Active Directory Connector

Version: 8.2

Copyright and Trademark Notices

Copyright © 2021 SailPoint Technologies, Inc. All Rights Reserved.

Contents

Supported Features

The Active Directory Connector provides the ability to provision users, groups, contacts, and entitlements. 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, Delete • Enable, Disable, Unlock, Change Password • Add/Remove Entitlements (includes Foreign Security Principals) • Terminal Services, Dial-in Attributes • Create, Update, Delete Exchange User Mail Box • Create, Update, Delete Exchange Mail User • Create, Update, Delete Skype for Business user • Enable/disable, setting policies for Skype for Business user • Reset Skype for Business user PIN • Password Interceptor
Active Directory Contacts	• Manages Active Directory Contacts as Accounts • Aggregation, Delta Aggregation, Partitioning Aggregation, Refresh Account • Create, Update, Delete • Add/Remove Entitlements • Create, Update, Delete Exchange Mail Contact
Active Directory Service Accounts (Managed Service Accounts/Group Managed Service Accounts)	• Aggregation, Partitioning Aggregation, Refresh Account • Create, Update, Delete • Add/Remove Entitlements

Account - Group Management

- Manages Active Directory Groups as Account-Groups
- Aggregation, Delta Aggregation, Refresh Group
- Create, Update, Delete
- Create, 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 Additional Information.
- 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 Configuration Parameters.
- Supports reusing of Ticket Granting Tickets (TGT) for Kerberos authentication during aggregation tasks.
  To resort back to the earlier implementation (non-cached) an additional attribute named **adSystemConfUseUpdatedSASLCommunication** attribute can be added to the system configuration. For more information, see Additional Configuration Parameters.

References

- "Unstructured Target Collector” on page 39
- “Delta Aggregation”

- “Partitioning Aggregation”
- IQService

Supported Managed System

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
- Microsoft Windows Server 2008
- Microsoft Windows Server 2003

> With this release of IdentityIQ, SailPoint IdentityIQ Active Directory Connector now provides support for Microsoft Windows Server version 2019 without any change in the Domain Services (AD DS) functional levels.

AWS Managed Microsoft Active Directory Service

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

1. Before you start using the connector, it is required that the IQService is installed and registered on any Windows system with any of the supported Operating System. For more information on installing and registering IQService, see IQService.

   > If 'Authentication Type' is set to 'Strong' then IQService host must be in the same domain or in trusted domain.

2. For Exchange and Skype for Business management, Windows PowerShell version 3.0 or above must be configured on system running IQService.
3. For managing Terminal Services (Remote Desktop Services profile) attributes, install the IQService on a Server class Windows Operating System.
4. For application managing multiple domain trees either from same or different forests, there must be two way trust relationship between them.
5. For managing Managed Service Accounts (MSA)/Group Managed Service Accounts (GMSA) following prerequisites are required:
   a. For reading **msDS-GroupMSAMembership** and **msDS-AllowedToActOnBehalfOfOtherIdentity** GMSA object properties, IQService is required and Active Directory PowerShell Module must be enabled on the IQService Host.
   b. For Provisioning operations of MSA and GMSA objects, IQService is required and Active Directory PowerShell Module must be enabled on the IQService Host.

Securing Active Directory Application

Securing Active Directory application can be obtained using the following communication paths that are involved based on the various 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 (*) symbol represents:

* 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 allows 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 is persisted on 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 “IQService public key exchange task.

From there on out IdentityIQ and IQService switch to use TLS for encrypting the XML data payload.

> SailPoint recommends to secure every communication path for Active Directory application by following the configurations mentioned below.

Securing communication path between IdentityIQ and Active Directory Domain Controller/ Target system

To secure TLS connection for Active Directory, TLS communication must be enabled between Active Directory Connector and 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.

To create the TLS communication:

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. Click **Test Connection** and then **Save**.

> As part of recommended practice, FQDN of Active Directory host must be specified (instead of IP address) in 'Servers' field when 'Use TLS' option is selected under “Domain Configuration”.

