Omada Identity

CyberArk Connectivity Guide

Version 14.0.6

Publish date: 30-09-2020

Omada A/S
Oesterbrogade 135
DK-2100 Copenhagen
Denmark
Tel: +45 7025 0069
Fax: +45 7025 0089
www.omada.net

Table of Contents

Introduction

This document specifies the details of connectivity between CyberArk and Omada Identity or Omada Identity Cloud. Within this document Omada Identity name covers both Omada Identity and Omada Identity Cloud solutions, unless stated otherwise. The details include such information as basic prerequisites of the target CyberArk systems, possible operations, as well as objects and types that these operations can be performed at, and configuration of the connectivity on the Omada Identity side, including its settings and customizations.

1.1 Version history

Version	Release Date	Description
Revision 3	12-02-2020	Improvements to the **Data model**, **Relevant objects and types** and additional explanations in section **Setting up CyberArk Rights filter expression (optional)**.
Revision 2	28-01-2020	Test connection functionality updated.
Revision 1	02-01-2020	**Network requirements** section updated.
V14	23-09-2019	Initial release

1.2 Related documentation

- Omada Identity – Import and Onboarding Guide
- Omada Identity – Configuration Guide
- Omada Identity – SCIM 2.0 Connectivity Guide

Connectivity Capabilities

2.1 Connector/collector capabilities

CyberArk connectivity is based on the System for Cross-domain Identity Management (SCIM) connector 2.0, which is built on top of the generic REST connector.

This means that the CyberArk connector has the same functionalities as the generic REST connector, that is: Create, Read, Update, Delete (CRUD) on Users and Groups using **the core schema**.

Note
See section **3.2.1 Additional information** for a link to core schema documentation.

2.2 Relevant objects and types

In a provisioning context, the CyberArk connector works the same way as the generic REST connector. By default, the CyberArk connectivity can perform the following operations:

Resource	Possible operations
Users	Create, Read, Update, Delete
Groups	Read, Update Add or remove users' assignments

2.2.1 Supported data types

The CyberArk collector enables the import of Identity data, which entails:

- **Accounts**: CyberArk users
- **Resources**: CyberArk Groups, CyberArk Safes (Containers in target system API), CyberArk Privileged Accounts (PrivilegedData in target system API), relationships between PrivilegedData and Containers
- **Resource Assignments**: CyberArk Groups assigned to Safes, CyberArk Users assigned to Safes, CyberArk Users assigned to Groups
- **Resource Parent/Child**: Relationships between Groups

2.2.2 Supported import modes

Only **Full import** is supported.

2.3 Supported versions of CyberArk

CyberArk connectivity is compatible with the currently available versions of CyberArk.

Target System

3.1 Target system prerequisites

Note
Before onboarding a CyberArk system, it is essential that you have general knowledge of SCIM 2.0, as well as the SCIM core schema and the protocol specification. See section **3.2.1 Additional information** for links to the official SCIM 2.0 documentation on these topics.

3.1.1 Minimum required permissions

You should have access to the SCIM 2.0 API, including relevant permissions and network firewall/antivirus settings.

3.1.2 Network requirements

The following ports need to be open in firewalls:

Port number	Protocol
443	HTTPS

In addition, the SCIM Server for CyberArk has the following requirements:

- Vault v9.x
- AIM Credential Provider v9.x
- PACLI v7.2

3.2 Target system configuration

To collect data from SCIM 2.0, you must construct a URL that contains the address of the SCIM server and the entity name that you want to collect, for example, *https://mywebsite.com/scim/users*.

3.2.1 Additional information

For official SCIM 2.0 documentation, see the links below:

- General information on SCIM 2.0, see this link.
- Overview, definitions, concepts and requirements, see this link.
- Information on the SCIM core schema, see this link.

- Information on the SCIM protocol specification, see this link.

Omada Identity Configuration

4.1 Collector configuration

This subchapter describes the prerequisites and proper configuration of the CyberArk collector in Omada Identity.

