Oracle® Identity Governance

Configuring the Generic SCIM Application

Release 12.2.1.3.0
F78259-04
February 2026

ORACLE

Oracle Identity Governance Configuring the Generic SCIM Application, Release 12.2.1.3.0

F78259-04

Primary Author: Oracle Corporation

Contents

List of Figures

List of Tables

Preface

This guide describes the connector that is used to integrate Oracle Identity Manager with SCIM-based target systems.

Audience

This guide is intended for resource administrators and target system integration teams.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Related Documents

For information about installing and using Oracle Identity Manager, visit the following Oracle Help Center page:

http://docs.oracle.com/cd/E52734_01/index.html

For information about Oracle Identity Manager Connectors documentation, visit the following Oracle Help Center page:

http://docs.oracle.com/cd/E22999_01/index.htm

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

About the Generic SCIM Connector

The Generic SCIM connector integrates Oracle Identity Governance with SCIM-based target systems.

> **Note**
>
> In this guide, the connector that is deployed using the **Applications** option on the **Manage** tab of Identity Self Service is referred to as an **AOB application**. The connector that is deployed using the **Manage Connector** option in Oracle Identity System Administration is referred to as a **CI-based connector** (Connector Installer-based connector).

From Oracle Identity Governance release 12.2.1.3.0 onward, connector deployment is handled using the application onboarding capability of Oracle Identity Self Service. This capability lets business users to onboard applications with minimum details and effort. The connector installation package includes a collection of predefined templates (XML files) that contain all the information required for provisioning and reconciling data from a given application or target system. These templates also include basic connectivity and configuration details specific to your target system. The connector uses information from these predefined templates allowing you to onboard your applications quickly and easily using only a single and simplified UI.

Application onboarding is the process of registering or associating an application with Oracle Identity Governance and making that application available for provisioning and reconciliation of user information.

The following sections provide a high-level overview of the connector:

Introduction to the Connector

The Generic SCIM connector is a solution to integrate Oracle Identity Manager with SCIM-based identity-aware applications. A **SCIM-based identity-aware application** is any application that exposes its SCIM APIs or interfaces for identity management.

> **Note**
>
> A SCIM-based identity-aware application has been referred to as the **target system** or **SCIM-based target system.**

The Generic SCIM connector provides a centralized system to streamline delivery of services and assets to your company's consumers, and manage those services and assets in a simple, secure, and cost efficient manner by using automation. The Generic SCIM connector standardizes service processes and implements automation to replace manual tasks.

In order to connect with a SCIM-based target system, the Generic SCIM connector supports HTTP Basic Authentication and OAuth 2.0 authentication mechanisms. This connector also supports authenticating to the target system by using access token and refresh token as an input from the user. This authentication mechanism can be useful if your target system does not provide a programmatic approach to obtain access tokens.

The connector supports the following OAuth 2.0 grant types:

- JWT
- Client Credentials
- Resource Owner Password

If your target system does not support any of the authentication types supported by this connector, then you can implement the custom authentication that your target system supports. You can connect this custom implementation to the connector by using the plug-ins exposed by this connector.

The Generic SCIM connector synchronizes data between Oracle Identity Governance and SCIM-based target systems by performing reconciliation and provisioning operations that parse data in the JSON format. If your target system does not support request or response payload in JSON format, then you can create your own implementation for parsing data. You can connect this custom implementation to the connector by using the plug-ins exposed by this connector.

The Generic SCIM connector is a connector for a discovered target system. This is because the schema of the SCIM-based target system with which the connector integrates is not known in advance. The Generic SCIM connector is not shipped with any artifacts. So during application creation, you must specify the schema of your target system, and this helps the connector understand the schema of the SCIM-based target system and then generate the artifacts.

Certified Components

These are the software components, and their versions required for installing and using the connector.

Table 1-1 Certified Components

Item	Requirement for AOB Application	Requirement for CI-Based Connector
Oracle Identity Governance or Oracle Identity Manager	You can use one of the following releases of Oracle Identity Governance: • Oracle Identity Governance 12*c* PS4 (12.2.1.4.0) • Oracle Identity Governance 14*c* (14.1.2.1.0)	You can use one of the following releases of Oracle Identity Governance or Oracle Identity Manager: • Oracle Identity Governance 12*c* PS4 (12.2.1.4.0) • Oracle Identity Governance 14*c* (14.1.2.1.0)
Target System	Any identity-aware application that supports SCIM service	Any identity-aware application that supports SCIM service
Connector Server	• 12.2.1.3.1 or 12.2.1.3.0 **Note**: Connector server is optional, if you have deployed the Generic SCIM connector in the Connector Server, then you can download the necessary Java Connector Server 12.2.1.3.1 or 12.2.1.3.0 from the Oracle Technology Network web page.	• 12.2.1.3.1 or 12.2.1.3.0 **Note**: Connector server is optional, if you have deployed the Generic SCIM connector in the Connector Server, then you can download the necessary Java Connector Server 12.2.1.3.1 or 12.2.1.3.0 from the Oracle Technology Network web page.
Connector Server JDK	• For Connector Server 12.2.1.3.1, use JDK 17 or later • For Connector Server 12.2.1.3.0, use JDK 1.8 or later	• For Connector Server 12.2.1.3.1, use JDK 17 or later • For Connector Server 12.2.1.3.0, use JDK 1.8 or later

Certified Languages

The connector will support the languages that are supported by Oracle Identity Governance.

These are the languages that the connector supports.

- Arabic
- Chinese (Simplified)
- Chinese (Traditional)
- Czech
- Danish
- Dutch
- English (US)
- Finnish
- French
- French (Canadian)
- German
- Greek
- Hebrew

- Hungarian
- Italian
- Japanese
- Korean
- Norwegian
- Polish
- Portuguese
- Portuguese (Brazilian)
- Romanian
- Russian
- Slovak
- Spanish
- Swedish
- Thai
- Turkish

Resource bundles are not part of the connector installation package as the resource bundle entries vary depending on the target system being used.

Usage Recommendation

These are the recommendations for the Generic SCIM Connector versions that you can deploy and use depending on the Oracle Identity Governance version that you are using.

If you are using Oracle Identity Governance 12*c* (12.2.1.3.0) or Oracle Identity Governance 14*c* (14.1.2.1.0) or later, then use the latest 12.2.1.*x* version of this connector. Deploy the connector using the **Applications** option on the **Manage** tab of Identity Self Service.

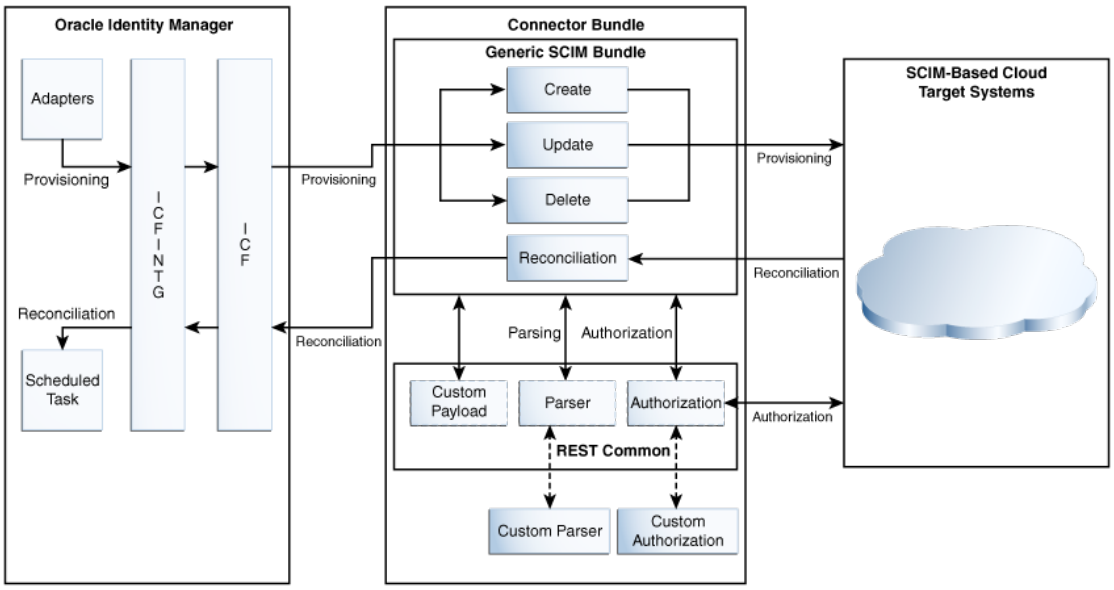
Architecture of Generic SCIM Connector

The Generic SCIM connector is implemented using the Identity Connector Framework (ICF).

The ICF is a component that provides basic reconciliation and provisioning operations that are common to all Oracle Identity Manager connectors. In addition, ICF provides common features that developers would otherwise need to implement on their own, such as connection pooling, buffering, time outs, and filtering. The ICF is shipped along with Oracle Identity Governance.

Below figure shows the architecture of the connector.

Figure 1-1 Connector Architecture



The primary function of the Generic SCIM connector is to connect to any target system that exposes its SCIM APIs and then synchronize user identity data between this target system and Oracle Identity Governance.

This connector is not shipped with any metadata as it is a connector for target system that is not known in advance. Depending on the schema of your target system, the connector artifacts are generated after you create the application for your target system. After the connector artifacts are created, Oracle Identity Governance communicates with your target system through the connector bundle by various provisioning and reconciliation operations.

The SCIM Common layer contains all the plug-ins and logic required by the connector to authenticate to the target system and parse data. Any custom implementation for authorization and data parsing can also be hooked as a plug-in in the SCIM Common layer.

During provisioning, adapters carry provisioning data submitted through the process form to the target system. The adapters establish a connection with the corresponding Create, Update, or Delete operations in the connector bundle which in turn establishes a connection with a target system by leveraging the SCIM Common layer. After the adapters establish a connection with the target system, SCIM calls are made to the endpoints, and the required provisioning operation is performed. Subsequently, the response from the target system is returned to the adapters.

During reconciliation, a schedule task is run which calls the SearchOp operation of the connector bundle. The connector bundle establishes a connection with the target system by using the SCIM Common layer. Then, the connector retrieves all records that match the reconciliation criteria by calling the specific SCIM endpoint. This result is then passed to Oracle Identity Governance.

Connector Features

The features of the connector include support for full and incremental reconciliation, limited reconciliation, custom authentication, custom parsing, custom payload, handling multiple endpoint URLs, and SSL communication.

The following are the features of the connector:

- Support for Both Trusted Source and Target Resource Reconciliation
- Full and Incremental Reconciliation
- Limited (Filtered) Reconciliation
- Custom Authentication
- Custom Parsing
- Custom Payload
- Support for Additional HTTP Headers
- Support for Handling Multiple Endpoint URLs
- SSL Communication

Trusted Source and Target Resource Reconciliation

You can configure your SCIM-based application as a Target application or an Authoritative application for reconciliation of records into Oracle Identity Governance.