Securing communication path between IdentityIQ and IQService

- TLS Communication
- Client Authentication

TLS Communication

The Active Directory Connector supports TLS communication for IQService.

> Client Authentication is mandatory for operations that use the IQService when the TLS Communication. Before setting up the 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>`

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.

Client Authentication

The Active Directory Connector supports client authentication for IQService which ensures that IQService is communicating with legitimate IdentityIQ.

> For client authentication, configure the IQService with the following command:
> `IQService.exe -a <Domain User/s>`

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

For more information on the client authentication, see IQService.

Securing communication path between IQService and Active Directory Domain Controller/ Target system

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

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

Administrator Permissions

Service Account

The Service Account must have appropriate rights on the Active Directory. The Domain Controller must be accessible from the IQService host computer.

> The rights 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 rights 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 rights 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 rights will depend on the use cases that are implemented, but could include:

- Read All Properties
- Write All Properties
- Create User Objects
- Delete User Objects
- Change Password
- Reset Password
- Read Members
- Write Members

Strong Authentication (SASL)

For Strong authentication (SASL), single service account can be used for multiple domains/forest. For this:

- 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 Service Account using Delegation Control Wizard as follows:

  a. Open **Active Directory Users and Computers**.

  b. Right click on the **domain** and select **Delegate Control to open Delegation of Control Wizard** and click **Next**.

  c. Add Service Account user using **Add** button and click on **Next**.

  d. Select **Create a custom task to delegate** and click **Next**.

e. Select Only the following objects in the folder option and select User Objects, Contact Objects and Group Objects and Create/Delete the selected objects in the folder.

f. On the next screen, select **Full Control** under **Permissions** and click on **Next** and then **Finish**.

Foreign Security Principals

For Foreign Security Principals (FSPs) to be aggregated, created, and modified single service account must have full delegated permissions on FSP container. Permissions must be delegated to Service Account using **Delegation Control Wizard** as follows:

1. Open **Active Directory Users and Computers**.
2. Right click on the **ForeignSecurityPrincipals** container and select **Delegate Control** to open **Delegation of Control Wizard** and click **Next**.
3. Add Service Account user using **Add** button and click on **Next**.
4. Select **Create a custom task to delegate** and click **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 click on **Next** and then **Finish**.

Exchange Server

For managing Exchange Server, the Service Account must be a member of Recipient Management group.

> Application user for provisioning of Exchange Server must be Remote shell enabled. To enable remote Shell for a user, set the **RemotePowerShellEnabled** parameter to **$True** on the `Set-User cmdlet`.
> For example, `Set-User UserName -RemotePowerShellEnabled $True`

Contacts

For managing contacts, the contacts must be delegated to **Account Operators** group using Delegation Control Wizard as follows:

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

Microsoft Skype for Business Server

For Microsoft Skype for Business Server user management, service account must be a member of one of the following and **CSUserAdministrator** domain groups:

- RTCUniversalServerAdmins
  Or
- custom group with SQL permission

The account must also be a member of local Administrator group on the system running IQService and ensure that:

- IQService can access the Lync/Skype for Business Server through port 1433 of the SQL Server.
- IQService Accounts have direct access to the database to successfully provision the account.

Permissions required for **Custom group** and **CSUserAdministrator** domain group in SQL:

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

Managed Service Account and Group Managed Service Account

For managing Managed Service Account and Group Managed Service Account following permissions are required:

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

Configuration Parameters

This section contains the information that the connector uses to connect and interact with the application. Each application type requires different information to create and maintain a connection.

> Attributes with * are mandatory attributes.

The Active Directory connector uses the following connection parameters:

IQService Configuration

IQService Host

FQDN/IP of the system where IQService is installed.

IQService Port

The TCP/IP port on which IQService is listening for requests.

If 'Use TLS' is enabled, then ensure to configure corresponding IQService TLS port.

IQService User

User registered with IQService for Client Authentication.