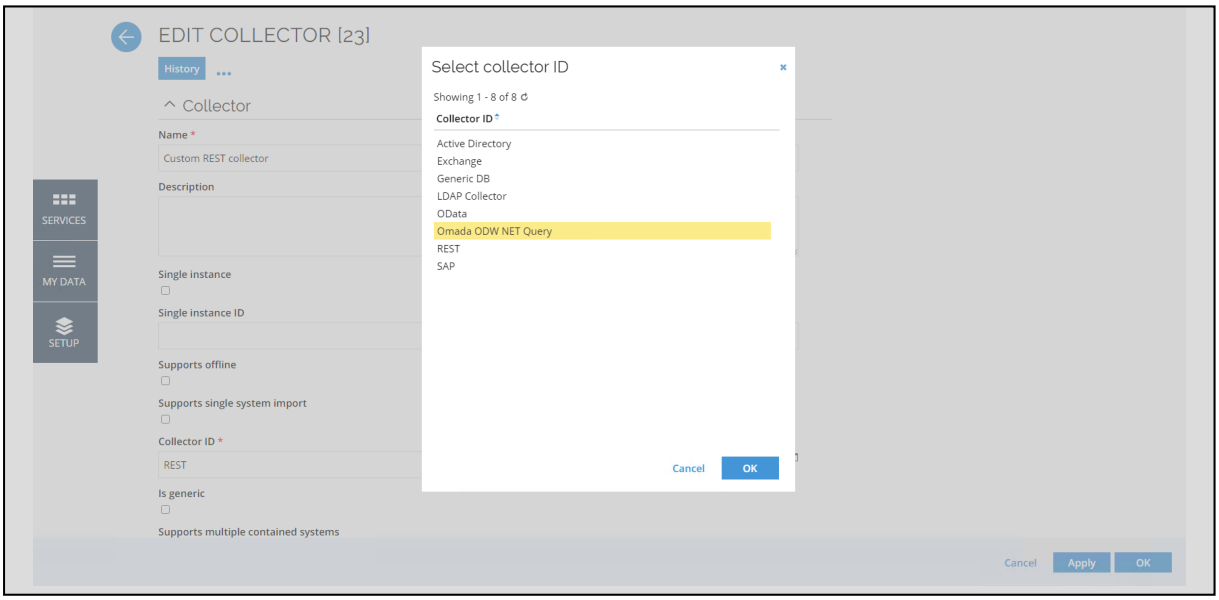
Since Omada Identity 14.0.6 (Update 6) all collectors based on the **generic REST** collector have their **Collector ID** migrated from *REST* to *Omada ODW NET Query*.

For the existing REST-based template collectors the migration process is performed during the update process, and it is transparent to the users.

In case of the existing custom REST-based collectors, i.e., the customer created collectors with REST as a Collector ID, some additional actions need to be taken by the customer to migrate the REST-based collector to a .NET Query collector.

In order to switch your custom REST-based collector to .NET:

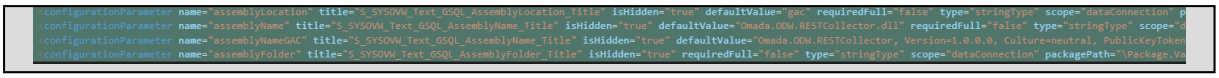
1. Go to **Setup** -> **Administration** -> **Connectivity** -> **Collectors**
2. Select your custom REST-based collector and click **Edit**.
3. Change the **Collector ID** to *Omada ODW NET Query* and click **Apply**.



In addition to the Collector ID, configuration changes must be added to the collector configuration file for the following configuration parameters:

- assemblyLocation
- assemblyName
- assemblyNameGAC

- assemblyFolder



These configuration details can be found in the **generic REST** collector configuration details.

Note
Please bear in mind the following relations of the configuration parameters:

If **assemblyLocation==FILE** then **assemblyFolder + assemblyName** needs to be filled; else, it can be left unset.

If **assemblyLocation==GAC** then **assemblyNameGAC** needs to be filled; else, it can be left unset.

4.1.1 Register new system

1. In **Systems** view, click **New** to register a new system.
2. Enter the name of the new system in the **Name** field of the pop-up screen.
3. Enter the unique system ID in the **System ID** field.
4. From the **Category** dropdown, select **New**.
5. Pick the **CyberArk Collector technology** radio button.
6. In the **Content** field, select the type of data that you want to import from and provision to the system.

4.1.2 Connection details

Important
If the addresses for query and for entities are the same, you must leave the **Entity root** and **Query address** fields blank or empty. Instead, enter the full address in the **Base URL** field.

Parameter	Description
Base URL	This field is optional. You can specify the Base URL of the service. When you specify a Base URL, this URL will be used for all defined queries if these do not specify a full URL of their own. The Base URL is part of the data connection data, and the Base URL should not be transported between environments.

