

Oracle® Identity Governance

Configuring a Generic REST-Based Application



Release 12.2.1.3.0

F33076-02

May 2021

The Oracle logo, consisting of a solid red square with the word "ORACLE" in white, uppercase, sans-serif font centered within it.

ORACLE®

Oracle Identity Governance Configuring a Generic REST-Based Application, Release 12.2.1.3.0

F33076-02

Copyright © 2021, Oracle and/or its affiliates.

Primary Author: Alankrita Prakash

Contributors: Sakshi Gambhir

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software" or "commercial computer software documentation" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface

Audience	vii
Documentation Accessibility	vii
Related Documents	vii
Conventions	viii

1 About the Generic REST Connector

1.1	Introduction to the Connector	1-1
1.2	Certified Components	1-2
1.3	Certified Languages	1-3
1.4	Architecture of the Generic REST Connector	1-4
1.5	Connector Features	1-6
1.5.1	Trusted Source and Target Resource Reconciliation	1-6
1.5.2	Full and Incremental Reconciliation	1-6
1.5.3	Limited (Filtered) Reconciliation	1-7
1.5.4	Custom Authentication	1-7
1.5.5	Custom Parsing	1-7
1.5.6	Custom Payload	1-7
1.5.7	Support for Additional HTTP Headers	1-8
1.5.8	Support for Handling Multiple Endpoint URLs	1-8
1.5.9	SSL Communication	1-8
1.6	Use Cases Supported by the Generic REST Connector	1-8

2 Creating an Application By Using the Generic REST Connector

2.1	Prerequisites for Creating an Application	2-1
2.1.1	Process Flow for Creating an Application By Using the Connector	2-1
2.1.2	Downloading the Connector Installation Package	2-3
2.2	Navigating to the Create Application Screen	2-3
2.3	Understanding and Updating the Basic Information	2-3
2.3.1	Basic Configuration Parameters	2-4
2.3.2	Advanced Settings Parameters	2-6

2.3.3	Authentication Parameters	2-16
2.3.4	Providing Basic Information	2-20
2.4	Understanding and Updating the Schema Page	2-20
2.4.1	Understanding the Schema Page for a Target Application	2-21
2.4.2	Understanding the Schema Page for an Authoritative Application	2-21
2.4.3	Updating the Schema Page	2-27
2.5	Updating Settings	2-28
2.5.1	Correlation Rules, Responses, and Situations	2-28
2.5.1.1	Correlation Rules, Responses, and Situations for a Target Application	2-28
2.5.1.2	Correlation Rules, Responses, and Situations for an Authoritative Application	2-29
2.5.2	Reconciliation Jobs	2-30

3 Performing the Postconfiguration Tasks for the Generic REST Connector

3.1	Creating a UI Form for an Application	3-1
3.1.1	Creating and Activating a Sandbox	3-1
3.1.2	Creating a New UI Form	3-1
3.1.3	Associating the Form with the Application Instance	3-2
3.1.4	Publishing a Sandbox	3-2
3.2	Updating an Existing Application Instance with a New Form	3-2
3.3	Harvesting Entitlements and Sync Catalog	3-3
3.4	Managing Logging for the Generic REST Connector	3-3
3.4.1	Understanding Log Levels	3-3
3.4.2	Enabling Logging	3-4
3.5	Localizing Field Labels in UI Forms	3-6
3.6	Configuring SSL	3-8

4 Using the Generic REST Connector

4.1	Configuring Reconciliation	4-1
4.1.1	Performing Full Reconciliation and Incremental Reconciliation	4-1
4.1.2	Performing Limited Reconciliation	4-2
4.2	Configuring Reconciliation Jobs	4-2
4.3	Performing Provisioning Operations	4-3
4.4	Uninstalling the Connector	4-3

5 Extending the Functionality of the Generic REST Connector

5.1	Implementing Custom Authentication	5-1
-----	------------------------------------	-----

5.2	Implementing Custom Parsing	5-3
5.3	Configuring Transformation and Validation of Data	5-5
5.4	Configuring Action Scripts	5-5
5.5	Configuring the Connector for Multiple Installations of the Target System	5-6

6 Upgrading the Generic REST Connector

6.1	Preupgrade Steps	6-1
6.2	Upgrade Steps	6-2
6.3	Post upgrade Steps	6-2
	Target Resource User Reconciliation	6-3
	Target User Delete Reconciliation	6-4

7 Files and Directories of the Generic REST Connector

List of Tables

2-1	Basic Configuration Parameters	2-4
2-2	Advanced Settings Parameters	2-6
2-3	HTTP Basic Authentication Parameters	2-17
2-4	OAuth 2.0 JWT Parameters	2-17
2-5	OAuth2.0 Client Credentials Parameters	2-18
2-6	OAuth 2.0 Resource Owner Password Parameters	2-19
2-7	REST Trusted User Account Schema Attributes for an Authoritative Application	2-22
2-8	Predefined Identity Correlation Rule for a Target Application	2-28
2-9	Predefined Situations and Responses for a Target Application	2-29
2-10	Predefined Identity Correlation Rule for an Authoritative Application	2-29
2-11	Predefined Situations and Responses for an Authoritative Application	2-30
2-12	Parameters of the User Reconciliation Jobs	2-31
2-13	Parameters of the Delete User Reconciliation Jobs	2-32
2-14	Parameters of the Reconciliation Jobs for Entitlements	2-32
3-1	Log Levels and ODL Message Type:Level Combinations	3-4
7-1	Files and Directories in the Connector Installation Package	7-1

Preface

This guide describes the connector that is used to integrate Oracle Identity Manager with any REST-based identity-aware application.

Audience

This guide is intended for resource administrators and target system integration teams. This guide assumes that you are familiar with REST-based architecture and standards.

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 Governance 12.2.1.4.0, visit the following Oracle Help Center page:

<https://docs.oracle.com/en/middleware/idm/identity-governance/12.2.1.4/index.html>

For information about installing and using Oracle Identity Governance 12.2.1.3.0, visit the following Oracle Help Center page:

<https://docs.oracle.com/en/middleware/idm/identity-governance/12.2.1.3/index.html>

For information about Oracle Identity Governance Connectors 12.2.1.3.0 documentation, visit the following Oracle Help Center page:

<https://docs.oracle.com/en/middleware/idm/identity-governance-connectors/12.2.1.3/index.html>

For information about Oracle Identity Manager Connectors 11.1.1 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.
<i>italic</i>	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 REST Connector

The Generic REST connector integrates Oracle Identity Manager with REST-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 topics provide a high-level overview of the connector:

- [Introduction to the Connector](#)
- [Certified Components](#)
- [Certified Languages](#)
- [Architecture of the Generic REST Connector](#)
- [Connector Features](#)
- [Use Cases Supported by the Generic REST Connector](#)

1.1 Introduction to the Connector

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

 **Note:**

In this guide:
A REST-based identity-aware application has been referred to as the **target system** or **REST-based target system**.

The Generic REST 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 REST connector standardizes service processes and implements automation to replace manual tasks.

In order to connect with a REST-based target system, the Generic REST connector supports HTTP Basic Authentication and OAuth 2.0 authentication mechanisms. This connector also supports authenticating to the target system by using access 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 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 REST connector synchronizes data between Oracle Identity Governance and REST-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 REST connector is a connector for a discovered target system. This is because the schema of the REST-based target system with which the connector integrates is not known in advance. The Generic REST 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 REST-based target system and then generate the artifacts.

