forest Application configuration parameters:

supportFSPsFromResourceForest

Set this Boolean attribute to `true` to enable aggregating and provisioning FSPs from the Resource Forest Domains for the Master Account. By default, the value is False.

retainShadowAccountOnDelete

To retain shadow account on delete of the master account set this Boolean attribute to true. Default: False

Configuring searchDNs with Resource Forest

This section lists additional searchDN specific configuration parameters:

- **Account searchDNs**: For aggregating Linked Mailbox data, no additional Account Search Scope required. The connector by default considers domains from Resource Forest as search scope for Shadow Accounts.
- Account Search scope can contain the **searchDNs** from the Resource Forest domains if Mail User from the Resource Forest is to be managed.
- **Contact searchDNs**: Adding **searchDNs** from the Resource Forest will allow managing contact objects from the Resource Forest.
- **Group searchDNs**: To manage groups from the Resource Forest domains, add **searchDN** entries from the Resource Forest Domain.

Microsoft Exchange Schema Attributes

In addition to the existing Microsoft Exchange account schema attributes, the following new attributes are added in the schema for new applications. For existing Active Directory applications, the following schema attributes must be added manually if required:

msExchRecipientTypeDetails

Type of the Microsoft Exchange recipient object. Value `2` indicates that the mailbox type is a Linked Mailbox.

shadowAccountDN

The distinguished name of the Linked Mailbox Shadow Account (Disable Account which was created while creating Linked Mailbox).

shadowAccountGuid

The objectGuid of the Linked Mailbox Shadow Account.

Linked Mailbox Provisioning Attribute

In addition to the existing Provisioning Policy Attributes for Microsoft Exchange in the Create Account Profile, the connector provides support for the following attributes for a Linked Mailbox:

shadowAccountDN

The distinguishedName of the **Linked Mailbox** shadow account (Disable Account which was created while creating Linked Mailbox).

Active Directory PowerShell Commands

You can enable Windows PowerShell to execute commands on your Active Directory system. All standard Active Directory commands are supported on your system. For more information on Active Directory PowerShell commands, refer to Microsoft Active Directory.

> **Note**
> You can use PowerShell commands to find your configuration information if it's running on a computer connected to the domain.

To get your forest and domain information, complete the following:

1. Select the Windows Start button, search for *PowerShell*, and then select **Windows PowerShell**.
2. To view your domain information, enter the following command: `Get -ADDomain`, and then press the **Enter** key.
3. To view your forest information, enter the following command: `Get -ADForest`, and then press the **Enter** key.

Enable Remote Shell for a User

For managing an Exchange Server, the service account must be a member of Recipient Management group.

The application user for provisioning of the Exchange server must be remote shell enabled. To enable remote shell for a user, complete the following:

1. Set the `RemotePowerShellEnabled` parameter to `$True` on the Set-User cmdlet. For example:

   ```
   Set-User UserName -RemotePowerShellEnabled $True
   ```

Active Directory Module for Windows PowerShell

To get the closest Domain Controller for a particular domain, the Active Directory source calls the IQService.

As a prerequisite, Active Directory module for Windows PowerShell on IQService machine must be installed and the Active Directory DS role must be installed on a machine.

To enable the Active Directory module for Windows PowerShell, complete the following:

1. Go to **Server manager > Features > Remote Server Administration Tools > AD DS & LDS tools** and select **Active Directory module for Windows PowerShell**.

To enable the Active Directory module for Windows PowerShell using PowerShell commands, complete the following:

1. Select the Windows Start button, search for *PowerShell*, and then select **Windows PowerShell**.

2. In the PowerShell console window, use the `Install-WindowsFeature` cmdlet to install the module. Specify the feature `-Name` (RSAT-AD-PowerShell) and add the `-IncludeAllSubFeature` parameter to add any child features. For example, `Install-WindowsFeature -Name RSAT-AD-PowerShell -IncludeAllSubFeature`

Microsoft Exchange Shared Mailbox

Shared Mailbox are special type of mailbox where multiple users can read and send email from the common email address. A shared mailbox is a type of user mailbox that does not have its own username and password. As a result, users cannot log into them directly. To access a shared mailbox, users must first be granted **Send As** or **Full Access** permissions to the mailbox after which, the user signs into their own mailboxes and then can access the shared mailbox by adding it to their Outlook profile.

The Active Directory connector supports managing Shared Mailbox as Account Group object. For this feature, the schema attributes and provisioning plan for the Shared Mailbox must be added in the application xml file.

Supported Operations

Operations	Features
Aggregation	• Aggregate Shared Mailbox as Account Group Object • Aggregation of User’s Shared Mailbox assignment as an entitlement.
Create, Update, Delete	Supports creating and updating attributes of the Shared Mailbox along with assigning and removing permissions of the Shared Mailbox. For more information, refer to Microsoft Exchange Shared Mailbox

Prerequisites

- IQService must be configured in the application
- Exchange configuration details are required for aggregation and provisioning operations

Administrator Permissions

- For aggregation of Shared Mailbox and aggregating user's Shared Mailbox Membership, the service account must be a member of **Account Operator Group** and **Recipient Management Group**.
- For Create, Update, and Delete operations on a Shared Mailbox and when assigning a Shared Mailbox to a user account:
  - Service account must be a member of **Account Operator Group** and **Recipient Management Group**.
  - Updating **Send As** permission of the Shared Mailbox, service account must have **Active Directory Permissions** Exchange Role. By default, a member of an Organization Management group has an

Exchange Role with higher capabilities that are not required for this operation. It is recommended that you create a custom **Exchange Admin Role Group**.

Complete the following to create a custom Exchange Admin Role Group:

1. On the **Exchange admin center** page, select **Permissions** in the left pane.
2. Under the **admin roles** tab, click **+** icon to create new Role Group.
3. On the **Role Group** window that appears, enter the **Name** and **Description**.
4. From the list of displayed Roles, search and select **Active Directory Permissions Role** and select **Save**.

This creates a Universal Security Group with the given name under **Microsoft Exchange Security Groups** organizationUnit. Add the service account to this group.

Enabling Shared Mailbox Management

> **Note**
> By default, Shared Mailbox management is not enabled for the Active Directory application.

To enable shared mailbox management, complete the following:

1. Import the Shared Mailbox Schema to the Active Directory application.
   Copy the following schema and paste it below Group Schema:

```
<Schema aggregationType="group" descriptionAttribute="description" displayAttribute="msDS-PrincipalName" featuresString="PROVISIONING" hierarchyAttribute="" identityAttribute="distinguishedName" instanceAttribute="" nativeObjectType="Group" objectType="sharedMailbox">
      <AttributeDefinition name="cn" type="string">
        <Description>common name(s) for which the entity is known by</Description>
      </AttributeDefinition>
      <AttributeDefinition name="distinguishedName" type="string">
        <Description>distinguished name for which the entity is known by</Description>
      </AttributeDefinition>
      <AttributeDefinition name="description" type="string">
        <Description>descriptive information</Description>
      </AttributeDefinition>
      <AttributeDefinition name="objectSid" type="string">
        <Description>Windows Security Identifier</Description>
      </AttributeDefinition>
      <AttributeDefinition name="objectguid" type="string">
```

```
        <Description>Object globally unique identifier </Description>
      </AttributeDefinition>
      <AttributeDefinition name="mailNickname" type="string">
        <Description>Exchange alias for the Shared Mailbox</Description>
      </AttributeDefinition>
      <AttributeDefinition name="msDS-PrincipalName" type="string">
        <Description>Name of the entity in the format "NetBIOS domain
name\sAMAccountName"</Description>
      </AttributeDefinition>
      <AttributeDefinition multi="true" name="fullAccess" type="string">
        <Description>List of user or group having full access permission on the
Shared Mailbox</Description>
      </AttributeDefinition>
      <AttributeDefinition multi="true" name="sendAs" type="string">
        <Description>List of user or group having 'Send As' permission on the
Shared Mailbox</Description>
      </AttributeDefinition>
      <AttributeDefinition multi="true" name="sendOnBehalf" type="string">
        <Description>List of user or group having 'Send on behalf' permission
on the Shared Mailbox</Description>
      </AttributeDefinition>
      <AttributeDefinition multi="true" name="memberOf" schem-
aObjectType="group" type="string"/>
      <AttributeDefinition name="sAMAccountName" type="string"/>
      <AttributeDefinition name="homeMDB" type="string"/>
      <Attributes>
        <Map>
          <entry key="groupMemberAttribute" value="[fullAccess, sendOnBehalf,
sendAs]" />
        </Map>
      </Attributes>
    </Schema>
```

2. Update the User Account schema to represent an assigned Shared Mailbox.

   Copy the following attribute definition and paste it in User Schema:

```
<AttributeDefinition entitlement="true" managed="true" multi="true" name-
e="memberOfSharedMailbox" schemaObjectType="sharedMailbox" type="string">
     <Description>List of Shared Mailboxes to which user is has per-
missions</Description>
</AttributeDefinition>
```

   The value of `schemaObjectType` can be set to **string** if the Shared Mailbox object schema is not added in the application.

