MICRO FOCUS

NetIQ® Identity Manager

Driver for SCIM Implementation Guide

August 2020

Legal Notice

For information about NetIQ trademarks, see https://www.netiq.com/company/legal/.

Contents

About NetIQ Corporation

We are a global, enterprise software company, with a focus on the three persistent challenges in your environment: Change, complexity and risk—and how we can help you control them.

Our Viewpoint

Adapting to change and managing complexity and risk are nothing new

In fact, of all the challenges you face, these are perhaps the most prominent variables that deny you the control you need to securely measure, monitor, and manage your physical, virtual, and cloud computing environments.

Enabling critical business services, better and faster

We believe that providing as much control as possible to IT organizations is the only way to enable timelier and cost effective delivery of services. Persistent pressures like change and complexity will only continue to increase as organizations continue to change and the technologies needed to manage them become inherently more complex.

Our Philosophy

Selling intelligent solutions, not just software

In order to provide reliable control, we first make sure we understand the real-world scenarios in which IT organizations like yours operate—day in and day out. That's the only way we can develop practical, intelligent IT solutions that successfully yield proven, measurable results. And that's so much more rewarding than simply selling software.

Driving your success is our passion

We place your success at the heart of how we do business. From product inception to deployment, we understand that you need IT solutions that work well and integrate seamlessly with your existing investments; you need ongoing support and training post-deployment; and you need someone that is truly easy to work with—for a change. Ultimately, when you succeed, we all succeed.

Our Solutions

- Identity & Access Governance
- Access Management
- Security Management
- Systems & Application Management
- Workload Management
- Service Management

Contacting Sales Support

For questions about products, pricing, and capabilities, contact your local partner. If you cannot contact your partner, contact our Sales Support team.

Worldwide:	www.netiq.com/about_netiq/officelocations.asp
United States and Canada:	1-888-323-6768
Email:	info@netiq.com
Web Site:	www.netiq.com

Contacting Technical Support

For specific product issues, contact our Technical Support team.

Worldwide:	www.netiq.com/support/contactinfo.asp
North and South America:	1-713-418-5555
Europe, Middle East, and Africa:	+353 (0) 91-782 677
Email:	support@netiq.com
Web Site:	www.netiq.com/support

Contacting Documentation Support

Our goal is to provide documentation that meets your needs. The documentation for this product is available on the NetIQ Web site in HTML and PDF formats on a page that does not require you to log in. If you have suggestions for documentation improvements, click **Add Comment** at the bottom of any page in the HTML version of the documentation posted at www.netiq.com/documentation. You can also email Documentation-Feedback@netiq.com. We value your input and look forward to hearing from you.

Contacting the Online User Community

NetIQ Communities, the NetIQ online community, is a collaborative network connecting you to your peers and NetIQ experts. By providing more immediate information, useful links to helpful resources, and access to NetIQ experts, NetIQ Communities helps ensure you are mastering the knowledge you need to realize the full potential of IT investments upon which you rely. For more information, visit community.netiq.com.

About This Guide

This guide explains how to install and configure the Identity Manager Driver for SCIM.

Audience

This guide is intended for administrators implementing Identity Manager, application server developers, Web services administrators, and consultants. You should also have an understanding of DSML/SPML, SCIM, JSON, and HTML.

Feedback

We want to hear your comments and suggestions about this manual and the other documentation included with this product. Please use the User Comments feature at the bottom of each page of the online documentation, or go to www.novell.com/documentation/feedback.html and enter your comments there.

Documentation Updates

For more information about the library for Identity Manager, see the following resources:

- Identity Manager documentation website (https://www.netiq.com/documentation/identity-manager-48/)
- Identity Manager drivers documentation website (https://www.netiq.com/documentation/identity-manager-48-drivers/)

1 Understanding the SCIM Driver

SCIM (System for Cross-domain Identity Management) protocol is designed to simplify user management operations. The SCIM driver provides a common user schema and an extension model which helps you to provision or deprovision identities to and from connected applications seamlessly. Use case based operations on resources, such as modifications, deletions, polling, querying, etc., can also be performed as required.

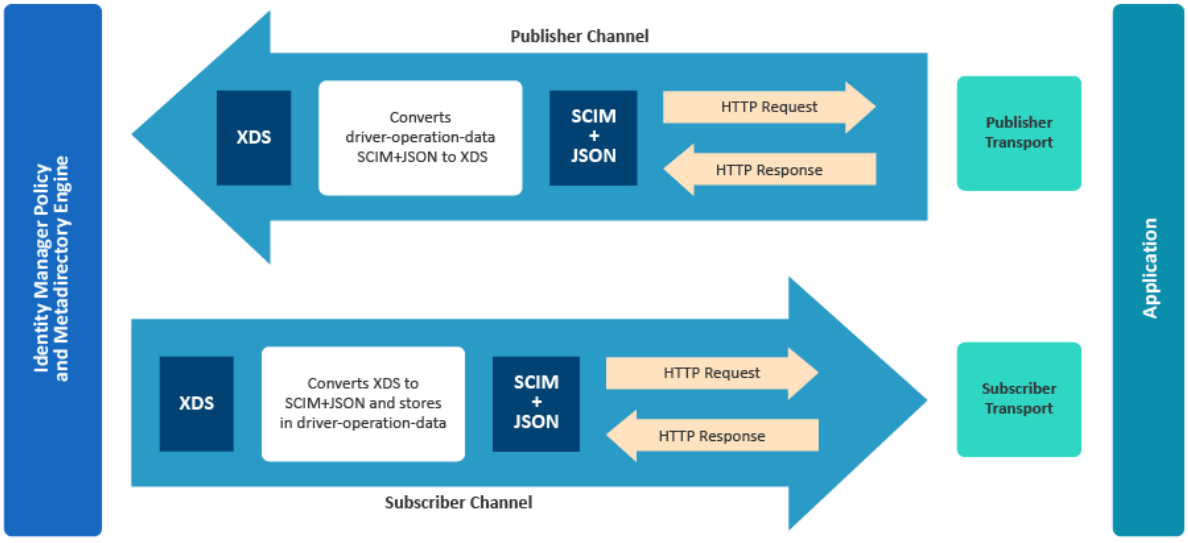
Apart from setting up and configuring the SCIM driver, this document also explains its various use cases. The use case operations that you can perform on resources such as users, groups, etc., include the following:

- Creating
- Modifying
- Deleting
- Migrating
- Polling
- Querying

SCIM Driver Architecture

The following diagram illustrates the bi-directional relationship between Identity Manager and the connected application.

Figure 1-1 *Architecture of SCIM Driver*



The Identity Manager engine and eDirectory reside on a single host computer. The SCIM Driver Module is Java based driver, that can execute on the same host system, or on a Remote Loader instance. The Identity Manager Engine loads the driver module dynamically.

The Identity Manager engine uses XDS (a specialized form of XML) to represent events in the Identity Manager. In the Subscriber channel, the XDS events from Identity Manager are converted to SCIM compliant JSON using policy conversions. The driver operation data in the driver shim, converts the SCIM compliant JSON to the appropriate HTTP requests or responses to communicate with the connected application.

In the Publisher channel, the driver fetches data from the connected application using the conversion policy. The connected application processes the request, and returns a HTTP response to the driver shim. The Publisher channel periodically polls for additions and modifications of the objects in the connected application.

SCIM Driver Packages

In Designer, navigate to **Help > Package Updates** to update the SCIM driver packages. When you update the required driver packages in designer, the designer updates the policies, rules and the parameters that are associated with the driver object. These rules, policies, and associated parameters are used to establish communication and synchronize data between Identity Manager and the connected application.

The SCIM driver packages and the details are:

- **SCIM Base (NETQSCIMBASE**): Contains the mandatory basic SCIM configurations required for SCIM driver. The base configurations include:
  - Driver Authentication methods such as Basic and OAuth2.0.
  - Subscriber settings with HTTPS error codes to retry such as 307, 400 etc,.
  - Publisher settings with Polling interval, Heartbeat interval, and Polling Resource option.
  - Advanced settings with the Schema and Modifier setting options.
  - Options for Remote Loader settings.
- **SCIM Default (NETQSCIMDCFG)**: Contains the mandatory default configurations required for configuring the SCIM driver. The default configurations include the policies as shown below:
  - **Matching policy**: This policy finds the matches for objects based on attributes.
  - **Creation policy**: The creation policy defines the conditions that must be met to create a new object. The creation policy is of two types, Subscriber Creation policy and Publisher Creation policy. The policy definitions can be same or different for the respective channels.

    For example, if you try to create a new user in Identity Manager by providing only the user's name and user ID, the user is created in Identity Manager but does not sync to the connected application. This happens when the definitions for creating the user are not specified completely in the creation policy. You can add templates in the creation policy to ensure that all the required definitions are specified.

    The Creation Policies are commonly used to:
    - Reject the creation of objects that don’t qualify, possibly because of a missing attribute.
    - Provide default attribute values.

- **Placement policy**: This policy specifies the containers where objects are to be placed.
- **Command Transformation policy**: This policy is to provide the final processing commands that are sent to the Identity Manager or to the connected application.
- **Schema Mapping policy**: The Schema Mapping policies store the definition of the class and attribute mappings between the Identity Manager and the connected application.
- **Filter**: Filter allows the object and its specific attributes to synchronize between the Identity Manger and the connected application.

- **SCIM JSON (NETQSCIMJSON)**: (Optional) This package contains the JSON configurations for SCIM driver to implement XDS to JSON conversion. The JSON that is created by using this package will be compatible with the connected application, to perform required operations.

2 Installing and Configuring SCIM Driver

You can install and configure a SCIM driver in Identity Manager to connect to SCIM based external applications. You must download and install the driver related files from the required driver build available in the Micro Focus Download site. You must also update the corresponding packages in Designer to install the driver.

IMPORTANT: The configuration parameters, sample values and examples mentioned in this chapter are for reference purposes only. You should modify them as required to suit your environment.

The following sections explain the details that are required to help you set up and configure the SCIM Driver.

Plan Your Installation

Prior to installing the driver, ensure all the prerequisites and system versions are updated as shown below:

- Prerequisites:
  - Designer version: IDM 4.8.1.1
  - Install REST binaries: REST 1.1.1
- System Requirements:
  - Identity Manager 4.7.4, or later
  - Identity Manager 4.8.1, or later

Installing the SCIM Driver

To start with installation, you must first:

- Download and install the SCIM driver files, see "Installing the SCIM Driver Files" on page 14.