1.2 Certified Components

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

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: <ul style="list-style-type: none"> Oracle Identity Governance 12c PS4 (12.2.1.4.0) Oracle Identity Governance 12c PS3 (12.2.1.3.0) 	You can use one of the following releases: <ul style="list-style-type: none"> Oracle Identity Governance 12c PS4 (12.2.1.4.0) Oracle Identity Governance 12c PS3 (12.2.1.3.0) Oracle Identity Manager 11g Release 2 PS3 (11.1.2.3.0) Oracle Identity Manager 11g Release 2 PS2 (11.1.2.2.0)
Target System	Any identity-aware application that supports REST service	Any identity-aware application that supports REST service
Connector Server	12.2.1.3.0 Note: Connector server is optional, If you have deployed the Generic REST connector in the Connector Server, then you can download the necessary Java Connector Server 12.2.1.3.0 from the Oracle Technology Network web page.	12.2.1.3.0 Note: Connector server is optional, If you have deployed the Generic REST connector in the Connector Server, then you can download the necessary Java Connector Server 12.2.1.3.0 from the Oracle Technology Network web page.
Connector Server JDK	JDK 1.8 or later	JDK 1.8 or later

1.3 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
- 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.

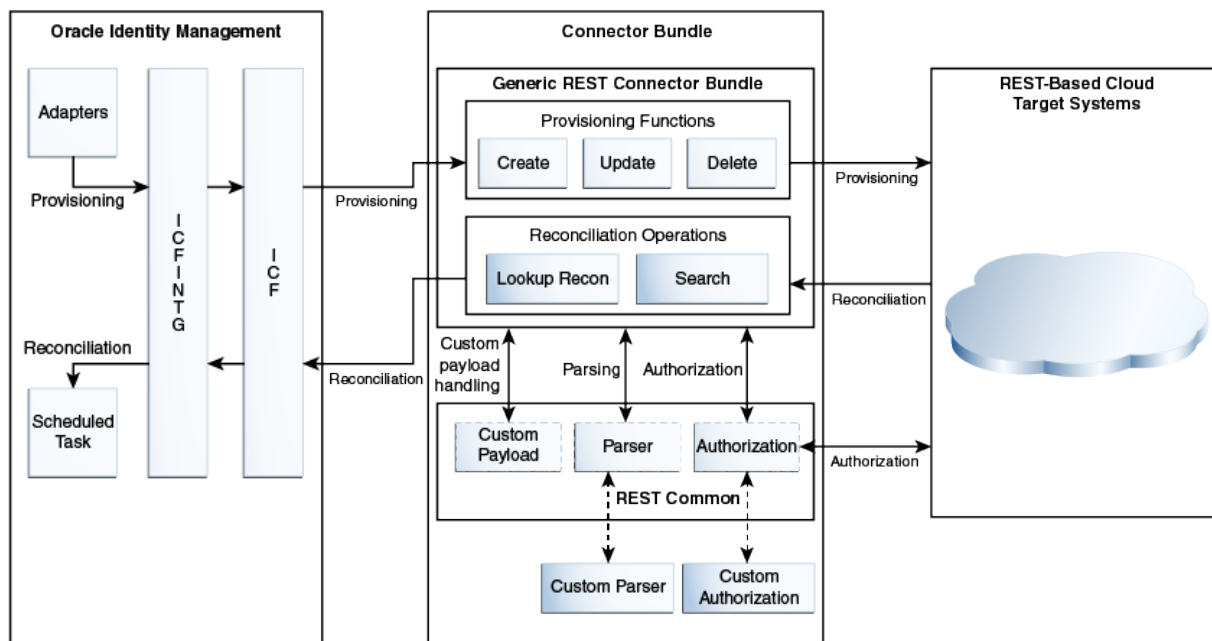
1.4 Architecture of the Generic REST Connector

The Generic REST 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 Manager.

[Figure 1-1](#) shows the architecture of the connector.

Figure 1-1 Connector Architecture



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

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 REST 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 REST 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 REST Common layer. After the adapters establish a connection with the target system, REST 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 REST Common layer. Then, the connector retrieves all records that match the reconciliation criteria by calling the specific REST endpoint. This result is then passed to Oracle Identity Governance.

1.5 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](#)

1.5.1 Trusted Source and Target Resource Reconciliation

You can configure your REST-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 for trusted source (authoritative application) and target resource (Target application) reconciliation.

You can use the Generic REST authoritative connector to integrate Eloqua as a trusted source of Oracle Identity Governance. In this mode, the connector reconciles all the person types that are supported by the Eloqua application.

In the target resource mode, you can use the Generic REST target connector to create a Target application to provision and reconcile user records from the Eloqua application.

See [Configuring Reconciliation Jobs](#) for more information.

1.5.2 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 Manager. 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 Manager.

 **Note:**

If the target system contains an attribute, such as `updatedAt` for target Eloqua, that holds the timestamp at which an object is created or modified, the connector supports incremental reconciliation .

You can perform a full reconciliation any time. See [Performing Full Reconciliation and Incremental Reconciliation](#) for more information.

1.5.3 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 Reconciliation](#) for more information about performing limited reconciliation.

1.5.4 Custom Authentication

By default, the Generic REST 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.

1.5.5 Custom Parsing

By default, the Generic REST 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.

1.5.6 Custom Payload

The Generic REST 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.

1.5.7 Support for Additional HTTP Headers

If your target system requires additional or custom HTTP headers in any REST 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.

1.5.8 Support for Handling Multiple Endpoint URLs

The Generic REST 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.

1.5.9 SSL Communication

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

See [Configuring SSL](#) for information about configuring secure communication.

1.6 Use Cases Supported by the Generic REST Connector

The Generic REST connector can be used to integrate OIM with any target system that supports REST services. This connector can be used to load identity data into OIM from a REST 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 OIM 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 OIM. 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 REST connector that can be used to integrate all the cloud applications with OIM. The Generic REST 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 REST connector is a hybrid approach that helps enterprises leverage on-premise OIM deployment to integrate with target systems for identity governance. These target systems include any application that exposes REST APIs such as SaaS, PaaS, home-grown applications and so on.

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

- **User Management**

The Generic REST 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 REST 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 OIM administrators. For example, to create a new user in the target system, fill in and submit the OIM 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 REST 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 REST connector to automatically assign or revoke groups to users based on predefined access policies in OIM. Similarly, you can use the Generic REST 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 REST 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](#)
- [Navigating to the Create Application Screen](#)
- [Understanding and Updating the Basic Information](#)
- [Understanding and Updating the Schema Page](#)
- [Updating Settings](#)