There are two versions of the connectors available to provide support fortrusted source (authoritative application) and target resource (Target application) reconciliation.

See Configuring Reconciliation Jobs for more information.

Full and Incremental Reconciliation

After you create the application, you can perform full reconciliation to bring all existing user data from the target system to Oracle Identity Governance. After the first full reconciliation run, you can configure your connector for incremental reconciliation. In incremental reconciliation, only records that are added or modified after the last reconciliation run are fetched into Oracle Identity Governance.

> **Note**
>
> If the target system contains an attribute that holds the timestamp at which an object is created or modified, the connector supports incremental reconciliation.

You can perform a full reconciliation run at any time. See Performing Full Reconciliation and Incremental Reconciliation for more information.

Limited (Filtered) Reconciliation

You can set a reconciliation filter as the value of the Filter Suffix attribute of the scheduled jobs. This filter specifies the subset of newly added and modified target system records that must be reconciled.

See Performing Limited (Filtered) Reconciliation for more information about performing limited reconciliation.

Custom Authentication

By default, the Generic SCIM connector supports HTTP Basic Authentication and OAuth 2.0 authentication mechanisms. The connector also supports an authentication mechanism in which the user provides access token as an input. The supported grant types for OAuth 2.0 authentication mechanism are JWT, Client Credentials, and Resource Owner Password. If your target system uses any of the authentication mechanisms that is not supported by the connector, then you can write your own implementation for custom authentication by using the plug-ins exposed by this connector.

See Implementing Custom Authentication for more information about creating your own implementation for the custom authentication.

Custom Parsing

By default, the Generic SCIM connector supports request and response payloads only in the JSON format. If your target system does not support request or response payload in JSON format, then you can implement a custom parsing logic by using plug-ins exposed by this connector.

See Implementing Custom Parsing for more information about custom parsing.

Custom Payload

The Generic SCIM connector provides support for handling custom formats for any attributes in the payload that do not adhere to the standard JSON format.

This can be achieved by specifying a value for the `customPayload` parameter of Advanced Settings. See Advanced Settings Parameters for more information about this parameter.

Support for Additional HTTP Headers

If your target system requires additional or custom HTTP headers in any SCIM call, then you can insert these HTTP headers as the value of the customAuthHeaders configuration parameter.

See Authentication Parameters for more information about this parameter.

Support for Handling Multiple Endpoint URLs

The Generic SCIM connector allows you to handle attributes of an object class (for example, a User object class) that can be managed only through endpoints other than the base endpoint URL of the object class. For example, in certain target systems, there are attributes of the User object class that can be managed using the base endpoint URL. However, some attributes (for

example, email alias) can be managed only through a different endpoint URL. The connector provides support for handling all endpoint URLs associated with an object class.

This can be achieved by providing endpoint URL details of such attributes in the relURIs IT resource parameter. See Advanced Settings Parameters for more information about this parameter.

SSL Communication

You can configure SSL to secure data communication between Oracle Identity Governance and the SCIM-based target system.

See Configuring SSL for information about configuring secure communication.

Use Cases Supported by the Generic SCIM Connector

The Generic SCIM connector can be used to integrate OIM with any target system that supports SCIM services. This connector can be used to load identity data into OIM from a SCIM service and then efficiently manage identities in an integrated cycle with the rest of the identity-aware applications in your enterprise.

As a business use case example, consider a leading logistics company that has 20+ cloud applications. Most of these cloud applications are now inefficient because data in these applications are manually entered and are managed using spreadsheets or custom-coded process flows. Therefore, this company wants to integrate its cloud applications with Oracle Identity Manager to streamline its operations, increase its organizational efficiency, and at the same time, lower its operational costs. There are two approaches for integrating these cloud applications with Oracle Identity Manager. One approach would be to deploy a point-to-point connector for each of these applications. The drawbacks of this approach are as follows:

- Increased time and effort to identify and deploy a point-to-point connector for each application.
- Increased administration and maintenance overheads for managing connectors for each application.
- Unavailability of point-to-point connectors for all applications. In such a scenario, one needs to develop custom connectors which increases time and effort to develop, deploy and test the custom connector.

An alternative to this approach is to use the Generic SCIM connector that can be used to integrate all the cloud applications with Oracle Identity Manager . The Generic SCIM connector provides the ability to manage accounts across all cloud applications without spending additional resources and time on building custom connectors for each cloud application.

The Generic SCIM connector is a hybrid approach that helps enterprises leverage on-premise Oracle Identity Manager deployment to integrate with target systems for identity governance. These targets systems include any application that exposes SCIM APIs such as SaaS, PaaS, home-grown applications and so on.

The following are some example scenarios in which the Generic SCIM connector is used:

- **User Management**

  The Generic SCIM Connector manages individuals who can access Cloud service by defining them as users in the system and assigning them to groups. This connector allows new users to self-provision on a Generic SCIM Cloud Service, while having it be controlled by IT. Users can request and provision from a catalog of cloud-based resources that is established by Oracle Identity Manager administrators. For example, to create a new user

in the target system, fill in and submit the Oracle Identity Manager process form to trigger the provisioning operation. The connector executes the create operation against your target system and the user is created on successful execution of the operation. Similarly, operations such as delete and update can be performed.

- **Entitlement Management**

  The Generic SCIM Connector manages Cloud services objects (if exposed by the target system) as entitlements. Depending on the target system being used, this connector can be used to manage entitlements such as Groups, Roles, Licenses, Folders, Collaboration and so on. For example, you can use the Generic SCIM connector to automatically assign or revoke groups to users based on predefined access policies in Oracle Identity Manager . Similarly, you can use the Generic SCIM Connector to manage role memberships that provide selective access to certain Cloud Service functionality or groups. Therefore, as new users are added to a specific role, they automatically gain corresponding access in the applications.

2

Creating an Application By Using the Generic SCIM Connector

Learn about onboarding applications using the connector.

You can onboard an application for your target system into Oracle Identity Governance from the connector package by creating a Target application or an Authoritative application. To do so, you must log in to Identity Self Service and then choose the **Applications** box on the **Manage** tab.

Prerequisites for Creating an Application

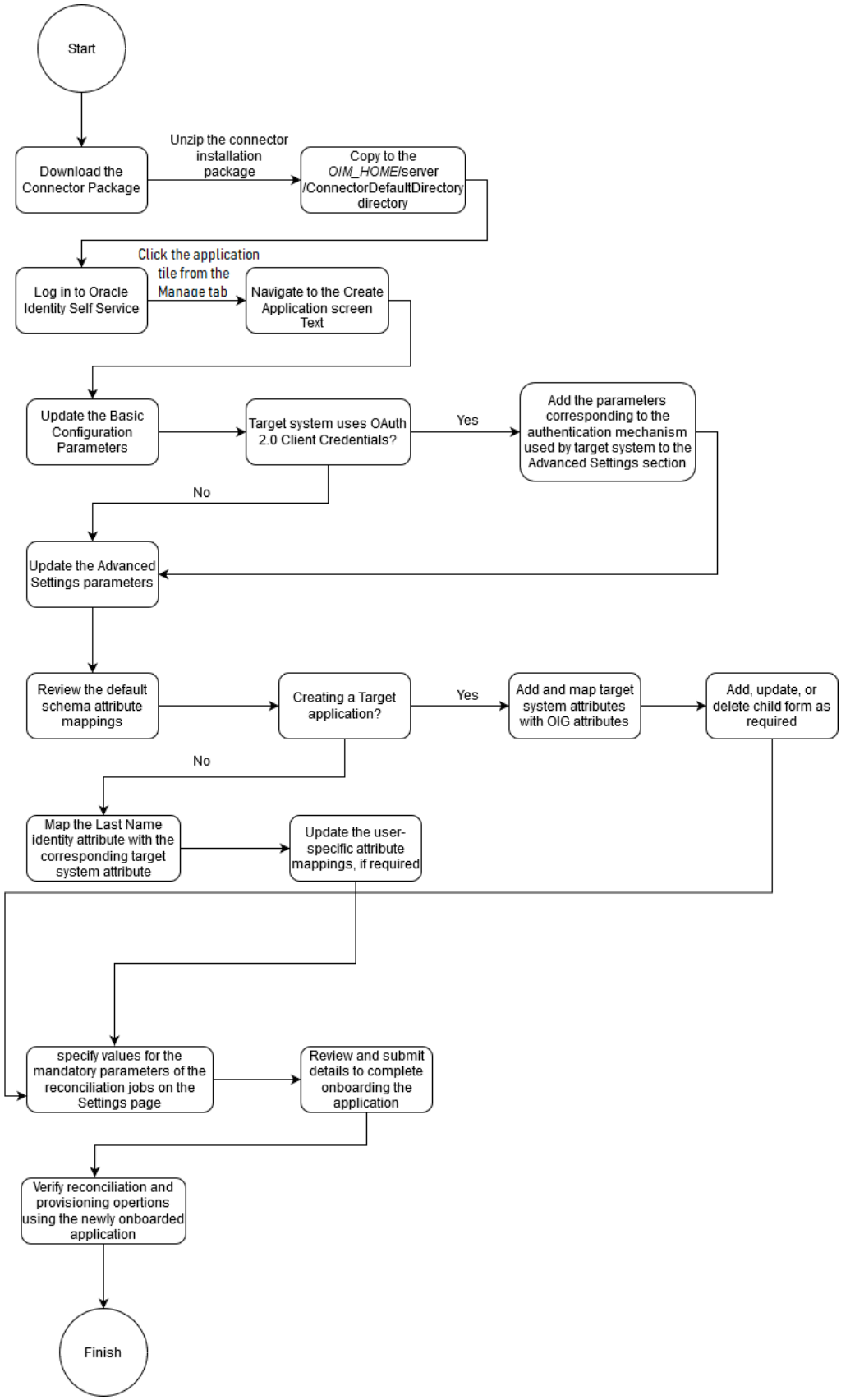
Learn about process flow for onboarding applications using the connector and the prerequisites for doing so.

Process Flow for Creating an Application By Using the Connector

From Oracle Identity Governance release 12.2.1.3.0 onward, connector deployment is handled using the application onboarding capability of Identity Self Service.

Below figure is a flowchart depicting high-level steps for creating an application in Oracle Identity Governance by using the connector installation package.

Figure 2-1 Overall Flow of the Process for Creating an Application By Using the Connector



Downloading the Connector Installation Package

You can obtain the installation package for your connector on the Oracle Technology Network (OTN) website.

To download the connector installation package:

1. Navigate to the OTN website at http://www.oracle.com/technetwork/middleware/id-mgmt/downloads/connectors-101674.html.
2. Click **OTN License Agreement** and read the license agreement.
3. Select the **Accept License Agreement** option.