Parameter	Description
	You must enter the URL that ends before the SCIM entity's name. For example, the URL for users is http://10.16.2.1/scim/users and the base URI is http://10.16.2.1/scim.
Authentication type	Choose the type of authentication to use for the REST system. The available options are: • **AWS Signature** – adds authentication information to the HTTP header of Amazon Web Services requests. • **Basic** – uses Base64-encoded string that contains a user name and password. • **Negotiate** – automatically selects between NTLM and Kerberos, depending on availability. • **NTLM** – NT LAN Manager authentication uses Windows credentials to transform the challenge data. • **Digest** – a challenge-response authentication that uses a nonce which is a string of random data. • **Kerberos** – uses ticket granting system for authenticating users. • **OAuth2 Client Credentials** – uses access token. • **OAuth2 Password** – uses user's credentials to acquire access token. • **OAuth2 JWT** – uses JSON web token. • **OAuth2 Static Token** - uses statically generated bearer token. **Note** Depending on the type of authentication that you choose to use with the system, you may see more or fewer settings appear in the dialog box.
User	This is the user name.
Password	This is the password. Each time you make a change to any of the settings in the Connection details dialog box, you must provide your password again.

Parameter	Description
Domain	This field is **optional**. Here, you can specify the domain name for the user.
Token end-point	This field applies only to **OAuth2** authentication options. Enter the URL used to exchange an authorization grant for an access token.
Client ID	This field applies only to **OAuth2** authentication options. Enter the client registered with the service.
Client secret	This field applies only to **OAuth2** authentication options. Enter the generated secret for the Client ID.
OAuth static token	This field applies only to **OAuth2 Static Token** authentication option. Enter a statically generated bearer token. The value will be encrypted upon storage.
Resource	This field is **optional**. Here, you can specify the Resource for which authorization will be granted.
JWT Encryption algorithm	This field applies only to **OAuth2 JWT** authentication options. Select the encryption algorithm used when signing the token. For example, RSASSA-PKCS1-v1_5 using SHA-256 (RS256)
JWT Type	This field applies only to **OAuth2 JWT** authentication options. Enter a JWT Type (Header parameter)
JWT Public key ID	This field applies only to **OAuth2 JWT** authentication options. Enter a public key ID for signing the JWT (Header parameter)
JWT Issuer	This field applies only to **OAuth2 JWT** authentication options. Enter a JWT Issuer claim.
JWT Subject	This field applies only to **OAuth2 JWT** authentication options. Enter a JWT Subject claim.
JWT Audience	This field applies only to **OAuth2 JWT** authentication options.

Parameter	Description
	Enter a JWT Audience claim.
JWT Expiration time	This field applies only to **OAuth2 JWT** authentication options. Set a JWT Expiration time.
JWT Additional claims	This field applies only to **OAuth2 JWT** authentication options. Enter a JWT Additional claims in the format: `key1,value1;key2,value2;...`
JWT Private key (PEM format)	This field applies only to **OAuth2 JWT** authentication options. Enter a JWT Private key is in the PEM format: -----BEGIN PRIVATE KEY----- .... -----END PRIVATE KEY----- or -----BEGIN ENCRYPTED PRIVATE KEY----- .... -----END ENCRYPTED PRIVATE KEY----- and then Passphrase for Private Key needs to be provided.
JWT Passphrase for private key	This field applies only to **OAuth2 JWT** authentication options. Enter a Passphrase for the provided Private key.
Test connection	Enable this setting to allow the system to check if the connection details are correct. **Important** If you want to use this functionality must install Omada Provisioning Service and make sure it has the necessary permissions to communicate with the target system.

4.1.3 Queries and mappings

This collector supports any number of queries and has the following query parameters which should be specified when creating of editing a query.

1. In the **URL is a DynamicExpresso expression** field, you can specify whether the URL is generated from a DynamicExpresso expression (if "Yes" is selected) or interpreted directly (if "No" is selected).

2. In the **URL** field, enter the URL for the resource. If "No" is selected in **URL is a DynamicExpresso expression**, you can specify a full URL or the part of the URL

which should be appended to the Base URL. If "Yes" is selected in **URL is a DynamicExpresso expression**, the URL enables the user to specify a DynamicExpresso expression, which is used to generate the URL dynamically.