2.1 Prerequisites for Creating an Application

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

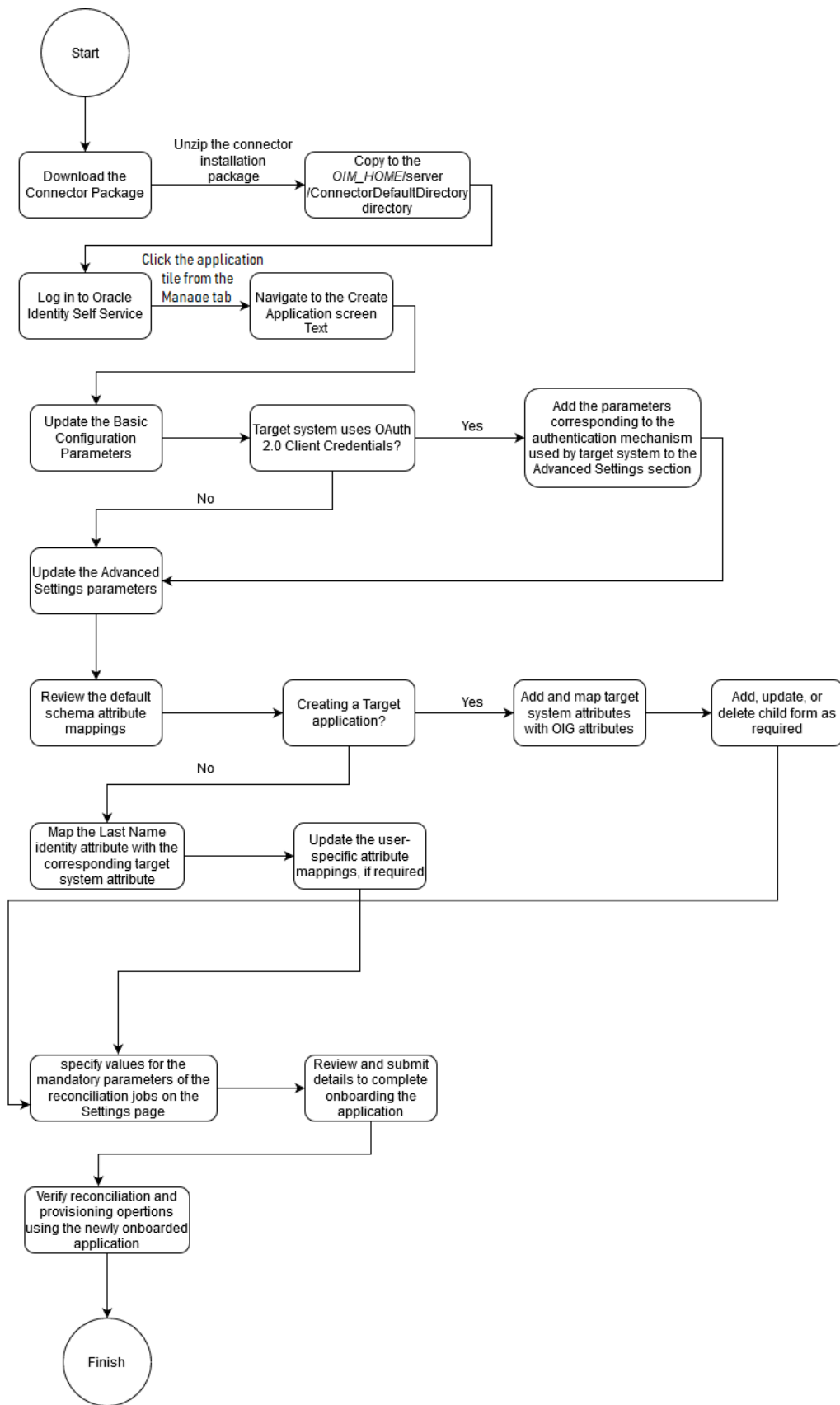
- [Process Flow for Creating an Application By Using the Connector](#)
- [Downloading the Connector Installation Package](#)

2.1.1 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.

[Figure 2-1](#) 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



2.1.2 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 Governance.
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.

2.2 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.

2.3 Understanding and Updating the 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](#)
- [Advanced Settings Parameters](#)
- [Authentication Parameters](#)
- [Providing Basic Information](#)

2.3.1 Basic Configuration Parameters

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

Table 2-1 Basic Configuration Parameters

Parameter	Mandatory?	Description
authenticationType	Yes	<p>Enter the type of authentication used by your target system.</p> <p>The following are the possible values for this parameter:</p> <ul style="list-style-type: none"> • For HTTP Basic Authentication: <code>basic</code> • For OAuth 2.0 JWT: <code>jwt</code> • For OAuth 2.0 Client Credentials: <code>client_credentials</code> • For OAuth 2.0 Resource Owner Password: <code>password</code> • For manual input of access token and refresh token: <code>other</code> • For custom authentication implementation: <code>custom</code> <p>Default value: <code>client_credentials</code></p> <p>Note: If you are using a value other than the default value, <code>client_credentials</code>, then ensure to add the parameters corresponding to the authentication type that your target system is using.</p>
host	Yes	<p>Host name or IP address of the computer hosting the target system.</p> <p>Sample value: <code>www.example.com</code></p>
port	Yes	<p>Port number at which the target system is listening.</p> <p>Sample value: <code>80</code></p>

Table 2-1 (Cont.) Basic Configuration Parameters

Parameter	Mandatory?	Description
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: <code>https://api.example.com/oauth2/token</code>
clientId	No	Enter the client identifier (a unique string) issued by the authorization server to the client during the registration process. Sample value: <code>XDWTh0r2eWuULCDVt</code>
clientSecret	No	Enter the value used to authenticate the identity of your client application. Sample value: <code>clZsdZisTOoYN5NITirarIDepDki JTGHdzNFT0m</code>
Connector Server Name	No	If you have deployed the Generic REST 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: <code>www.example.com</code>
proxyPort	No	Proxy port number Sample value: <code>80</code>
proxyUser	No	Proxy user name of the target system user account that Oracle Identity Manager uses to connect to the target system.
proxyScheme	No	
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.
sslEnabled	No	If the target system requires SSL connectivity, then set the value of this parameter to <code>true</code> . Otherwise set the value to <code>false</code> . Default value: <code>true</code>

Table 2-1 (Cont.) Basic Configuration Parameters

Parameter	Mandatory?	Description
uriPlaceholder	No	<p>Enter the key-value pair for replacing place holders in the reURIs. The URI place holder consists of values which are repeated in every relative URL. Values must be comma separated.</p> <p>For example, tenant ID and API version values are a part of every request URL. Therefore, we replace it with a key-value pair.</p> <p>Sample value: "tenant_id;<domain name>", "api_version;api-version=1.6"</p>

2.3.2 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	<p>This parameter holds the name of the connector class.</p> <p>Default value: org.identityconnectors.genericrest</p>
Bundle Version	Yes	<p>This parameter holds the version of the connector bundle class.</p> <p>Default value: 12.3.0</p>
Connector Name	Yes	<p>This parameter holds the name of the connector bundle package.</p> <p>Default value: org.identityconnectors.genericrest.GenericRESTConnector</p>

Table 2-2 (Cont.) Advanced Settings Parameters

Parameter	Mandatory?	Description
relURIs	Yes	<p>Enter the relative URLs for all operations of each object class.</p> <p>Enter a value for this parameter in one of the following formats:</p> <ul style="list-style-type: none"> • For attributes: <i>OBJ_CLASS.OP=REL_URL</i> • For special attributes: <i>OBJ_CLASS.ATTR_NAME.OP=REL_URL</i> • For attributes that have the same relative URL for multiple operations: <i>OBJ_CLASS=REL_URL</i> or <i>OBJ_CLASS.ATTR_NAME=REL_URL</i> • If you want to filter records based on a filter criteria during reconciliation, then use <i>\$(Filter Suffix)\$</i>. Here, replace <i>\$(Filter Suffix)\$</i> with the filter criteria that you specify as part of the Filter Suffix attribute of the scheduled job. • If you have to pass the unique ID of the user as part of endpoint URL, use <i>\$(__UID__\$)</i>. Here, replace <i>\$(__UID__\$)</i> with the unique ID of the user. • If you have to pass any attribute other than the unique ID of the user, then represent it in one of the following formats: <ul style="list-style-type: none"> – For a single-valued attribute: <i>\$(attr_name)\$</i> Here, replace <i>\$(attr_name)\$</i> with the name of the attribute in the target system. – For an embedded object: <i>\$(OBJ_CLASS.ATTR_NAME)\$</i> For example, <i>\$(__GROUP__.id)\$</i>. • End point specified in relURI's should be without a host name. Host name should be specified in basic configuration and it will be appended through the code in run time. <p>Sample value: "<u>__ACCOUNT__</u>.CREATEOP=/admin/directory/v1/users", "<u>__ACCOUNT__</u>.SEARCHOP</p>