IQService Password

Password of registered user for Client Authentication.

Use TLS

Indicates whether this is a TLS communication between IdentityIQ and IQService.

If 'Use TLS' is enabled, 'IQService User' and 'IQService Password' attributes are mandatory.

> For more information on enabling the Client Authentication and TLS communication, see IQServices.

Forest Configuration

Forest Configuration consists of the details of all the forests that must be managed.

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, else uses Server configured for respective domains to authenticate the users.

Global Catalog configuration also facilitates domain discovery within that forest.

Following are the list of attributes that must be configured for each domain that the application is managing:

Forest Name

Logical name of the forest used in the organization.

GC Server

Global Catalog Server information in following format:

```
GC IP/FQDN:Port
```

User

Service account to manage forests with appropriate permissions mentioned in the Administrator Permissions.

For Strong Authentication (SASL) to work, username must be in userPrincipalName format that is, **User-Name@DNSDomainName.com**. For more information, see Using Strong Authentication (SASL).

Password

Password of service account.

Authentication and Security

Authentication Type to be used when binding to Active Directory.

- **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. Recommend to **Use TLS** with Simple Authentication
- **Strong**: Strong Authentication bind is performed which uses Kerberos or NTLM depending upon whether the IdentityIQ is in a network (of service account domain) or outside network. SASL has implicit security layer for Data Encryption.

Use TLS

(*Applicable only when **Authentication and Security** is **Simple***) Indicates whether this is a TLS communication.

For more information on enabling the TLS communication, see Securing Active Directory Application.

Resource Forest

Indicates whether this Forest is Resource Forest from Resource Forest deployment model.

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

Manage All Domains

Manages all domains under that forest using the forest credential. If selected, domain configuration section is not required to be configured. In this case, domains that the application will manage can be previewed with **Preview** button.

If not selected, domains in this forest can be enumerated in the Domain Configuration by clicking **Discover** button.

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

Domain Configuration

Domain Configuration consist of details to connect domain/s such as distinguished name of the domain, Username and Password. Domain settings must be configured for all domains that must be managed by this application.

The following list of attributes must be configured for each domain that the application is managing.

*Forest Name**

Name of the forest of the domain. This forest must also be configured in Forest Configuration.

*Domain**

Distinguished name of the domain.

*User**

User of the domain in **Domain\User** format with appropriate rights required to read and provision.

For Strong Authentication (SASL) to work, username must be in userPrincipalName format that is, **User-Name@DNSDomainName.com**. For more information, see Using Strong Authentication (SASL).

*Password**

Password of the user mentioned for **User** field.

Servers

List of IPs or FQDNs of Domain Controller (DC) Servers belonging to the domain. Connector will bind to domain controllers in the defined ordered in case of any failures to bind to particular DC.

Unless there is a requirement to connect to particular DC/s, It is recommended to leave this field blank (server-less bind) to let the connector decide appropriate Domain Controller and have better resiliency and fault tolerance.

Authentication and Security

Authentication Type to be used when binding to Active Directory.

- **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. Recommend to **Use TLS** with Simple Authentication.
- **Strong**: Strong Authentication bind is performed which uses Kerberos or NTLM depending upon whether the IdentityIQ is in a network (of service account domain) or outside network. SASL has implicit security layer for Data Encryption.

Use TLS

(*Applicable only when **Authentication and Security** is **Simple***) Select this option to secure the channel between IdentityIQ and Domain Controller using TLS protocol.

For more information on enabling the TLS communication, see Securing Active Directory Application.

Exchange Configuration

Exchange Configuration consist of details to connect Exchange Server such as Exchange Forest, Exchange Hosts, UserName, Password, Account Forests and whether the communication should be over TLS. If this application needs to manage Exchange mailboxes, mail users or distribution lists, following Exchange Configurations must be defined.

*Exchange Forest**

Name of the forest where exchange is installed. This forest should also be configured in Forest Configuration.