3. Optionally, in the **Nested URL** field, you can provide any attribute returned from the **URL** and use it as a **nested query**. The attribute must be enclosed in {} brackets, for example, */groups/{PARENT_id}/members?roles=MEMBER*.

Important
If the **URL** returns a collection (multi-value), the **Nested URL** will only be called using the first element of each collection.

Moreover, as the **Nested URL** is called for the number of rows returned from the **URL**, employing this feature causes a **performance penalty**.

4. Optionally, in the **Append** field, enter some query parameters which should be appended.

5. Optionally, in the **Distinct** field, specify if the collector should remove possible duplicate rows.

6. Optionally, in the **Filter** field under the **Parameters** heading, you can provide a Dynamic Expresso expression that is used for filtering the data imported into OIS. It returns a TRUE/FALSE result for each imported data row. If the expression returns "FALSE" for the given row that row is skipped during import.

   The filter can be supplied with special functions **#MinRow()** or **#MaxRow()**. The #MinRow()/#MaxRow() are custom functions that can be combined with regular DynamicExpresso expressions thanks to the **#** prefix. For example, in

```
#MinRow(col1, col2)#col=="active"
```

   the custom function is encapsulated with **#** at the start and end.

   The MinRow()/MaxRow() functions take two parameters. The intention is similar to a "Group by function" in SQL server that allow you to eliminate duplicates and to take the **lowest** or **highest** *[order by column]* for each *[unique column] row*, i.e., *MinRow ([unique column], [order by column])*.

7. Optionally, in the **Description** field, enter a description for what this query is doing.

The collector performs a `GET` request on a specific URL. The final URL is created by adding the Query URL to the base URL. The Query URL must contain at least the name of the entity that you would like to collect data from.

The schema for an entity confirms that properties are present in a specific entity and describes how to access them.

Important
You must investigate the schema for the response as well as the schema for each of the entities that comes out of the CyberArk API.

You can do this by reading the documentation for the CyberArk API or by running a manual `GET` request.

When you onboard a system based on SCIM 2.0, a number of default queries and mappings will be set up when you register the first system. They are:

- **Accounts** – users are imported into Omada Identity as accounts.
- **Resource** – groups are imported into Omada Identity as resources with the resource type "CyberArk Group".
- **Resource Assignment** – user memberships in groups are imported into Omada Identity as resource assignments.
- **Resource Parent/Child** – group memberships in other groups are imported into Omada Identity as resource parent/child relationships.

4.1.3.1 Ad group filtering

A filtering functionality in the query mapping form is managed using the **Filter** field.

The **Filter** field is used for Dynamic Expresso expressions. See the procedure above for details.

By default, the **Filter** field in the "Group" mapping is filled in with the following expression:

```
urnietfparamsscimschemascyberark10Group_ldapFullDN == null
```

It filters out all AD-originating Groups from the data imported to Omada Identity.

4.1.3.2 Mapping of resource owners

If you create a query to import resource owners, it is possible to specify the resource's owner in two ways. You can do it either by directly importing the **UID** of the identity or by specifying the **account** from which the resolved owner is imported as a resource owner.

When mapping directly to the UID of identity, please ensure that identities are already **imported** to Omada Identity.

When mapping to an owned account, it is possible to either specify the **business key** of the account or the **composed businesskey**. The former should be used if the account is in the same system as the resource; the latter should be used if the account is imported in any of the trusted systems.