   You must accept the license agreement before you can download the installation package.
4. Download and save the installation package to any directory on the computer hosting Oracle Identity Manager.
5. Extract the contents of the installation package to any directory on the computer hosting Oracle Identity Governance. This creates a directory named *CONNECTOR_NAME-RELEASE_NUMBER*.
6. Copy the CONNECTOR_NAME-RELEASE_NUMBER directory to the *OIG_HOME/server/ConnectorDefaultDirectory* directory.

Navigating to the Create Application Screen

To navigate to the Create Application screen, you must log in to Identity Self Service and then choose the **Applications** box on the **Manage** tab.

1. Log in to Identity Self Service either by using the **System Administration** account or an account with the **ApplicationInstanceAdministrator** admin role.
2. On the Applications page, click the **Create** menu on the toolbar, and then select one of the following options:
   - **Target** - to create a Target application.
   - **Authoritative** - to create an Authoritative application.

The Create Application screen with the Basic Information page is displayed.

Understanding and Updating Basic Information

Learn about the configuration-related details that you must enter on the Basic Information page. The connector uses these details to establish a connection with the target system and while performing reconciliation or provisioning operations.

Creating an application (whether Target or Authoritative) involves providing relevant details of the application such as connectivity information, schema details, and so on. On the Basic Information page, you provide various details of application that you wish to onboard. For example, you select the connector bundle, enter the name of the application to be created, and enter connection-related parameters that the connector uses to connect Oracle Identity Governance with your target system and perform connector operations.

This following topic discuss parameters on the various sections of the Basic Information page in detail and the process for specifying values for these parameters:

- Basic Configuration Parameters

Basic Configuration Parameters

These are the connection-related parameters that Oracle Identity Governance requires to connect to the SCIM-based target system.

Table 2-1 Basic Configuration Parameters

Parameter	Mandatory?	Description
grantType	Yes	Enter the type of authentication used by your target system. The following are the possible values for this parameter: • For HTTP Basic Authentication: `basic` • For OAuth 2.0 JWT: `jwt` • For OAuth 2.0 Client Credentials: `client_credentials` • For OAuth 2.0 Resource Owner Password: `password` • For manual input of access token and refresh token: `other` • For custom authentication implementation: `custom` **Default value**: `client_credentials` **Note:** If you are using a value other than the default value, client_credentials, then ensure to add the parameters corresponding to the authentication type that your target system is using.
authenticationServerUrl	No	Enter the URL of the authorization server that authenticates the client (by validating the client ID and client secret), and if valid, issues an access token. **Sample value**: `https://api.example.com/oauth2/token`
baseURI	No	This is a mandatory attribute while creating an application. Do not modify the value of the parameter. **Default value**: `/api/v1`
clientId	No	Enter the client identifier (a unique string) issued by the authorization server to the client during the registration process. **Sample value**: `XDWTh0r2eWuULCDVt`

Table 2-1 (Cont.) Basic Configuration Parameters

Parameter	Mandatory?	Description
clientSecret	No	Enter the value used to authenticate the identity of your client application. **Sample value**: `clZsdZisTOoYN5NITirarIDepDkiJTGHdzNFT0m`
acceptType	No	This is a mandatory attribute while creating an application. Do *not* modify the value of the parameter. **Default value**: `application/scim+json`
contentType	No	This entry holds the type of the body of the request. This is a mandatory attribute while creating an application. Do *not* modify the value of the parameter. **Default value**: `application/scim+json`
host	Yes	Host name or IP address of the computer hosting the target system. **Sample value**: `www.example.com`
username	Yes	Enter the username of the target system that you create for performing connector operations. **Sample value**: johnsmith
password		Enter the password of the target system user account that you create for connector operations. **Sample value**: password
Connector Server Name	No	If you have deployed the Generic SCIM connector in the Connector Server, then enter the name of the IT resource for the Connector Server.
proxyHost	No	Name of the proxy host used to connect to an external target system. Sample value: `www.example.com`
proxyPort	No	Proxy port number Sample value: `80`
proxyUser	No	Proxy user name of the target system user account that Oracle Identity Governance uses to connect to the target system.
proxyPassword	No	Password of the proxy user ID of the target system user account that Oracle Identity Governance uses to connect to the target system.

Table 2-1 (Cont.) Basic Configuration Parameters

Parameter	Mandatory?	Description
sslEnabled	No	If the target system requires SSL connectivity, then set the value of this parameter to `true`. Otherwise set the value to `false`. Default value: `true`

Advanced Settings Parameters

These are the configuration-related entries that the connector uses during reconciliation and provisioning operations.

Table 2-2 Advanced Settings Parameters

Parameter	Mandatory?	Description
Bundle Name	Yes	This parameter holds the name of the connector class. **Default value:** `org.identityconnectors.genericscim`
Bundle Version	Yes	This parameter holds the version of the connector bundle class. **Default value:** `12.3.0`
Connector Name	Yes	This parameter holds the name of the connector bundle package. **Default value:** `org.identityconnectors.genericscim.genericscimConnector`
defaultBatchSize		This entry holds the value of the number of records that can be retrieved from the target system in one go. **Default value**: `200`

Table 2-2 (Cont.) Advanced Settings Parameters

Parameter	Mandatory?	Description
customPayload		This entry holds the payloads for all operations that are not in the standard format. Enter a comma-separated list of request payload formats for target system attributes that do not adhere to the standard JSON format. Format: `OBJ_CLASS.ATTRNAME.OP=PAYLOAD_FORMAT` If you must pass the unique ID or Name attribute of the user as part of a custom payload, then represent it as $(__UID__)$ or $(__NAME__)$ respectively **Sample value**: `"__ACCOUNT__.__GROUP__.UPDATEOP={ \"user\": { \"id\": \"$(__UID__)$\"}, \"group\": { \"id\": \"$(id)$\" } }"`
nameAttributes	No	Enter the name attribute for all object classes that are handled by the connector. This value specifies the mapping between the _NAME_ connector attribute and the corresponding target system attribute for each object class that the connector handles. Format: *`OBJ_CLASS.ATTR_NAME`* Sample value: `"__ACCOUNT__.userName"` **Note:** All values in this parameter must be comma separated.
attrToOClassMapping		This is Attribute names to Other Object Class mapping. **Sample value**: `"__ACCOUNT__.groups=Groups"`

Table 2-2 (Cont.) Advanced Settings Parameters

Parameter	Mandatory?	Description

jsonResourcesTag	Yes, if you are implementing custom parsers	Enter the JSON tag value that is used for parsing a response payload. The connector will consider the value that you enter in this parameter as an unwanted outer tag while parsing responses. You can skip entering a value for this parameter if there is no unwanted outer tag in your response payload.
		Enter a value for this parameter in the following format:
		`OBJ_CLASS=OUTER_ATTR_NAME`
		In this format, `OBJ_CLASS` is the name of the object class for which a response payload is being parsed. `OUTER_ATTR_NAME` is the name of the outer tag in the response payload.
		For example, consider the following JSON value for a User object:
		`“Resources”:”{ “user”:” {value1}”, “user2”:” {value2}” }`
		Because the name of the object class for a User object is `__ACCOUNT__`, for the given example, the value of the jsonResourcesTag parameter is `__ACCOUNT__=Resources`.
		Note: You must enter a value for this parameter only if the data from your target system is in JSON format. For more than one JSON tag, the values must be comma separated.

scimVersion		This entry specifies the SCIM version. **Sample value**: 1
statusAttributes	No	Enter the name of the target system attribute that holds the status for each object class this connector handles. Format: `OBJ_CLASS.ATTR_NAME` **Sample value**: `"__ACCOUNT__.suspended"` **Note:** All values in this parameter must be comma separated.

Table 2-2 (Cont.) Advanced Settings Parameters

Parameter	Mandatory?	Description
uidAttributes	Yes	Enter the __UID__ attribute for each object class that the connector handles. A __UID__ attribute is a target system attribute that uniquely identifies an account in the target system. This target system attribute name must be unique and need not be autogenerated. Format: `OBJ_CLASS.ATTR_NAME` **Sample value**: `"__ACCOUNT__.id"` In this format, `OBJ_CLASS` is the connector object class and `ATTR_NAME` is the name of the attribute that uniquely identifies an account in the target system. **Note**: All values in this parameter must be comma separated.
passwordAttributes	No	Enter the name of the target system attribute that is mapped to the __PASSWORD__ attribute of the connector in OIM.

Authentication Parameters

Authentication parameters are used by the target system to authenticate an application. The set of parameters for which you must specify values depends on the value that you enter for the `grantType` parameter of the Basic Configuration section.

The `grantType` parameter holds the type of authentication used by your target system. The connector supports the following types of authentication:

- HTTP Basic Authentication
- OAuth 2.0 JWT
- OAuth 2.0 Client Credentials
- OAuth 2.0 Resource Owner Password
- Manually Input Access Token

If your target system uses an authentication type other than the ones listed above, then you must write your own implementation which requires development effort. By default, the UI includes parameters for the OAuth 2.0 Client Credentials authentication type. For any other authentication type, you must create and add the corresponding parameters in the Advanced Settings section.

The following are the possible values for the `grantType` parameter:

- For HTTP Basic Authentication: `basic`
- For OAuth 2.0 JWT: `jwt`
- For OAuth 2.0 Client Credentials: `client_credentials`
- For OAuth 2.0 Resource Owner Password: `password`

- For manual input of access token: `other`
- For custom authentication implementation: `custom`

> **Note**
>
> This section provides information about parameters for all authentication types. Enter values only for parameters corresponding to the authentication type you specify.

HTTP Basic Authentication

Table 2-3 lists the set of parameters for which you must enter values when the authenticationType parameter is set to `basic`.

Table 2-3 HTTP Basic Authentication IT Resource Parameters

Parameter	Description
username	Enter the user name or user ID of the account that Oracle Identity Governance must use to connect to and access the target system during reconciliation and provisioning operations. Sample value: `johnsmith`
password	Enter the password of the account that Oracle Identity Governance must use to connect to and access the target system during reconciliation and provisioning operations. Sample value: `password`

OAuth 2.0 JWT

Table 2-4 lists the set of parameters for which you must enter values when the grantType parameter is set to `jwt`.

Table 2-4 OAuth 2.0 JWT IT Resource Parameters

Parameter	Description
aud	Enter the intended audience of the JWT. The value can either be a URI or token endpoint URL of the authorization server. Sample value: `https://www.example.com/oauth2/v3/token`
iss	Enter a value that uniquely identifies the entity that issued the JWT. Sample value: `527901474-ugnvd5uh21p598cf9h6cd@developer.example.com`
scope	Enter the scope of the access token being issued. Sample value: `https://www.example.com/auth/adm.direct.group`, `https://www.example.com/auth/adm.direct.user`

Table 2-4 (Cont.) OAuth 2.0 JWT IT Resource Parameters

Parameter	Description
sub	Enter a value that identifies the principal to which the JWT is being issued. Sample value: `admin@example.com`
privateKeyLocation	Enter the absolute path to the private key used to sign the access token. Sample value: `C:\Users\jdoe\Desktop\Connector_Server_111210\connector_server_java-1.4.0\bundles\googleapps.p12`
privateKeySecret	Enter the secret key for the private key that is being used to sign the access token.
tokenLifespan	Enter the life span of the access token in milliseconds. Sample value: `3600`
signatureAlgorithm	Enter the algorithm used for signing the access token. Sample value: `RS265`
privateKeyFormat	Enter the format of the private key used to sign the access token. Sample value: `PKCS12`

OAuth 2.0 Client Credentials

Below table lists the set of parameters for which you must enter values when the grantType parameter is set to `client_credentials`.