*Exchange Hosts**

FQDN or IP of Exchange Server Host/s.

*User**

User in **Domain\User** format with appropriate rights.

*Password**

Password of the user mentioned for **User** field.

*Account Forest**

Name of accounts (user) forests served by this exchange.

Use TLS

Indicates whether this is a TLS communication between Active Directory and Exchange.

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 post script 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 there in the IdentityIQ database not then no further action would be 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 user has entitlement changes along with attribute change (s) or if users have 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 ==> SystemConfiguration**
Or
IdentityIQ ==> debug page ==> Configuration Object (dropdown) ==> 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.

> 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"/>
```

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

Following are examples with sample values:

```
<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"/>
```

Schema Attributes

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 which are used to hold entitlements shared across identities.

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 Entitlement, Multi-Valued and Indexed. For more information on the schema tab, see *SailPoint IdentityIQ Administration Guide*.

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

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

Attributes with asterisk mark (*) 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. Examples: “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.

Examples: “Martin K Smith”, “Marty Smith” and “printer12”.

distinguishedName

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

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.

Examples: “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.

Examples: “AASD” as a destination indicator for Sydney, Australia. “GBLD” as a destination indicator for London, United Kingdom.

The directory will 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.

employeeNumber

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

Examples: “John”, “Sue”, and “David”.

homePhone

This attribute contains the employees home phone number.

homePostalAddress

This attribute contains the employees mailing address.

homeMDB

Exchange mailbox store DN. Required for mailbox creation.

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.

Examples: “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.

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.

Examples: “Austin”, “Chicago”, and “Brisbane”.

mail

This attribute type contains the RFC822 mailbox for the user.

manager

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

mailNickname

Exchange Alias.

mobile

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

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.

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.

msRADIUSFramedIPAddress

Use this property to assign a specific IP address to a user 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.

ou

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

Examples: “Sales”, “Human Resources”, and “Information Technologies”.

objectguid

Globally unique identifier of the object.

pager

This attribute type contains the telephone number of this persons pager.

objectType

Indicates type of the Active Directory objects. For example, User, Contact

physicalDeliveryOfficeName

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

Examples: “Austin, Downtown Austin” and “Chicago, Finance Station E”.

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.

Example: “Box 27".

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.

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.

Example: “78664”, to identify Pflugerville, TX, in the USA.

preferredDeliveryMethod

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

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.

Example: “Receptionist$XYZ Technologies$6034 Courtyard Dr. $Austin, TX$USA”.

roomNumber

This attribute type contains the room or office number or this persons normal work location.

secretary

This attribute type contains the distinguished name of this persons 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.

Example: The person object “cn=Elvis Presley,ou=employee,o=XYZ\, Inc.” is related to the role objects “cn=Bowwling 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 will contain the distinguished name of each role object as separate values.

sn

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

Example: “Smith”.

st

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

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.

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.

Examples: “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.

Examples: “s9709015”, “admin”, and “Administrator”.

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”.

memberOf

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

objectSid

Windows Security Identifier

sAMAccountName

This attribute type contains the sAMAccoutName for this user.

msDS-PrincipalName

Name of the entity in the following format:

```
NetBIOS domain name\sAMAccountName
```

sIDHistory

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

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.

Managed Service Account Attributes

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

msDS-ManagedPasswordInterval

Interval in days after which Active Directory will change 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**

msDS-GroupMSAMembership

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

IQService is required to read and provision this property.

msDS-AllowedToActOnBehalfOfOtherIdentity

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

IQService is required to read and provision this property.

servicePrincipalName

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

For example, **MyService/Host1.example.com**

Provisioning Policy Attributes

The following table lists the provisioning policy attributes:

Provisioning Policy Attributes for User Creation

ObjectType

Type of the account to be created. Default value: User.

- For creating Contact, object type must be contact.
- For Group Managed Service Account, object type must be **msDS-GroupManagedServiceAccount**.
- For Managed Service Account, object type must be **msDS-ManagedServiceAccount**.

distinguishedName

Distinguished name of the user to be created.

sAMAccountName

sAMAccountName of the user to be created.

password

Password of the user to be created.

IIQ Disabled

A boolean attribute, set to true to create a disabled user.

primaryGroupDN

Default group of the user to be created.

description

Description of the user to be created.

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.

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.

msRADIUSFramedIPAddress

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

preferredServer

The preferred server (Domain Controller) on which this request must be executed.

For example,