3. Add Create and Update Provisioning policies. The connector supports updating attributes which are present in the Shared Mailbox schema.

   Copy the following policies under the **<ProvisioningForms>** tag:

- **Create Policy**

```
<Form name="Create Shared Mailbox" objectType="sharedMailbox" type-
e="Create">
      <Attributes>
        <Map>
          <entry key="pageTitle" value="Create Shared Mailbox"/>
        </Map>
      </Attributes>
      <Section>
        <Field displayName="con_prov_policy_ad_distinguishedName" helpKey-
y="help_con_prov_policy_ad_distinguishedName" name="distinguishedName"
required="true" type="string"/>
        <Field displayName="con_prov_policy_ad_mailNickname" helpKey-
y="help_con_prov_policy_ad_mailNickname" name="mailNickname" required-
d="true" reviewRequired="true" type="string"/>
        <Field displayName="con_prov_policy_ad_homeMDB" helpKey="help_
con_prov_policy_ad_homeMDB" name="homeMDB" reviewRequired="true" type-
e="string"/>
        <Field displayName="Full Access" multi="true" name="fullAccess"
type="string"/>
        <Field displayName="Send As" multi="true" name="sendAs" reviewRe-
quired="true" type="string"/>
        <Field displayName="Send On Behalf" multi="true" name-
e="sendOnBehalf" type="string"/>
      </Section>
</Form>
```

- **Update Policy**

```
    <Form name="Update Shared Mailbox" objectType="sharedMailbox" type="Update">
      <Attributes>
        <Map>
          <entry key="pageTitle" value="Update Shared Mailbox"/>
        </Map>
      </Attributes>
      <Section>
        <Field displayName="con_prov_policy_ad_distinguishedName" helpKey="help_con_prov_policy_ad_distinguishedName" name="distinguishedName" required="true" type="string">
          <Attributes>
            <Map>
              <entry key="readOnly" value="true"/>
            </Map>
          </Attributes>
        </Field>
        <Field displayName="con_prov_policy_ad_mailNickname" helpKey="help_con_prov_policy_ad_mailNickname" name="mailNickname" required="true" reviewRequired="true" type="string"/>
        <Field displayName="con_prov_policy_ad_homeMDB" helpKey="help_con_prov_policy_ad_homeMDB" name="homeMDB" reviewRequired="true" type="string"/>
        <Field displayName="msDS-PrincipalName" helpKey="msDS-PrincipalName" name="msDS-PrincipalName" reviewRequired="true" type="string">
          <Attributes>
            <Map>
              <entry key="readOnly" value="true"/>
            </Map>
          </Attributes>
        </Field>
        <Field displayName="sAMAccountName" helpKey="sAMAccountName" name="sAMAccountName" reviewRequired="true" type="string">
        </Field>
        <Field displayName="objectSid" helpKey="cn" name="objectSid" reviewRequired="true" type="string">
          <Attributes>
            <Map>
              <entry key="readOnly" value="true"/>
            </Map>
          </Attributes>
        </Field>
        <Field displayName="objectguid" helpKey="objectguid" name="objectguid" reviewRequired="true" type="string">
          <Attributes>
            <Map>
              <entry key="readOnly" value="true"/>
            </Map>
          </Attributes>
        </Field>
        <Field displayName="Full Access" multi="true" name="fullAccess" 
reviewRequired="true" type="string"/>
        <Field displayName="Send As" multi="true" name="sendAs" reviewRe-
quired="true" type="string"/>
        <Field displayName="Send On Behalf" multi="true" name-
e="sendOnBehalf" reviewRequired="true" type="string"/>
      </Section>
    </Form>
```

Microsoft Exchange Shared Mailbox Additional Configuration Parameters

This section lists additional Microsoft Exchange Shared Mailbox configuration parameters:

defaultSharedMBPermissions

Comma separated names of Shared Mailbox permissions which would be assigned to user when Shared Mailbox access is requested by using the **memberOfSharedMailbox** attribute. Default value: **fullAccess**, **sendAs**

For example:

```
<entry key="defaultSharedMBPermissions" value="sendAs,sendOnBehalf"/>
```

Permitted values: fullAccess, sendAs, sendOnBehalf

fetchSMBMembershipForUserFromAD (applicable for get individual account operation only)

To avoid the intensive time process of reading users Shared Mailbox assignment except for aggregation operation, connector returns user's Shared Mailbox values which were aggregated in the previous aggregation. This configuration attribute will get latest values of user's Shared Mailbox assignment.

> **Note**
> Setting this flag to true has a performance impact on get account operation.

Microsoft Exchange Shared Mailbox Schema Attributes

The following attributes are the only supported schema attributes for the Shared Mailbox object and User schema attribute (`memberOfSharedMailbox`).

Shared Mailbox Object Schema Attributes

In addition to the existing object attributes (`samAccountName`, `ObjectGUID`, `ObjectSID`, `distinguishedName`, `homeMDB`, `mailNickName`), the following schema attributes must be added manually if required:

fullAccess

Multi-valued attribute representing Active Directory objects having Full Access permission on Shared mailbox. Object’s name is represented in **msDs-PrincipalName** format.

sendAs

Multi-valued attribute representing Active Directory objects having **Send As** permission on Shared mailbox. Object’s name is represented in **msDs-PrincipalName** format.

sendOnBehalf

Multi-valued attribute representing Active Directory objects having **Send As** permission on Shared mailbox. Object’s name is represented in **distinguishedName** format.

User Schema Attribute

memberOfSharedMailbox

List of Shared Mailbox names that the user has **Full Access**, **Send As**, **Send On Behalf**, or all three permissions.

Microsoft Exchange Shared Mailbox Provisioning Attributes

In addition to the existing Provisioning Policy Attributes, the connector provides support for the following attributes for a Shared Mailbox:

> **Note**
> Attributes marked with an asterisk (*) are mandatory.

distinguishedName*

DistinguishedName of the shared Mailbox to be created.

mailNickname*

Exchange alias for the Shared Mailbox.

homeMDB

Exchange Mailbox store DN.

fullAccess

List of user/groups in msDS-PrincipalName format to whom to assign **Full Access** permission.

sendAs

List of user/group in msDS-PrincipalName to whom to assign **Send As** permission.

sendOnBehalf

List of user/group in DN format to whom to assign **Send On Behalf** permission.

Active Directory Recycle Bin

A new feature ‘Recycle Bin’ introduced by Microsoft provides support for restoring deleted users, groups with all their attributes and group memberships. SailPoint Active Directory Connector support this feature. Using this feature, any deleted objects (Accounts and Groups) can be restored.

Prerequisites

> **Note**
> Recycle Bin feature must be enabled on Active Directory.

- IQService can be installed on Windows system with one of the following Operating System:
  - Microsoft Windows Server 2019
  - Microsoft Windows Server 2016
  - Microsoft Windows Server 2012 R2
  - Microsoft Windows Server 2012

  For more information on installing and registering IQService, refer to IQService.
- Install **Active Directory module for Windows PowerShell** on the computer where IQService is installed.