- Extend the schema for SCIM Driver, see “Extending Schema For Supporting Custom Attributes Required By SCIM Driver” on page 15
- Import and Install the driver packages, see “Installing the SCIM Driver Packages in Designer” on page 16.

Installing the SCIM Driver Files

You can install the SCIM driver files as a root user or as a non-root user in your system. The procedure to install the driver files is similar for any connected application.

You must ensure that you have the required SCIM driver files such as, **`.zip, .rpm, and .jar`** etc., from the required driver build available in Micro Focus Download site to install the SCIM driver in your system.

For example:

- `.zip` file: `SCIMDriver.zip`
- `.rpm` file: `<netiq-DXMLscim.rpm>`
- `.jar` file: `<SCIMUtils.jar>`

This section explains the common procedure to install the driver files.

1. Download and unzip the contents of the **`SCIMDriver.zip`** file to a temporary location on your computer.
2. Install the driver files (for IDM 4.7.4 and above) based on your user role.

   To install as a:

   - root user, see “Installing Driver Files as a Root User” on page 14.
   - non-root user, see “Installing Driver Files as a Non-Root User” on page 14.

   Installing Driver Files as a Root User

   1. Login as a root user on the server where you want apply the driver jar file.
   2. Navigate to the extracted **`SCIMDriver.zip`** directory and perform one of the following actions based on your platform:
      - **Linux**: Install the new **`netiq-DXMLscim.rpm`** in your driver installation directory by running one of the following command in a terminal window:
        - If you are installing the binary, run the command: **`rpm -Ivh (binaries-path)/netiq-DXMLscim.rpm`**
      - **Windows**: Copy the **`SCIMShim.jar`** file to the driver’s installation folder. For example, **`\NetIQ\IdentityManager\NDS`** (local installation) or **`\Novell\RemoteLoader\64bit`** (remote installation).

   Installing Driver Files as a Non-Root User

   1. Verify that the **`/rpm`** directory exists and contains the **`_db.000`** file.
   2. The **`_db.000`** file is created during a non-root installation of the Identity Manager engine. The absence of this file indicates that the Identity Manager is not installed properly. In such a case, reinstall the Identity Manager to correctly place the file in the mentioned directory.

3. To set the root directory to the location of non-root in Identity Manager, enter the following command in the command prompt:

   ROOTDIR=<non-root eDirectory location>

   This will set the environmental variables to the directory where Identity Manager is installed as a non-root user.

4. For example, to install the SCIM driver rpm, use this command:

   rpm --dbpath $ROOTDIR/rpm -Ivh --relocate=/usr=$ROOTDIR/opt/novell/eDirectory --relocate=/etc=$ROOTDIR/etc --relocate=/opt/novell/eDirectory=$ROOTDIR/opt/novell/eDirectory --relocate=/opt/novell/dirxml=$ROOTDIR/opt/novell/dirxml --relocate=/var=$ROOTDIR/var --badreloc --nodeps --replacefiles /home/user/netiq-DXMLscim.rpm

   NOTE: In the above command **/opt/novell/eDirectory** is the location where non-root Identity Manager is installed, and **/home/user/** is the home directory of the non-root user.

3 (Conditional) If the driver is running locally, start the Identity Manager and the driver instance.

4 (Conditional) If the driver is running with a Remote Loader instance, start the Remote Loader instance and the driver instance.

Extending Schema For Supporting Custom Attributes Required By SCIM Driver

You can upload new attributes through the Identity Manager to extend the SCIM schema. The following steps explain the procedure to extend the SCIM schema:

1 Copy the following schema file to the system where Identity Manager is installed.

For example, `/root/schema/scim-schema.sch`

2 Execute the following `ndssch` command.

```
ndssch [-h hostname[:port]] [-t tree_name] [-d] admin_FDN schemafile [schema_description]
```

For example, `ndssch -h 10.71.131.123:524 -t SLES12SP3_Quality_131123_TREE -d admin.sa.system /root/schema/scim-schema.sch scim-Group`

3 The log file is created in the default location, `/root/schema.log` for troubleshooting.

4 Restart Identity Manager to see the schema changes.

Installing the SCIM Driver Packages in Designer

Once the driver files are installed, you must import and install the SCIM Base and SCIM Default Configuration packages. For more information on what is included in the SCIM driver package, see "SCIM Driver Packages" on page 10.

The generic steps to:

- import the driver packages, see Importing the Current Driver Packages in the "*NetIQ Identity Manager Driver Administration Guide*".
- install the driver packages, see Installing the Driver Files in the "*NetIQ Identity Manager Driver Administration Guide*".

IMPORTANT: You must ensure to select the following packages for the SCIM driver:

- **SCIM Base Package**:
  - **Package Name**: NETQSCIMBASE
  - **Version**: 1.0.0
  - **Build Date**: 20200812
  - **Build Number**: 172426
- **SCIM Default Configuration package** (mandatory for SCIM 1.0)
  - **Package Name**: NETQSCIMDCFG
  - **Version**: 1.0.0
  - **Build Date**: 20200806
  - **Build Number**: 185236
- **SCIM JSON Configuration package** (optional for SCIM 1.0)
  - **Package Name**: NETQSCIMJSON
  - **Version**: 1.0.0
  - **Build Date**: 20200721
  - **Build Number**: 184051

Configuring the SCIM Driver for a Connected Application

To begin with the configuration, you need to set up the SCIM driver object in the designer and configure the SCIM driver with the specific parameters to connect to an external SCIM based application.

- If you do not have the driver set and Identity Vault created in Designer, see Setting Up a New Driver Object in "*NetIQ Identity Manager Driver Administration Guide*".
- If you already have the driver set and Identity Vault in Designer, proceed with the following sections to configure the SCIM driver with a connected application.

You can configure a SCIM driver with authentication methods such as, Basic or OAuth2.0, as shown below:

- **OAuth 2.0**: The OAuth 2.0 authentication method uses query options and secret options that require token values to be configured for authentication. If the connected application supports OAuth2.0 authentication method, it is recommended to configure the SCIM driver with OAuth 2.0. For more information, see "Configuring SCIM Driver with OAuth 2.0 Authentication" on page 17.
- **Basic**: The basic authentication method uses a simple user name, a user password, and the connected application's login URL to authenticate a user to login to the application. Basic Authentication requires the password to be stored in the application itself, and this is can be accessed by other applications that are associated with it. To configure SCIM driver with basic authentication, see "Configuring SCIM Driver with Basic Authentication" on page 26.

Configuring SCIM Driver with OAuth 2.0 Authentication

The OAuth 2.0 authentication method is used for authenticating the driver with enhanced security to connect to an application. OAuth 2.0 authentication can be established using Bearer tokens or JWT's.

The following steps explain the procedure to configure the SCIM driver:

1. In the **Authentication Method field** select **OAuth 2.0**.
2. In the **OAuth2.0 Token Management field,** select the option as required. The available options are:
   - **Bearer**: A bearer token is a lightweight security token (a short string of hexadecimal characters) that grants the bearer access to a protected resource. The Bearer token is created for you by the connected application's authentication server. By selecting the **Bearer** option, you can generate a new bearer token to authorize the SCIM driver with connected application.

     To configure SCIM Driver using bearer token, see "Configure SCIM Driver with Bearer Token" on page 17.
   - **JWT**: A JSON Web Token (JWT) is part of OAuth authorization and authentication framework. A JWT securely authenticates the driver to connect to an external application to perform operations as required. By selecting the **JWT** option, you can generate a new JWT to authorize the SCIM driver with connected application.

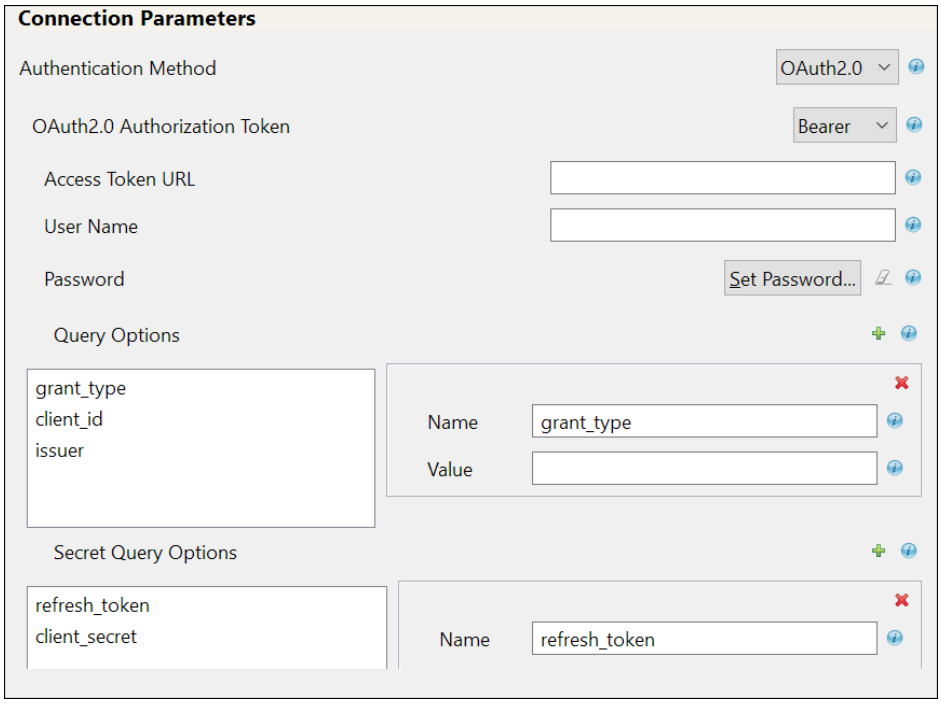
     To configure SCIM Driver using JWT, see "Configuring SCIM Driver with JWT" on page 20.
   - **Manual**: Select **Manual** if you already have a token available or created by an external application.

     To configure SCIM Driver using an available bearer token, see "Configuring SCIM Driver with an Available Token" on page 22.

NOTE: Configuring a JWT is preferred as it is more secured with a digital server certificate.

Configure SCIM Driver with Bearer Token

Bearer is an access token issued by connected application to achieve multi-server authentication.



If you select **Bearer**, the following fields appear. The sample values are shown in the following table, however you must enter values as applicable for your environment.

IMPORTANT: For any operation performed on the connected application using OAuth 2.0, an access token is sent for authorization of the user from the connected application. The access token expires post the session idle time set by the connected application, or in case of a system restart. The connected application displays the Unauthorized Access error or an Invalid Session error for any request initiated with an expired access token. The presence of a refresh token helps to re-establish the failed session internally by generating a new access token without user's intervention.