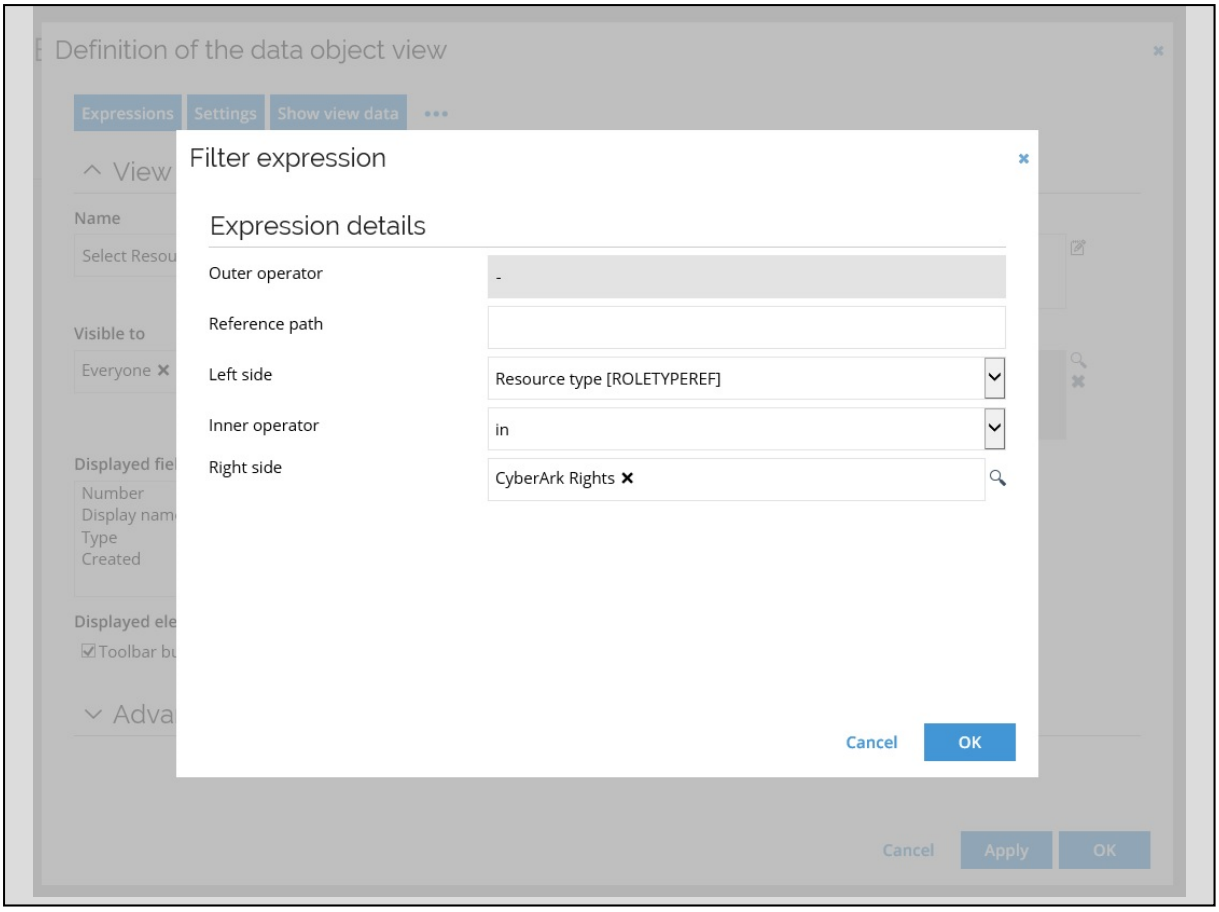
When the account stems from another system, you should use a **Lookup** mapping.

4.1.4 Advanced configuration

Parameter	Description
Security protocol	Choose a security protocol to use for an HTTPS connection: • TLS 1.1 • TLS 1.2
Paging mechanism	Select the type of paging the service uses. Services may result large datasets and will return data in chunks. The CyberArk collector is based on SCIM 2.0 / REST and offers the following options: • **None**: Select this option if the service offers no paging and all data is returned. • **Paging URL**: Use this option if the response contains a field with a URL for the next page. • **URL parameters**: Use this option if the paging must be specified as URL parameters.
Paging URL field	Use this option if the response contains a field with a URL for the next page. When **Paging mechanism** is selected as **Paging URL**, use the **Paging URL** field to specify the JSON field in the response that contains the URL for the next page of data. The collector will continue querying the service until this field is empty.
URL parameters	Use this option if the paging must be specified as URL parameters. When **Paging mechanism** is selected as URL parameters, use the **URL Parameters** field to enter the parameters that must be appended to the query to get the next page of data.
Total field	Enter here the JSON field in the response that indicates the total amount of records. This field is optional. If the field is left empty, the service will be called until an empty result set is returned.

4.1.5 Setting up CyberArk Rights filter expression (optional)

Optionally, after the initial import is completed, but before connector configuration, the user can add the "Resource type in CyberArk Rights" filter to the "CyberArk Rights" Property.



This is done to enable the filtering only for the relevant attributes when performing a Request Access for CyberArk Safes.

4.2 Connector configuration

4.2.1 Prerequisites

If you select the **CyberArk** as the provisioning connector when enabling provisioning, you must type a name for the connector.

4.2.2 Provisioning configuration

Note
If the addresses for query and for entities are the same, you must leave the **Entity root** and **Query address** fields blank or empty.

Instead, you must enter the full address in the **Base URI** field.