  > **Note**
  > By default, this module is installed on all DCs.

  For non-DC but server class Operating System computer, open Windows PowerShell Console and execute the following commands:
  - `Import-module servermanager`
  - `Add-WindowsFeature -Name "RSAT-AD-PowerShell" –IncludeAllSubFeature`
- Run the following PowerShell command on all domain controllers (DCs) in the forest which must be managed:

  ```
  Enable-PSRemoting
  ```

  > **Note**
  > If multiple servers are managed, run the above command on all the servers present under the "domainSettings".

Configuring Recycle Bin

1. Open the Console and `IIQ\HOME\WEB-INF\config\configManageDeletedObjects.xml` file. The `configManageDeletedObjects.xml` file creates the **Manage Recycle Bin** quick link on the dashboard and adds the **Restore Deleted Objects** workflow.

2. Modify `manageRecycleBin` attribute in the Active Directory application with the value set to **true**.

```
<entry key="manageRecycleBin">
   <value>
      <Boolean>true</Boolean>
   </value>
</entry>
```

3. After account and account-group aggregation, the deleted object would be visible under the **Manage Recycle Bin** quick link. Accounts/Groups can be restored individually or all together.

4. The **DirSync** delta aggregator also supports detecting deleted objects.

Unstructured Target Collector

Unstructured target information is used to define unstructured data sources from which the connector is to extract data. Unstructured data is any data that is stored in a format that is not easily readable by a machine. For example, information contained in an Excel spread sheet, the body of an email, a Microsoft Word document, or an HTML file is considered unstructured data. Unstructured targets pose a number of challenges for connectors, because not only is the data stored in a format that is hard to extract from, the systems and directory structures in which the files reside are often difficult to access.

The unstructured target collector that can be configured with Active Directory application is Windows file share.

> **Note**
> Active Directory Connector supports automated revocation of the Target Permissions.

Windows File Share

Windows file share target collector can be configured on Active Directory application to read and correlate file share permissions on Active Directory entities. To correlate the aggregated permissions, ensure that the following attribute is marked as Correlation Key in respective schema:

- **objectSid** for Accounts and Groups

This target collector requires a the IQService to be installed on a machine that has visibility to the directory or share to include in the target scan. Refer to the Installation Guide for information on installing and registering the IQService.

The unstructured targets defined on this tab are used by the Target Aggregation task to correlate targets with permissions assigned to identities and account groups for use in certifications.

The Unstructured Targets tab contains the following information:

Field	Description
Attributes: The required settings for connecting to the IQService.	
IQService Host	The host on which the IQService resides.
IQService Port	The TCP/IP port where the IQService is listening for requests.
IQService User	User registered with IQService for Client Authentication.
IQService Password	Password of registered user for Client Authentication.
Use TLS for IQService	Indicates whether this is a TLS communication between IdentityIQ and IQService. If **Use TLS** is enabled, the `IQService User` and `IQService Password` attributes are mandatory.

Field	Description
Number of targets per block	Number or targets (files) to include in each block of data returned.
File Shares: The required information for each share.	
Path	UNC Style path to a share or local directory. You can target a specific file or a directory and its sub-directories containing multiple files from which to extract the required data. If you target a directory, use the **Wildcard** and **Directory Depth** fields to narrow the query if possible.
Directories Only	Use to instruct to the collector to ignore files and just report back directory permission information.
Directory Depth	The sub-directory depth from which to extract data. The **Directory Depth** field enables you to extend your query up to ten (10) sub-directories below the one specified in the **Path** field.
Wildcard	Use wild cards to target a particular file type of naming scheme. For example, to search only Excel spread sheets, use `*.xls` or to search only files with names beginning with `finance_`, use `finance_*.*`
Include Inherited Permissions	Use to instruct the collector to not report permissions unless they are directly assigned. Only directly assigned permissions will be returned
Administrator	The administrator that has access to this share so you can collect permissions. This value should be the users principal user@xyz.com name or a fully qualified domain user name in the `domain\\user` format.
Password	The password associated with the specified administrator. The service will be running as System or can be configured to be run as any user, so the Administrator/Password fields may not be required in all cases.
Rules: Specify the rules used to transform and correlate the targets. **Note** Select the "**...**" icon to launch the Rule Editor to make changes to your rules if needed.	
Creation Rule	The rule used to determine how the unstructured data extracted from data source is transformed into data that can be read by IdentityIQ.
Correlation Rule	The rule used to determine how to correlate accounts (users and contacts) information from the application with identity cubes in IdentityIQ.
Provisioning related attributes: Select the settings for provisioning to the share.	
Override Default Pro-	Select it to override the default provisioning action for the collector.

Field	Description
visioning	
Provisioning Action	The overriding provisioning action for the collector.

To revoke permissions for Active Directory users and/or groups using Windows File Share Target Collector, perform the following:

1. Add the following attributes under target source configuration:

   ```
   <entry key="searchAttrForAcct" value="msDS-PrincipalName"/>
   <entry key="searchAttrForGrp" value="msDS-PrincipalName"/>
   ```

2. Remove the `NO_PERMISSIONS_PROVISIONING` feature string from the application configuration.

Partitioning Aggregation

With IdentityIQ version 8.1 Patch 4, 8.2, or 8.0p5, auto partitioning can be performed by going to **Configuration > Settings > Account** and selecting the **Allow Auto Partitioning** checkbox. For more information, refer to Account and Group Settings.

Configuring partitions manually

The Active Directory connector supports the Partitioning Aggregation feature to enable faster retrieval of Active Directory data.

In the Active Directory connector, data can be partitioned by specifying a `searchDN` and/or a `searchFilter` as a partition entry. The The Active Directory connector partition entries are the application configuration searchDNs list with each entry of the list treated as a single partition.

Typically, for a container-based partitioning of data, define the searchDNs or partition list as follows:

```
<entry key="searchDNs">
  <value>
     <List>
        <Map>
           <entry key="searchDN" value="ou=test1,DC=test,DC=sailpoint,DC=com"/>
          <entry key="iterateSearchFilter" value="(&(objectclass=user) )"/>
          <entry key="searchScope" value="SUBTREE"/>
        </Map>
        <Map>
           <entry key="searchDN" value="ou=test2,DC=test,DC=sailpoint,DC=com"/>
           <entry key="iterateSearchFilter" value="(&(objectclass=user) )"/>
          <entry key="searchScope" value="SUBTREE"/>
        </Map>
     </List>
</entry>
```

And for filter-based partition, define the searchDNs list or partition list as follows:

```
<entry key="searchDNs">
  <value>
     <List>
       <Map>
          <entry key="searchDN" value="DC=test,DC=sailpoint,DC=com"/>
         <entry key="iterateSearchFilter" value="(&(objectclass=user)(sn=a*))"/>
          <entry key="searchScope" value="SUBTREE"/>
     </Map>
       <Map>
         <entry key="searchDN" value="DC=test,DC=sailpoint,DC=com"/>
         <entry key="iterateSearchFilter" value="(&(objectclass=user) (sn=b*))"/>
          <entry key="searchScope" value="SUBTREE"/>
       </Map>
    </List>
</entry>
```

As seen above, in the first example, the OUs on which the search is performed are different although the `searchFilter` is the same. Whereas, in the second partitions entry, the OUs are same, but the `iterateSearchFilter` values are different. Since the required key values are similar, you could have both the above examples coupled together into the application configuration of a single Active Directory Connector application. Active Directory Connector combines the `searchDN` value and the `iterateSearchFilter` value and considers it as the partition context, avoiding any additional required configurations.

> **Note**
>
> Each of the partitions specified must be unique by way of the `searchDN` value or the `iterateSearchFilter` value. If not, the first partition is aggregated skipping the subsequent duplicate ones.
>
> When there is no defined partition list, the aggregation executes over the `baseDN` and the `iteraterSearchFilter` only, even though the task definition has partitioning enabled. Similarly, with a partition list defined and partitioning is not enabled on the task definition, IdentityIQ retrieves data from each `searchDN` entry in a sequential manner.