> **Note**
>
> By default, these parameters are available in the Basic Configuration section. Therefore, there is no need to add them manually.

Table 2-5 OAuth2.0 Client Credentials IT Resource Parameters

Parameter	Description
clientId	Enter the client identifier (a unique string) issued by the authorization server to the client during the registration process. Sample value: `XDWTh0r2eWuULCDVt`
clientSecret	Enter the value used to authenticate the identity of your client application. Sample value: `clZsdZisTOoYN5NITirarIDepDkiJTGHdzNFT0m`

Table 2-5 (Cont.) OAuth2.0 Client Credentials IT Resource Parameters

Parameter	Description
authenticationServerURL	Enter the URL of the authorization server that authenticates the client (by validating the client ID and client secret), and if valid, issues an access token. Sample value: `https://api.example.com/oauth2/token`

OAuth 2.0 Resource Owner Password

Below table lists the set of IT resource parameters for which you must enter values when the grantType parameter is set `password`.

Table 2-6 OAuth 2.0 Resource Owner Password IT Resource Parameters

Parameter	Description
username	Enter the user name or user ID of the resource owner. Sample value: `johnsmith`
password	Enter the password of the resource owner. Sample value: `password`
clientId	Enter the client identifier issued to the client during the registration process. Sample value: `XDWTh0r2eWuULCDVt` **Note:** This is an optional parameter.
clientSecret	Enter the client secret used to authenticate the identity of the client application. Sample value: `clZsdZisTOoYN5NITirarIDepDkiJTGHdzNFT0m` **Note:** This is an optional parameter.
authenticationServerUrl	Enter the URL of the authorization server (token endpoint) that authenticates the client (by validating client ID and client secret) and the resource owner credentials, if valid, issues an access token. Sample value: `https://api.example.com/oauth2/token`

Manual Input of Access Tokens

This section discusses the parameter for which you must enter a value when the grantType parameter is set to `other`.

In this authentication mechanism, the connector expects the value of the access token to be directly passed through the `customAuthHeaders` parameter.

The `customAuthHeaders` parameter must hold the access token value that must be passed through an HTTP authorization header, for example,`access_token=<value>`.

Custom Authentication

This section discusses the parameter for which you must enter a value when the grantType parameter is set to `custom`.

If you have implemented custom authentication, then you must enter a value for the `customAuthClassName` parameter. The `customAuthClassName` parameter must hold the name of the class implementing the custom authentication logic that you created while performing the procedure described in Implementing Custom Authentication.

Providing Basic Information

You must provide configuration-related details on the Basic Information page. The connector uses these details while performing reconciliation.

On the Basic Information page, you provide the application details and configuration info that the connector uses during reconciliation. Depending on the authentication type that your target system uses, you may need to manually add the corresponding attributes for providing authentication details.

1. On the Basic Information page, ensure that the **Connector Package** option is selected.
2. From the **Select Bundle** drop-down list, select **Generic SCIM Connector 12.2.1.3.0**.
3. Enter the **Application Name**, **Display Name**, and **Description** for the application.
4. In the Basic Configuration section, enter values for parameters as required. If you set the value of the `grantType` parameter to a value other than the default value (`client_credentials`), then you must add and enter values for the corresponding authentication attributes in the Advanced Settings section. For example, if you set the value of the `grantType` parameter to `basic` (for basic HTTP authentication), then this type of authentication requires a username and password. As there are no such attributes available in the UI, you must manually add them as follows:
   a. In the Advanced Settings section, click **Add Attribute**.
   b. In the New Attribute dialog box, enter values for the **Name** and **Value** fields. Optionally, enter a value for the **Display Name** field.

      For example, in the **Name** field, enter `username` as the attribute name. In the **Value** field, enter `jdoe`.
   c. Click **OK**.

      The attribute is displayed under the Custom section.
   d. Repeat Steps 4.a through 4.c for each attribute that you need to add for your authentication type.
5. In the Advanced Settings section, enter values for the parameters as required.
6. Click **Next** to proceed to the Schema page.

> **See Also**
>
> Authentication Parameters for the list of supported authentication types and their corresponding parameters

Understanding and Updating the Schema Page

Use the Schema page to add or update attributes to help the connector understand the underlying structure of your target system.

Understanding the Schema Page for a Target Application

The Schema page for a Target application displays the default schema that maps Oracle Identity Governance attributes with corresponding target system attributes. The connector uses these mappings during reconciliation and provisioning operations.

The table under the SCIM User section displays a basic set of user-specific attribute mappings between Oracle Identity Governance and the target system. The table also lists the data type for a given attribute and specifies whether it is mandatory for reconciliation and whether it is a matching key field for fetching records during reconciliation. By default, the Schema page displays mappings for the __UID__, __NAME__, and __ENABLE__ connector attributes with the corresponding Oracle Identity Governance attributes. If required, you can edit the default attribute mappings.

The Schema page also provides a child form for the Group object class by default. Similar to the table in the SCIM User section, you can use this table to specify Group attribute mappings, data type for a given attribute, whether it is a key field for reconciliation, and so on.

As the connector is unaware of the target system schema, you must manually add and map your target system attributes with corresponding Oracle Identity Governance attributes. as described in Creating a Target Application of *Oracle Fusion Middleware Performing Self Service Tasks with Oracle Identity Governance*.

Understanding the Schema Page for an Authoritative Application

The Schema page for an Authoritative application displays the default schema that maps Oracle Identity Governance attributes with corresponding target system attributes. The connector uses these mappings during reconciliation runs.

The table under the SCIM Trusted section displays a basic set of user-specific attribute mappings between Oracle Identity Governance and the target system. The table also lists the data type for a given attribute and specifies whether it is mandatory for reconciliation and whether it is a matching key field for fetching records during reconciliation. By default, the Schema page displays mappings for the __UID__, __NAME__, and __ENABLE__ connector attributes with the corresponding Oracle Identity Governance attributes. If required, you can edit the default attribute mappings.

The schema page lists the default set of user-specific attribute mappings between the reconciliation fields in Oracle Identity Governance and target system attributes. The table also lists the data type for a given attribute and specifies whether it is a mandatory attribute for reconciliation.

If required, you can edit these attributes mappings by adding new attributes or deleting existing attributes on the Schema page as described in Creating an Authoritative Application in *Oracle Fusion Middleware Performing Self Service Tasks with Oracle Identity Governance*.

The Organization Name, Xellerate Type, and Role identity attributes are mandatory fields on the OIG User form that cannot be left blank during reconciliation. As your target system might not contain corresponding attributes for the Organization Name, Xellerate Type, and Role identity attributes, they have been mapped to attributes in Oracle Identity Governance. In addition, the connector provides default values (as listed in the “Default Value for Identity Display Name” column of below table) that it can use during reconciliation. For example, the default target attribute value for the Organization Name attribute is Xellerate Users. This implies that the connector reconciles all target system user accounts into the Xellerate Users organization in Oracle Identity Governance. Similarly, the default attribute value for Xellerate Type attribute is End-User, which implies that all reconciled user records are marked as end users.

Table 2-7 SCIM Trusted User Account Schema Attributes for an Authoritative Application

Identity Display Name	Target Attribute	Data Type	Mandatory Reconciliation Property?	Recon Field?	Default Value for Identity Display Name
User Login	__NAME__	String	No	Yes	NA
SCIM User GUID	__UID__	String	No	Yes	NA
usr_locale	preferredLanguage	String	No	Yes	NA
Display Name			No	Yes	NA
Status	__ENABLE__	String	No	Yes	NA
Xellerate Type		String	No	Yes	End-User
Organization Name		String	No	Yes	Xellerate Users
Role		String	No	Yes	Full-Time

Updating the Schema Page

Update the Schema page to review and update the attribute mappings between Oracle Identity Governance and target system.

To update the schema:

1. On the Schema page, review the default mapping provided by the connector.
2. If you are creating a Target application, then:
   a. In the SCIM User section, click **Add Attribute** to add a new row and enter a target system user-specific attribute and its mapping with the corresponding Oracle Identity Governance attribute. Then, select the checkboxes for the **Mandatory**, **Provision Field**, **Recon Field**, **Key Field**, and **Case Insensitive** fields to specify the provisioning and reconciliation properties for the newly added user-specific attribute as required.

      For example, if you are using Eloqua as the target system, and you want to add the emailAddress attribute, then click **Add Attribute**. In the newly added row, enter `Email` and `emailAddress` in the **Display Name** and **Target Attribute** columns, respectively. Then select the Provision Field, Recon Field, and Key Field checkboxes.

   b. Repeat Step 2.a for add all the user-specific attribute mappings.
   c. Expand the Groups section, and review the default child form details.

d. If your target system uses Groups as a child form, then update the table to match the attributes in your target system. Otherwise, delete the child form by clicking **Delete Form**.

e. (Optional) Click **Add Child Form** and enter the required details to add any other child forms that your target system uses. For example, if you are using Eloqua as the target system, then add a child form for Licenses.

3. If you are creating an Authoritative application, then for the Last Name attribute in the Identity Display Name column, enter the corresponding target system attribute name in the Target Attribute column. Then, specify the reconciliation properties in columns such as **Mandatory**, **Recon Field**, and so on. Repeat this for any other attributes that you need to update. If required, add new attributes by clicking **Add Attribute**.

4. Click **Next** to proceed to the Settings page.

Updating Settings

Review the default provisioning and reconciliation settings for the application being created, and customize it if required.

- Correlation Rules, Responses, and Situations
- Reconciliation Jobs

Correlation Rules, Responses, and Situations

Learn about the predefined rules, responses and situations for Target and Authoritative applications. The connector uses these rules and responses for performing reconciliation.

- Correlation Rules, Responses, and Situations for a Target Application
- Correlation Rules, Responses, and Situations for an Authoritative Application

Correlation Rules, Responses, and Situations for a Target Application

When you create a Target application, the connector uses correlation rules to determine the identity to which Oracle Identity Governance must assign a resource.

Predefined Identity Correlation Rules

By default, the Generic SCIM connector provides a simple correlation rule when you create a Target application. The connector uses this correlation rule to compare the entries in Oracle Identity Governance repository and the target system repository, determine the difference between the two repositories, and apply the latest changes to Oracle Identity Governance.

Below table lists the default simple correlation rule for a SCIM-based target system. If required, you can edit the default correlation rule or add new rules. You can create complex correlation rules also. For more information about adding or editing simple or complex correlation rules, see Updating Identity Correlation Rule in *Oracle Fusion Middleware Performing Self Service Tasks with Oracle Identity Governance*.