```
<ProvisioningPlan targetIntegration="AD-Direct">
 <AccountRequest op="Modify" nativeIdentity="CN=Adam,CN=Users,DC=SPDomain,DC=local">
  <Attributes>
   <Map>
    <entry key="preferredServer" value="DC2.SPDOMAIN.LOCAL"/>
   </Map>
  </Attributes>
  <AttributeRequest op="Add" name="displayName" value="Adam Gilchrist"/>
 </AccountRequest>
</ProvisioningPlan>
```

accountExpires

The attribute **accountExpires** should be defined as String.

The value of **accountExpires** could be set in Microsoft defined timestamp which represents the number of 100-nanosecond intervals since January 1, 1601 (UTC). The value can also be entered in human readable format which is `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.

Provisioning Policy 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.[a]

Delete Provisioning Policy Attribute for Non-Leaf User Object

deleteSubTree

To delete the non-leaf user objects set the value of the attribute to true (boolean).

For example:

```
<ProvisioningPlan nativeIdentity="Adam" targetIntegration="AD-Direct">
  <AccountRequest application="AD-Direct" nat-
iveIdentity="CN=Adam,CN=Users,DC=SPDomain,DC=local" op="Delete">
     <Attributes>
        <Map>
          <entry key="deletesubtree">
            <value>
              <Boolean>true</Boolean>
            </value>
          </entry>
        </Map>
     </Attributes>
  </AccountRequest>
</ProvisioningPlan>
```

[a]* - Attributes with asterisk mark (*) 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.

Special Provisioning Attributes for Move/Rename Request

AC_NewName

A string attribute to rename the user. For example, CN=abc

AC_NewParent

A string attribute to move the user to new OU. For example, OU=xyz,DC=pqr,DC=com

The **AC_NewName** and **AC_NewParent** are special attributes to handle the move and rename operations and can be sent in Attributes Map and AccountRequest instead of AttributeRequest.

For example:

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE ProvisioningPlan PUBLIC "sailpoint.dtd" "sailpoint.dtd">
<ProvisioningPlan>
   <AccountRequest application="AD App" nat-
iveIdentity="CN=SampleUser,CN=Users,DC=Example,DC=Com" op="Disable">
    <Attributes>
     <Map>
         <entry key="AC_NewParent" value="OU=DsiabledUsers,DC=Example,DC=Com"/>
     </Map>
    </Attributes>
   </AccountRequest>
</ProvisioningPlan>
```

Provisioning Policy Attributes for Microsoft Exchange

homeMDB

(*Optional*) Exchange mailbox store DN. Required for mailbox creation. Send this attribute with new mailbox store DN to move the mailbox to another mailbox store.

mailNickname

Exchange alias for mailbox, mailuser, or mailcontact. Required for mailbox creation and to update or disable the mailbox. Send this attribute with no value to disable the mailbox.

exch_externalEmailAddress

External email address. Required for mailuser and mailcontact creation and to update or disable the mailuser and mailcontact, Send this attribute with no value to disable the mailuser and mailcontact.

msExchHideFromAddressLists

(*Optional*) Hide from Exchange address lists.

DomainController

(*Optional*) Fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.

> Active Directory Connector also supports provisioning other Microsoft Exchange attributes other than mentioned above.

Provisioning Policy Attributes for Microsoft Lync\Skype for Business User

DomainController

(*Optional*) Fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.