IQService Before/After Scripts

IdentityIQ provides most of the provisioning functionality for many systems through its connectors. Some systems provide better integration interface from Windows platform compared to other platforms. Hence connectors for such systems require IQService deployed on a Windows system. The IQService implementation performs the provisioning functions (such as Add User, Connect User to a Group) that are supported by the respective System. The IQService functions are called by the IdentityIQ connector implementation.

In addition to the basic action, some organizations may require supplementary actions performed by each function from Windows system. The IQService supports customization of the functions by allowing integrating before / after scripts implemented in any language. Following are some features of the IQService Before/After script:

- Centralized configurations (in IdentityIQ) for setting up Before/After scripts
- Supports Object Oriented scripting
- Script refers SailPoint library to get the request, result classes
- Can be executed with specific context
- Script can modify request/result

A script is a group of statements that perform one or more actions and manipulate request / result attributes. Scripts can be used to automate any required actions that are currently performed manually. Scripts called before processing the request are referred to as native before scripts and scripts called after processing the request are referred to as native after scripts.

The scripts needs to be defined in a Rule and then configured for an Application in IdentityIQ. Based on the rule type, the connector would send the scripts to IQService that needs to be executed before / after processing the request. The IQService supports executing before and after Rules for Create, Modify, and Delete request operations.

Writing a Script

IQService divides scripts in the following categories:

- Scripts with Object Oriented Support
- Scripts without Object Oriented Support

This guide provides the following sample before scripts:

- Sample PowerShell before script that modifies the value of an attribute and adds a new attribute to the request.
- Sample script that demonstrates how to read `AccountRequest` metadata (additional attributes passed with Account request/objectRequest object).
- Sample `connectorBeforeModify` script that demonstrates how to read `AccountRequest` arguments.
- Sample PowerShell after script that ensures that the request was processed successfully and creates home directory at the path specified in the request.
- Sample after script that displays how to access any application configuration attribute in the script.
- Sample batch after script that ensures that the request was processed successfully and creates home directory at the path specified in the request.

Scripts with Object Oriented Support

Scripting languages with Object Oriented capabilities (for example, PowerShell) are popular because of their simplistic nature and easy to use. These scripts can form objects of any type by referring any library/assembly implemented in any language and call its methods.

Native scripts implemented in these languages have easier and more powerful access to request and result objects. IQService comes with a class library named Utils.dll which bundles all required classes to access the request and result objects. The inputs provided to the script would be in the form of process environment variables. The following table describes the environment variables created by IQService:

Name	Type	Before Script	After Script
Application	System.Collections.Hashtable	Read Only	Read Only
Request	SailPoint.Utils.objects.AccountRequest	Read/Write	Read Only
Result	SailPoint.Utils.objects.ServiceResult	Not Available	Read/Write

The data in the environment variables is in XML. The script creates respective objects using Utils.dll. Once the object is modified, the script should convert it to XML by calling toxml() method of the object and write the xml to a file at the path that is passed as the only argument to the script. The script returns non-zero value in case of error and writes the error message in the file at the path that is passed as the argument to the script. This failure is communicated to IdentityIQ as part of result.

Sample PowerShell Before Scripts

- The following is a sample PowerShell before script that modifies the value of an attribute and adds one new attribute to the request:

```
# Refer to SailPoint class library Requires PowerShell v2 installed on the system.
Add-type -path utils.dll
# Read the environment variables
$sReader = New-Object System.IO.StringReader([System.String]$env:Request);

# Form the xml reader object
$xmlReader = [System.xml.XmlTextReader]([SailPoint.Utils.xml.XmlUtil]::getReader($sReader));

# Create SailPoint Request object
$requestObject = New-Object SailPoint.Utils.objects.AccountRequest($xmlReader);

# Loop through the attributes from the request for each ($attribute in $requestObject.AttributeRequests){
if($attribute.Name -eq "description"){
$attribute.value = "my description"; #change value of the attribute
}
}

# Add a new attribute to request
$attributeObject = New-Object SailPoint.Utils.objects.AttributeRequest;
$attributeObject.Name = "otherMobile";
$otherMobileValues = New-Object System.Collections.ArrayList;
$otherMobileValues.Add("222-292-2929");
$otherMobileValues.Add("333-292-2929");
$attributeObject.Value= $otherMobileValues;
$attributeObject.Operation = "Set";
$requestObject.AttributeRequests.Add($attributeObject);

# Write the request xml to file at the path passed as argument
$requestObject.toxml()|out-file $args[0];
```

- The following before script demonstrates how to read `AccountRequest` metadata (additional attributes passed with Account request/objectRequest object).

  The before/after script requires additional information to determine execution flow of the script that cannot be passed as `AttributeRequest` obejct because the out-of-the-box connector is not able to provision those attributes. Therefore attributes that are only used in scripts, must be sent as a metadata (Attributes) of `AccountRequest` instead of `AttributeRequests`.

  The following example of the `AccountRequest` xml displays how to pass this metadata in the request. In the following example the `country`, `city`, and `jobTitlepassing` attributes are passed as metadata:

```
<ProvisioningPlan nativeIdentity="TESTDOMAIN\User100" targetIntegration="AD">
  <AccountRequest application="AD" nativeIdentity="CN=User100,CN=Users,DC=TESTDOMAIN,DC=LOCAL" op="Enable">
```

```
    <Attributes>
      <Map>
        <entry key="flow" value="AccountsRequest"/>
        <entry key="interface" value="LCM"/>
        <entry key="operation" value="Enable"/>
        <entry key="provisioningPolicies">
          <value>
            <List>
              <String>ChangePassword</String>
            </List>
          </value>
        </entry>
        <entry key="country" value="US" />
        <entry key="city" value="Austin"/>
        <entry key="jobTitle" value="Sr Manager" />
      </Map>
    </Attributes>
    <AttributeRequest name="displayName" op="Add" value="SailorX"/>
  </AccountRequest>
</ProvisioningPlan>
```

- The following `connectorBeforeModify` script demonstrates how to read `AccountRequest` arguments:

  `Add-type -path "C:\Program Files\SailPoint\IQService\utils.dll"`

```
# Read the environment variables
$sReader = New-Object System.IO.StringReader([System.String]$env:Request);

# Form the xml reader object
$xmlReader = [System.xml.XmlTextReader]([SailPoint.Utils.xm-
l.XmlUtil]::getReader($sReader));

# Create SailPoint Request object
$requestObject = New-Object SailPoint.Utils.objects.AccountRequest($xmlReader);

# Read NativeIdentity of the AccountRequest
$nativeIdentity = $($requestObject.NativeIdentity)

# Get all attributes from the request
$attributesMap = $requestObject.Attributes

# Retrieve values from attributesMap
$city = $attributesMap["city"]
$country = $attributesMap["country"]
$jobTitle = $attributesMap["jobTitle"]

# Create AttributeRequest and assign group membership to user based on the loc-
ation and job title of the user
$memberOfList = @("OU=Enabled Users,DC=TESTDOMAIN,DC=LOCAL")
if( $city -eq "Austin" -And $jobTitle -eq "Sr Manager") {
$memberOfList.Add("CN=AustinManagers,OU=Managers,DC=TESTDOMAIN,DC=LOCAL")
}
if( $country -eq "US") {
$memberOfList.Add("CN=USEmployees,OU=Employees,DC=TESTDOMAIN,DC=LOCAL")
}
if( $memberOfList.Count -gt 0) {
$attributeObject = New-Object SailPoint.Utils.objects.AttributeRequest;
$attributeObject.Name = "memberOf";
$attributeObject.Value= $memberOfList;
$attributeObject.Operation = "Add";
$requestObject.AttributeRequests.Add($attributeObject);
}

# Write the request xml to file at the path passed as argument
$requestObject.toxml()|out-file $args[0];
```

Sample PowerShell After Scripts

- The following is a sample PowerShell after script that ensures that the request was processed successfully and creates home directory at the path specified in the request:

```
# Refer to SailPoint class library. Requires PowerShell v2 installed on the system.
Add-type -path E:\SVN\trunk\src\WinRPCGateway\IQService\bin\Debug\utils.dll

# Read the environment variables
$sReader = New-Object System.IO.StringReader([System.String]$env:Request);
$sResult = New-Object System.IO.StringReader([System.String]$env:Result);

# Form the xml reader objects
$xmlReader = [ System.xml.XmlTextReader]([sailpoint.utils.xml.XmlUtil]::getReader($sReader));
$xmlReader_Result = [ System.xml.XmlTextReader]([sailpoint.utils.xml.XmlUtil]::getReader($sResult));

# Create SailPoint objects
$requestObject = New-Object Sailpoint.Utils.objects.AccountRequest($xmlReader);
$resultObject = New-Object Sailpoint.Utils.objects.ServiceResult($xmlReader_Result);

#Check if the request was processed successfully
if($resultObject.Errors.count -eq 0){

#Get Home directory path
foreach ($attribute in $requestObject.AttributeRequests){

#Create Home directory
if($attribute.Name -eq "TS_TerminalServicesHomeDirectory"){
new-item $attribute.Value -itemtype directory;
}
}
}
```

- The following after script displays how to access any application configuration attribute in the script.