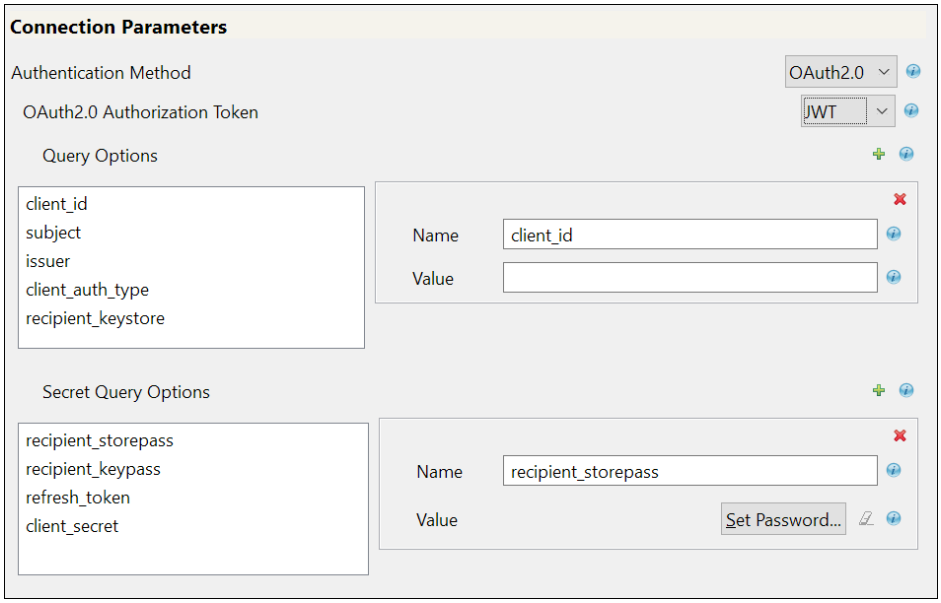
Field	Sample Field Value
Access Token URL	*<https://login.salesforce.com/services/oauth2/token>*
User Name	The user name to login to the connected application.
Password	The password to login to connected application.

Field	Sample Field Value
Query Options: The following fields appear: • **grant_type**: It is the method used by the application to procure an access token. • **client_id**: The client_id is a public identifier for connected application. • **issuer**: The authorization server's URL that uses the https protocol.	• **grant_type**: password • **client_id**: *<3MVG97quAmFZJfVwk3ylU.8elhRYBqG9h25m3TWewozjKnFIY0HrhOEJl7LMET9HHocaHnTB1k04kophr1CgW>* • **issuer**: *<https://login.Salesforce.com>* **NOTE:** In case of a driver upgrade, the **issuer** field does not auto populate the earlier configured value. You must enter the issuer field value manually.
Secret Query Options: The values specified in these options are hidden for security purposes. • **refresh_token**: Refresh Token is a web token which is used to acquire new access tokens when current access tokens expire or become invalid. The authorization server provides refresh tokens to Identity Manager to obtain new access token without user's interaction in the backend. • **client_secret**: The client secret is used to establish the ownership of the client_id.	• **refresh_token**: 5Aep861Xq7VoDavIt6UxKW62EAmfy0hKFv1T_X8yhb9PRQWtsOCrr97CYDrVasefykdl_f.DTVaJGKxjmz50XjQ • **client_secret**: E734505442694ECD0156D83F965B42C0F07601BB8BFDCA9879420C1FF23C8A87
Common fields in Connection Parameters **NOTE:** The fields mentioned in the below rows are common for **OAuth2.0** and **Basic** authentication methods.	
Application Truststore File: The path and the name of the keystore file that contains the trusted certificates for the remote server to achieve SSL handshake. **IMPORTANT:** For **Bearer**, add the public certificate to cacerts, present in the path /opt/netiq/common/jre/lib/security.	*</root/scim_configuration/trustSalesforce/Salesforce>* **NOTE:** Create the truststore file in .jks format for the connected application. For more information on how to create the truststore file, see Configuring the Subscriber Channel in "*NetIQ Identity Manager Driver Administration Guide*".
Mutual Authentication: Enable and specify this field if the authentication is supported by the connected application. You must ensure to have both the server certificates stored in Identity Manager and the connected applications. Defaults to **Hide**. Select **Show** if you want to set mutual authentication information.	• **IDM Keystore file**: Specify the path and the name of the keystore file that contains the trusted certificates for the connected application server to provide mutual authentication. For example, C:\security\keystore. • **IDM Keystore password**: Specify the password for the keystore file.

Field	Sample Field Value
Proxy Authentication: Defaults to **Hide**. Select **Show** if you want to set proxy authentication parameters. Specify the proxy host address and proxy host port in this field.	◆ **Proxy host name and port**: *<192.168.0.0:port>*. Choose an unused port number on the proxy server. ◆ **Username**: *<user name for proxy authentication>* ◆ **Password**: *<password for proxy authentication>* ◆ **Re-enter Password**: *<password for proxy authentication>*
HTTPS Connection Timeout: Specify the HTTPS connection time out value.	The timeout value must be greater than 0. **NOTE:** The driver waits for the time specified (in minutes) and terminates the HTTPS connection displaying the error code. The error codes are configured in the **Subscriber Options > HTTPS error codes for retry** field.
SCIM 2.0 URL: Enter the URL for the connected application. SCIM Resources like User, Group etc., will be appended to this URL.	`<https://salesforce.com/api/rest/scim/v2/339216517038085>`

Configuring SCIM Driver with JWT

This is a secured and digitally signed access token in the JWT format. A JWT is an encrypted data string consisting of a header, payload, and a signature, and is used to transfer authorization data in client-server applications to authenticate the identity of the resource.



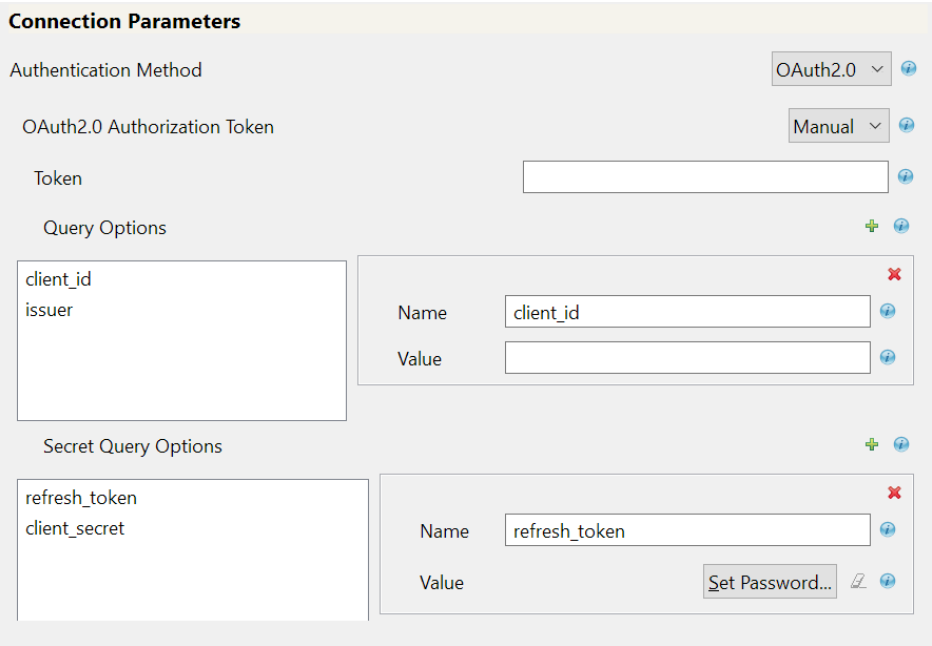
If you select **JWT**, the following fields appear:

Field	Sample Field Value
Query Options: The following fields appear: • **client_id**: The client_id is a public identifier for the connected application. • **subject**: The user's unique identity for which the access token is being requested. • **issuer**: The authorization server's URL that uses the https protocol. • **client_auth_type**: The client's authorization types configured for granting access to the application. • **recipient_keystore**: The keystore file that is used to search for the digital signature that contains the public key in the connected application. The following steps explain how to create the recipient_keystore. 1. Create the digital signature. For more information, see Create a Private Key and Self-Signed Digital Certificate. 2. Create the PKCS12 file by combining the server key and the server certificate, as shown below: **openssl pkcs12 -inkey <server key> -in <server certificate> -export -out <filename>.pkcs12** 3. Import the PKCS12 file into the recipient_keystore, as shown below: **/opt/netiq/common/jre/bin/keytool -importkeystore -srckeystore <filename>.pkcs12 -srcstoretype pkcs12 -destkeystore <recipient keystore>**	• **client_id**: *<3MVG97quAmFZJfVwk3ylU.8elhRYBqG9h25m3TWewozjKnFIY0HrhOEJl7LMET9HHocaHnTB1k04kophr1CgW>* • **subject**:*<username@microfocus.com>* • **issuer**: *<https://login.salesforce.com>* • **client_auth_type**: private_key_jwt • **recipient_keystore**: *</Soft/Certs/recipient.jks>*

Field	Sample Field Value
Secret Query Options: The values specified in these options are hidden for security purposes. • **recipient_storepass**: Password for the recipient_keystore file that is mentioned above. • **recipient_keypass**: Password for the server certificate that is available in the recipient_keystore file. • **refresh_token**: Refresh Token is a web token which is used to acquire new access tokens when current access tokens expire or become invalid. The authorization server provides refresh tokens to Identity Manager to obtain new access token without user's intervention. • **client_secret**: The client secret is used to establish the ownership of the client_id. For the other common fields such as Application Truststore File, Mutual Authentication, Proxy Authentication, HTTPS Connection Timeout, and SCIM 2.0 URL, see *"Common fields in Connection Parameters" on page 19*	• **recipient_storepass**: <*novell*> • **recipient_keypass**: <*novell*> • **refresh_token**: 5Aep861Xq7VoDavIt6UxKW62EAmfy0hKFv1T_X8yhb9PRQWtsOCrr97CYDrVasefykdl_f.DTVaJGKxjmz50XjQ • **client_secret**: E734505442694ECD0156D83F965B42C0F07601BB8BFDCA9879420C1FF23C8A87

Configuring SCIM Driver with an Available Token

Select **Manual** if you already have an access token available or created by an external application.