For example,

```
<ProvisioningPlan>
  <AccountRequest op="Modify" nativeIdentity="CN=Adam,CN=Users,DC=SPDomain,DC=local">
   <Attributes>
    <Map>
     <entry key="DomainController" value="DC1.SPDOMAIN.LOCAL"/>
    </Map>
   </Attributes>
     <AttributeRequest op="Add" name="msRTCSIP-UserEnabled">
        <Value>
          <Boolean>true</Boolean>
        </Value>
     </AttributeRequest>
  </AccountRequest>
</ProvisioningPlan>
```

SipAddress

To assign the user a specific SIP address.

RegistrarPool

Registrar pool where the user's Microsoft Lync\Skype for Business Server account will be homed. Mandatory attribute. Send this attribute with no value to remove the user from Microsoft Lync\Skype for Business Server.

SipAddressType

Select one of the SipAddressType from supported values: SamAccountName, FirstLastName, EmailAddress. Microsoft Lync\Skype for Business Server generates a SIP address for the new user when SipAddressType is provided in combination with SipDomain.

SipDomain

The SIP domain for the user account being enabled. Microsoft Lync\Skype for Business Server generates a SIP address for the new user when SipAddressType is provided in combination with SipDomain.

msRTCSIP-UserEnabled

Send this attribute with true/false to enable/disable Microsoft Lync\Skype for Business.

Pin

Dial-in Conferencing PIN number to be set for the Microsoft Lync\Skype for Business.

LyncPinLockedOut

Send this attribute with true/false to lock/unlock Microsoft Lync\Skype for Business user’s Dial-in Conferencing PIN.

DialPlan

Name to set dial plan for Microsoft Lync\Skype for Business user.

Active Directory Connector supports only User DialPlan.

Active Directory Connector also supports provisioning other Microsoft Lync\Skype for Business attributes other than mentioned above which can be provisioned using `Set-CsUser` command.
To set any other Microsoft Lync\Skype for Business attributes, edit application xml file to add **lyncAttributes** application attribute of string type with comma separated name of Microsoft Lync\Skype for Business attributes added in provisioning policy.

Provisioning Policy Attributes for Managed Service Account (MSA) Attributes

dNSHostName

(*Required andApplicable only for Group Managed Service Account only*) DNS host name of Service Account.

msDS-SupportedEncryptionTypes

Supported Encryption Types for the Service Account.

msDS-ManagedPasswordInterval

(*Applicable only for Group Managed Service Account only*) Number of days for the password change interval.

msDS-GroupMSAMembership

(*Applicable only for Group Managed Service Account only*) Principals allowed to retrieve Managed Password of this Group-Managed Service Account.

msDS-AllowedToActOnBehalfOfOtherIdentity

Accounts that can act on the behalf of this Group Managed Service Account.

servicePrincipalName

Service principal names for this Service Account.

Provisioning Policy Attributes for CreateGroup

distinguishedName

Group in the distinguished name format.

sAMAccountName

sAMAccountName

Provisioning Policy Attributes for UpdateGroup

description

A description of the group.

groupType

Group Type. Allowed values are:

1. Security
2. Distribution

groupScope

Group Scope. Allowed values are:

1. Domain local
2. Global
3. Universal

mailNickname

Alias and is required if want to create Distribution Group on exchange. Only Universal type of group can be created on exchange.

* - Attributes with asterisk mark (*) 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.

To pass additional information (metadata) in 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.

Active Directory Connector supports updating any other exchange mailbox attributes supported by **set-mailbox** cmdlet. To set any such parameter, prefix the parameter name of the **set-mailbox** cmdlet with **Exch_** while adding the attribute to the provisioning policy.

For example:

For the **HiddenFromAddressListsEnabled** attribute, the attribute name would be added as **Exch_HiddenFromAddressListsEnabled** in provisioning policy.

Alternatively, this can be done by editing the application xml file by adding an application attribute named **exchangeAttributes** of string type with comma separated name of the Exchange attributes added in provisioning policy.