Parameter	Description
Base URI	Specify the server address. This is the URI used to access SCIM server. Example: *https://127.0.0.1/*.
Entity root	This is the URI used when entities are not in the root server directory. For example, when users are accessed using the following address: *http://127.0.0.1:8888/v2/users*, the Entity root is "*v2*".
Authentication mode	The type of authentication to use for the REST system. Depending on the type of authentication that you choose to use, you may see more or fewer settings. The available options are: • **None** – there are no authentication headers. • **Basic** – username and password are required. • **OAuth2** – all OAuth mandatory * fields must be completed with proper value. These values can be found in your external vendor's documentation of your REST system.
OAuth grant type	The way the authorization token is requested: • Client credentials • JWT bearer • Password credentials • Static token
URL for authorization token	Enter the URL used to exchange an authorization grant for an access token. You can get it from your SCIM 2.0-based application.
OAuth client ID	The public client ID that is used for exchanging the authorization code for an access token. You should get it from your SCIM 2.0-based application.
OAuth client secret	The private client secret that is used for exchanging the authorization code for an access token. You should get it from your SCIM 2.0-based application.

Parameter	Description
OAuth Resource	The name of the OAuth resource owner.
OAuth Token Revoke URL	The token revocation endpoint URL.
Token cache policy	There are three options: • Always cache • Expiration interval (if you select this option, you need to set the OAuth token expiration time in seconds) • Never cache
Username	The user name for the selected type of authentication.
Password	The password used for authentication.
Security protocol	The security protocol of your REST system. • SSL 3.0 • TLS 1.0 • TLS 1.1 • TLS 1.2
Content type header	Choose the format of the content. The default value is application/json.
Accept header	The default value is application/json.
Use numeric values	Setting this to true ensures that intType, longType, and referenceType values are sent as numeric values, and not string, in the JSON payload. For referenceType properties, the value is only set as a numeric value if the resolved value can be converted to a numeric value.
Query address	This is the base address for querying objects. It is appended to Base URI when specified. Example: *https://127.0.0.1/query*.
Root resource path	This field controls the root resource path used when references and object ids are resolved. The default is $.Resources[0] as defined in the SCIM standard.

Parameter	Description
Test connection	Select this checkbox to perform a test of the connection before using the connector. **Important** If you want to use this functionality must install Omada Provisioning Service and make sure it has the necessary permissions to communicate with the target system.
Test query	This field only apprears if the **Test connection** field is enabled. Here you can enter an optional test query used to verify the connection. The query must be relative to the base address, e.g., 'Users.' Entering a test query is important for a proper test for authentication methods which doesn't access the target system, e.g., basic authentication or static bearer token.