Field	Sample Field Value
Token: Specify access token value that is generated using API calls. For example, call REST API using Postman and specify the Bearer Token.	*<00D2v000002mBdQ!ARQAQAzAXhpgilDpcvN3RDgCkrfh4pyzCOv2G1Iq5kEMh0TRi>*
Query Options: The following fields appear. ◆ **client_id**: The client_id is a public identifier for the connected application. ◆ **issuer**: The authorization server's URL that uses the https protocol.	◆ **client_id**: *<3MVG97quAmFZJfVwk3ylU.8elhRYBqG9h25m3TWewozjKnFIY0HrhOEJl7LMET9HHocaHnTB1k04kophr1CgW>* ◆ **issuer**: *<https://login.Salesforce.com>* **NOTE:** In case of a driver upgrade, the **issuer** field does not auto populate the earlier configured value. You must enter the issuer field value manually.
Secret Query Options: The values specified in these options are hidden for security purposes. ◆ **refresh_token**: Refresh Token is a web token to acquire new access tokens when current access tokens expire or become invalid. The authorization server provides refresh tokens to the Identity Manager to obtain new access token without user's intervention. ◆ **client_secret**: The client secret is used to establish the ownership of the client_id. For the other common fields such as Application Truststore File, Mutual Authentication, Proxy Authentication, HTTPS Connection Timeout, and SCIM 2.0 URL, see *"Common fields in Connection Parameters" on page 19*	◆ **refresh_token**: 5Aep861Xq7VoDavIt6UxKW62EAmfy0hKFv1T_X8yhb9PRQWtsOCrr97CYDrVasefykdl_f.DTVaJGKxjmz50XjQ ◆ **client_secret**: E734505442694ECD0156D83F965B42C0F07601BB8BFDCA9879420C1FF23C8A87

3 In the **Install SCIM Base** page, specify the **Subscriber Options** and **Publisher Options**, and click **Next**.

Field	Description with Sample values
Subscriber Options	**HTTPS error codes for retry**: Specify the HTTPS errors that must return a retry status. Error codes must be a list of integers separated by spaces. For example: *<307 408 503 504>* **NOTE:** The operation is retried if these error codes are encountered.

Field	Description with Sample values
Publisher Options	◆ **Enable Publisher Channel**: Select **Yes** to enable the Publisher channel. ◆ **Polling interval in minutes**: Specify the polling interval in minutes For example: *<10>* ◆ **Heartbeat interval in minutes**: This option is used to configure the time interval for which the driver shim sends a periodic status message on the Publisher channel. By default, this is set to 10 minutes. **IMPORTANT:** **Polling Resource Options**: This field does not appear in this page when you are setting up the driver for the first time. These options are to be specified once the driver is configured. After configuring the driver, double click the connector line in the modeler window and navigate to **Driver Configuration > Publisher Options** tab to specify the polling resource options. ◆ Select the **Configured Resources** option to poll on all resources that are configured as part of the schema settings. ◆ Select the **Custom Resources** option and click  to configure customized polling **Resource ID** and **Resource URL**. ◆ For User: ◆ **Resource ID**: Example, `urn:ietf:params:scim:schemas:core:2.0:User` ◆ **Resource URL**: Example, `https://ap16.salesforce.com/services/scim/v2/Users?startIndex=1&count=100` **NOTE:** In the above URL's, the `startIndex` refers to the resource from where the poll must start and `count` refers to the number of resources from the `startIndex` for polling. ◆ For Group: ◆ **Resource ID**: Example, `urn:ietf:params:scim:schemas:core:2.0:Group` ◆ **Resource URL**: Example, `https://ap16.salesforce.com/services/scim/v2/Groups?startIndex=1&count=100`

4 In the **Install SCIM Base** page, specify the parameters as shown in the following table, and click **Next**.

Table 2-1 *Schema Settings*

Field	Description with Sample Values
Refresh Schema on Driver Startup	Defaults to **No**, specify **Yes** if you want to refresh the schema. **IMPORTANT:** Select this option as **Yes** to load the connected application's schema for the first time, or if the connected application's schema has changed. It is recommended to change this field to **No** once the schema is fetched successfully. If this field remains selected as **Yes**, the driver will fetch the schema from the connected application every time the driver restarts and might cause mapping issues. For more information on schema, see Chapter 6, "SCIM Schema Utility," on page 47.
Schema Options	Select required method to fetch the connected application's schema. The available options are: ◆ **SCIM 2.0**: SCIM 2.0 Schema for User and Group, as defined in RFC 7643. ◆ **Application URL**: The application's end point for SCIM schema. Example, `https://ap17.salesforce.com/services/scim/v2/Schemas`. ◆ **Import JSON File**: Import the user defined schema JSON file from the local file system. This file must comply to SCIM JSON format as per RFC 7643.
Resource Type	Specify the Resource ID and Resource Endpoint's of the resources in Uniform Resource Name (URN) Format. For example, Users, Groups, Roles, Entitlements etc. ◆ **Resource ID:** Resource ID in URN Format. For example, `urn:ietf:params:scim:schemas:core:2.0:Users` ◆ **Resource Endpoint**: The resource endpoint of the Resource ID. For example, `Users`. ◆ **Modify Method Operation**: Specify the method of operation to be performed on the resources. Select the option as supported by the connected application. The available options are: ◆ **PUT**: This option is used to modify an entire resource which is already a part of the collection of resources in the connected application. ◆ **PATCH**: This option is used to make partial updates to resources in the connected application. Similarly for Groups: ◆ **Resource ID**: Example, `urn:ietf:params:scim:schemas:core:2.0:Group` ◆ **Resource Endpoint**: `Groups` ◆ **Modify Method Operation**: Select the option as required.

Table 2-2 Modifier Settings

Field	Description
Custom Java Class	The custom Java classes that are used to extend the driver's functionality. Defaults to Hide, select Show to configure document modifiers.
Document Handling:	Select this option to customize the Java classes for processing the data as JSON objects. Defaults to No, select Yes. The **Class** and **Init Parameter** fields appear. ◆ Class: Specify the class using a full package identifier. For example, `com.example.MyNewClass` ◆ Init Parameter: Specify the parameters in string format that you want to pass to the init() method of your class. The init method of your class is responsible for parsing the information contained in this string. Leave this field blank if your class does not require a configuration string to be passed to the init method.

5 In the Remote Loader page, if you are configuring the driver with a remote loader instance select yes, else select no. Click Next.

For more information about configuring the driver with Remote Loader, see Deciding Whether to Use the Remote Loader in "*NetIQ Identity Manager Driver Administration Guide*".

6 Review the summary of tasks and click Finish. The configured driver appears in the designer screen.

Configuring SCIM Driver with Basic Authentication

The basic authentication method uses a simple user name, a user password, and the connected application's login URL, to authenticate a user to login to an application. If you select Basic in the Authentication Method field, the following fields appear:

Field	Description
User Name	Specify the name of the user.
Password	Specify the password.
Application URL	Specify the URL of the connected application.
For the other common fields such as Application Truststore File, Mutual Authentication, Proxy Authentication, HTTPS Connection Timeout, and SCIM 2.0 URL, see *"Common fields in Connection Parameters" on page 19*	

After you have specified the fields that are required for Basic authentication, continue with Step 3 on page 23.

Deploying, Starting and Activating the SCIM Driver

After installing and configuring the driver you must deploy, start and activate it. To perform the respective operations see:

- Deploying the Driver in "*NetIQ Identity Manager Driver Administration Guide*"
- Starting the Driver in "*NetIQ Identity Manager Driver Administration Guide*"
- Activating Drivers in "*NetIQ Identity Manager Driver Administration Guide*"

3 Customizing the Driver for SCIM Services

The SCIM driver customizations enable you to create and configure Java extensions, or modify the JSON/XML payload in the publisher and subscriber channels. The following sections explain the customizations that are available to establish seamless communication with the connected application.

Creating and Configuring Java Extensions

In some cases, the connected application could implement the SCIM interface to exchange information in a method that might deviate from the RFC standards. To create Java extensions, the modifier class file, for example <*SFDocModifier.jar*> must be available in the path `/opt/novell/eDirectory/lib/dirxml/classes`.

You can modify the following requests and responses using Java extensions:

- Subscriber request document to the connected application.
- Subscriber response document for Identity Manager.
- Publisher request document sent through the Publisher channel to the connected application.
- Publisher response document received through the publisher channel to Identity Manager.

You should name your modifier class using any Java package and a class name that is convenient for your environment.

For example, if you are writing your own class that implements the DocumentModifiers interface, and you named your class as *MyDocumentModifiers* within a package called `com.microfocus.idm`, then you perform the following steps to compile `.jar`, and deploy your class:

1. Prepare your environment.

   Make sure that you have a current Java Development Kit (JDK) installed on your computer. Visit the Java Web Site if you need to download one.

2. Gather your source code in the proper directory structure as defined by your package naming.

   In the above example, navigate to **com > microfocus > idm > MyDocumentModifiers.java** to find the source code file.

3. Make sure you have the jar files you need to compile your class.

   At a minimum, you need `SCIMUtils.jar`, available in:

   - Linux path: `/opt/novell/eDirectory/lib/dirxml/classes`
   - Windows path: `C:\NetIQ\IDM\NDS\lib`

   Also, if you are using XML documents within your class, you also need `nxsl.jar`, that is available in `/opt/novell/eDirectory/lib/dirxml/classes`.

4. Place the `.jar` file in the root directory. For example, out of the `com` directory.

5. Execute the command prompt or shell prompt with the above path.

6 Compile your class by entering one of the following commands:

- **For Windows:** `javac -classpath SCIMUtils.jar;nxsl.jar com\novell\idm\*.java`
- **For Linux or UNIX:** `javac -classpath SCIMUtils.jar:nxsl.jar com/novell/idm/*.java`

7 Create a Java archive file containing your class by entering one of the following commands:

- **For Windows:** `jar cvf mydriverextensions.jar com\microfocus\idm\*.class`
- **For Linux:** `jar cvf mydriverextensions.jar com/microfocus/idm/*.class`

8 Place the jar file you created in Step 7 into the same directory that contains the `SCIMShim.jar`.

- **In Windows**: `C:\NetIQ\IDM\NDS\lib`.
- **In Linux**: `/opt/novell/eDirectory/lib/dirxml/classes`

9 In iManager, edit the driver settings:

9a Select **Custom Java Extension** to **Show**.

9b Select **Document Handling** to **Implemented**.

9c Specify *`com.microfocus.idm.MyDocumentModifiers`* as the value for Class and a relevant string value for Init Parameter.

NOTE: The init parameter is the string that is passed to the init method of your class. You can add all the required information for class initialization in this file.