Table 2-2 (Cont.) Advanced Settings Parameters

Parameter	Mandatory?	Description
		<pre> =/admin/directory/v1/users/\$ (Filter Suffix)\$&page=\$ (PAGE_OFFSET)\$&pagesize=\$ (PAGE_SIZE)\$", "__ACCOUNT__=/ admin/directory/v1/users/\$ (__UID__\$", "__GROUP__.CREAT EOP=/admin/directory/v1/ groups", "__GROUP__=/admin/ directory/v1/groups/\$ (__UID__\$", "__GROUP__.SEARC HOP=/admin/directory/v1/ groups/(Filter Suffix)\$&page=\$ (PAGE_OFFSET)\$&pagesize=\$ (PAGE_SIZE)\$", "__ACCOUNT__.a lias=/admin/directory/v1/ users/\$(__UID__\$/ aliases", "__ACCOUNT__.alias. DELETEOP=/admin/ directory/v1/users/\$ (__UID__\$/aliases/\$ (alias)\$", "__GROUP__.alias.D ELETEOP=/admin/directory/v1/ groups/\$(__UID__\$/aliases/\$ (alias)\$", "__ACCOUNT__.__GRO UP__=/admin/directory/v1/ groups/\$(__GROUP__.id)\$/ members", "__ACCOUNT__.__GROU P__.DELETEOP=/admin/ directory/v1/groups/\$ (__GROUP__.id)\$members/\$ (__UID__\$", "__ACCOUNT__.__G ROUP__.SEARCHOP=/admin/ directory/v1/groups? userKey=\$(__UID__\$" </pre> <p>Note: Depending on the target system, you can perform pagination by using either offset value or page count. Below are sample values for both options.</p> <ul style="list-style-type: none"> • Sample value for offset value: <pre> "__ACCOUNT__.SEARCHOP=/ admin/directory/v1/ users/?\$(Filter Suffix)\$&pagesize=\$ (PAGE_SIZE)\$&offset=\$ (PAGE_OFFSET)\$" </pre> • Sample value for page count: <pre> "__ACCOUNT__.SEARCHOP=/ admin/directory/v1/ users/?\$(Filter Suffix)\$&page=\$ </pre>

Table 2-2 (Cont.) Advanced Settings Parameters

Parameter	Mandatory?	Description
		<p><code>(PAGE_INCREMENT)&pagesize= e=\$(PAGE_SIZE)\$"</code></p> <p>To support pagination using page size we have introduced one configurable attribute page increment as same as page offset. It should we specified in rel URI as <code>\$(PAGE_INCREMENT)\$</code>.</p> <p>Page increment attribute count starts from 1 and increment by one till it reaches last page. For example, <code>relURI:/API/REST/2.0/system/users?page=\$(PAGE_INCREMENT)&pagesize=\$(PAGE_SIZE)\$</code></p> <p>Since <code>PAGE_SIZE</code> is a configurable parameter, let's consider it to be 100. Then the request URI looks as below:</p> <ul style="list-style-type: none"> • 1st request : <code>/API/REST/2.0/system/users?page=1&pagesize=100</code> • 2nd request : <code>/API/REST/2.0/system/users?page=2&pagesize=100</code> • 3rd request : <code>/API/REST/2.0/system/users?page=3&pagesize=100</code> • 4th request : <code>/API/REST/2.0/system/users?page=4&pagesize=100</code> and so on <p>Here every page has 100 users and it continues to bring the data till the last page. If the target system has 1000 accounts, then 10 calls will be triggered.</p> <p>To support pagination using <code>OFFSET</code> is configurable attribute page offset. It should we specified in rel URI as <code>\$(PAGE_OFFSET)\$</code>.</p> <p>Page offset attribute count starts from 0 and increment by page size till it reaches last page. For example, <code>relURI:/API/REST/2.0/system/users?page=\$(PAGE_OFFSET)&pagesize=\$(PAGE_SIZE)\$</code></p> <p>Since <code>PAGE_SIZE</code> is a configurable parameter, let's consider it to be 100. Then the request URI looks as below:</p>

Table 2-2 (Cont.) Advanced Settings Parameters

Parameter	Mandatory?	Description
		<ul style="list-style-type: none"> • 1st request : /API/REST/2.0/system/users?offset=0&pagesize=100 • 2nd request : /API/REST/2.0/system/users?offset=101&pagesize=100 • 3rd request : /API/REST/2.0/system/users?offset=201&pagesize=100 • 4th request : /API/REST/2.0/system/users?offset=301&pagesize=100 and so on <p>Here every page has 100 users and it continues to bring the data till the last page. If the target system has 1000 accounts, then 10 calls will be triggered.</p>
opTypes	No	<p>Enter the HTTP operation type for each object class that is to be managed by the connector.</p> <p>Values must be comma separated and in the following format: <code>"OBJ_CLASS.OPERATION=HTTP_OPERATION"</code></p> <p>In this format, <i>OBJ_CLASS</i> is the connector object class, <i>OPERATION</i> is the connector operation (for example, CreateOp, UpdateOp, or SearchOp), and <i>HTTP_OPERATION</i> is the HTTP operation (for example, GET, PUT, or POST).</p> <p>Sample value: <code>"__ACCOUNT__.CREATEOP=POST", "__ACCOUNT__.DELETEOP=DELETE", "__GROUP__.CREATEOP=POST", "__ACCOUNT__.alias.CREATEOP=POST", "__ACCOUNT__.alias.UPDATEOP=POST", "__ACCOUNT__.alias.DELETEOP=DELETE", "__GROUP__.alias.CREATEOP=POST", "__GROUP__.alias.UPDATEOP=POST", "__GROUP__.alias.DELETEOP=DELETE", "__ACCOUNT__.__GROUP__.CREATEOP=POST", "__ACCOUNT__.__GROUP__.UPDATEOP=POST", "__ACCOUNT__.__GROUP__.DELETEOP=DELETE"</code></p>

Table 2-2 (Cont.) Advanced Settings Parameters

Parameter	Mandatory?	Description
nameAttributes	Yes	<p>Enter the name attribute for all object classes that are handled by the connector. This value specifies the mapping between the <code>__NAME__</code> connector attribute and the corresponding target system attribute for each object class that the connector handles.</p> <p>Format: <code>OBJ_CLASS.ATTR_NAME</code></p> <p>Sample value: "<code>__ACCOUNT__.userName</code>"</p> <p>Note: All values in this parameter must be comma separated.</p>
uidAttributes	Yes	<p>Enter the <code>__UID__</code> attribute for each object class that the connector handles. A <code>__UID__</code> 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.</p> <p>Format: <code>OBJ_CLASS.ATTR_NAME</code></p> <p>Sample value: "<code>__ACCOUNT__.id</code>"</p> <p>In this format, <code>OBJ_CLASS</code> is the connector object class and <code>ATTR_NAME</code> is the name of the attribute that uniquely identifies an account in the target system.</p> <p>Note: All values in this parameter must be comma separated.</p>
statusAttributes	No	<p>Enter the name of the target system attribute that holds the status for each object class this connector handles.</p> <p>Format: <code>OBJ_CLASS.ATTR_NAME</code></p> <p>Sample value: "<code>__ACCOUNT__.suspended</code>"</p> <p>Note: All values in this parameter must be comma separated.</p>
passwordAttribute	No	<p>Enter the name of the target system attribute that is mapped to the <code>__PASSWORD__</code> attribute of the connector in OIM.</p>
pageSize	No	<p>The number of resources/users that appears on a page for a search operation.</p> <p>Default value: 100</p>