Table 2-8 Predefined Identity Correlation Rule for a Target Application

Target Attribute	Element Operator	Identity Attribute	Case Sensitive?
__NAME__	Equals	User Login	No

In this identity rule:

- __NAME__ is a single-valued attribute on the target system that identifies the user account.
- User Login is the field on the OIM User form.

Predefined Situations and Responses

The connector provides a default set of situations and responses when you create a Target application. These situations and responses specify the action that Oracle Identity Governance must take based on the result of a reconciliation event.

Below table lists the default situations and responses for this connector. If required, you can edit these default situations and responses or add new ones. For more information about adding or editing situations and responses, see Updating Situations and Responses in *Oracle Fusion Middleware Performing Self Service Tasks with Oracle Identity Governance*.

Table 2-9 Predefined Situations and Responses for a Target Application

Situation	Response
No Matches Found	None
One Entity Match Found	Establish Link
One Process Match Found	Establish Link

Correlation Rules, Responses, and Situations for an Authoritative Application

When you create an Authoritative application, the connector uses correlation rules to determine the identity that must be reconciled into Oracle Identity Governance.

Predefined Identity Correlation Rules

By default, the Generic SCIM connector provides a simple correlation rule when you create an Authoritative application. The connector uses this correlation rule to compare the entries in Oracle Identity Governance repository and the target system repository, determine the difference between the two repositories, and apply the latest changes to Oracle Identity Governance.

Below table lists the default simple correlation rule for an authoritative application. If required, you can edit the default correlation rule or add new rules. You can create complex correlation rules also. For more information about adding or editing simple or complex correlation rules, see Updating Identity Correlation Rule in *Oracle Fusion Middleware Performing Self Service Tasks with Oracle Identity Governance*.

Table 2-10 Predefined Identity Correlation Rule for an Authoritative Application

Target Attribute	Element Operator	Identity Attribute	Case Sensitive?
__UID__	Equals	SCIM User GUID	No
__NAME__	Equals	User Login	No

The identity correlation rule for an Authoritative application is as follows:

(__UID__ Equals SCIM User GUID) OR (__NAME__ Equals User Login)

In the first identity rule component:

- __UID__ is an attribute on the target system that uniquely identifies the user account.
- SCIM User GUID is the unique identifier of the resource assigned to the OIG User.

In the second identity rule component:

- __NAME__ is a single-valued attribute on the target system that identifies the user account.
- User Login is the field on the OIG User form.

Both the rule components are joined using the OR logical operator.

Predefined Situations and Responses

The connector provides a default set of situations and responses when you create an Authoritative application. These situations and responses specify the action that Oracle Identity Governance must take based on the result of a reconciliation event.

Below table lists the default situations and responses for an authoritative application. If required, you can edit these default situations and responses or add new ones. For more information about adding or editing situations and responses, see Updating Situations and Responses in *Oracle Fusion Middleware Performing Self Service Tasks with Oracle Identity Governance*.

Table 2-11 Predefined Situations and Responses for an Authoritative Application

Situation	Response
No Matches Found	Create User
One Entity Match Found	Establish Link
One Process Match Found	Establish Link

Reconciliation Jobs

Apart from reviewing the provisioning, reconciliation, and organization settings for your application and customizing them if required, you must specify values for the mandatory parameters of the reconciliation jobs.

The Settings page provides a preview of all settings related to provisioning, reconciliation, and organizations. You can review these settings and customize them if required. On the Reconciliation tab of the Settings page, expand the **Reconciliation Jobs** section to view the reconciliation jobs that the connector automatically creates after you create a Target or an Authoritative application. At this point, you can delete any reconciliation job that you do not want to use. If required, you can also edit the reconciliation jobs or create custom reconciliation jobs to meet your requirements. For information about editing these predefined jobs or creating new ones, see Updating Reconciliation Jobs in *Oracle Fusion Middleware Performing Self Service Tasks with Oracle Identity Governance*.

> **Note**
>
> Ensure that you enter values for the mandatory parameters (marked by the asterisk (*) symbol) of all the reconciliation jobs and then click **Apply**.

By default, jobs for user, delete user, and entitlement reconciliation are available for use after you create your application.

User Reconciliation Jobs

The following reconciliation jobs are available for reconciling user data:

- SCIM Application Target User Reconciliation: Use this reconciliation job to reconcile user data from a Target application.
- SCIM Application Trusted User Reconciliation: Use this reconciliation job to reconcile user data from an Authoritative application.

The parameters for both these jobs are the same.

Table 2-12 Parameters of the User Reconciliation Jobs

Parameter	Description
Application Name	Name of the application you created for your target system. This value is the same as the value that you provided for the Application Name field while creating your application. Do *not* modify this value.
Filter	Enter the expression for filtering records that the scheduled job must reconcile. Sample value: `search=loginName=<uid_name>`
Incremental Recon Attribute	Name of the target system attribute that holds the timestamp at which the user record was modified.
Object Type	Type of object you want to reconcile. Default value: `User`
Latest Token	The parameter holds the value of the target system attribute that is specified as the value of the Incremental Recon Attribute parameter. The Latest Token parameter is used for internal purposes. By default, this value is empty. **Note:** Do not enter a value for this attribute. The reconciliation engine automatically enters a value in this attribute.
Scheduled Task Name	Name of the scheduled job. **Note:** For the scheduled job included with this connector, you must not change the value of this attribute. However, if you create a new job or create a copy of the job, then enter the unique name for that scheduled job as the value of this attribute.

Delete User Reconciliation Jobs

The following reconciliation jobs are available for reconciling data about deleted user accounts:

- SCIM Application Target User Delete Reconciliation: Use this reconciliation job to reconcile data about deleted user accounts from a Target application.
- SCIM Application Trusted User Delete Reconciliation: Use this reconciliation job to reconcile data about deleted user accounts from an Authoritative application.

The parameters for both these jobs are the same.

Table 2-13 Parameters of the Delete User Reconciliation Jobs

Parameter	Description
Application Name	Name of the application you created for your target system. This value is the same as the value that you provided for the Application Name field while creating your target application. Do *not* modify this value.
Object Type	Type of object you want to reconcile. Default value: `User`
Scheduled Task Name	Name of the scheduled job. **Note:** For the scheduled job included with this connector, you must not change the value of this attribute. However, if you create a new job or create a copy of the job, then enter the unique name for that scheduled job as the value of this attribute.

Reconciliation Jobs for Entitlements

Depending on the child forms (of Lookup type) that you use or added on the Schema page, the corresponding reconciliation jobs for entitlements are displayed. For example, if you use the Groups child form on the Schema page, then the SCIM Application Group Lookup Reconciliation job is available for reconciling entitlements.

These reconciliation jobs are available only for a Target application. The parameters for all such reconciliation jobs are the same.

Table 2-14 Parameters of the Reconciliation Jobs for Entitlements

Parameter	Description
Application Name	Name of the application you created for your target system. This value is the same as the value that you provided for the Application Name field while creating your target application. Do not modify this value.
Lookup Name	This parameter holds the name of the lookup definition that maps each lookup definition with the data source from which values must be fetched. Depending on the reconciliation job you are using, the default values is as follows: `Lookup.GenericSCIM.`*`ENTITLEMENT_NAME`* For example, for the Groups entitlement job, the default value is `Lookup.GenericSCIM.Groups`.
Object Type	Enter the type of object whose values must be synchronized. For example, for reconciling Groups entitlement, enter `__GROUPS__`. **Note:** Do not change the value of this attribute.
Code Key Attribute	Enter the name of the connector or target system attribute that is used to populate the Code Key column of the lookup definition (specified as the value of the Lookup Name attribute). Default value: `__NAME__` **Note:** Do not change the value of this attribute.

Table 2-14 (Cont.) Parameters of the Reconciliation Jobs for Entitlements

Parameter	Description
Decode Attribute	Enter the name of the connector or target system attribute that is used to populate the Decode column of the lookup definition (specified as the value of the Lookup Name attribute). Default value: __NAME__

3

Performing Postconfiguration Tasks for the Generic SCIM Connector

These are the tasks that you can perform after creating an application in Oracle Identity Governance.

- Creating a UI Form for an Application
- Harvesting Entitlements and Sync Catalog
- Managing Logging for the Generic SCIM Connector
- Localizing Field Labels in UI Forms
- Configuring SSL

Creating a UI Form for an Application

During application creation, if you did not choose to create a default form, then you must create a UI form for the application that you created by using the connector.

> **Note**
>
> Perform the procedures described in this section only if you did not choose to create the default form during creating the application.

The following topics describe the procedures to create a UI form an application:

- Creating and Activating a Sandbox
- Creating a New UI Form
- Associating the Form with the Application Instance
- Publishing a Sandbox
- Updating an Existing Application Instance with a New Form

Creating and Activating a Sandbox

You must create and activate a sandbox to begin using the customization and form management features.

Creating a Sandbox and Activating a Sandbox in *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Governance*

Creating a New UI Form

You can use Form Designer in Oracle Identity System Administration to create and manage application instance forms.

See Creating Forms By Using the Form Designer in *Oracle Fusion Middleware Administering Oracle Identity Governance*.

While creating the UI form, ensure that you select the resource object corresponding to the newly created application that you want to associate the form with. In addition, select the **Generate Entitlement Forms** check box.

Associating the Form with the Application Instance

By default, an application instance is automatically created after you install the connector.

You must associate this application instance with the form created in Creating a New UI Form.

See Managing Application Instances in *Oracle Fusion Middleware Administering Oracle Identity Governance* for instructions on modifying an application instance to associate it with a form.

Publishing a Sandbox

Before you publish a sandbox, perform the following procedure as a best practice to validate all sandbox changes made till this stage as it is hard to revert changes once a sandbox is published:

1. In the System Administration, deactivate the sandbox.
2. Log out of the System Administration.
3. Log in to the Self Service using the xelsysadm user credentials and then activate the sandbox that you deactivated in Step 1.
4. In the Catalog, ensure that the Generic SCIM application instance form appears with correct fields.
5. Publish the sandbox.

See Publishing a Sandbox in *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Governance*.

Updating an Existing Application Instance with a New Form

For any changes that you do in the schema of your application in Identity Self Service, you must create a new UI form and update the changes in an application instance.

To update an existing application instance with a new form:

1. Create and activate a sandbox.
2. Create a new UI form for the resource.
3. Open the existing application instance.
4. In the Form field, select the new UI form that you created.
5. Save the application instance.
6. Publish the sandbox.

See Also

- Creating a Sandbox and Activating a Sandbox in *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Governance*
- Creating Forms By Using the Form Designer in *Oracle Fusion Middleware Administering Oracle Identity Governance*
- Publishing a Sandbox in *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Governance*

Harvesting Entitlements and Sync Catalog

You can populate Entitlement schema from child process form table, and harvest roles, application instances, and entitlements into catalog. You can also load catalog metadata.