For example:

Provisioning policy attribute name: **HiddenFromAddressListsEnabled**

Add the attribute in application debug page as follows:

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

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 it's 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, 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 , 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, see Managing Shared Mailbox Permissions

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, Service Account must be a member of **Account Operator Group** and **Recipient Management Group**.
- For Create, Update, Delete operation on Shared Mailbox and assigning Shared Mailbox to 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, member of Organization Management group has Exchange Role with higher capabilities which is not required for this operation. Hence it is recommend to create custom **Exchange Admin Role Group**.

Perform the following steps to create custom Exchange Admin Role Group:

a. On the **Exchange admin center** page, select **Permissions** in the left pane.

b. Under the **admin roles** tab click **+** icon to create new Role Group.

c. On the **Role Group** window that appears enter the **Name** and **Description**.

d. From the list of Roles that are displayed, search and select **Active Directory Permissions Role** and click on **Save.**

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

Additional 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.

> Setting this flag to true will have performance impact on get account operation

Enabling Shared Mailbox Management

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

Perform the following steps to enable Shared Mailbox management:

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

```
<Schema aggregationType="group" descriptionAttribute="description" dis-
playAttribute="msDS-PrincipalName" featuresString="PROVISIONING" hier-
archyAttribute="" 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</De-
scription>
      </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\sAMAc-
countName"</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" schemaObjectType="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 User Account schema to represent assigned Shared Mailbox.
   Copy the following 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 Shared Mailbox object schema is not added in the application.

3. Add Create and Update Provisioning policies. Connector supports updating attributes which are present in the Shared Mailbox schema.
   Copy the following respective policies under **<ProvisioningForms>** tag

- **Create Policy**

```
<Form name="Create Shared Mailbox" objectType="sharedMailbox" type="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="help_
con_prov_policy_ad_mailNickname" name="mailNickname" required="true" reviewRe-
quired="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="Full Access" multi="true" name="fullAccess" type-
e="string"/>
        <Field displayName="Send As" multi="true" name="sendAs" reviewRe-
quired="true" type="string"/>
        <Field displayName="Send On Behalf" multi="true" name="sendOnBehalf"
type="string"/>
      </Section>
</Form>
```

- **Update Policy**

```
    <Form name="Update Shared Mailbox" objectType="sharedMailbox" type-
e="Update">
      <Attributes>
        <Map>
          <entry key="pageTitle" value="Update 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">
          <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" reviewRe-
quired="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-
e="sAMAccountName" reviewRequired="true" type="string">
        </Field>
        <Field displayName="objectSid" helpKey="cn" name="objectSid" reviewRe-
quired="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="sendOnBehalf"
reviewRequired="true" type="string"/>
      </Section>
    </Form>
```

Supported Schema Attributes

Following are the only supported schema attributes for 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) 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 name to which user has **Full Access**, **Send As** or **Send On Behalf** or all three permission.

Provisioning Policy Attributes

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

*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.

Managing Shared Mailbox Permissions

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 'memberOfSharedMailbox' attribute in provisioning plan for the user object

1. Assign **Full Access** and **Send As** permission 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>
```

   Above request will assign **Full Access** and **Send As** permission to the user. These are the default permissions which would be assigned if permission names are not provided explicitly in the request. 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.
   User can request for specific permission (other than the default) by passing additional information in the AttributeRequest by using **sharedMailboxPermission** attribute in the request.
   For example, The following example assigns only 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>
```

Connector supports only assigning of specific permissions. To remove specific permission on Shared Mailbox, update those permissions properties on Shared Mailbox entitlement object.

Assign permission by updating Shared Mailbox permission attributes

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

For example, following plan would execute modify operation on the shared mailbox and assign 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>
```

Active Directory Resource Forest Topology Exchange Management

In Active Directory **Account Forest - Resource Forest** topology, all user accounts exist in one or more Forest(s) called as **Account Forest(s)** while resources have a dedicated Active Directory Forest called as a **Resource Forest**. The **Resource Forest** may have deployments like Microsoft Exchange, Skype Server and so on.