---

10 Restart the driver. You can now use your custom class.

Modifying the JSON/XML Payload

You can also customize the JSON/XML payload received in the Subscriber and Publisher channels through conversion policies. The conversion policies modify the format of the request and response documents for compatibility.

The conversion can be done using any of the following three methods:

- Use the default XDS to JSON conversion policy to transform the payload generated in the `<driver-operation-data>` to a format supported by your SCIM service.
- Create your own XDS to JSON conversion policy, and ensure to keep the payload in `<driver-operation-data>` so that the driver transfers the same to the connected application.
- Driver shim automatically performs the conversion without any conversion policies.

4 Managing the SCIM Driver

As you work with the SCIM driver, there are a variety of management tasks you might need to perform, including the following:

- Starting, and stopping the driver
- Viewing driver version information
- Using Named Passwords to securely store passwords associated with the driver
- Monitoring the driver's health status
- Backing up the driver
- Inspecting the driver's cache files
- Viewing the driver's statistics
- Using the DirXML Command Line utility to perform management tasks through scripts
- Securing the driver and its information

The above mentioned tasks along with several others, are common to all Identity Manager drivers, they are included in one reference guide, the "*NetIQ Identity Manager Driver Administration Guide*".

Securing the Driver

The procedure to secure the communication between Identity Manager and the connected application is common for all drivers. For more information, see Securing Communication in the "*NetIQ Identity Manager Driver Administration Guide*".

Upgrading the Driver

This is the initial release of the driver so the upgrade path is not available.

5 Sample Deployment of SCIM Driver for Salesforce

You can configure a SCIM driver in Identity Manager to connect to external applications complying to SCIM. The following section explains how to setup and configure the SCIM Driver for Salesforce.

IMPORTANT: All the field values shown in this chapter are just sample values. You must ensure not use them directly when configuring the driver.

Creating a Connected App for Identity Manager in Salesforce

Salesforce can be integrated with Identity Manager using API's and standard OAuth2.0 protocols. For more information to create a connected app in Salesforce, see Connected Apps section in Salesforce help pages.

Creating SCIM Driver Object for Connecting to Salesforce in Designer

To begin with the configuration, you need to set up the SCIM driver object in the designer, and configure the SCIM driver with the specific parameters to connect to Salesforce application.

The generic steps to set up a driver object and the configuration parameters is shown below. If you already have the driver object setup in designer, you can skip to Step 20 on page 35 to proceed with Salesforce specific configuration.

1. Open Designer.
2. In the toolbar, click **Help** > **Check for Package Updates**.
3. Select the required versions of the SCIM Base and SCIM Default packages as mentioned below:
   - **SCIM Base Package**:
     - **Package Name**: NETQSCIMBASE
     - **Version**: 1.0.0

- **Build Date**: 20200812
- **Build Number**: 172426

- **SCIM Default Package** (Mandatory):
  - **Package Name**: NETQSCIMDCFG
  - **Version**: 1.0.0
  - **Build Date**: 20200806
  - **Build Number**: 185236
- **SCIM JSON Package** (Optional):
  - **Package Name**: NETQSCIMJSON
  - **Version**: 1.0.0
  - **Build Date**: 20200721
  - **Build Number**: 184051

4 Click **OK** to update the packages.

5 In the Outline view, right-click the **Package Catalog**.

6 Click **Import Package** and scroll to find the **SCIM Salesforce Configuration** package.

- **SCIM Salesforce Configuration Package** (Mandatory):
  - **Package Name**: NETQSCIMSFCG
  - **Version**: 1.0.0
  - **Build Date**: 20200721
  - **Build Number**: 184139

7 Click **OK** to import the selected packages, then click **OK** in the successfully imported packages message. The designer is now updated with the selected package.

8 In **Designer > Outline** view, open your project.

9 Right click project > **New** > **Identity Vault**, or drag and drop **Identity Vault** from the **Palette** to **Modeler** window.

10 In the **Add Server Association** screen, select the following field values and click **OK**.

- Server DN
- Identity Manager Version
- Identity Manager Edition

The Identity Vault Credentials window appears.

11 In Identity Vault Credentials window, enter:

Field	Description
Host	The IP address of the Identity Vault's host machine.
Username	The name of the user.
Password	The password of the user to login to the identity vault.

12 Select **Save Password**, if you want to save your password for easy logins in the future.

13 Click **OK**.

The Identity Vault with the Driver Set appears in the **Modeler** window.

14 In the right pane, drag and drop the **SCIM** driver icon from the **Tools** tab in the **Modeler** window, to the Identity Vault.

15 In the **Driver Configuration Wizard**, select **SCIM Base Package** (Contains the base functionality for a driver. You must install a driver base configuration package first), and click **Next**.

NOTE: You can only select one base package.

---

16 In the **Select Mandatory Features** page, select the **SCIM Default Package**, and click **Next**.

17 (Optional) In the **Select Optional Features** page, select **SCIM JSON Package**, and click **Next**.

18 Verify if the required **Important Note** items are met, and click **Next**.

19 On the **Driver Information** page, specify a name for the driver, then click **Next**.

20 Select **OAuth 2.0** in the Authentication Method field, as the SCIM driver should be configured to connect to Salesforce with **OAuth 2.0** as the authentication method.

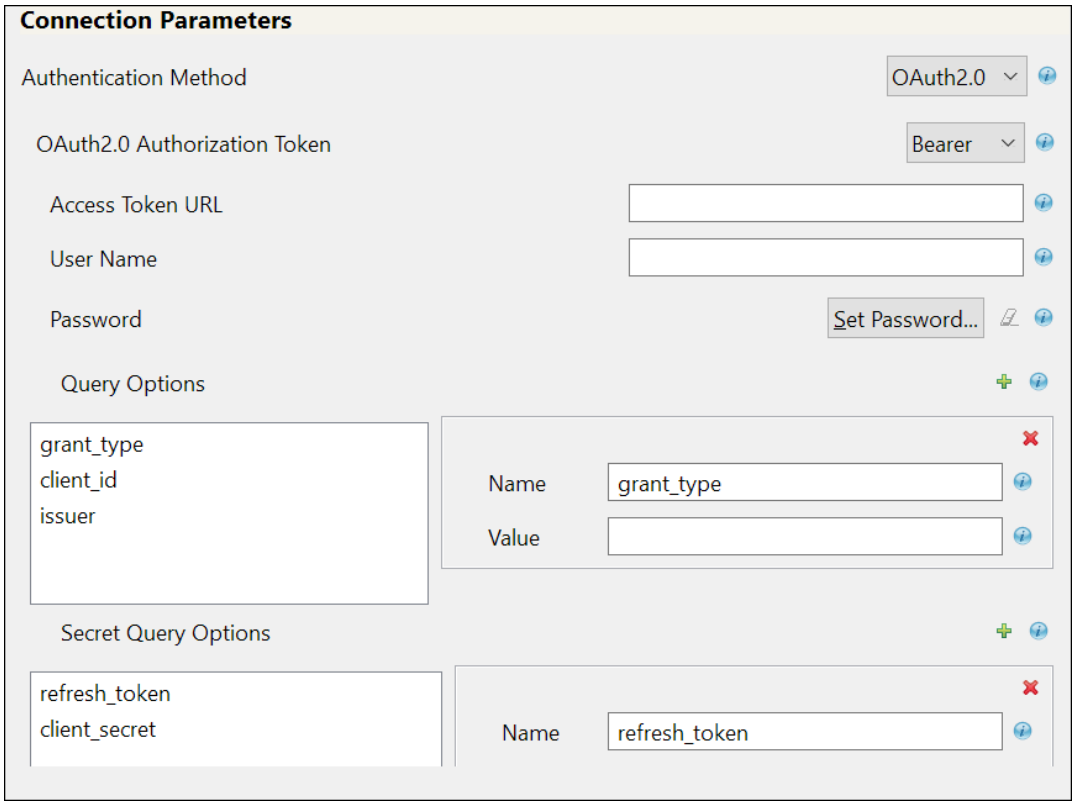
21 In the **OAuth2.0 Authorization Token** field, select the option as required. The available options are:

- **Bearer**: To configure SCIM Driver with new bearer token, see "Configuring SCIM Driver with Bearer Token" on page 35.
- **JWT**: To configure SCIM Driver using **JWT**, see "Configuring SCIM Driver with JWT" on page 38
- **Manual**: To configure SCIM Driver using an available bearer token, see "Configuring SCIM Driver Manually with an Available Token" on page 39

NOTE: Configuring a **JWT** is recommended as it is more secured with a digital server certificate.

Configuring SCIM Driver with Bearer Token

Bearer is an access token issued by servers (Salesforce) to achieve multi-server authentication.



If you select **Bearer**, the following fields appear. Enter the values as shown in the following table.

IMPORTANT: For any operation performed on the Salesforce application using OAuth 2.0, an access token is sent for authorization of the user from Salesforce. The access token expires post the session idle time set for Salesforce, or in case of a system restart. Salesforce displays Unauthorized Access error or an Invalid Session error for any request initiated with an expired access token. The presence of a refresh token helps to re-establish the failed session internally by generating a new access token without user's intervention.

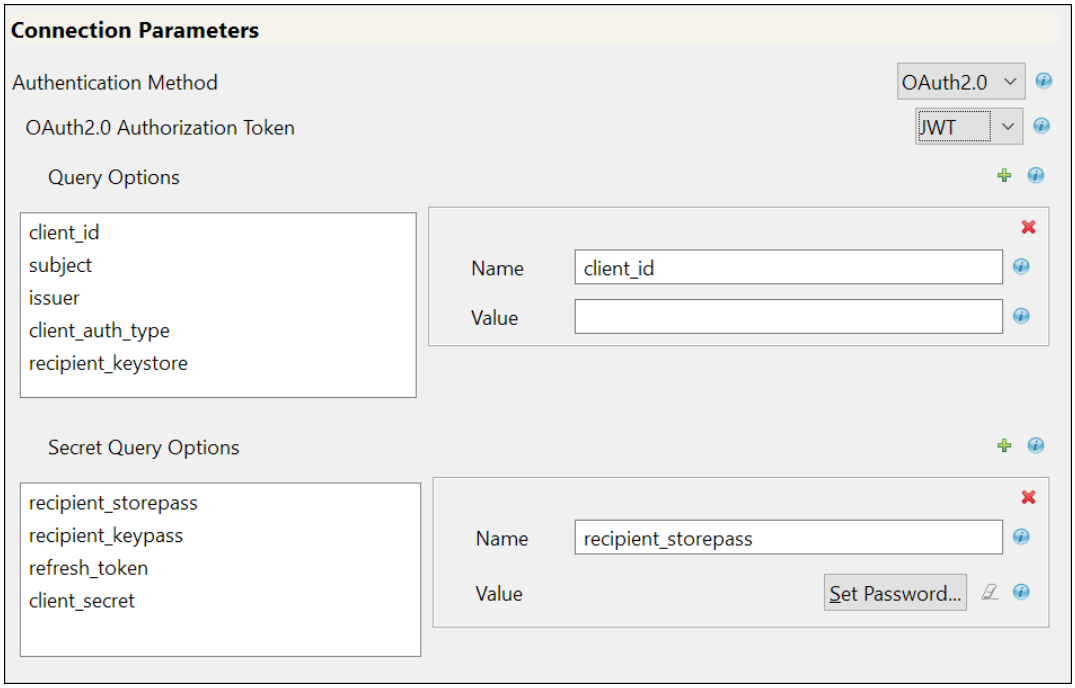
Field	Sample Field Value
Access Token URL	*<https://login.salesforce.com/services/oauth2/token>*
User Name	The user name to login to Salesforce.
Password	The password to login to Salesforce.