4.2.3 Data model

The following data model is available out of the box:

```
<?xml version="1.0"?>
<connectorDataModel xmlns="http://schemas.omada.net/ops/2015/ConnectorDataModelML"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" modelNamespace="CYBERARKSCIM">
  <properties>
    <property name="schemas"/>
    <property name="active" dataType="booleanType"/>
    <property name="id"/>
    <property name="urn:ietf:params:scim:schemas:cyberark:1.0:Safe.location"/>
    <property name="urn:ietf:params:scim:schemas:cyberark:1.0:User.profession"/>
    <property name="urn:ietf:params:scim:schemas:cyberark:1.0:Safe.supportOLAC"/>
    <property name="displayName"/>
    <property name="description"/>
    <property name="groupId" dataType="referenceType"/>
    <property name="containerId" dataType="referenceType"/>
    <property name="employeeNumber"/>
    <property name="externalId"/>
    <property name="userName"/>
    <property name="name.givenName"/>
    <property name="name.familyName"/>
    <property name="name"/>
    <property name="type"/>
    <property name="rights" multiValued="true"/>
    <property name="userId" dataType="referenceType"/>
    <property name="Operations[].op"/>
    <property name="Operations[].path"/>
```

```
    <property name="Operations[].value[].id" dataType="referenceType"/>
    <property name="members[].value" multiValued="true" dataType="referenceType"/>
    <property name="members[].$ref" multiValued="true" dataType="referenceType"/>
    <property name="privilegedData[].value" multiValued="true"/>
    <property name="privilegedData[].$ref" multiValued="true"
dataType="referenceType"/>
    <property name="user.value"/>
    <property name="user.name"/>
    <property name="user.$ref" multiValued="true" dataType="referenceType"/>
 <property name="group.value" dataType="referenceType"/>
    <property name="group.$ref" dataType="referenceType"/>
    <property name="group.name" dataType="referenceType"/>
    <property name="container.value"/>
       <property name="container.name" dataType="referenceType"/>
       <property name="container.$ref" dataType="referenceType"/>
  </properties>
  <objects>
    <object name="CyberArkUser">
      <objectDetails>
          <objectDetail name="VerbForUpdate" value="PUT"/>
        <objectDetail name="VerbForDelete" value="PUT"/>
        <objectDetail name="ReconcileOnUpdate" value="True"/>
        <objectDetail name="ReconcileOnDelete" value="True"/>
      </objectDetails>
      <objectProperties>
        <objectProperty>schemas</objectProperty>
        <objectProperty>active</objectProperty>
        <objectProperty>userName</objectProperty>
        <objectProperty>employeeNumber</objectProperty>
        <objectProperty>name.givenName</objectProperty>
        <objectProperty>name.familyName</objectProperty>


<objectProperty>urn:ietf:params:scim:schemas:cyberark:1.0:User.profession</objectProp
erty>
        <objectProperty isKey="true" referenceObject="Users" referenceKeyProperty="id"
referenceLookupProperty="userName">userId</objectProperty>
      </objectProperties>
    </object>
    <object name="CyberArkGroup">
      <objectDetails>
      </objectDetails>
      <objectProperties>
        <objectProperty>schemas</objectProperty>
        <objectProperty>displayName</objectProperty>
        <objectProperty>externalId</objectProperty>
        <objectProperty isKey="true" referenceObject="Groups"
referenceKeyProperty="id"
referenceLookupProperty="displayName">groupId</objectProperty>
      </objectProperties>
    </object>
    <object name="CyberArkContainer">
      <objectDetails>
      </objectDetails>
```

```
      <objectProperties>

<objectProperty>schemas</objectProperty>

<objectProperty>urn:ietf:params:scim:schemas:cyberark:1.0:Safe.location</objectProper
ty>

<objectProperty>urn:ietf:params:scim:schemas:cyberark:1.0:Safe.supportOLAC</objectPro
perty>
        <objectProperty>displayName</objectProperty>
        <objectProperty>description</objectProperty>
        <objectProperty>externalId</objectProperty>
        <objectProperty>name</objectProperty>
        <objectProperty>type</objectProperty>
        <objectProperty>privilegedData[].value</objectProperty>
        <objectProperty isKey="true" referenceObject="Containers"
referenceKeyProperty="id"
referenceLookupProperty="displayName">containerId</objectProperty>
      </objectProperties>
    </object>
     <object name="CyberArkGroupAssignment">
      <objectDetails>
        <objectDetail name="VerbForUpdate" value="PUT"/>
        <objectDetail name="VerbForDelete" value="PUT"/>
        <objectDetail name="ReconcileOnUpdate" value="True"/>
        <objectDetail name="ReconcileOnDelete" value="True"/>
      </objectDetails>
      <objectProperties>
      <objectProperty referenceObject="Users" referenceKeyProperty="id"
referenceLookupProperty="userName">members[].value</objectProperty>
      </objectProperties>
    </object>
    <object name="CyberArkContainerAssignment">
      <objectDetails>
        <objectDetail name="VerbForUpdate" value="PUT"/>
        <objectDetail name="VerbForDelete" value="DELETE"/>
        <objectDetail name="ReconcileOnUpdate" value="True"/>
        <objectDetail name="ReconcileOnDelete" value="False"/>
      </objectDetails>
      <objectProperties>
        <objectProperty>schemas</objectProperty>
        <objectProperty>rights</objectProperty>
               <objectProperty>container.value</objectProperty>
               <objectProperty>user.value</objectProperty>
      </objectProperties>
    </object>
  </objects>
</connectorDataModel>
```

The CyberArk connector is built on top of SCIM 2.0 and a generic REST connector. This connector overrides the default REST connector behavior of the *Resolve Reference Property* method and *Resolve Object ID*. The base address for both requests is taken from the *URI for accessing REST query* configuration parameter.