  The following example reads application attributes with name `Office365Username` and `password` and uses them to connect to Exchange Online and sets Mailbox properties:

```
# Refer to SailPoint class library.
Add-type -path C:\Program files\IQService\bin\Debug\utils.dll

# Read the environment variables
$sReader = New-Object System.IO.StringReader([System.String]$env:Request);
$sResult = New-Object System.IO.StringReader([System.String]$env:Result);

# Form the xml reader objects
$xmlReader = [ System.xml.XmlTextReader]([sailpoint.utils.xml.XmlUtil]::getReader($sReader));
$xmlReader_Result = [ System.xml.XmlTextReader]([sailpoint.utils.xml.XmlUtil]::getReader($sResult));

# Create SailPoint objects
$requestObject = New-Object Sailpoint.Utils.objects.AccountRequest($xmlReader);
$resultObject = New-Object Sailpoint.Utils.objects.ServiceResult($xmlReader_Result);

# Retrive nativeIdentity from request object
$nativeIdentity = $requestObject.NativeIdentity

# Get xmlFactory object to retive application configuration
$xmlFactory = [sailpoint.Utils.xml.XmlFactory]::Instance;

# Read the environment variables
$sReader1 = $env:Application

# Retrive application configuration object
$appObject = $xmlFactory.parseXml($sReader1)

#Retrive application configuration entries named Office365username and password
value from AzureAD application config
$office365AdminUsername = $appObject.Office365username

#Retrive password attribute value
$o365Password = $appObject.password

#create Credential object
$secpasswd = ConvertTo-SecureString $o365Password -AsPlainText -Force
$cred = New-Object System.Management.Automation.PSCredential ($office365AdminUsername, $secpasswd)

```

```
#Connect to Office365
Import-Module msonline
Connect-MsolService -Credential $cred

#Connect Exchange-Online
$msoExchangeURL = "https://ps.outlook.com/powershell/"
$session = New-PSSession -ConfigurationName Microsoft.Exchange -ConnectionUri
$msoExchangeURL -Credential $cred -Authentication Basic -AllowRedirection
Import-PSSession $session

# Set mailbox properties
Set-MailBox -identity $nativeIdentity -UseDatabaseQuotaDefaults $false -
IssueWarningQuota "200MB" -ProhibitSendQuota "250MB" -ProhibitSendReceiveQuota
"280MB"
```

Scripts without Object Oriented Support

Non Object Oriented scripts do not support referring to the class library or a way of parsing XML. To have easy access to each attribute along with their operation and values, IQService creates process environment variables for each of the application and request attributes with names in the form `SP_<OPERATION>_<NAME>` for requests and `SP_APP_<NAME>` for applications. For native identity, the environment variable would be `SP_NativeIdentity`. These types of scripts have limited access to descriptive results and get only *SUCCESS* or *FAIL* in the `Result` environment variable. Therefore the after scripts implemented using these scripting languages cannot modify any attribute/result. The before scripts can add, modify, or remove any attribute from the request. The script needs to write the newly added or modified attribute to the file at the path passed as an argument to the script in the form `SP_<OPERATION>_<NAME>->=<VALUE>`. For removing the attribute from the request, write `/~<ATTRIBUTE_NAME>` to the file. Value for the multivalued attribute is delimited by **/#/**.

The following is a sample batch after script that ensures that the request was processed successfully and creates home directory at the path specified in the request:

```
IF %Result% ==SUCCESS md %SP_Set_TS_TerminalServicesHomeDirectory%
```

Creating a Rule

IdentityIQ (6.0) user interface does not have facility to create Native Rule applicable for IQService. Create a rule with any supported type from the user interface. Add the script to the Rule source and save the Rule. Navigate to the debug page, open the newly created Rule and perform the following steps:

1. Change the rule type to one of the following types as appropriate:

Type Name	Description
ConnectorBeforeCreate	Before script for creation operation.
ConnectorAfterCreate	After script for create operation.
ConnectorBeforeModify	Before script for modify operation that includes enable, disable, and unlock.
ConnectorAfterModify	After script for modify operation that includes enable, disable, and unlock.
ConnectorBeforeDelete	Before script for deletion operation.
ConnectorAfterDelete	After script for delete operation.

2. Use the following format to add attributes to the Rule in the form:

```
<Attributes>
    <Map>
        <entry key=<Name> value=<Value>/>
    </Map>
</Attributes>
```

Add the following attributes to the Rule in the form:

Name	Description	Default Value
objectOrientedScript	Whether the rule source uses object oriented scripting.	False
disabled	Set to true if the rule should not be executed on the IQService side.	False
extension	Extension of the script.	.bat
program	Program/application that can execute this type of script. Ensure that this program is installed on the system where IQService is running and i properly configured to execute the scripts.	cmd.exe or cmd
timeout	Time interval (in seconds) that IQService should wait for script to return. After this interval, IQService aborts the script.	10

Configuring the Rules in Application

With this releases, IdentityIQ user interface does not have facility to configure Native Rule applicable for IQService in Application. Navigate to the debug page, open the application and add `<nativeRules>` under Attributes map with list of names of the Rules that must be configured for this application.

For example:

```
<entry key="nativeRules">
```

```
  <value>
    <List>
      <String>AfterCreate-Powershell</String>
      <String>BeforeCreate-Powershell</String>
      <String>BeforeModify-Batch</String>
    </List>
  </value>
</entry>
```

Delta Aggregation

> **Note**
> This includes changes such as when a user or group has been added, updated, or deleted on the managed system.

By default, Active Directory supports the **DirSync** mode of delta aggregation which is based on DirSync feature of Active Directory.

Testing Delta Aggregation

For delta aggregation to work properly, a start point is required from where the system detects changes. To retrieve changes from the last iteration, a full aggregation must be performed first during which the reference point is maintained. Once the full aggregation completes, create a separate delta aggregation task to retrieve delta changes that occurred post the full aggregation.

Perform the following steps to test delta Aggregation:

1. Execute Account and Account - Group Aggregation task.

2. Create a task with delta aggregation flag set for Account and Account - Group Aggregation.

3. Perform Create/Update/Delete/Revoke operations for Accounts/Groups on the directory server.

4. Execute the respective delta aggregation task.

5. Confirm the changes have been retrieved into IdentityIQ.

Parameter References

The following topics contain the parameter reference topics. These provide details on the important and default parameters for configuration, provisioning, or schemas.

Identity Attributes

SailPoint requires certain attributes remain in your configuration. These attributes are referred to as Identity Attributes, and they must not be updated. If you update these attributes from their default values, the connector may fail. To resolve any issues caused by changing Identity Attributes, re-configure them to their default values. The following table lists the Identity Attributes for this connector:

Identity Attribute	Schema Object Type
distinguishedName	Account
distinguishedName	Group

Default Provisioning Attributes Reference

This page details the default provisioning attributes for your connector.

Account Creation

> **Note**
>
> - For an account that has been moved or renamed in Active Directory since last aggregation, ensure that the change is aggregated before performing any provisioning operation on the account.