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 for a mailbox, a Linked Mailbox is created on the Resource Forest Exchange server with 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 corresponding **Master Account**.

The connector relies on the connection details provided under the Configuration Parameters, Configuration Parameters and Configuration Parameters 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.

Additional Configuration Parameters

This section lists the **Resource Forest** specific additional configuration parameters:

Resource Forest specific domain configuration attributes

disableShadowAccountMembership

By default, the connector considers the memberships of shadow account as master accounts memberships.

To discard membership of shadow account, set this attribute to true under domainSettings of respective domain in Boolean (or as a String) 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, 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 type 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

supportFSPsFromResourceForest

To enable aggregating and provisioning FSPs from the Resource Forest Domains for the Master Account set this Boolean attribute to true. Default: False

retainShadowAccountOnDelete

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

Configuring searchDNs

- **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.

Schema Attributes

In addition to the existing Microsoft Exchange account schema attributes, following new attributes are added in the schema for the new application. For existing Active Directory application 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 **Linked Mailbox**.

shadowAccountDN

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

shadowAccountGuid

The objectGuid of the **Linked Mailbox** shadow account.

Provisioning Policy Attribute

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

shadowAccountDN

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

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

> Recycle Bin feature must be enabled on Active Directory.

1. 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, see IQService.
2. Install **Active Directory module for Windows PowerShell** on the computer where IQService is installed.

   > 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`
3. Run the following PowerShell command on all domain controllers (DCs) in the forest which must be managed:

   ```
   Enable-PSRemoting
   ```

   > 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.

Additional Information

This section describes the additional information related to the Active Directory Connector.

Delta Aggregation

> This includes changes such as user/group has been added/updated/deleted on the managed system. This version now supports aggregation of delta changes for Move and Rename operations.

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

Prerequisites

To provide Replicating directory changes permissions to the user, perform the following actions:

1. In the Active Directory Users and Computers browser menu, select the **View** option, right-click and ensure that **Advanced features** check box is enabled.
2. Right-click the domain node and select **property** option and open the Security tab.
3. Add user to the list of Security Principals.
4. Select the user and select **Allow** checkbox for Replicating Directory Changes permission.

To provide Read permissions on Deleted Objects Container to user, perform the following actions:

1. Log on to any domain controller in the target domain with a user account that is a member of the Domain Administrators group.
2. Open a command prompt: navigate to Start, enter `cmd` and click **Enter**.
   Enter the following command and press **Enter**:

   ```
   dsacls <Deleted objects container DN> /<takeownership>
   ```

   In the above command line, `Deleted objects container DN` is the distinguished name of the deleted objects container.

   For example, `dsacls "CN=Deleted Objects,DC=SailPoint,DC=Com" /takeownership`
3. To grant **Read** permission to the objects in the **Deleted Objects container** to a user type, enter the following command and press **Enter**:

   ```
   dsacls < Deleted objects container DN > /G <domainName\userName >: LCRP
   ```

   In the above command line, `LCRP` stands for the list object and read properties permission.

   For example, `dsacls "CN=Deleted Objects, DC=SailPoint,DC=Com" /G Sailpoint\John:LCRP`

To execute delta partitioned aggregation, set the value of the enableCache attribute to true as follows:

Delta partitioning aggregation requires caching to be enabled.

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

Testing Delta Aggregation

For delta aggregation to work properly, a start point would be required from where it would detect 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.

Partitioning Aggregation

With IdentityIQ version 8.2, auto partitioning can be performed using the **Allow Auto Partitioning** checkbox (See Configuration Parameters).

The following section describes the manual configuration for partitions.

Configuring partitions manually

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

In Active Directory Connector, data can be partitioned by specifying a **searchDN** and/or a **searchFilter** as a partition entry. 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, we 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.

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

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.

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, 'IQService User' and 'IQService Password' attributes are mandatory.
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.

Field	Description
	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. Click 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 Provisioning	Select it to override the default provisioning action for the collector.
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.