To harvest entitlements and sync catalog:

1. Run the reconciliation jobs for entitlements discussed in Reconciliation Jobs.
2. Run the Entitlement List scheduled job to populate Entitlement Assignment schema from child process form table.
3. Run the Catalog Synchronization Job scheduled job.

See Also

Predefined Scheduled Tasks in *Oracle Fusion Middleware Administering Oracle Identity Manager*

Managing Logging for the Generic SCIM Connector

Oracle Identity Governance uses the Oracle Diagnostic Logging (ODL) logging service for recording all types of events pertaining to the connector.

The following topics provide detailed information about logging:

- Understanding Logging on the Connector Server
- Enabling Logging for the Connector Server
- Understanding Log Levels
- Enabling Logging

Understanding Logging on the Connector Server

When you enable logging, the connector server stores in a log file information about events that occur during the course of provisioning and reconciliation operations for different statuses. By default, the connector server logs are set at INFO level, and you can change this level to any one of these.

- Error

This level enables logging of information about errors that might allow connector server to continue running.

- WARNING

This level enables the logging of information about potentially harmful situations.

- INFO

This level enables logging of messages that highlight the progress of the operation.

- FINE, FINER, FINEST

These levels enable logging of information about fine-grained events, where FINEST logs information about all events.

Enabling Logging for the Connector Server

Edit the logging properties file located in the `CONNECTOR_SERVER_HOME/Conf` directory to enable logging.

To do so:

1. Navigate to the `CONNECTOR_SERVER_HOME/Conf` directory.
2. Open the logging.properties file in a text editor.
3. Edit the following entry by replacing INFO with the required level of logging:
   `.level=INFO`
4. Save and close the file.
5. Restart the connector server.

Understanding Log Levels

When you enable logging, Oracle Identity Manager automatically stores in a log file information about events that occur during the course of provisioning and reconciliation operations.

ODL is the principal logging service used by Oracle Identity Governance and is based on java.util.logger. To specify the type of event for which you want logging to take place, you can set the log level to one of the following:

- SEVERE.intValue()+100

  This level enables logging of information about fatal errors.

- SEVERE

  This level enables logging of information about errors that might allow Oracle Identity Manager to continue running.

- WARNING

  This level enables logging of information about potentially harmful situations.

- INFO

  This level enables logging of messages that highlight the progress of the application.

- CONFIG

  This level enables logging of information about fine-grained events that are useful for debugging.

- FINE, FINER, FINEST

  These levels enable logging of information about fine-grained events, where FINEST logs information about all events.

These message types are mapped to ODL message type and level combinations as shown in Table 3-1.

Table 3-1 Log Levels and ODL Message Type:Level Combinations

Java Level	ODL Message Type:Level
SEVERE.intValue()+100	INCIDENT_ERROR:1
SEVERE	ERROR:1
WARNING	WARNING:1
INFO	NOTIFICATION:1
CONFIG	NOTIFICATION:16
FINE	TRACE:1
FINER	TRACE:16
FINEST	TRACE:32

The configuration file for OJDL is logging.xml, which is located at the following path:

DOMAIN_HOME/config/fmwconfig/servers/*OIM_SERVER*/logging.xml

Here, *DOMAIN_HOME* and *OIM_SERVER* are the domain name and server name specified during the installation of Oracle Identity Governance.

Enabling Logging

To enable logging in Oracle WebLogic Server:

1. Edit the logging.xml file as follows:
   a. Add the following blocks in the file:

```
<log_handler name='genericSCIM-handler' level='[LOG_LEVEL]'
class='oracle.core.ojdl.logging.ODLHandlerFactory'>
<property name='logreader:' value='off'/>
<property name='path' value='[FILE_NAME]'/>
<property name='format' value='ODL-Text'/>
<property name='useThreadName' value='true'/>
<property name='locale' value='en'/>
<property name='maxFileSize' value='5242880'/>
<property name='maxLogSize' value='52428800'/>
<property name='encoding' value='UTF-8'/>
</log_handler>


<logger name="ORG.IDENTITYCONNECTORS.GENERICSCIM"
level="[LOG_LEVEL]"useParentHandlers="false">
<handler name="genericSCIM-handler"/><handler name="console-handler"/>
</logger>
<logger
name="ORG.IDENTITYCONNECTORS.SCIMCOMMON.UTILS.SCIMCOMMONUTILS"level="[LOG_LEVEL]"
 useParentHandlers="false">
<handler name="genericSCIM-handler"/><handler name="console-handler"/>
</logger>
```

   b. Replace all occurrences of **[LOG_LEVEL]** with the ODL message type and level combination that you require. Table 3-1 lists the supported message type and level combinations.

Similarly, replace **[FILE_NAME]** with the full path and name of the log file in which you want log messages specific to connector operations to be recorded.

The following blocks show sample values for **[LOG_LEVEL]** and **[FILE_NAME]** :

```
<log_handler name='genericSCIM-handler' level='NOTIFICATION:1'
class='oracle.core.ojdl.logging.ODLHandlerFactory'>
<property name='logreader:' value='off'/>
<property name='path' value='/<%OIM_DOMAIN%>/servers/oim_server1/ logs/
genericSCIMLogs.log>"
<property name='format' value='ODL-Text'/>
<property name='useThreadName' value='true'/>
<property name='locale' value='en'/>
<property name='maxFileSize' value='5242880'/>
<property name='maxLogSize' value='52428800'/>
<property name='encoding' value='UTF-8'/>
</log_handler>

<logger name="ORG.IDENTITYCONNECTORS.GENERICSCIM" level="NOTIFICATION:1"
useParentHandlers="false">
<handler name="genericSCIM-handler"/>
<handler name="console-handler"/>
</logger>
<logger name="ORG.IDENTITYCONNECTORS.SCIMCOMMON.UTILS.SCIMCOMMONUTILS"
level="NOTIFICATION:1" useParentHandlers="false">
<handler name="genericSCIM-handler"/>
<handler name="console-handler"/>
</logger>
```

With these sample values, when you use Oracle Identity Manager, all messages generated for this connector that are of a log level equal to or higher than the `NOTIFICATION:1` level are recorded in the specified file.

2. Save and close the file.
3. Set the following environment variable to redirect the server logs to a file:

   For Microsoft Windows:

   ```
   set WLS_REDIRECT_LOG=FILENAME
   ```

   For UNIX:

   ```
   export WLS_REDIRECT_LOG=FILENAME
   ```

   Replace ***FILENAME*** with the location and name of the file to which you want to redirect the output.
4. Restart the application server.

Localizing Field Labels in UI Forms

You can localize UI form field labels by creating and using a file containing localized versions for your target system fields.

To localize field label that you add to in UI forms:

1. Create a properties file (for example, GR_ja.properties) containing localized versions for the column names in your target system (to be displayed as text strings for GUI elements and messages in Identity System Administration and Identity Self Service).
2. Log in to Oracle Enterprise Manager.

3. In the left pane, expand **Application Deployments** and then select **oracle.iam.console.identity.sysadmin.ear.**
4. In the right pane, from the Application Deployment list, select **MDS Configuration.**
5. On the MDS Configuration page, click **Export** and save the archive to the local computer.
6. Extract the contents of the archive, and open the following file in a text editor:

   SAVED_LOCATION\xliffBundles\oracle\iam\ui\runtime\BizEditorBundle_en.x
7. Edit the BizEditorBundle.xlf file in the following manner:
   a. Search for the following text:

      ```
      <file source-language="en"
      original="/xliffBundles/oracle/iam/ui/runtime/BizEditorBundle.xlf"
      datatype="x-oracle-adf">
      ```

   b. Replace with the following text:

      ```
      <file source-language="en" target-language="LANG_CODE"
      original="/xliffBundles/oracle/iam/ui/runtime/BizEditorBundle.xlf"
      datatype="x-oracle-adf">
      ```

      In this text, replace *LANG_CODE* with the code of the language that you want to localize the form field labels. The following is a sample value for localizing the form field labels in Japanese:

      ```
      <file source-language="en" target-language="ja"
      original="/xliffBundles/oracle/iam/ui/runtime/BizEditorBundle.xlf"
      datatype="x-oracle-adf">
      ```

   c. Search for the application instance code. This procedure shows a sample edit for Generic SCIM application instance. The original code is:

      ```
      <trans-unit id="$
      {adfBundle['oracle.adf.businesseditor.model.util.BaseRuntimeResou
      rceBundle']
      ['persdef.sessiondef.oracle.iam.ui.runtime.form.model.user.entity
      .userEO.UD_GENERIC_NAME_GIVEN_NAME  c_description']}">
      <source>Name Givenname</source>
      <target/>
      </trans-unit>
      <trans-unit
      id="sessiondef.oracle.iam.ui.runtime.form.model.GRGAForm1.entity.
      GRGAForm1EO.UD_GENERIC_NAME_GIVEN_NAME c_LABEL">
      <source>Name Givenname</source>
      <target/>
      ```

   d. Open the properties file created in Step 1 and get the value of the attribute, for example, global.udf.UD_GENERIC_NAME_GIVEN_NAME = \u4567d.
   e. Replace the original code shown in Step c with the following:

      ```
      <trans-unit id="$
      {adfBundle['oracle.adf.businesseditor.model.util.BaseRuntimeResou
      rceBundle']
      ```

```
['persdef.sessiondef.oracle.iam.ui.runtime.form.model.user.entity
.userEO.UD_GENERIC_NAME_GIVEN_NAME  c_description']}">
<source>Name Givenname</source>
<target>\u4567d</target>
</trans-unit>
<trans-unit
id="sessiondef.oracle.iam.ui.runtime.form.model.GRGAForm1.entity.
GRGAForm1EO.UD_GENERIC_NAME_GIVEN_NAME c_LABEL">
<source>Name Givenname</source>
<target>\u4567d</target>
```

f. Repeat Steps 7.a through 7.d for all attributes of the process form.

g. Save the file as BizEditorBundle_*LANG_CODE*.xlf. In this file name, replace *LANG_CODE* with the code of the language to which you are localizing. Sample file name: BizEditorBundle_ja.xlf.

8. Repackage the ZIP file and import it into MDS.

> **See Also**
>
> Deploying and Undeploying Customizations in *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Governance* for more information about exporting and importing metadata files

9. Log out of and log in to Oracle Identity Governance.

Configuring SSL

You must configure SSL to secure data communication between Oracle Identity Governance and your target system.

To configure SSL:

1. Obtain the SSL public key certificate for the SCIM-based target system.
2. Copy the public key certificate of the SCIM-based target system to the computer hosting Oracle Identity Governance.
3. Run the following `keytool` command to import the target system certificate into the Oracle WebLogic Server keystore:

   ```
   keytool -import -keystore KEYSTORE_NAME -storepass PASSWORD -file
   CERT_FILE_NAME -alias ALIAS
   ```

   In this command:

   - *KEYSTORE_NAME* is the full path and name of the DemoTrust keystore.
   - *PASSWORD* is the password of the keystore.
   - *CERT_FILE_NAME* is the full path and name of the certificate file.
   - *ALIAS* is the target system certificate alias.