Table 2-2 (Cont.) Advanced Settings Parameters

Parameter	Mandatory?	Description
pageTokenAttribute	No	The attribute in response payload that denotes the next page token. Default value: odata.nextLink
pageTokenRegex	No	This attribute is used in the URL while reconciliation to support pagination. Default value: (?<=skiptoken=).*
Any Incremental Recon Attribute Type	No	By default, during incremental reconciliation, Oracle Identity Governance accepts timestamp information sent from the target system only in Long datatype format. Setting the value of this parameter to True indicates that Oracle Identity Governance will accept timestamp information in any datatype format. Default value: True

Table 2-2 (Cont.) Advanced Settings Parameters

Parameter	Mandatory?	Description
jsonResourcesTag	Yes, if you are implementing custom parsers	<p>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.</p> <p>Enter a value for this parameter in the following format: <i>OBJ_CLASS=OUTER_ATTR_NAME</i></p> <p>In this format, <i>OBJ_CLASS</i> is the name of the object class for which a response payload is being parsed. <i>OUTER_ATTR_NAME</i> is the name of the outer tag in the response payload.</p> <p>For example, consider the following JSON value for a User object:</p> <pre>"Resources" : "{ "user1" : "{value1}" , "user2" : "{value2}" }"</pre> <p>Because the name of the object class for a User object is <code>__ACCOUNT__</code>, for the given example, the value of the <code>jsonResourcesTag</code> parameter is <code>__ACCOUNT__=Resources</code>.</p> <p>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.</p>
httpHeaderContentType	Yes	<p>This parameter holds the content type expected by the target system in the header.</p> <p>Default value: <code>application/json</code>.</p>
httpHeaderAccept	Yes	<p>This parameter holds the accept type expected by the target system in the header.</p> <p>Default value: <code>application/json</code>.</p>

Table 2-2 (Cont.) Advanced Settings Parameters

Parameter	Mandatory?	Description
specialAttributeTargetFormat	No	<p>Enter the format in which a special attribute is present in the target system.</p> <p>Attributes in an object class that can be managed only through a separate rest endpoint rather than the same endpoint of the base object class are referred to as special attributes.</p> <p>Format: <i>OBJ_CLASS.ATTR_NAME=TARGET_FORMAT</i></p> <p>In this example, <i>TARGET_FORMAT</i> is the format in the format of the special attribute in the target system.</p> <p>For example, consider the following target endpoint value:</p> <pre>{ "kind": "admin#directory#aliases", "etag": etag, "aliases": ["kind": "admin#directory#alias", "id": string, "etag": etag, "primaryEmail": string, "alias": string] }</pre> <p>In this example, the alias attribute is present as <i>aliases.alias</i> in the target system endpoint. Therefore, you must set the value of this parameter to <i>__ACCOUNT__.alias=aliases.alias</i>.</p> <p>Sample value: <i>"__ACCOUNT__.alias=aliases.alias", "__GROUP__.alias=aliases.alias", "__ACCOUNT__.__GROUP__=groups"</i></p>

Table 2-2 (Cont.) Advanced Settings Parameters

Parameter	Mandatory?	Description
specialAttributeHandling	No	<p>Enter the list of special attributes whose values must be sent to the target system in separate calls, one at a time. If you do not specify a value for this parameter, then the connector will send all values for a given special attribute in a single call.</p> <p>Format: <i>OBJ_CLASS.ATTR_NAME.OP=SINGLE</i></p> <p>Note: All values in this parameter must be comma separated.</p> <p>Sample value: "<u>ACCOUNT</u>.alias.CREATEOP=SINGLE", "<u>GROUP</u>.alias.CREATEOP=SINGLE", "<u>ACCOUNT</u>.alias.UPDATEOP=SINGLE", "<u>GROUP</u>.alias.UPDATEOP=SINGLE", "<u>ACCOUNT</u>.<u>GROUP</u>.CREATEOP=SINGLE", "<u>ACCOUNT</u>.<u>GROUP</u>.UPDATEOP=SINGLE"</p>
customPayload	No	<p>Enter a comma-separated list of request payload formats for target system attributes that do not adhere to the standard JSON format.</p> <p>Format: <i>OBJ_CLASS.ATTRNAME.OP=PAYLOAD_FORMAT</i></p> <p>Note: If you must pass the unique ID or Name attribute of the user as part of a custom payload, then represent it as <i>\$(<u>UID</u>)\$</i> or <i>\$(<u>NAME</u>)\$</i> respectively"</p> <p>Sample value: "<u>ACCOUNT</u>.<u>GROUP</u>.UPDATEOP={ \"user\": { \"id\": \"\$(<u>UID</u>)\$\"}, \"group\": { \"id\": \"\$(id)\$\" } }"</p>
targetObjectIdentifier	No	<p>Enter the name of the attribute used for identifying the required target object during reconciliation.</p> <p>Format: <i>OBJ_CLASS.ATTR_NAME=TARGET_OBJ_ATTR_NAME;VAL</i></p>
statusEnableValue	No	<p>This entry holds the name attribute for all target values indicating the status as enabled.</p>
statusDisableValue	No	<p>This entry holds the name attribute for all target values indicating the status as disabled.</p>

2.3.3 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 `authenticationType` parameter of the Basic Configuration section.

The `authenticationType` 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 `authenticationType` 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 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 authenticationType parameter is set to jwt.

Table 2-4 OAuth 2.0 JWT 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
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.

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

Parameter	Description
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: RS256
privateKeyFormat	Enter the format of the private key used to sign the access token. Sample value: PKCS12

OAuth 2.0 Client Credentials

Table 2-5 lists the set of parameters for which you must enter values when the `authenticationType` 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 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: clZsdZisT0oYN5NITirarIDepDkiJTGHdzNFT0m
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: <code>https://api.example.com/oauth2/token</code>

OAuth 2.0 Resource Owner Password

Table 2-6 lists the set of parameters for which you must enter values when the `authenticationType` parameter is set to `password`.

Table 2-6 OAuth 2.0 Resource Owner Password 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: clZsdZisTOoYN5NITirarIDepDkiJTGHdzNF T0m 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 `authenticationType` 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 `authenticationType` 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](#).

2.3.4 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 REST 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 `authenticationType` 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 `authenticationType` 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

2.4 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](#)

- [Understanding the Schema Page for an Authoritative Application](#)
- [Updating the Schema Page](#)

2.4.1 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 REST 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 REST 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. You can

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*.

2.4.2 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 REST 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 [Table 2-7](#)) 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 REST 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
REST User GUID	__UID__	String	No	Yes	NA
Status	__ENABLE_	String	No	Yes	NA
Last Name		String	No	Yes	NA
Xellerate Type		String	No	Yes	End-User
Role		String	No	Yes	Full-Time
Organization Name		String	No	Yes	Xellerate Users

The following is EloquaAppsUserSchema.properties, a sample schema file property used in the Eloqua target system:

```
#Schema file for Eloqua
#List of fields
FieldNames=id,login,name,firstName,Active,lastName,senderDisplayName,emailAddress

#Unique ID Attribute
UidAttribute=id
#Account Name attribute
NameAttribute=login

#Account Status Attribute and Mapping
StatusAttribute=Active
Active.True=Enabled
Active.False=Disabled
```