Account Attribute	Description
ObjectType	The type of account to be created. The default is **User**. • For users, the object type must be **User**. • For contacts, the object type must be **Contact**.

Account Attribute	Description
	- For group managed service accounts, the object type must be **msDS-GroupManagedServiceAccount**. - For managed service accounts, the object type must be **msDS-ManagedServiceAccount**.
distinguishedName	Distinguished name of the new account.
sAMAccountName	sAMAccountName of the new account.
manager	Manager for the new account.
mail	Email address of the new account.
password	Password for the new account.
givenName	First name associated with the account.
sn	Last name associated with the account.
pwdLastSet	This attribute can only be set as `true` or `false`. - When set to `true`, the `pwdLastSet` attribute value is set to `0` and it selects the **User must change password on logon** checkbox for the Active Directory user object's account in ADUC. - When set to `false`, the `pwdLastSet` attribute value is set to `-1` and sets this attribute to the current time, and it deselects the **User must change password on logon** checkbox. The default Static Value is false.
IIQ Disabled	This attribute can only be set as `true` or `false`. Set to `true` to create a disabled user.
primaryGroupDN	Default group of the new account.
description	Description of the new account.
telephoneNumber	Telephone number of the new account. The default Attribute is Alternate Phone Number (phone).

Exchange Mailbox Attributes

Note the following when working with mailbox attributes:

- If you send an email address in the mail attribute, the exchange may not use it, if the E-mail Policy in the exchange is set to create it differently. The email address is not taken and sent back to Active Directory after it is created, based on the policy.

- For the Active Directory source, the `mailNickname`, `homeMBD`, and `msExchHideFromAddressLists` attributes are case insensitive when processed by the IQService.
- The Active Directory source sets the MS-Exchange attributes - `homeMDB` and `mailNickname` as AD attributes, if MS-Exchange is not enabled.

The following are additional attributes required to create a mailbox:

Attribute	Description
homeMDB	The exchange mailbox store domain name required to create a mailbox. For example: `SomeExchangeDB`. The `homeMDB` attribute is in the format of `SomeExchangeDB', not 'CN=SomeExchangeDB`.
mailNickname	The exchange alias that you can use to update or disable the mailbox. For example: `firstname.lastname`
msExchHideFromAddressList	The attribute to hide from the Exchange address lists.
externalEmailAddress	The external email address, required for mail contact creation.

Updating Exchange Mailbox Attributes

The Active Directory connector supports updating any Exchange mailbox attributes supported by set-mailbox cmdlet, using the following methods:

1. Add the attribute in the provisioning policy with `Exch_` as a prefix. For example, to set the `HiddenFromAddressListsEnabled` exchange attribute, add the attribute name as `Exch_HiddenFromAddressListsEnabled` in the provisioning policy.
2. Alternatively, this can be done by editing the application xml file by adding an application attribute named `exchangeAttributes` of string type with a comma separated name of the Exchange attributes added in provisioning policy.

   For example, for the HiddenFromAddressListsEnabled attribute, add the following to the debug page:

   ```
   <entry key="exchangeAttributes" value="HiddenFromAddressListsEnabled,
   UseDatabaseQuotaDefaults"/>
   ```

Attributes for Skype for Business

The `msRTCSIP-UserEnabled` attribute must be updated as part of the Create Profile section.

By default, provisioning of the following attributes is supported:

Attribute	Description
SipAddress	This attribute contains the SIP address of a given user.
SipDomain	This attribute contains the SIP domain of a given user.
SipAddressType	This attribute contains the SIP address type of a given user. Skype for Business Server generates a SIP address for the new user when SipAddressType is provided in combination with SipDomain.
Registrar Pool	This attribute contains the Registrar pool of a given user.
msRTCSIP-UserEnabled	This attribute indicates whether the user is currently enabled for Microsoft Lync\Skype for Business Server.

Schema and Provisioning Attributes of Group Managed Service Accounts (gMSA) Object

For the provisioning of the following gMSA attributes, you must add them manually for the existing sources. By default, they are available for new sources.

Account Attribute	Description
dNSHostName	The DNS host name of the service account. This attribute is mandatory for gMSA provisioning.
msDS-SupportedEncryptionTypes	The supported encryption types for the service account. This is a multi-valued attribute.
msDS-ManagedPasswordInterval	The number of the days for the password change interval.
msDS-GroupMSAMembership	The principals that are allowed to retrieve Managed Password of this Group-Managed Service Account. This is a multi-valued attribute.
msDS-AllowedToActOnBehalfOf OtherIdentity	The accounts that can act on the behalf of this Group Managed Service Account. This is a multi-valued attribute.
servicePrincipalName	The service principal names for the service account. This is a multi-valued attribute.

Provisioning Attributes for Contacts

Add the `displayAttributeForContacts` attribute as additional parameter for Contacts. CN is used as the default value for display name of Contact objects. The Display attribute can be set using the `connector_displayAttributeForContact` config attribute.

Provisioning Attribute for Resource Forest Topology Exchange Management

The following String-type attribute required for creating Linked Mailbox, is available by default, for the new sources. For existing sources, add manually in the **Create Profile** section.

Account Attribute	Description
shadowAccountDN	Distinguished Name of the Linked Mailbox Shadow Account to be created. It is required for creating new Linked Mailbox.

Provisioning Attributes for Terminal Services

By default the Terminal Services/Remote Desktop Services attributes are not added to the schema and provisioning policy for performance optimization. To manage Terminal Services attributes, add these attributes to schema and provisioning policy. Alternatively, you can uncomment these attributes from the connector registry and import it again.

Account Attribute	Description
TS_TerminalServicesProfilePath	The roaming or mandatory profile path to be used when the user logs on to the RD Session Host server.
TS_TerminalServicesHomeDrive	The root drive for the user.
TS_TerminalServicesHomeDirectory	The root directory for the user.
TS_TerminalServicesInitialProgram	The path and file name of the application that the user wants to start automatically when the user logs on to the RD Session Host server.
TS_TerminalServicesWorkDirectory	The working directory path for the user.
TS_EnableRemoteControl	A value that specifies whether to allow remote observation or remote control of the user's Remote Desktop Services session.
TS_AllowLogon	A value that specifies whether the user is allowed to log on to the RD Session Host server.
TS_BrokenConnectionAction	A value that specifies the action to be taken when a Remote Desktop Services session limit is reached.
TS_ReconnectionAction	A value that specifies if reconnection to a disconnected Remote Desktop Services session is allowed.
TS_ConnectClientDrivesAtLogon	A value that specifies if mapped client drives should be reconnected when a Remote Desktop Services session is started.
TS_ConnectClientPrintersAtLogon	A value that specifies whether to reconnect to mapped client printers at logon. The value is one if reconnection is enabled, and zero if reconnection is disabled.

Account Attribute	Description
TS_DefaultToMainPrinter	A value that specifies whether to print automatically to the client's default printer. The value is one if printing to the client's default printer is enabled, and zero if it is disabled.
TS_MaxConnectionTimeout	The maximum duration of the Remote Desktop Services session, in minutes. After the specified number of minutes have elapsed, the session can be disconnected or terminated.
TS_MaxDisconnectionTime	The maximum amount of time, in minutes, that a disconnected Remote Desktop Services session remains active on the RD Session Host server. After the specified number of minutes have elapsed, the session is terminated.
TS_MaxIdleTime	The maximum amount of time that the Remote Desktop Services session can remain idle, in minutes. After the specified number of minutes has elapsed, the session can be disconnected or terminated.

Individual Attribute Notes

accountExpires Attribute

For the Active Directory source, the `accountExpires` attribute must be defined as a string. The value of the `accountExpires` attribute can be set in the Microsoft defined timestamp that represents the number of 100-nanosecond intervals since January 1, 1601 (UTC).

The value can also be entered in a human readable format: `MM/DD/YYYY HH:MM:SS AM TimeZone`. For example, 05/11/2019 12:00:00 AM IST. A value of `0`, `never`, or `9223372036854775807` indicates that the account never expires.