4.2.3.1 The Resolve Reference Property method

The *Resolve Reference Property* method only calculates those properties that do not have the *isKey* property set to *true*. The following is the request that is sent to the SCIM server:

```
GET https://<scim server address>/A?filter=B eq 'C'
```

These are the *DataModel* attributes:

- A – *referenceObject*
- B – *ReferenceLookupProperty*
- C – property value

4.2.3.2 The Resolve Object ID method

The *Resolve Object ID* method is similar to the *Resolve Reference Property* method. However, it only works on the properties where *isKey* is set to true.

You can also have multiple properties of this kind. If you have multiple properties, the *referenceKeyProperty* and the *referenceObject* must be the same in each of them. In that case, only the **AND** operator is used. For example:

```
GET https://<scim server address>/A?filter=B eq 'C'
```

or

```
GET https://<scim server address>/A?filter=B eq 'C' and D eq ‘E’
```

The following are the *DataModel* attributes:

- A – *referenceObject*
- B – *ReferenceLookupProperty*
- C – property value

The following are optional attributes:

- D – another *ReferenceLookupProperty*
- E – another property value

4.2.4 Schemas

Schemas describe which properties you can expect in a specific request. The difference between CyberArk and other REST systems are that schemas in CyberArk are an integral part of any request to the SCIM server.

For every operation, for example, **Create a User**, **Add/Delete assignments**, one of the JSON body properties must be **schemas**, and this defines how the request looks like.

4.2.5 Reconciliation

The CyberArk Connector has a feature called **Object Reconciliation** which is triggered in the data model by setting *objectDetails* with names *ReconcileOnUpdate* or *ReconcileOnDelete* to have the value *True*. For example:

```
<objectDetail name="ReconcileOnUpdate" value="True"/>

<objectDetail name="ReconcileOnDelete" value="True"/>
```

This setting works on a per object basis, so it is possible to have it enabled for certain objects and disabled for others.

If the operation is *Update* or *Delete*, and the relevant property is set to True, the connector merges between the current object in the external system and operations defined in **Task Mapping**.

Important
Please be aware that correct Update or Deletion of **container assignments** can be performed only if there was a successful import after such an assignment has been created, its ID has been retrieved, and provisioning has been finished.

When **Reconciliation** is turned off, a request is created based on task mapping alone. Each task mapping rule is converted to a JSON property (or array entry). The request is then sent out to a server.

When **Reconciliation** is turned on, the first step is taking the object/entity that is being read from an external system (SCIM server). Then, each task mapping rule is merged into this object based on the *action* and *multivalue* attributes.

The result of this operation is then sent to the external system.

4.2.6 Task Mappings

You can apply the following action attributes to task mappings: *Add*, *Modify*, *Delete*.

The *Multivalue* attribute can be set to *true* or *false*.

In the entity taken from an external system, the task mapping name attribute points to a JSON property. The dot character (“.”) indicates a child of a parent property.

In the following example, the name attribute performs an operation on the *members* array item and in the *value* property.

```
<fieldMapping name="members[].value"
```

4.2.6.1 Addition

In this action attribute, the CyberArk Connector searches for a property on an existing structure. If the parent property is an array, it adds elements to it. If the parent property is an object, it adds the property to this object.

The property is only added when there is no property of that name and it does not add duplicates. If the *Multivalue* attribute is set to *true*, the CyberArk Connector does not gather all the properties, which also has the *Multivalue* attribute set to *true* and has the same parent path.

The addition of properties to the *Multivalue* attribute works only on array. When you add properties, they create an entry based on all the multivalue fields that have the same parent path. If there is a duplicate entry, a new one is not added.

This is the addition action attribute: *action="add"*.

4.2.6.2 Modification

In this action attribute, the CyberArk Connector searches for a property on the existing structure and modifies it accordingly.

Important
Modification works only on objects, not on the array.

This is the modification action attribute: *action="modify"*.

4.2.6.3 Deletion

In this action attribute, the CyberArk Connector searches for a property on the existing structure. If the parent property is array, it searches for an array element with the property that matches the value.

If the CyberArk Connector finds the array element, it removes the entire entry. If the parent property is an object, it removes the property if it exists in the object.

This is the deletion action attribute: *action="remove"*.

4.2.6.4 Mappings configuration

During configuration of a CyberArk connector, it is recommended that the task mappings are set up using the following mappings:

Parameter	Description
CyberArk Account	Contains mappings of RoPE Accounts to CyberArk Accounts.
CyberArk Container Assignment	Contains mappings of the RoPE assignments to CyberArk Container assignments.
CyberArk Group Assignment	Contains mappings of RoPE assignments to Cyber-Ark Group assignments.