   The following is a sample value for this command:

   ```
   keytool -import -keystore WEBLOGIC_HOME/server/lib/DemoTrust.jks -storepass
   DemoTrustKeyStorePassPhrase –file /home/target.cert –alias serverwl
   ```

Note

- Change the parameter values passed to the `keytool` command according to your requirements. Ensure that there is no line break in the keytool arguments.
- Ensure that the system date for Oracle Identity Governance is in sync with the validity date of the SSL certificate to avoid any errors during SSL communication.

4

Using the Generic SCIM Connector

You can use the connector for performing reconciliation and provisioning operations after configuring it to meet your requirements.

The following topics discuss information related to using the connector for performing reconciliation and provisioning operations:

- Configuring Reconciliation
- Configuring Reconciliation Jobs
- Performing Provisioning Operations

Configuring Reconciliation

You can configure the connector to specify the type of reconciliation and its schedule.

This section discusses the following topics related to configuring reconciliation:

- Performing Full Reconciliation and Incremental Reconciliation
- Performing Limited (Filtered) Reconciliation

Performing Full Reconciliation and Incremental Reconciliation

Full reconciliation involves reconciling all existing user records from the target system into Oracle Identity Governance.

In **incremental reconciliation**, only records created or modified after the latest date or timestamp the last reconciliation was run are considered for reconciliation.

After you deploy the connector, you must first perform full reconciliation.

You can perform a full reconciliation run by removing or deleting any value currently assigned to the Filter attribute and then run the scheduled job for user data reconciliation. See Reconciliation Jobs for more information about the user reconciliation scheduled job and Filter attribute. In this scheduled job, you can include the timestamp attributes available in the Incremental Recon Attribute field.

At any given point in time, youcan switch from incremental reconciliation to full reconciliation. All you need to do is perform a full reconciliation run.

To perform incremental reconciliation, you must update and run the scheduled job for user data reconciliation to include the following attributes:

- Incremental Recon Attribute — Name of the target system attribute that holds the time stamp at which the record was last modified. The value in this attribute is used to determine the newest or latest record reconciled from the target system.
- Latest Token — Holds the value of the attribute that is specified as the value of the Incremental Recon Attribute attribute. The Latest Token attribute is used for internal purposes. Do *not* enter a value for this attribute. The reconciliation engine automatically enters a value in this attribute. Sample value: `1354753427000`

Performing Limited (Filtered) Reconciliation

Limited or filtered reconciliation is the process of limiting the number of records being reconciled based on a set filter criteria.

By default, all target system records that are added or modified after the last reconciliation run are reconciled during the current reconciliation run. You can customize this process by specifying the subset of added or modified target system records that must be reconciled. You do this by creating filters for the reconciliation module.

You can perform limited reconciliation by creating filters that your target system supports. This connector provides the Filter attribute (scheduled task attributes) that allows you to use any of the attributes of the target system to filter target system records.

For detailed information about ICF Filters, see ICF Filter Syntax in *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Governance*.

Configuring Reconciliation Jobs

Configure Reconciliation jobs to perform reconciliation runs that check for new information on your target system periodically and replicates the data in Oracle IdentityGovernance.

You can apply this procedure to configure the reconciliation jobs for users and entitlements.

To configure a reconciliation job:

1. Log in to Oracle Identity System Administration.
2. In the left pane, under System Management, click **Scheduler.**

> **Note**
>
> If you are using OIG 12cPS4 with 2022OCTBP or later version, log in to Identity Console, click **Manage**, under **System Configuration**, click **Scheduler**.

3. Search for and open the scheduled job as follows:
   a. In the Search field, enter the name of the scheduled job as the search criterion. Alternatively, you can click **Advanced Search** and specify the search criterion.
   b. In the search results table on the left pane, click the scheduled job in the Job Name column.
4. On the Job Details tab, you can modify the parameters of the scheduled task:
   - **Retries:** Enter an integer value in this field. This number represents the number of times the scheduler tries to start the job before assigning the Stopped status to the job.
   - **Schedule Type:** Depending on the frequency at which you want the job to run, select the appropriate schedule type. See Creating Jobs in *Oracle Fusion Middleware Administering Oracle Identity Governance*.

   In addition to modifying the job details, you can enable or disable a job.
5. On the **Job Details** tab, in the Parameters region, specify values for the attributes of the scheduled task.

> **Note**
>
> Values (either default or user-defined) must be assigned to all the attributes. If even a single attribute value is left empty, then reconciliation is not performed.

6. Click **Apply** to save the changes.

> **Note**
>
> You can use the Scheduler Status page in Identity System Administration to either start, stop, or reinitialize the scheduler.

Performing Provisioning Operations

To create a new user in Oracle Identity Self Service by using the **Create User** page, you must provision or request for accounts on the **Accounts** tab of the **User Details** page.

To perform provisioning operations in Oracle Identity Governance, perform the following steps:

1. Log in to **Identity Self Service**.
2. Create a user as follows:
   a. In Identity Self Service, click **Manage**. The **Home** tab displays the different Manage option. Click **Users**. The **Manage Users** page is displayed.
   b. From the Actions menu, select **Create**. Alternatively, you can click **Create** on the toolbar. The **Create User** page is displayed with input fields for user profile attributes.
   c. Enter details of the user in the **Create User** page.
3. On the Account tab, click **Request Accounts.**
4. In the Catalog page, search for and add to cart the application instance for the connector that you configured earlier and then click **Checkout**.
5. Specify value for fields in the application form and then click **Ready to Submit.**
6. Click **Submit.**

> **See Also**
>
> Creating a User in *Oracle Fusion Middleware Performing Self Service Tasks with Oracle Identity Governance* for details about the fields on the Create User page

5

Extending the Functionality of the Connector

You can extend the functionality of the connector to address your specific business requirements.

The following topics discuss information related to using the connector for performing reconciliation and provisioning operations:

- Implementing Custom Authentication
- Implementing Custom Parsing
- Configuring Transformation and Validation of Data
- Configuring Action Scripts
- Configuring the Connector for Multiple Installations of the Target System

Implementing Custom Authentication

If your target system uses an authentication mechanism that is not supported by this connector, then you must implement the authentication that your target system uses and then attach it to the connector by using the plug-ins exposed by this connector. Implementing custom authentication involves creating a Java class, overriding the Map<String, String> getAuthHeaders(Map<String, Object> authParams) method that returns the authorization header in the form of a map, and updating the connector installation media to include the new Java class.

All the target system configuration and authentication details that may be required for obtaining the authorization header are passed to the Map<String, String> getAuthHeaders(Map<String, Object> authParams) method through specific IT resource parameters. All the configuration properties exposed by this connector are accessible within this method as a part of "authParams".

To implement a custom authentication:

1. Create a Java class for implementing custom authentication. This class must implement the org.identityconnectors.scimcommon.auth.spi.AuthenticationPlugin interface.

   Note down the name of this Java class. You will provide the name of the Java class while configuring the IT resource for your target system which is described later in this guide.

2. Override the **Map<String, String> getAuthHeaders(Map<String, Object> authParams)** method in the custom Java class.

   This method must implement the custom authentication logic that returns the authorization header in the form of a map For example, `{ Authorization = Bearer XXXXXXXXXX }`. The authorization header contains the access token received from the target.

3. Package the Java class implementing the custom authentication into a JAR file.

4. Package the JAR file containing the custom authentication implementation with the connector bundle JAR as follows:

> **Note**
>
> Ensure to package all the JARs for any other custom implementations that you may have.

a. Extract the contents of the org.identityconnectors.genericscim-12.3.0.jar file into a temp directory. This file is located in the GenericSCIM-***RELEASE_NUMBER***\bundle directory.

b. Copy the JAR file containing the custom authentication (from Step 3) to the lib directory.

c. Regenerate the connector bundle (org.identityconnectors.genericscim-12.3.0.jar) by running the following command:

```
jar -cvfm org.identityconnectors.genericscim-12.3.0.jar META-INF/
MANIFEST.MF *
```

> **Note**
>
> While updating the connector bundle, ensure that META-INF\MANIFEST.MF file is unchanged.

5. Run the Oracle Identity Governance Delete JARs utility to delete any existing JARs in Oracle Identity Governance database before you upload the regenerated connector bundle. This utility is copied into the following location when you install Oracle Identity Governance:

> **Note**
>
> Before you use this utility, verify that the `WL_HOME` environment variable is set to the directory in which Oracle WebLogic Server is installed.

- **For Microsoft Windows**:

  OIM_HOME/server/bin/DeleteJars.bat

- **For UNIX**:

  OIM_HOME/server/bin/DeleteJars.sh

When you run the utility, you are prompted to enter the login credentials of the Oracle Identity Governance administrator, URL of the Oracle Identity Governance host computer, context factory value, type of JAR file being deleted, and the location from which the JAR file is to be deleted. Specify 4 (ICF Bundle) as the value of the JAR type.

6. Run the Oracle Identity Governance Upload JARs utility to upload the regenerated connector bundle to Oracle Identity Governance database. This utility is copied into the following location when you install Oracle Identity Governance:

> **Note**
>
> Before you use this utility, verify that the `WL_HOME` environment variable is set to the directory in which Oracle WebLogic Server is installed.

- **For Microsoft Windows**:

  OIM_HOME/server/bin/UploadJars.bat

- **For UNIX**:

  OIM_HOME/server/bin/UploadJars.sh

When you run the utility, you are prompted to enter the login credentials of the Oracle Identity Governance administrator, URL of the Oracle Identity Governance host computer, context factory value, type of JAR file being uploaded, and the location from which the JAR file is to be uploaded. Specify 4 (ICF Bundle) as the value of the JAR type.

7. Restart Oracle Identity Governance.

This completes the procedure for implementing a custom authentication.

Implementing Custom Parsing

By default, the connector supports only JSON parsing during reconciliation runs. If the reconciliation data from your target system is not in JSON format, then you must write a custom parser implementation for your data format.

To implement custom parsing:

1. Create a Java class for implementing the custom parser. This class must implement the org.identityconnectors.scimcommon.parser.spi.ParserPlugin interface.

   Note down the name of this Java class. You will provide the name of the Java class while configuring the IT resource for your target system which is described later in this guide.

2. Override the **String parseRequest(Map<String, Object> attrMap)** and **List<Map<String, Object>> parseResponse(String response, Map<String, String> parserConfigParams)** methods in the custom Java class.

   The String parseRequest(Map<String, Object> attrMap) method implements the logic for parsing an attribute and generates a string request payload.

   The List<Map<String, Object>> parseResponse(String response, Map<String, String> parserConfigParams) method implements the logic for parsing the string response received from the target in this class.