The value of the `accountExpires` attribute is displayed in the MM/DD/YYYY hh:mm:ss aa Z format. For example, if previously the time of account expiry was displayed as 5/14/2019 12:0:0 AM IST, it will now be displayed as 05/14/2019 12:00:00 AM IST.

'Never' as a Value of accountExpires Attribute

The Active Directory source supports `never` as a value of the `accountExpires` attribute in provisioning, when the `timeZone` attribute is present in the source configuration.

> **Note**
> SailPoint recommends that the `accountExpires` attribute must be defined as a string. However, the Active Directory source accepts an integer value for the `accountExpires` attribute in account provisioning if it is not a string.

Rollback of Created Account

The Active Directory source supports rollback of created account in case provisioning of one or more requested attributes fails during the provisioning operation. Set the `rollbackCreatedAccountOnError` attribute to True.

Schema Attributes Reference

The out-of-the-box schema attributes must be defined as string if not specified.

> **Note**
> Attributes with an asterisk (*) are the Terminal Services/Remote Desktop Services attributes. By default, these attributes are not added to the schema and provisioning policy for performance optimization. To manage Terminal Services attributes, add these attributes to schema and provisioning policy. Alternatively, you can uncomment these attributes from the connector registry and import it again.

Account Attributes

businessCategory

The types of business performed by an organization. Each type is one value of this multi-valued attribute. For example, "engineering", "finance", and "sales".

carLicense

This attribute type contains the license plate or vehicle registration number associated with the user.

cn

This attribute type contains names of an object. Each name is one value of this multi-valued attribute. If the object corresponds to a person, it is typically the person's full name.

For example, "Martin K Smith", "Marty Smith" and "printer12".

departmentNumber

This attribute contains a numerical designation for a department within your enterprise.

description

This attribute type contains human-readable descriptive phrases about the object. Each description is one value of this multi-valued attribute.

For example, "Updates are done every Saturday, at 1am.", and "distribution list for sales".

destinationIndicator

This attribute type contains country and city strings associated with the object (the addressee) needed to provide the Public Telegram Service. The strings are composed in accordance with CCITT Recommendations F.1 [F.1] and F.31 [F.31]. Each string is one value of this multi-valued attribute.

For example, "AASD" as a destination indicator for Sydney, Australia. "GBLD" as a destination indicator for London, United Kingdom.

The directory does not ensure that values of this attribute conform to the F.1 and F.31 CCITT recommendations. It is the application's responsibility to ensure destination indicators that it stores in this attribute are appropriately constructed.

displayName

This attribute contains the preferred name to be used for this person throughout the application.

distinguishedName

This attribute contains the distinguished name by which the user is known. This default attribute must not be changed for a provisioning operation.

employeeNumber

This attribute contains the numerical identification key for this person within your enterprise.

employeeType

This attribute contains a descriptive type for this user.

For example, contractor, full time, or part time.

externalEmailAddress

This attribute contains external email address of the mail user. Mail user is an AD user having mailbox outside of organization.

facsimileTelephoneNumber

This attribute type contains telephone numbers and any required parameters for facsimile terminals. Each telephone number is one value of this multi-valued attribute.

givenName

This attribute type contains name strings that are the part of a person's name that is not their surname. Each string is one value of this multi-valued attribute.

For example, "John", "Sue", and "David".

homeMDB

Exchange mailbox store DN. Required for mailbox creation.

homePhone

This attribute contains the employees home phone number.

homePostalAddress

This attribute contains the employees mailing address.

initials

This attribute type contains strings of initials of some or all of an individual's names, except the surname(s). Each string is one value of this multi-valued attribute.

For example, "J. A." and "J"

internationalISDNNumber

This attribute type contains Integrated Services Digital Network (ISDN) addresses, as defined in the International Telecommunication Union (ITU) Recommendation E.164 [E.164]. Each address is one value of this multi-valued attribute.

For example, "0198 444 444".

l

This attribute type contains names of a locality or place, such as a city, county, or other geographic region. Each name is one value of this multi-valued attribute.

For example, "Austin", "Chicago", and "Brisbane".

mail

This attribute type contains the RFC822 mailbox for the user.

mailNickname

Exchange Alias.

manager

This attribute type contains the distinguished name of the manager to whom this person reports.

memberOf

This attribute type contains the account group membership for this person on the application.

mobile

This attribute type contains the mobile telephone number of this person.

msDS-PrincipalName

Name of the entity in the following format:

```
NetBIOS domain name\sAMAccountName
```

msExchHideFromAddressLists

Hide from Exchange address lists.

msNPAllowDialin

Indicates whether the account has permission to dial in to the RAS server.

msNPCallingStationID

If this property is enabled, the server verifies the caller's phone number. If the caller's phone number does not match the configured phone number, the connection attempt is denied.

msRADIUSCallbackNumber

The phone number that is used by the server is set by either the caller or the network administrator. If this property is enabled, the server calls the caller back during the connection process.

msRADIUSFramedIPAddress

Use this property to assign a specific IP address to a user when a connection is made.

msRADIUSFramedRoute

Define a series of static IP routes that are added to the routing table of the server running the Routing and Remote Access service when a connection is made.

o

This attribute type contains the names of an organization. Each name is one value of this multi-valued attribute.

objectClass

The values of the objectClass attribute describe the kind of object which an entry represents. The objectClass attribute is present in every entry, with at least two values. One of the values is either "top" or "alias".

objectguid

Globally unique identifier of the object.

objectSid

Windows Security Identifier.

objectType

Indicates type of the Active Directory objects.

For example, User, Contact

ou

This attribute type contains the names of an organizational unit. Each name is one value of this multi-valued attribute.

For example, "Sales", "Human Resources", and "Information Technologies".

pager

This attribute type contains the telephone number of this person's pager.

physicalDeliveryOfficeName

This attribute type contains names that a Postal Service uses to identify a specific post office.

For example, "Austin, Downtown Austin" and "Chicago, Finance Station E".

postalAddress

This attribute type contains addresses used by a Postal Service to perform services for the object. Each address is one value of this multi-valued attribute.

For example, "1111 Elm St.$Austin$Texas$USA".

postalCode

This attribute type contains codes used by a Postal Service to identify postal service zones. Each code is one value of this multi-valued attribute.

For example, "78664", to identify Pflugerville, TX, in the USA.

postOfficeBox

This attribute type contains postal box identifiers use by a postal service to locate a box on the premises of the Postal Service rather than a physical street address. Each postal box identifier is a single value of this multi-valued attribute.

For example, "Box 27".

preferredDeliveryMethod

This attribute type contains an indication of the preferred method of getting a message to the object.

For example, If the mhs-delivery Delivery Method is preferred over telephone-delivery, which is preferred over all other methods, the value would be: "mhs $ telephone".

preferredLanguage

This attribute type contains the preferred written or spoken language of this person.

registeredAddress

This attribute type contains postal addresses to be used for deliveries that must be signed for or require a physical recipient. Each address is one value of this multi-valued attribute.

For example, "Receptionist$XYZ Technologies$6034 Courtyard Dr. $Austin, TX$USA".

roomNumber

This attribute type contains the room or office number or this person's normal work location.

sAMAccountName

This attribute type contains the sAMAccountName for this user.

secretary

This attribute type contains the distinguished name of this person's secretary.

seeAlso

This attribute type contains the distinguished names of objects that are related to the subject object. Each related object name is one value of this multi-valued attribute.

For example, the person object "cn=Elvis Presley,ou=employee,o=XYZ\, Inc." is related to the role objects "cn=Bowling Team Captain,ou=sponsored activities,o=XYZ\, Inc." and "cn=Dart Team,ou=sponsored activities,o=XYZ\, Inc.". Since the role objects are related to the person object, the `seeAlso` attribute contains the distinguished name of each role object as separate values.

sIDHistory

(*Optional*) User can add this attribute manually to view the data in a readable string format.

sn

This attribute type contains name strings for surnames, or family names. Each string is one value of this multi-valued attribute.