The following is GenericRestConfiguration.groovy, a sample groovy file:

```
/*
 * Run like:
 * In Windows:
GenericRestGenerator.cmd ..\resources\GenericRestConfiguration.groovy
trusted
 * In Linux/Unix: sh GenericRestGenerator.sh ../resources/
```

```
GenericRestConfiguration.groovy trusted
*/
trusted {
  /*
   * ITResource name
   */
  itResourceDefName='resttrust2' // This will be used as a base name
for all metadata across the connector
  // itResourceName="$itResourceDefName" //the same as
itResourceDefName by default

  /*
   * Output files
   */
  // connectorDir="..${itResourceDefName}" //
output dir of the connector, is the same as it resource name by default
  // xmlFile='GenericRest-ConnectorConfig.xml' // name
of the dm xml of the connector
  // configFile='GenericRest-CI.xml' // name
of the config xml
  // propertiesFile='GenericRest-generator.properties' // name
of the resources/properties file
  // version='11.1.1.6.0' //
connector version

  /*
   * Trusted/Target mode
   * For trusted, we will not create forms, dataobjects and event
handlers
   * For target, we will create all above metadata
   */
  trusted=true // Flag
to denote if the mode is trusted or not

  /*
   * Location of the generic rest bundle jar
   */
  bundleJar='../lib/org.identityconnectors.genericrest-12.3.0.jar'

  /*
   * The Configuration used to run the generic rest bundle mentioned
above, get the schema from below mentioned schemaFile, which is
required for generating metadata
   */
  config = [
    'schemaFile' : '../schema/EloquaAppsUserSchema.properties',
    'authenticationType' : '',
    'username' : '',
    'password' : '',
    'host' : '',
    'port' : '',
    'proxyHost' : '',
    'proxyPort' : '',
    'proxyUser' : '',
```



```
'proxyPassword' : '',
'sslEnabled' : '',
'relURIs' : '',
'nameAttributes' : '',
'uidAttributes' : '',
'opTypes' : '',
'jsonResourcesTag' : '',
'httpHeaderContentType' : '',
'httpHeaderAccept' : '',
'customPayload' : '',
'statusAttributes' : '',
'specialAttributeTargetFormat' : '',
'specialAttributeHandling' : '',
'statusEnableValue' : '',
'statusDisableValue' : '',
'passwordAttribute' : ''
]

/**
 * Provide the attribute list that need to be handled as Date on
process form
 * Make sure these fields datatype in schema should be long
 * dateAttributeList is not a mandatory field
 **/

//dateAttributeList = ["JoiningDate"]

/**
 * Alias are used to map the OIM User Form attributes with the
Connector Attributes.
 * The Format is of 'Connector Attribute':'OIM User Form
Attribute'
 * Mandatory alias shouldn't be removed. Customer can update
these mandatory attributes but should not be removed
 * Customer can add other aliases to the OIM User form fields
 **/
// Mapping is mandatory for attributes User Login, Last
Name,Organization, Xellerate Type and Role. One can modify the required
mappings but shouldn't delete them.
//UID field is not required in trusted but if customer wanted to
add UID field then one can map it to a valid OIM User Form Label
alias = ['__NAME__':'User Login', 'lastName':'Last
Name','Organization':'Organization Name', 'Employee Type':'Xellerate
Type', 'Role':'Role']
//Extend the aliases to include more connector attributes for
trusted by mapping
alias += ['__ENABLE__':'Status', 'emailAddress':'Email']
}

/*
 * Run like:
 * In Windows:
```

```
GenericRestGenerator.cmd ..\resources\GenericRestConfiguration.groovy
target
  * In Linux/Unix: sh GenericRestGenerator.sh ../resources/
GenericRestConfiguration.groovy target
  */
target {
  /*
   * ITResource name
   */
  itResourceDefName='restci2' // This will be used as a base name for
all metadata across the connector
  // itResourceName="$itResourceDefName" //the same as
itResourceDefName by default

  /**
   * Give the name of the Application Instance that need to
be generated. By default Application Instance name is taken as
itResourceDefName
   * Application Instance is not a mandatory field
   */
  //applicationInstanceName="$itResourceDefName"

  /*
   * Output files
   */
  // connectorDir="..\${itResourceDefName}" //
output dir of the connector, is the same as it resource name by default
  // xmlFile='GenericRest-ConnectorConfig.xml' // name
of the dm xml of the connector
  // configFile='GenericRest-CI.xml' // name
of the config xml
  // propertiesFile='GenericRest-generator.properties' // name
of the resources/properties file
  // version='11.1.1.6.0' //
connector version

  /*
   * Location of the generic rest bundle jar
   */
  bundleJar='../lib/org.identityconnectors.genericrest-12.3.0.jar'

  /*
   * The Configuration used to run the generic rest bundle mentioned
above, and get the schema from the below mentioned schemaFile, which is
required for generating metadata
   */
  config = [
    'schemaFile' : '../schema/GoogleAppsUserSchema.properties',
    'authenticationType' : '',
    'username' : '',
    'password' : '',
    'host' : '',
    'port' : '',
    'proxyHost' : '',
    'proxyPort' : '',
```

```
'proxyUser' : '',
'proxyPassword' : '',
'sslEnabled' : '',
'relURIs' : '',
'nameAttributes' : '',
'uidAttributes' : '',
'opTypes' : '',
'jsonResourcesTag' : '',
'httpHeaderContentType' : '',
'httpHeaderAccept' : '',
'customPayload' : '',
'statusAttributes' : '',
'specialAttributeTargetFormat' : '',
'specialAttributeHandling' : '',
'statusEnableValue' : '',
'statusDisableValue' : '',
'passwordAttribute' : ''
]

/**
 * Lookup List is the list of attributes for which we need to
create lookups and map those fields as lookup in form.
 * For Main Process Form Fields and a Multivalued field the
format is
 *      lookupAttributeList=["FieldName"]
 * For Embedded Multi Valued field the format is
 *      lookupAttributeList=["ObjectClassName.SubFieldName"]
 * lookups will be generated with the FieldNames in format
Lookup.${ITResource}.${FieldName}
 *
 * lookupList is not a mandatory field
 */

//lookupAttributeList=['Currency']

/* entitlementAttributeList is the list of fully Qualified field
names to which entitlements need to be tagged.
 * If you require entitlements for a multi valued attribute which
is embedded then the format should be as
 *      entitlementAttributeList=["${ObjectClass}.SubFieldName"]
 * If the attribute is just a MultiValued then the format is
 *      entitlementAttributeList=["MultiValuedFieldName"]
 *
 * entitlementAttributeList is not a mandatory attribute
 */

entitlementAttributeList=['__GROUPS__.id']

/**
 * Provide the attribute list that need to be handled as Date on
process form
 * Make sure these fields datatype in schema should be long
 * dateAttributeList is not a mandatory field
 */
```

```

//dateAttributeList = ["JoiningDate"]

/*
 * Target attribute to user fields alias
 */
//Mandatory alias
alias = ['__UID__':'id', '__NAME__':'login']
//Optional aliases if any (Can also be used to give short names for
lengthy attribute names)
//alias += ['Generic Rest Target':'GENREST', 'Family Name':'Last
name']

/*
 * Generate prepopulate adapters
 */
prepopulate = ['__UID__':'User Login']
}

```

2.4.3 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 REST 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.
 - f. If you
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.

2.5 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](#)

2.5.1 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](#)

2.5.1.1 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 REST 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.

[Table 2-8](#) lists the default simple correlation rule for a REST-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.

[Table 2-9](#) 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

2.5.1.2 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 REST 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.

[Table 2-10](#) 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	REST User GUID	No
__NAME__	Equals	User Login	No

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

(__UID__ Equals REST 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.
- REST 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.

[Table 2-11](#) 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

2.5.2 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:

- REST Application Target User Reconciliation: Use this reconciliation job to reconcile user data from a Target application.
- REST 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.
Filter	Do <i>not</i> modify this value. 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:

- REST Application Target User Delete Reconciliation: Use this reconciliation job to reconcile data about deleted user accounts from a Target application.
- REST 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 <i>not</i> 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 REST 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.