3. Package the Java class implementing the custom parser into a JAR file.

4. Package the JAR file containing the custom parser implementation with the connector bundle JAR as follows:

> **Note**
>
> Ensure to package all the JARs for any other custom implementations that you may have.

a. Extract the contents of the org.identityconnectors.genericscim-12.3.0.jar file into a temp directory. This file is located in the GenericSCIM-***RELEASE_NUMBER***\bundle directory.

b. Copy the JAR file containing the custom authentication (from Step 3) to the lib directory.

c. Regenerate the connector bundle (org.identityconnectors.genericscim-12.3.0.jar) by running the following command:

```
jar -cvfm org.identityconnectors.genericscim-12.3.0.jar META-INF/
MANIFEST.MF *
```

> **Note**
>
> While updating the connector bundle, ensure that META-INF\MANIFEST.MF file is unchanged.

5. Run the Oracle Identity Governance Delete JARs utility to delete any existing JARs in Oracle Identity Governance database before you upload the regenerated connector bundle. This utility is copied into the following location when you install Oracle Identity Governance:

> **Note**
>
> Before you use this utility, verify that the `WL_HOME` environment variable is set to the directory in which Oracle WebLogic Server is installed.

- **For Microsoft Windows**:

  OIM_HOME/server/bin/DeleteJars.bat

- **For UNIX**:

  OIM_HOME/server/bin/DeleteJars.sh

When you run the utility, you are prompted to enter the login credentials of the Oracle Identity Governance administrator, URL of the Oracle Identity Governance host computer, context factory value, type of JAR file being deleted, and the location from which the JAR file is to be deleted. Specify 4 (ICF Bundle) as the value of the JAR type.

6. Run the Oracle Identity Governance Upload JARs utility to upload the regenerated connector bundle to Oracle Identity Governance database. This utility is copied into the following location when you install Oracle Identity Governance:

> **Note**
>
> Before you use this utility, verify that the `WL_HOME` environment variable is set to the directory in which Oracle WebLogic Server is installed.

- **For Microsoft Windows**:

  OIM_HOME/server/bin/UploadJars.bat

- **For UNIX**:

OIM_HOME/server/bin/UploadJars.sh

When you run the utility, you are prompted to enter the login credentials of the Oracle Identity Governance administrator, URL of the Oracle Identity Governance host computer, context factory value, type of JAR file being uploaded, and the location from which the JAR file is to be uploaded. Specify 4 (ICF Bundle) as the value of the JAR type.

7. Restart Oracle Identity Governance.

This completes the procedure for implementing custom parsers.

Configuring Transformation and Validation of Data

Configure transformation and validation of user account data by writing Groovy script logic while creating your application.

You can configure transformation of reconciled single-valued user data according to your requirements. For example, you can use First Name and Last Name values to create a value for the Full Name field in Oracle Identity Governance.

Similarly, you can configure validation of reconciled and provisioned single-valued data according to your requirements. For example, you can validate data fetched from the First Name attribute to ensure that it does not contain the number sign (#). In addition, you can validate data entered in the First Name field on the process form so that the number sign (#) is not sent to the target system during provisioning operations.

To configure transformation or validation of user account data, you must write Groovy scripts while creating your application. For more information about writing Groovy script-based validation and transformation logic, see Validation and Transformation of Provisioning and Reconciliation Attributes of *Oracle Fusion Middleware Performing Self Service Tasks with Oracle Identity Governance*.

Configuring Action Scripts

You can configure **Action Scripts** by writing your own Groovy scripts while creating your application.

These scripts can be configured to run before or after the create, update, or delete an account provisioning operation. For example, you can configure a script to run before every user creation operation.

For information on adding or editing action scripts, see Updating the Provisioning Configuration in *Oracle Fusion Middleware Performing Self Service Tasks with Oracle Identity Governance*.

Configuring the Connector for Multiple Installations of the Target System

You must create copies of configurations of your base application to configure it for multiple installations of the target system.

The following example illustrates this requirement:

The London and New York offices of Example Multinational Inc have their own installations of the target system, including independent schema for each. The company has recently installed Oracle Identity Governance, and they want to configure it to link all the installations of the target system.

To meet the requirement posed by such a scenario, you must clone your application which copies all configurations of the base application into the cloned application. For more information about cloning applications, see Cloning Applications in *Oracle Fusion Middleware Performing Self Service Tasks with Oracle Identity Governance*.

6

Upgrading the Generic SCIM Connector

If you have already deployed the 11.1.1.5.0 version of the Generic SCIM connector, then you can upgrade the connector to version 12.2.1.3.0 by uploading the new connector JAR files to the Oracle Identity Governance database.

The following topics describe the procedures for upgrading the Generic SCIM Connector:

- Preupgrade Steps
- Upgrade Steps
- Post-Upgrade Steps

> **See Also**
>
> About Upgrading Connectors in *Oracle Fusion Middleware Administering Oracle Identity Governance* for detailed information on these steps.

Preupgrade Steps

The preupgrade steps for the connector involves performing a reconciliation run to fetch records from the target system, defining the source connector in Oracle Identity Governance, creating copies of the connector if you want to configure it for multiple installations of the target system, and disabling all the scheduled jobs.

> **Note**
>
> It is strongly recommended that you create a backup of the Oracle Identity Governance database and the connector JARs before you perform an upgrade operation. Refer to the database documentation for information about creating a backup.

As a best practice, first perform the upgrade procedure in a test environment. Perform the following pre upgrade steps:

1. Perform a reconciliation run to fetch all the latest updates to Oracle Identity Governance.
2. Perform the preupgrade procedure as documented in the Managing Connector Lifecycle in *Oracle Fusion Middleware Administering Oracle Identity Governance*.
3. Disable all the scheduled jobs.

Upgrade Steps

Update the following Decode entries in the `Lookup.<Connector name>.Configuration` and `Lookup.<Connector name>.Configuration.Truested lookup definition: Code Key: Bundle Version; Decode: 1.0.1115`
New Decode: 12.3.0

Post-Upgrade Steps

Post upgrade steps involve uploading new connector JAR to Oracle Identity Governance database.

To perform the post-upgrade, perform the following procedure:

1. Delete the old Connector JARs. Run the Oracle Identity Governance Delete JARs (`$ORACLE_HOME/bin/DeleteJars.sh`) utility to delete the existing ICF bundle`org.identityconnectors.genericscim-1.0.11150.jar` from the Oracle Identity Governance database.

   When you run the Delete JARs utility, you are prompted to enter your log in credentials for the Oracle Identity Governance administrator, URL of the Oracle Identity Governance host computer, context factory value, type of JAR file being deleted, and the name of the JAR file to be removed. Specify *4* as the value of the JAR type.

2. Upload the new connector JARs by performing the following steps:

   a. Run the Oracle Identity Governance Upload JARs (`$ORACLE_HOME/bin/UploadJars.sh`) utility to upload the connector JARs.

   b. Upload the `org.identityconnectors.genericscim-12.3.0.jar` bundle as an ICF Bundle. Run the Oracle Identity Governance Upload JARs utility to post the new ICF bundle.`org.identityconnectors.genericscim-12.3.0.jar` file to the Oracle Identity Governance database.

      When you run the Upload JARs utility, you are prompted to enter the log in credentials of the Oracle Identity Governance administrator, URL of the Oracle Identity Governance host computer, context factory value, type of JAR file being uploaded, and the location from which the JAR file is to be uploaded. Specify *4* as the value of the JAR type.

   c. Reconfigure the IT resource of the connector if the IT resource details are updated.

3. Restart the Oracle Identity Governance.
4. Perform full reconciliation and delete reconciliation (if applicable).
5. If the connector is deployed on a Connector Server, then perform the following:

   a. Stop the connector server.

   b. Replace the existing bundle JAR file `org.identityconnectors.genericscim-1.0.1115.jar` with the new bundle JAR file `org.identityconnectors.genericscim-12.3.0.jar`.

   c. Start the connector server.

> **Note**
>
> If you have configured the connector for multiple versions of target system, see Configuring the Connector for Multiple Installations of the Target System.

> **See Also**
>
> Configuring Oracle Identity Governance for information about creating, activating, and publishing a sandbox and creating a new UI form.

A

Files and Directories of Generic SCIM Connector

These are the components of the connector installation package and connector metadata package that comprise the Generic SCIM connector.

Below table describes the files and directories in the connector installation package.

Table A-1 Files and Directories in the Connector Installation Package

File in the Installation Package Directory	Description
bundle/ org.identityconnectors.genericscim-12.3.0.jar	This JAR file is the ICF connector bundle.
configuration/GenericSCIM-CI.xml	This file is used for installing a CI-based connector. This XML file contains configuration information that is used by the Connector Installer during connector installation.
xml/GenericSCIM-auth-template.xml	This file contains definitions for the connector objects required for creating an Authoritative application. It includes certain details required to connect Oracle Identity Governance with the target system. It also includes configuration details specific to your target system, attribute mappings, correlation rules, and reconciliation jobs.
xml/GenericSCIM-ConnectorConfig.xml	This XML file contains definitions for the connector components. These components include the following: • IT resource type • Process form • Process task and adapters (along with their mappings) • Resource object • Provisioning process • Prepopulate rules • Lookup definitions • Scheduled tasks
xml/GenericSCIM-target-template.xml	This file contains definitions for the connector objects required for creating a Target application. It includes certain details required to connect Oracle Identity Governance with the target system. It also includes configuration details specific to your target system, attribute mappings, correlation rules, and reconciliation jobs.
metadata-generator/bin/classpath_append.cmd metadata-generator/bin/classpath.cmd	These files contain the commands that add the JAR files (located in the lib directory) to the classpath on Microsoft Windows.

Table A-1 (Cont.) Files and Directories in the Connector Installation Package

File in the Installation Package Directory	Description
metadata-generator/bin/ GenericScimGenerator.cmd	This file contains commands to run the metadata generator.
metadata-generator/bin/GenericScimGenerator.sh logging.properties	Note that the .cmd file is the Microsoft Windows version of the metadata generator. Similarly, the .sh file is the UNIX version of the metadata generator.
metadata-generator/lib/connector-framework-internal.jar groovy-ant.jar groovy-sql.jar	This JAR file contains class files that define the ICF Application Programming Interface (API). This API is used to communicate between Oracle Identity Governance and this connector.
metadata-generator/lib/connector-framework.jar groovy-groovydoc.jar groovy-xml.jar	This JAR file contains the groovy libraries required for running the metadata generator.
metadata-generator/lib/GenericSCIM-oim-integration.jar groovy.jar	This JAR file contains the class files of the metadata generation utility.
metadata-generator/lib/ org.identityconnectors.genericscim-12.3.0.jar	These files contain the commands that add the JAR files (located in the lib directory) to the classpath on Microsoft Windows.
metadata-generator/resources/ GenericSCIMConfiguration.groovy	This file contains properties that store basic information about the target system schema, which is used to configure the mode (trusted source or target resource) in which you want to run the connector. In addition, it stores information about the manner in which the connector must connect to the target system.