Field	Sample Field Value
Query Options: The following fields appear. • **grant_type** • **client_id** • **issuer**	• **grant_type**: password • **client_id**: *<3MVG97quAmFZJfVwk3ylU.8elhRYBqG9h25m3TWewozjKnFIY0HrhOEJl7LMET9HHocaHnTB1k04kophr1CgW>* • **issuer**: *<https://login.Salesforce.com>* • **username**: *<username to login to Salesforce>* **NOTE:** In case of a driver upgrade, the **issuer** field does not auto populate the earlier configured value. You must enter the issuer field manually.
Secret Query Options: The values specified in these options are hidden for security purposes. • **refresh_token** • **client_secret**	• **refresh_token**: 5Aep861Xq7VoDavIt6UxKW62EAmfy0hKFv1T_X8yhb9PRQWtsOCrr97CYDrVasefykdl_f.DTVaJGKxjmz50XjQ • **client_secret**: E734505442694ECD0156D83F965B42C0F07601BB8BFDCA9879420C1FF23C8A87 • **password**: *<password to login to Salesforce>*
Header Fields	• **Name**: Content-Type • **Value**: application/x-www-form-urlencoded
Common fields in Connection Parameters **NOTE:** The fields mentioned in the below rows are common for OAuth2.0 and Basic Authentication.	
Application Truststore File: The path and the name of the keystore file that contains the trusted certificates for the remote server to achieve SSL handshake. **IMPORTANT:** For **Bearer**, add the public certificate to cacerts, present in the path /opt/netiq/common/jre/lib/security.	*</root/scim_configuration/trustSalesforce/Salesforce>* **NOTE:** Create the truststore file in .jks format for the connected application. For more information on how to create the truststore file, see Configuring the Subscriber Channel in "*NetIQ Identity Manager Driver Administration Guide*".
Mutual Authentication	Not supported in Salesforce
Proxy Authentication	• **Proxy host name and port**: *<192.168.0.0:port>*. Choose an unused port number on the proxy server. • **Username** • **Password** • **Re-enter Password**

Field	Sample Field Value
HTTPS Connection Timeout	The timeout value must be greater than 0. **NOTE:** The driver waits for the time specified (in minutes) and terminates the HTTPS connection displaying the error codes that are configured in the **Subscriber Options > HTTPS error codes for retry** field.
SCIM 2.0 URL	*<https://salesforce.com/api/rest/scim/v2/339216517038085>*

Configuring SCIM Driver with JWT

The JSON Web token is an access request token in the JSON Web Token (JWT) format. It is an encrypted data string consisting of a header, payload, and a signature, and is used to transfer authorization data in client-server applications to authenticate the identity of the resource.

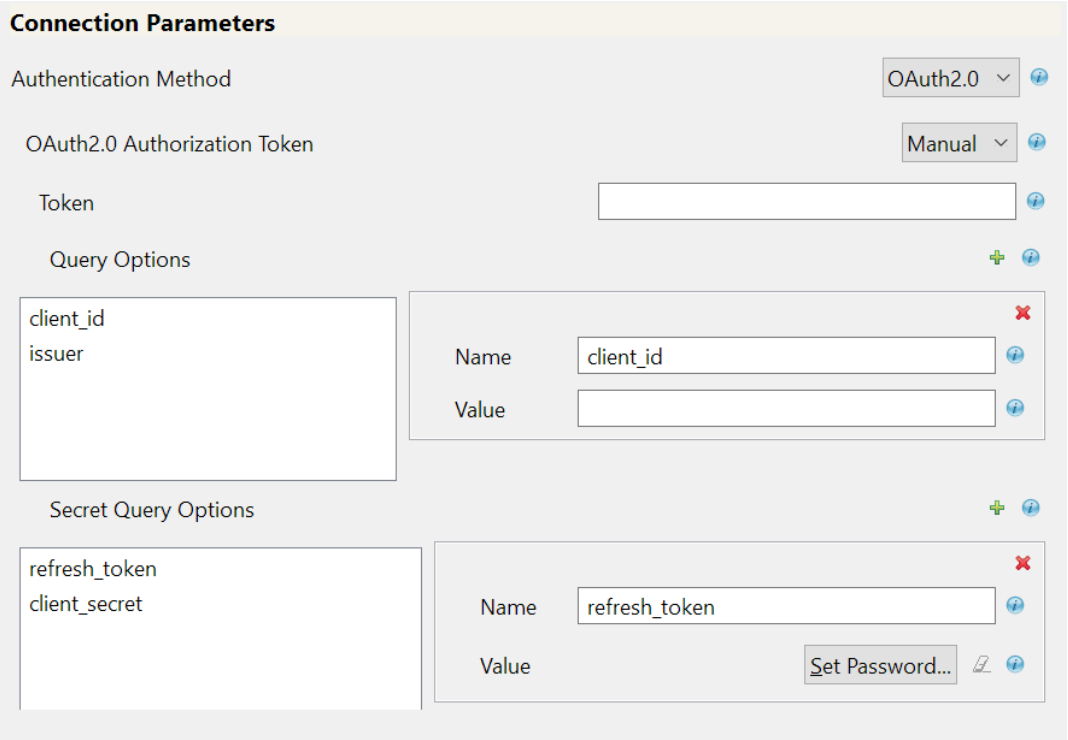


If you select **JWT,** the following fields appear:

Field	Sample Field Value
Query Options: The following fields appear: • **client_id** • **subject** • **issuer** • **client_auth_type** • **recipient_keystore**	• **client_id**: *<3MVG97quAmFZJfVwk3ylU.8elhRYBqG9h25m3TWewozjKnFIY0HrhOEJl7LMET9HHocaHnTB1k04kophr1CgW>* • **subject**:*<username@microfocus.com>* • **issuer**: *<https://login.salesforce.com>* • **client_auth_type**: private_key_jwt • **recipient_keystore**: *</Soft/Certs/recipient.jks>*
Secret Query Options: The values specified in these options are hidden for security purposes. • **recipient_storepass** • **recipient_keypass** • **refresh_token** • **client_secret**	• **recipient_storepass**: *<novell>* • **recipient_keypass**: *<novell>* • **refresh_token**: 5Aep861Xq7VoDavIt6UxKW62EAmfy0hKFv1T_X8yhb9PRQWtsOCrr97CYDrVasefykdl_f.DTVaJGKxjmz50XjQ • **client_secret**: E734505442694ECD0156D83F965B42C0F07601BB8BFDCA9879420C1FF23C8A87
For the other common fields such as Application Truststore File, Mutual Authentication, Proxy Authentication, HTTPS connection Timeout, and SCIM 2.0 URL, see "Common fields in Connection Parameters" on page 37	

Configuring SCIM Driver Manually with an Available Token

Select **Manual** if you already have an access token available or created by an external application.



Field	Sample Field Value
Token	*<00D2v000002mBdQ!ARQAQAzAXhpgilDpcvN3RDgCkrfh4pyzCOv2G1Iq5kEMh0TRi>*
Query Options: The following fields appear. • **client_id** • **issuer**	• **client_id**: *<3MVG97quAmFZJfVwk3ylU.8elhRYBqG9h25m3TWewozjKnFIY0HrhOEJl7LMET9HHocaHnTB1k04kophr1CgW>* • **issuer**: *<https://login.Salesforce.com>*
Secret Query Options: The values specified in these options are hidden for security purposes. • **refresh_token** • **client_secret**	• **refresh_token**: 5Aep861Xq7VoDavIt6UxKW62EAmfy0hKFv1T_X8yhb9PRQWtsOCrr97CYDrVasefykdl_f.DTVaJGKxjmz50XjQ • **client_secret**: E734505442694ECD0156D83F965B42C0F07601BB8BFDCA9879420C1FF23C8A87
For the other common fields such as Application Truststore File, Mutual Authentication, Proxy Authentication, HTTPS connection Timeout, and SCIM 2.0 URL, see "Common fields in Connection Parameters" on page 37	

22 In the **Install SCIM Base** page, specify the **Subscriber Options** and **Publisher Options**, and click **Next**.

Field	Sample Field Value
Subscriber Options	HTTPS error codes for retry: *<307 408 503 504>* **NOTE:** The operation is retried if these errors are encountered.
Publisher Options	◆ Enable Publisher Channel: Select Yes to enable the Publisher channel. ◆ Polling interval in minutes: *<10>* ◆ Heartbeat interval in minutes: *<10>* **IMPORTANT:** Polling Resource Options: After configuring the driver, double click the connector line in the modeler window and navigate to Driver Configuration > Publisher Options tab to specify the polling resource options. Select the option as required: ◆ Configured Resources: to poll all resources that are configured as part of the schema settings. ◆ Custom Resources: Click ✚ to configure customized polling Resource ID and Resource URL, as shown below: ◆ For User: ◆ Resource ID: Example, `urn:ietf:params:scim:schemas:core:2.0:User` ◆ Resource URL: Example, `https://ap16.salesforce.com/services/scim/v2/Users?startIndex=1&count=100` ◆ For Group: ◆ Resource ID: Example, `urn:ietf:params:scim:schemas:core:2.0:Group` ◆ Resource URL: Example, `https://ap16.salesforce.com/services/scim/v2/Groups?startIndex=1&count=100`

23 In the Install SCIM Base page, specify the parameters as shown in the following table, and click Next.

Table 5-1 *Schema Settings*

Field	Sample Field Value
Refresh Schema on Driver Startup	Defaults to No, specify Yes if you want to refresh the schema. For more information on schema, see Chapter 6, “SCIM Schema Utility,” on page 47.