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

Parameter	Description
Lookup Name	<p>This parameter holds the name of the lookup definition that maps each lookup definition with the data source from which values must be fetched.</p> <p>Depending on the reconciliation job you are using, the default values is as follows:</p> <p><code>Lookup.GenericREST.ENTITLEMENT_NAME</code></p> <p>For example, for the Groups entitlement job, the default value is <code>Lookup.GenericREST.Groups</code>.</p>
Object Type	<p>Enter the type of object whose values must be synchronized.</p> <p>For example, for reconciling Groups entitlement, enter <code>__GROUPS__</code>.</p> <p>Note: Do not change the value of this attribute.</p>
Code Key Attribute	<p>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).</p> <p>Default value: <code>__NAME__</code></p> <p>Note: Do not change the value of this attribute.</p>
Decode Attribute	<p>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).</p> <p>Default value: <code>__NAME__</code></p>

3

Performing the Postconfiguration Tasks for the Generic REST Connector

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

- [Creating a UI Form for an Application](#)
- [Harvesting Entitlements and Sync Catalog](#)
- [Managing Logging for the Generic REST Connector](#)
- [Localizing Field Labels in UI Forms](#)
- [Configuring SSL](#)

3.1 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](#)

3.1.1 Creating and Activating a Sandbox

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

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

3.1.2 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.

3.1.3 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 Manager* for instructions on modifying an application instance to associate it with a form.

3.1.4 Publishing a Sandbox

Before publishing a sandbox, perform this procedure as a best practice to validate all sandbox changes made till this stage as it is difficult to revert the changes after a sandbox is published.

1. In Identity System Administration, deactivate the sandbox.
2. Log out of Identity System Administration.
3. Log in to Identity 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 application instance form for your resource appears with correct fields.
5. Publish the sandbox. See *Publishing a Sandbox in Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Governance*.

3.2 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*

3.3 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

3.4 Managing Logging for the Generic REST 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 contain detailed information:

- [Understanding Log Levels](#)
- [Enabling Logging](#)

3.4.1 Understanding Log Levels

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

ODL is the principle 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 Governance 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.

3.4.2 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='genericrest-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.GENERICREST" level="[LOG_LEVEL]"
useParentHandlers="false">
  <handler name="genericrest-handler" />
  <handler name="console-handler" />
</logger>
<logger name="ORG.IDENTITYCONNECTORS.RESTCOMMON.UTILS.RESTCOMMONUTILS"
level="[LOG_LEVEL]" useParentHandlers="false">
  <handler name="genericrest-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 3-2 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='genericrest-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/genericrestLogs.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.GENERICREST" level="NOTIFICATION:1"
useParentHandlers="false">
  <handler name="genericrest-handler" />
  <handler name="console-handler" />
</logger>
<logger name="ORG.IDENTITYCONNECTORS.RESTCOMMON.UTILS.RESTCOMMONUTILS"
level="NOTIFICATION:1" useParentHandlers="false">
  <handler name="genericrest-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.

3.5 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 Rest 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.

3.6 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 REST-based target system.
2. Copy the public key certificate of the REST-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 REST 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](#)
- [Uninstalling the Connector](#)

4.1 Configuring Reconciliation

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

This section provides information on the following topics related to configuring reconciliation:

- [Performing Full Reconciliation and Incremental Reconciliation](#)
- [Performing Limited Reconciliation](#)

4.1.1 Performing Full Reconciliation and Incremental Reconciliation

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

After you create the application, you must first perform full reconciliation. In addition, you can switch from incremental reconciliation to full reconciliation whenever you want to ensure that all target system records are reconciled in Oracle Identity Manager.

To perform a full reconciliation run, ensure that no values are specified for the following parameters of the jobs for reconciling user records:

- Filter
- Incremental Recon Attribute
- Latest Token

 **Note:**

If the target is Eloqua, for example, you can use the attribute `updatedAt` for incremental reconciliation.

4.1.2 Performing Limited Reconciliation

By default, all target system records are reconciled during the current reconciliation run. You can customize this process by specifying the subset of target system records that must be reconciled.

You can perform limited reconciliation by creating filters for the reconciliation module. This connector provides a Filter parameter that allows you to use the Generic REST resource attributes to filter the target system records.

To create a filter for the Eloqua target system, the sample value would be "search=loginName=<uid_name>".

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

4.2 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 Identity Governance.

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

To configure a reconciliation job:

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

4.3 Performing Provisioning Operations

You create a new user in Identity Self Service by using the Create User page. You provision or request for accounts on the Accounts tab of the User Details page.

To perform provisioning operations in Oracle Identity Governance:

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

4.4 Uninstalling the Connector

Uninstalling the connector deletes all the account-related data associated with its resource objects.

If you want to uninstall the connector for any reason, then run the Uninstall Connector utility. Before you run this utility, ensure that you set values for `ObjectType` and `ObjectValues` properties in the `ConnectorUninstall.properties` file. For example, if you want to delete resource objects, scheduled tasks, and scheduled jobs associated with the connector, then enter "ResourceObject", "ScheduleTask", "ScheduleJob" as the value of the `ObjectType` property and a semicolon-separated list of object values corresponding to your connector as the value of the `ObjectValues` property.

 **Note:**

If you set values for the `ConnectorName` and `Release` properties along with the `ObjectType` and `ObjectValue` properties, then the deletion of objects listed in the `ObjectValues` property is performed by the utility and the Connector information is skipped.

For more information, see Uninstalling Connectors in *Oracle Fusion Middleware Administering Oracle Identity Governance*.

5

Extending the Functionality of the Generic REST 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](#)

5.1 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.restcommon.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 XXXXXXXXXXXX }`. 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.genericrest-1.0.1115.jar` file into a temp directory. This file is located in the `GenericREST-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.genericrest-1.0.1115.jar`) by running the following command:

```
jar -cvfm org.identityconnectors.genericrest-1.0.1115.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 Manager Delete JARs utility to delete any existing JARs in Oracle Identity Manager database before you upload the regenerated connector bundle. This utility is copied into the following location when you install Oracle Identity Manager:

 **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 Manager administrator, URL of the Oracle Identity Manager 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 Manager Upload JARs utility to upload the regenerated connector bundle to Oracle Identity Manager database. This utility is copied into the following location when you install Oracle Identity Manager:

 **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 Manager administrator, URL of the Oracle Identity Manager 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 Manager.

This completes the procedure for implementing a custom authentication.

5.2 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.restcommon.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.genericrest-1.0.1115.jar` file into a temp directory. This file is located in the `GenericREST-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.genericrest-1.0.1115.jar`) by running the following command:

```
jar -cvfm org.identityconnectors.genericrest-1.0.1115.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 Manager Delete JARs utility to delete any existing JARs in Oracle Identity Manager database before you upload the regenerated connector bundle. This utility is copied into the following location when you install Oracle Identity Manager:

 **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 Manager administrator, URL of the Oracle Identity Manager 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 Manager Upload JARs utility to upload the regenerated connector bundle to Oracle Identity Manager database. This utility is copied into the following location when you install Oracle Identity Manager:

 **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 Manager administrator, URL of the Oracle Identity Manager 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 Manager.

This completes the procedure for implementing custom parsers.

5.3 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*.

5.4 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 enable, disable, or delete an account provisioning operations. 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*.

5.5 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 REST Connector

If you have already deployed the 11.1.1.5.0 version of the Generic REST 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 Manager database.

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

- [Pre upgrade Steps](#)
- [Upgrade Steps](#)
- [Post upgrade Steps](#)

See Also:

For detailed steps on upgrading Connectors in Oracle Fusion Middleware Administering, see the Oracle Fusion Middleware Administering Oracle Identity Manager guide.