For example, "Smith".

st

This attribute type contains the full names of states or provinces. Each name is one value of this multi-valued attribute.

For example, "Texas".

street

This attribute type contains site information from a postal address (that is, the street name, place, avenue, and the house number). Each street is one value of this multi-valued attribute.

For example, "15 Main St.".

telephoneNumber

This attribute type contains telephone numbers that comply with the ITU Recommendation E.123 [E.123]. Each number is one value of this multi-valued attribute.

teletexTerminalIdentifier

The withdrawal of Recommendation F.200 has resulted in the withdrawal of this attribute.

telexNumber

This attribute type contains sets of strings that are a telex number, country code, and answer back code of a telex terminal. Each set is one value of this multi-valued attribute

title

This attribute type contains the persons job title. Each title is one value of this multi-valued attribute.

For example, "Vice President", "Software Engineer", and "CEO".

uid

This attribute type contains computer system login names associated with the object. Each name is one value of this multi-valued attribute.

For example, "s9709015", "admin", and "Administrator".

> **Note**
> **NetBIOSName** is the domain NetBIOS Name of the account. To aggregate this attribute, add NetBIOSName as schema attribute under Schema.

Group Attributes

cn

This attribute type contains names of an object. Each name is one value of this multi-valued attribute. If the object corresponds to a person, it is typically the person's full name.

For example, "Martin K Smith", "Marty Smith" and "printer12".

description

This attribute type contains human-readable descriptive phrases about the object. Each description is one value of this multi-valued attribute.

For example, "Updates are done every Saturday, at 1am.", and "distribution list for sales".

distinguishedName

This attribute contains the distinguished name by which the user is known.

This is an Identity Attribute which must not be changed.

GroupScope

This attribute type contains the group scope.

GroupType

This attribute type contains the group type.

mailNickname

Exchange distribution group name.

memberOf

This attribute type contains the group membership for this person on the application.

msDS-PrincipalName

Name of the entity in the following format:

```
NetBIOS domain name\sAMAccountName
```

objectSid

Windows Security Identifier.

objectguid

Globally unique identifier of the object.

owner

This attribute type contains the owner of the object.

sAMAccountName

This attribute type contains the sAMAccoutName for this group.

> **Note**
> **NetBIOSName** is the domain NetBIOS Name of the group. To aggregate this attribute, add NetBIOSName as schema attribute under Group Schema.

Attributes for Terminal Services

TS_TerminalServicesProfilePath*

The roaming or mandatory profile path to be used when the user logs on to the RD Session Host server.

TS_TerminalServicesHomeDrive*

The root drive for the user.

TS_TerminalServicesHomeDirectory*

The root directory for the user.

TS_TerminalServicesInitialProgram*

The path and file name of the application that the user wants to start automatically when the user logs on to the RD Session Host server.

TS_TerminalServicesWorkDirectory*

The working directory path for the user.

TS_EnableRemoteControl*

A value that specifies whether to allow remote observation or remote control of the user's Remote Desktop Services session.

TS_AllowLogon*

A value that specifies whether the user is allowed to log on to the RD Session Host server.

TS_BrokenConnectionAction*

A value that specifies the action to be taken when a Remote Desktop Services session limit is reached.

TS_ReconnectionAction*

A value that specifies if reconnection to a disconnected Remote Desktop Services session is allowed.

TS_ConnectClientDrivesAtLogon*

A value that specifies if mapped client drives should be reconnected when a Remote Desktop Services session is started.

TS_ConnectClientPrintersAtLogon*

A value that specifies whether to reconnect to mapped client printers at logon. The value is one if reconnection is enabled, and zero if reconnection is disabled.

TS_DefaultToMainPrinter*

A value that specifies whether to print automatically to the client's default printer. The value is one if printing to the client's default printer is enabled, and zero if it is disabled.

TS_MaxConnectionTime*

The maximum duration of the Remote Desktop Services session, in minutes. After the specified number of minutes have elapsed, the session can be disconnected or terminated.

TS_MaxDisconnectionTime*

The maximum amount of time, in minutes, that a disconnected Remote Desktop Services session remains active on the RD Session Host server. After the specified number of minutes have elapsed, the session is terminated.

TS_MaxIdleTime*

The maximum amount of time that the Remote Desktop Services session can remain idle, in minutes. After the specified number of minutes has elapsed, the session can be disconnected or terminated.

Microsoft Lync\Skype for Business Server attributes

msRTCSIP-UserEnabled

Whether the user is currently enabled for Microsoft Lync\Skype for Business Server.

DialPlan

Name of the user DialPlan.

LineServerURI

The line server URL.

EnabledForFederation

Whether a user is enabled for federation.

PublicNetworkEnabled

Whether a user is enabled for access outside network.

EnterpriseVoiceEnabled

Whether a user EnterpriseVoiceEnabled service is enabled.

LineURI

The line Uniform Resource Identifier (URI).

SipAddress

This attribute contains the SIP address of a given user.

VoicePolicy

The name of Voice Policy.

MobilityPolicy

The name of Mobility Policy.

ConferencingPolicy

The name of Conferencing Policy.

PresencePolicy

The name of Presence Policy.

VoiceRoutingPolicy

The name of VoiceRouting Policy.

RegistrarPool

The name of registrar pool.

LocationPolicy

The name of Location Policy.

ClientVersionPolicy

The name of ClientVersion Policy.

ClientPolicy

The name of Conferencing Policy.

ExternalAccessPolicy

The name of ExternalAccess Policy.

HostedVoicemailPolicy

The name of HostedVoicemail Policy.

PersistentChatPolicy

The name of PersistentChat Policy.

UserServicesPolicy

The name of UserServices Policy.

ExperiencePolicy

The name of Experience Policy.

ArchivingPolicy

The name of Archiving Policy.

LegalInterceptPolicy

The name of LegalIntercept Policy.

PinPolicy

The name of Pin Policy.

LyncPinSet

Whether a user pin is set.

LyncPinLockedOut

Whether a user pin is locked. m

Managed Service Account Attributes

Only these attributes are certified for provisioning and read operations for managing Managed Service Accounts and Group Managed Service Accounts.

msDS-AllowedToActOnBehalfOfOtherIdentity

Accounts that can act on the behalf of this Group Managed Service Account. Values of this multi valued attribute must be in Distinguished Name format.

IQService is required to read and provision this property.

msDS-GroupMSAMembership

Principals allowed to use this Group Managed Service Account. Values of this multi valued attribute must be in Distinguished Name format.

IQService is required to read and provision this property.

msDS-ManagedPasswordInterval

Interval in days after which Active Directory changes the password of the Managed Service Account.

msDS-SupportedEncryptionTypes

Supported Encryption Types for the Managed Service Account. This attribute can have multiple values.

For example, RC4, AES128, AES25

servicePrincipalName

Service principal names for the Managed Service Account. This attribute is multi valued.

For example, MyService/Host1.example.com

Troubleshooting

If you encounter any of the following issues or errors, SailPoint recommends that you follow the guidance provided below to resolve the error before contacting SailPoint Support.

For more information, refer to the Active Directory Connector - FAQ and Troubleshooting document.

Authentication Error

Error occurred while authentication user: Xyz

```
Unable to use PTA with SASL
```

Resolution: Check the UPN name being used.

Kerberos Authentication requires `userPrincipalName` as the primary requirement, and due to differences in the domain for `userPrincipalName`, the PTA authentication is failing. You must use the correct domain.

For example, if you have two domains:

```
Abc.com
```

```
xyz.abc.com
```

and you want to perform PTA for users present at `xyz.abc.com`, the format should be `username@xyz.abc.com`.

Organizational Units (OU) Child Objects Don't Update Following a Delta Aggregation

If you make a change to an OU which contains accounts or groups, such as renaming or moving it, a delta aggregation doesn't pick up the changes. This is a limitation in Microsoft DirSync Control explained here: https://docs.microsoft.com/en-us/windows/win32/ad/polling-for-changes-using-the-dirsync-control.

Resolution: Perform a full aggregation to capture the changes and update the child objects. You might have to do this regularly to ensure the data is up to date.