Field	Sample Field Value
Schema Options	Select the option as required, the default value is SCIM 2.0. The available options are: • SCIM 2.0 • Application URL: <*https://ap17.salesforce.com/services/scim/v2/Schemas*> • Import JSON File: Import the user defined schema JSON file from the local file system.
Resource Type	• Resource ID: Resource ID in URN Format. For example, `urn:ietf:params:scim:schemas:core:2.0:Users` • Resource Endpoint: The resource endpoint for the Resource ID. For example, `Users`. • Modify Method Operation: Select PUT when you want to modify a resource in Salesforce. Similarly for Groups: • Resource ID: Example, `urn:ietf:params:scim:schemas:core:2.0:Group` • Resource Endpoint: `Groups` • Modify Method Operation: Select PUT.

Table 5-2 *Modifier Settings*

Field	Sample Field Value
Custom Java Class	Defaults to Hide, select Show to configure Modifiers.
Document Handling: Defaults to No, select Yes.	• Class: *com.example.MyNewClass* • Init Parameter: Specify the parameters in string format that you want to pass to the init() method of your class.

24 In the Remote Loader page, if you are configuring the driver with a remote loader select yes, else select no. Click Next.

For more information about installing Remote Loader, see Deciding Whether to Use the Remote Loader in "*NetIQ Identity Manager Driver Administration Guide*".

25 Review the summary of tasks, and click Finish. The configured driver appears in the designer screen.

Global Configuration Values

After configuring the SCIM driver, you can set the Global Configuration Values (GCVs) as required. For more information, see "Global Configuration Values" on page 51.

The SCIM driver includes the predefined GCV as shown below:

- **Validate Resource with Required Attributes**: Select as **true**, to validate resources and the required attributes that are available in the schema.

For more information on GCVs, see When and How to Use Global Configuration Values in "*NetIQ Identity Manager Driver Administration Guide*".

Sample SCIM Driver Use Cases for Salesforce

This section explains the sample use cases that you can perform in Identity Manager to execute the required operation on resources available in Salesforce.

IMPORTANT: All the field values shown in this section are just sample values. You must ensure not use them directly to perform the use case operations.

The following operations can be performed on the subscriber channel:

NOTE: You must replace the variable values in the SCIM end point URL as per Salesforce specifications. These are just sample values, replace them as applicable for the SCIM end point examples mentioned in other sections.

- *<tenant name>* with *ap16, ap17,* etc.
- *<current version>* with *v2,* etc.
- *<association>* with *salesforce-userid, salesforce-groupid*, etc.

- **Operations performed on a user**

Operation	Sample SCIM endpoint	Method
Adding a user: A user is added in Identity Manager and synchronized to Salesforce through the SCIM driver. For example, the details of the user such as, user's first name, last name, contact details, email ID, location, department, user name, initial login password are added and synchronized with Salesforce. **IMPORTANT:** Ensure to add the auxiliary class `scim-User` to the object class attribute. In case the auxiliary class is not added, the scim related attributes such as, `scim-Entitlementsvalue, scim-Address` will not be displayed, and the user created in iManager will not sync to Salesforce. To sync the created user to Salesforce, you must mandatorily provide the **`scim-Entitlementsvalue`** attribute value. For example, ***<00e2x000000K4Yv>***.	`https://<tenantname>.salesforce.com/services/scim/<current version>/Users`	POST
Deleting a user: Deleting a user in Identity Manager disables the user in Salesforce.	`https://<tenantname>.salesforce.com/services/scim/<current version>/Users/<salesforce-userid>`	DELETE
Modifying a user: If there are any changes made to the user details such as, contact details, email ID etc, they will be synchronized with Salesforce. **NOTE:** Salesforce does not support renaming a user.	`https://<tenantname>.salesforce.com/services/scim/<current version>/Users/<salesforce-userid>`	PUT
Migrating a user: You can migrate an individual or multiple users from Identity Manager to Salesforce and vice-versa.	`https://<tenantname>.salesforce.com/services/scim/<current version>/Users`	GET/PUT
Polling a user: You can poll a user or multiple users from Salesforce to Identity Manager.	`https://<tenantname>.salesforce.com/services/scim/<current version>/Users/`	GET

Operation	Sample SCIM endpoint	Method
Querying a User: You can query the synced attributes of resource such as user from Salesforce through iManager. Also, you can query through dxcmd utility to fetch required resources or attributes using specific conditions.	https://<tenantname>.salesforce.com/services/scim/<current version>/Users/<*salesforce-userid*>	GET **NOTE:** Complex JSON attributes cannot be queried from SCIM compliant applications through dxcmd utility.

- **Operations performed on public groups**

Operation	Sample SCIM endpoint	Method
Adding a group: A group is added in Identity Manager to manage multiple users with same set of access permissions, rather than managing them individually.	The SCIM end point for Salesforce to add a group: https://<*tenantname*>.salesforce.com/services/scim/<*current version*>/Groups	POST
Adding member to a group: A member is added to a group based on the user's role, department and access permissions that the user qualifies for, so that the access permissions for that designated user role are provisioned accordingly.	The SCIM end point for Salesforce to add a member to a group: https://<*tenantname*>.salesforce.com/services/scim/<*current version*>/Groups	PUT
Removing member from a group: A user can be removed from a group if the user's role or designation, or access permissions provided do not qualify a user to belong to that group. This happens in case of a role or designation change of the user, or separation or termination of the user.	The SCIM end point for Salesforce to remove a member from a group: https://<*tenantname*>.salesforce.com/services/scim/<*current version*>/Groups/<*salesforce-groupid*>	PUT
Renaming group object: The group name can be renamed as required.	The SCIM end point for Salesforce to renaming a group: https://<*tenantname*>.salesforce.com/services/scim/<*current version*>/Groups/<*salesforce-groupid*>	PUT
Deleting a group: Duplicate groups, redundant groups, empty groups or groups that are not required can be deleted, and the group members will be moved to another group as required.	The SCIM end point for Salesforce to delete a group: https://<*tenantname*>.salesforce.com/services/scim/<*current version*>/Groups/<*salesforce-groupid*>	DELETE

Operation	Sample SCIM endpoint	Method
Migrating a Group: You can migrate an individual or multiple groups from Identity Manager to Salesforce and vice-versa.	The SCIM end point for Salesforce to add a member to a group: `https://<tenantname>.salesforce.com/services/scim/<current version>/Groups`	PUT/GET
Polling a Group: You can poll groups from Salesforce to Identity Manager.	The SCIM end point for Salesforce to poll groups: `https://<tenantname>.salesforce.com/services/scim/<current version>/Groups`	GET
Querying a Group: You can query the synced attributes of resource such as group from Salesforce through iManager. Also, you can query through `dxcmd` utility to fetch required resources or attributes using specific conditions.	The SCIM end point for Salesforce to query groups: `https://<tenantname>.salesforce.com/services/scim/<current version>/Groups`	GET **NOTE:** Complex JSON attributes cannot be queried from SCIM compliant applications through `dxcmd` utility.

Known Observations from Salesforce

The following are a few observations when some specific operations are performed in Salesforce:

- If you try to modify an email ID in Identity Manager and sync with Salesforce, the email ID does not get updated in Salesforce. The success code 200 is returned which appears in the driver log when this operation is performed.
- Salesforce does not support renaming a user from Identity Manager.
- Salesforce does not support the canonical type attribute for the phone number.
- Renaming a group in Identity Manager changes only the label attribute and not the name attribute.
- By default, you can poll only up to 10 users from Salesforce using the GET method.

Mapping Attributes for Salesforce

The attributes of Identity Manager and Salesforce must be mapped as per the schema mapping policy. After fetching the schema from Salesforce, the attributes of Identity Manager and Salesforce are mapped in the backend by default. You can modify the attributes if any changes are required.

For the procedure to modify or change any attribute mapping, see “Refreshing the Fetched Connected Application’s Schema” on page 47.

You can also refer to Appendix C, “Mapping Attributes for Identity Manager and Connected Application,” on page 55 for the list of attributes that are available for mapping.

For more information on the terminologies and conventions of Salesforce, see Connected App and OAuth Terminology.

6 SCIM Schema Utility

The SCIM schema utility is used to fetch the connected application's schema. Using the schema mapping policy, the resource attributes of connected application are mapped with the respective resource attributes of Identity Manager.

You can fetch schema of the connected application using one of the following methods:

- **SCIM 2.0**: The default schema for Users and Groups as defined in RFC 7643, which holds core users and group along with extended user schema definition.
- **Application URL**: This utility fetches the schema by querying an application URL. For example, *<https://ap17.salesforce.com/services/scim/v2/Schemas>*
- **Import JSON**: If the schema endpoint of a connected application is not available, you can provide a user defined schema file from your local file system. For example, *<NIdM_Driver_SCIM\schema\scim_default_schemas>*

After the schema is fetched successfully, the connected application's resources and its attributes are available in the schema mapping policy. You can now use the new schema for mapping the resources and its attributes accordingly.

Refreshing the Fetched Connected Application's Schema

When you configure the driver for the first time, you must set the **Refresh Schema on Driver Startup** to **Yes** and specify the **Schema Options** for fetching the connected application's schema. Once these parameters are set and you start the driver, the driver fetches the connected applications schema and stores it in the driver storage (DirXML-DriverStorage:), which is available in **iManager > Driver Properties > General** tab.

In iManager, the procedure to refresh the schema, or fetch a new schema for mapping is shown below:

1. Login to iManager.
2. Select **Identity Manager Overview**.
3. Click **Driver Sets** tab, all the configured drivers appear.

   NOTE: If the driver set is not listed on the **Driver Sets** tab, use the **Search In** field to search for and display the driver set.
4. Click the driver name, the **Driver Set Overview** page appears.
5. Select **Schema Mapping Policies** in the SCIM driver diagram.
6. Open the schema mapping policy that is available.
7. Click **Refresh Application Schema** button, and click **OK** to confirm. A confirmation message appears displaying the successful schema refresh action.
8. Select fetched schema's resource type with the corresponding Identity Manager's resource and click **Add**. Perform this step for all the resource types that are to be mapped.

9 After mapping all the resource types, Select the **Resource Type** and click **Attribute** button.

10 In the **Identity Manager Schema Mapping Policy Editor** window, select the corresponding attribute value for the resource type and click **Add** > **OK**.

Similarly, map all the resource types with their corresponding attributes. For more information on mapping attributes see, Appendix C, “Mapping Attributes for Identity Manager and Connected Application,” on page 55.