6.1 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 Manager, 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 Manager 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 Manager.
2. Perform the preupgrade procedure as documented in the Managing Connector Lifecycle in Oracle Fusion Middleware Administering Oracle Identity Manager.
3. Disable all the scheduled jobs.

6.2 Upgrade Steps

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

6.3 Post upgrade Steps

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

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

1. Delete the old Connector JARs. Run the Oracle Identity Manager Delete JARs (`$ORACLE_HOME/bin/DeleteJars.sh`) utility to delete the existing ICF bundle `org.identityconnectors.genericrest-1.0.1115.jar` from the Oracle Identity Manager database.

When you run the Delete JARs utility, you are prompted to enter your log in credentials for the Oracle Identity Manager administrator, URL of the Oracle Identity Manager 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 Manager Upload JARs (`$ORACLE_HOME/bin/UploadJars.sh`) utility to upload the connector JARs.
 - b. Upload the `org.identityconnectors.genericrest-12.3.0.jar` bundle as an ICF Bundle. Run the Oracle Identity Manager Upload JARs utility to post the new ICF bundle `org.identityconnectors.genericrest-12.3.0.jar` file to the Oracle Identity Manager database.

When you run the Upload JARs utility, you are prompted to enter the log in credentials of the Oracle Identity Manager administrator, URL of the Oracle Identity Manager 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 Manager.
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.genericrest-1.0.1115.jar` with the new bundle JAR file `org.identityconnectors.genericrest-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.](#)

 **Note:**

1. After upgrading to the 12.x Generic REST connector, if you require Target Resource User Reconciliation and Delete Reconciliation support (if target supports), perform the steps given in the below sections 6.3.1 and 6.3.2.
2. For incremental reconciliation support update the target Resource User Reconciliation, by following the steps in [Target Resource User Reconciliation](#).
3. For delete Reconciliation support, refer to the [Target User Delete Reconciliation](#).

Target Resource User Reconciliation

For Incremental Recon, update the Target Resource User Reconciliation.

You must update the Target Resource User Reconciliation, by performing the following procedure:

1. Log in to the **System Administration** and expand **System Configuration >Export Form OIG > Target Resource User Reconciliation> Scheduler** and save xml as Target Resource User Reconciliation.xml.
2. Open Target Resource User Reconciliation.xml in edit plus.
3. Add the following code text before `</parameters>`:

```
<string-param required="false" encrypted="false" helpText="Incremental Recon Attribute">Incremental Recon Attribute</string-param><string-param required="false" encrypted="false" helpText="Latest Token">Latest Token</string-param>
```
4. Save XML.
5. Log in to the System Administration and expand **System Configuration**, then import to OIG Target Resource User Reconciliation.xml. Restart the Oracle Identity Manager Server for changes to take effect.

Target User Delete Reconciliation

Create the Target User Delete Reconciliation Job by performing the following steps:

1. Rename Target Resource User Reconciliation.xml as Target User Delete Reconciliation.xml.
2. Open Target User Delete Reconciliation.xml in edit plus and perform the following changes:
 - a. scheduledTask name=Target User Delete Reconciliation & mds-file as Target User Delete Reconciliation.xml
 - b. Update the name, class, description as given below:


```
<name> Target User
Delete Reconciliation</name>
<class>oracle.iam.connectors.icfcommon.recon.SearchReconDeleteTask<
/class> <description> Target User Delete Reconciliation</
description>
```
 - c. Update <parameters> as given below:


```
<parameters> <string-param
required="false" encrypted="false" helpText="IT Resource Name">IT
Resource Name</string-param> <string-param required="false"
encrypted="false" helpText="Object Type">Object Type</string-param>
<string-param required="false" encrypted="false" helpText="Resource
Object Name">Resource Object Name</string-param> </parameters>
```
 - d. Add the below Job tag before </xl-ddm-data>:

```
<Job repo-type="API" name=" Target User Delete Reconciliation " >
<retrycount>0</retrycount>
<method>executeJob</method>
<taskKey>250</taskKey>
<taskClassName>oracle.iam.connectors.icfcommon.recon.SearchReconD
eleteTask</taskClassName>
<concurrent>>false</concurrent>
<jobScheduleType>None</jobScheduleType>
<attributes>
<object>
<key>Resource Object Name</key>
<value type="jobparameter">
<name type="string">Resource Object Name</name>
<required type="boolean">>false</required>
<encrypted type="boolean">>false</encrypted>
<helpText type="string">Resource Object Name</helpText>
<dataType type="string">String</dataType>
<paramKey type="string">37</paramKey>
<paramValue type="string">0365 Target User</paramValue>
</value>
</object>
<object>
<key>IT Resource Name</key>
<value type="jobparameter">
<name type="string">IT Resource Name</name>
<required type="boolean">>false</required>
<encrypted type="boolean">>false</encrypted>
<helpText type="string">IT Resource Name</helpText>
```



```
<dataType type="string">String</dataType>
<paramKey type="string">35</paramKey>
<paramValue type="string">0365 Target</paramValue>
</value>
</object>
<object>
<key>Object Type</key>
<value type="jobparameter">
<name type="string">Object Type</name>
<required type="boolean">>false</required>
<encrypted type="boolean">>false</encrypted>
<helpText type="string">Object Type</helpText>
<dataType type="string">String</dataType>
<paramKey type="string">36</paramKey>
<paramValue type="string">User</paramValue>
</value>
</object>
</attributes>
<lastModifyDate>1523209246378</lastModifyDate>
<taskStatus>>false</taskStatus>
<source-scope type="scheduledTask" name=" Target User Delete
Reconciliation "/>
<taskName scheduledTask=" Target User Delete Reconciliation "/>
</Job>
```

3. Save the XML file.
4. Log in to the **System Administration** and expand **System Configuration** and import to OIG Target User Delete Reconciliation.xml.
5. Update the Scheduler job parameters based on your requirement.

7

Files and Directories of the Generic REST Connector

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

[Table 7-1](#) describes the files and directories in the connector installation package.

Table 7-1 Files and Directories in the Connector Installation Package

File in the Installation Package Directory	Description
bundle/ org.identityconnectors.genericrest-12.3.0.jar	This JAR file is the ICF connector bundle.
configuration/GenericREST-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.
Files in the resources directory	Each of these resource bundles contains language-specific information that is used by the connector. During connector installation, these resource bundles are copied to Oracle Identity database. Note: A resource bundle is a file containing localized versions of the text strings that are displayed on the Administrative and User Console. These text strings include GUI element labels and messages.
xml/GenericREST-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/GenericREST-ConnectorConfig.xml	This XML file contains definitions for the connector components. These components include the following: <ul style="list-style-type: none">• IT resource type• Process form• Process task and adapters (along with their mappings)• Resource object• Provisioning process• Prepopulate rules• Lookup definitions• Scheduled tasks

Table 7-1 (Cont.) Files and Directories in the Connector Installation Package

File in the Installation Package Directory	Description
xml/GenericREST-pre-config.xml	This XML file contains definitions for the connector objects associated with any non-User objects such as Groups, Organizations, and so on. Also, it contains definitions of Lookups and schedule tasks.
xml/GenericREST-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	These files contain the commands that add the JAR files (located in the lib directory) to the classpath on Microsoft Windows.
metadata-generator/bin/classpath.cmd	
metadata-generator/bin/GenericRestGenerator.cmd	This file contains commands to run the metadata generator.
metadata-generator/bin/GenericRestGenerator.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 Manager 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/GenericREST-oim-integration.jar groovy.jar	This JAR file contains the class files of the metadata generation utility.
metadata-generator/lib/org.identityconnectors.genericrest-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/resource/GenericRestConfiguration.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.