11 Click **Apply** > **OK**.

12 Now, click the **Driver Filter** in the SCIM driver diagram.

13 In the **Filter** window, scroll to find the mapped attribute and select it. The fields associated with the selected attribute appears in the right pane.

14 Select the **Synchronize** radio button in the **Publish** and **Subscribe** options.

15 Click **Apply > OK**.

Adding a New Resource to Schema Mapping Policy

You can add resources and map the corresponding attributes to Identity Manager in the schema mapping policy.

The procedure to add a resource and map the attributes is shown below:

1 Ensure that the schema definition is available for the required resource and attributes either in the application schema or in the specified JSON schema file.

2 Add the resource/entry in the resource type.

3 Refresh the connected application’s schema in schema mapping policy.

4 Map the resources and attributes.

5 Modify the filter for the resource and attribute.

SCIM JSON Attribute Representation Using SCIM Schema Utility

The schema mapping policy comprises all the resource attributes. The attributes are represented with their sub-attributes, or their canonical types, or both. These attributes are modified to the required format as explained in the following sections.

Attribute Representation Using SCIM Utility Grammar With Delimiters

The resource attributes in the Identity Manager are in the JSON format. These attributes are of singular, complex, complex multivalued types. For schema mapping, the resource attributes can be from a core class or from an extension. The SCIM driver modifies the JSON format to a linear SCIM format using delimiters, as shown below:

- **+** as the urn(Resource) delimiter
- **:** as the attribute-Sub attributes delimiter

The delimiters as mentioned earlier are used to represent the SCIM attributes as shown below:

- Core attributes: The core attributes are of three types and are delimited by `:`, as shown below:
  - Singular: *<attribute>*
  - Complex Singular: *<attribute>:<subattribute>*
  - Complex Multi-valued: *<attribute>:<cannonicalType>:<subattribute>*
- Extensions attributes: The extension attributes are associated to the URN with a **+**, as shown below:
  - Singular: *<urn>+<attribute>*
  - Complex Singular: *<urn>+<attribute>:<subattribute>*
  - Complex Multi-valued: *<urn>+<attribute>:<cannonicalType>:<subattribute>*

Formatting JSON Structures to SCIM Attributes

The SCIM driver formats the JSON structure into a linear format SCIM attribute using delimiters. The following table shows how the JSON structures are transformed into linear SCIM attributes using utility grammar.

JSON Structure	SCIM Attributes	Grammar
Singular Attribute `"userName"`:johndoe@microfocus.com	`username`	<attribute>
Complex Attribute `"phoneNumbers":[ {"type":"work", "value":"09663502443"}, ],`	◆ `phoneNumbers` ◆ `phoneNumbers:work` ◆ `phoneNumbers:value` ◆ `phoneNumbers:work:value`	◆ <attribute> ◆ <attribute>:<cannonicalType> ◆ <attribute>:<subattribute> ◆ <attribute>:<cannonicalType>:<subattribute>
Extension Attribute `"urn:ietf:params:scim:schemas:extension:enterprise:2.0:User": { "organization":"00D2v000002mBdQEAU","employeeNumber":"21212" },`	◆ `urn:ietf:params:scim:schemas:extension:enterprise:2.0:User+Organization` ◆ `urn:ietf:params:scim:schemas:extension:enterprise:2.0:User+employeeNumber`	<urn>+<attribute>

JSON Structure	SCIM Attributes	Grammar

Complex Values in Extension Attribute
 	
`"urn:ietf:params:scim:schemas:extension:enterprise:2.0:User":`
`{`
`"manager":`
`[`
`{ "displayName":"John Doe"`
`},`
`],`
`},`
	`urn:ietf:params:scim:schemas:extension:enterprise:2.0:User+Manager:displayName`	◆ <urn>+<attribute>:<cannonicalType>:<subattribute>
		◆ <urn>+<attribute>:<subattribute>

A Driver Properties

This section provides information about the Driver Configuration and Global Configuration Values properties for the SCIM driver. These are the only unique properties for drivers. All other driver properties (Named Password, Engine Control Values, Log Level, and so forth) are common to all drivers.

For more information, see Driver Properties in the "*NetIQ Identity Manager Driver Administration Guide*".

Global Configuration Values

Global Configuration Values (GCVs) are values that can be used by the driver to control its functionality. GCVs are defined in the driver or in the driver set. Driver set GCVs can be used by all drivers in the driver set.

The SCIM driver includes predefined GCVs. You can also add your own GCVs as required for the additional policy implementation in the driver. The configured SCIM driver's GCV is:

- **Validate Resource with Required Attributes**: Select as **true**, to validate resources and the required attributes that are available in the schema.

For more information on GCVs, see When and How to Use Global Configuration Values in "*NetIQ Identity Manager Driver Administration Guide*".

B Trace Levels

The driver supports the following trace levels:

Table B-1 *Supported Trace Levels*

Level	Description
0	Driver status messages. All warnings and failure status is captured.
1	Driver status and Driver initialization messages. The success, warnings and failure status are captured.
2	Previous levels plus all other error details.
3 and 4	Previous levels plus XDS to JSON parser processing details.
5	Previous level plus all configured debug messages.
6	Previous levels plus HTTPS request documents.
7	Previous levels plus HTTPS response documents.

For information about setting driver trace levels, see “Viewing Identity Manager Processes” in the *NetIQ Identity Manager Driver Administration Guide*.

C Mapping Attributes for Identity Manager and Connected Application

The following table shows the mapping of the User class and Group class SCIM attributes between the SCIM compliant connected application and the Identity Manager.

Table C-1 *Mapping of User Attributes*

Identity Manager Attribute	SCIM Attribute	Description
User	urn:ietf:params:scim:schemas :core:2.0:User	User class
CN	userName	The user’s name.
displayName	displayName	The name of the user that is displayed in the application.
Surname	name:familyName	User’s last name or family name. For example, Jensen, given the full name Ms. Barbara J Jensen, III.
Full Name	name:formatted	User’s full name. For example, the full name Ms. Barbara J Jensen, III.
GivenName	name:givenName	User’s first name or given name. For example, Barbara, given the full name Ms. Barbara J Jensen, III.
initials	name:middleName	The user’s initials. For example, J, given the full name Ms. Barbara J Jensen, III.
personalTitle	name:honorificPrefix	The user’s honorofic prefixes. For example, Ms, in the name Ms. Barbara J Jensen, III.
Generational Qualifier	name:honoroficSuffix	The user’s honorofic suffix. For example, III, given the full name Ms. Barbara J Jensen, III.
LoginDisabled	active	A boolean value indicating the user's administrative status.
InternetEmailAddress	emails:work:value	The user's email address.

Identity Manager Attribute	SCIM Attribute	Description
scim-id	id	The user's unique identifier.
preferredName	nickName	The user's preferred name.
title	title	The user's designation.
employeeType	userType	The user's employment type.
scim-Emails	emails	The user's email ID. Example: {"type":"work", "primary":true, "value":"username@mf.com"}
scim-WorkEmails	emails:work	The user's work email ID. Example: username@mf.com
Internet EMail Address	emails:work:value	The user's work email ID. Example: username@mf.com
scim-HomeEmails	emails:home	The user's home email ID.
scim-PrimaryEmail	emails:primary	The user's primary email ID.
Telephone Number	phonenumbers:work:value	The user's phone number.
scim-Addresses	addresses	The user's address. Example: {"type":"work","primary":true,"streetAddress":"12 Market","locality":"Washington","region":"BC","postalCode":"810022","country":"USA","formatted":"12 Market\nWashington, BC 810022 USA"}
scim-UserGroups	groups	Group object
scim-EntitlementsValue	entitlements:value	The user's who are entitled with the required set of permissions. For example, chatter free user, identity user, force.com user (entitlements specific to Salesforce).
workforceID	urn:ietf:params:scim:schemas:extension:enterprise:2.0:User+employeeNumber	The user's employee identification number.
costCenter	urn:ietf:params:scim:schemas:extension:enterprise:2.0:User+costCenter	The cost center to which the user belongs.
scim-UserManager	urn:ietf:params:scim:schemas:extension:enterprise:2.0:User+manager	The manager details of a user. Example: { "value": "0052v00000gUpxlAAC", "$ref": "/Users/0052v00000gUpxlAAC", "displayName": "John Doe"}

Identity Manager Attribute	SCIM Attribute	Description
scim-UserManagerName	urn:ietf:params:scim:schemas:extension:enterprise:2.0:User+manager:value	The manager's name of the user. Example: John Doe.
OU	urn:ietf:params:scim:schemas:extension:enterprise:2.0:User+organization	Organization detail of the user.
departmentNumber	urn:ietf:params:scim:schemas:extension:enterprise:2.0:User+department	Department name of the user. Example: Sales, HR, Finance, etc.

Table C-2 *Mapping of Group Attributes*

Identity Manager Attribute	Keeper Security Attribute	Description
Group	urn:ietf:params:scim:schemas:core:2.0:Group	Group class
scim-members	members	The list of members in the Group. For example, a member detail in a particular group is as follows: `{"value": "0052x000002jjQ3AAI", "type": "User", "$ref": "https://ap17.salesforce.com/services/scim/v2/Users/0052x000002jjQ3AAI"}`
scim-UserMembers	members:User:value	Member ID value. For example: 0052x000002jjQ3AAI, as shown in the above example.
scim-GroupMembers	members:Group:value	Group ID value. For example: 00G2x000000m85wEAA

D Troubleshooting the Driver

Hidden JSON Content in Output Transformation Policy Channels

For security reasons, the content of JSON in the traces are hidden by default. This is done as there may be sensitive information and sensitive attribute values present in the JSON traces. This occurs due to the presence of `Is sensitive` attribute in the output transformation policy channel which suppresses the JSON content.

To troubleshoot and see the hidden JSON content, you must remove the `Is sensitive` attribute.

Troubleshooting Driver Processes

Viewing driver processes is necessary to analyze unexpected behavior. To view the driver processing events, use DSTrace. You should only use it during testing and troubleshooting the driver. Running DSTrace while the drivers are in production increases the utilization on the Identity Manager server and can cause events to process very slowly. For more information, see "Viewing Identity Manager Processes" in the *NetIQ Identity Manager Driver Administration Guide*.

Resource Attributes Modification Conflicts During Migration Operation

According to the migration rule, during the migration operation the driver merges the attribute values of resources which results in a conflict. This happens if the Merge Authority value is set to default in the driver filter.

Workaround: Set the Merge Authority value to IDV.